Tesis Doctoral

Herramientas bioinformáticas para el análisis estructural de proteínas a escala genómica

Radusky, Leandro Gabriel

2017-03-10

UNIVERSIDAD DE BUENOS AIRES
Facultad de Ciencias Exactas y Naturales
Departamento de Química Biológica

Herramientas bioinformáticas para el análisis estructural de proteínas a escala genómica

Tesis para optar al título de Doctor de la Universidad de Buenos Aires en el Área Química Biológica

Leandro Gabriel Radusky

Directores de tesis: Dr. Marcelo Adrián Martí - Dr. Adrián Gustavo Turjanski
Consejero de estudios: Dr. Adrián Gustavo Turjanski

Buenos Aires, 2017

Fecha de Defensa: 10/3/2017

Resumen

El desarrollo de herramientas computacionales para el cálculo y análisis de datos se encuentra actualmente en constante expansión en diferentes campos de la ciencia, particularmente en el campo de las ciencias de la vida, donde la cantidad de datos disponibles, generados por nuevas técnicas experimentales convierte en fundamental la implementación de técnicas computacionales para el análisis y manejo de datos y su transformación en conocimiento.

En este sentido, focalizándose en el campo de la bioinformática estructural de proteínas y la quimioinformática, nos hemos concentrado en la generación de herramientas y aplicación de las mismas en problemas relacionados con la salud humana.

El presente trabajo de tesis tiene como primer objetivo proponer nuevos procedimientos para el descubrimiento de blancos proteicos relevantes en organismos bacterianos, usando como caso de estudio *Mycobacterium tuberculosis*.

El segundo objetivo de esta tesis es el de, seleccionado el blanco proteico dentro de un genoma de interés, estudiar cuáles son las características que debiera cumplir una molécula para tener buenas probabilidades unirse a dicho blanco y a partir de la misma proponer posibles ligandos derivados de bases de datos de compuestos.

El tercer objetivo de esta tesis es poder comprender y predecir cuáles serán los efectos en la función de una proteína determinada (potencialmente cualquiera que sea de interés del usuario) de mutaciones no sinónimas que se produzcan en su secuencia.

Los tres objetivos han sido abordados desde un punto de vista computacional y todos los métodos desarrollados pueden considerarse herramientas *in-silico*. Más allá de su aplicación en organismos o proteínas puntuales como casos de estudio, todos los

desarrollos pueden ser extendidos y reutilizados de manera directa y automática sobre otros organismos o proteínas.

Los resultados obtenidos han sido validados contra la literatura existente, permitiendo reproducir resultados experimentales y/o manualmente curados de una manera automática, lo que supone una reducción de tiempo y recursos en los procesos en los que estas herramientas están involucradas.

Bioinformatic tools for a genomic scale analysis of protein structure

Abstract

The development of computational tool for data calculation and analysis is actually in constant expansion through the different fields of science, particularly in the field of the life sciences, where the amount of available data produced by novel experimental techniques makes indispensable the implementation of computational techniques to handle and analyze the data and its transformation into knowledge.

In this sense, focusing in the field of the protein's structural bioinformatics and the cheminformatics, we concentrated our efforts in building tools and apply them in problems related to human health.

The present thesis work has as first objective to propose novel procedures to discover relevant protein targets in bacterial organisms, using as case of study *Mycobacterium tuberculosis*.

The second objective of this thesis is to, once selected the protein target within a genome, to study which are the properties that a molecule has to fulfill to have good chances of binding to this target, and based in it to propose possible ligands derived from a compound database.

The third propound objective is to comprehend and predict which will be the effects in the protein function (potentially any protein of user interest) of non-synonymous mutations produced in their sequence.

All the three objectives has been addressed from a computational point of view and all the developed methods can be considered *in-silico* tools. Besides of its application in punctual organisms or protein targets as cases of study, all this developments can be extended and reused in a direct manner over other organisms/proteins.

The obtained results has been validated against the existent literature, allowing to reproduce experimental and/or manually curated results in an automatic way, which supposes a saving of time and resources in the processes where this tools are involved.

Índice

1. Introducción

1.1 La Bioinformática como área del conocimiento

Existen discusiones alrededor de la definición y alcance de términos como bioinformática[1], biología computacional, química computacional, etc., de las cuales no participaremos en esta tesis. En un sentido amplio, puede entenderse el área de la bioinformática como el conjunto de técnicas computacionales para el almacenamiento, análisis y procesamiento de cualquier tipo de datos biológicos.

Si nos remitimos a los orígenes de la bioinformática como campo de estudio, podemos hablar de que los primeros algoritmos de procesamiento de información biológica estén probablemente asociados al tratamiento de secuencias de ADN[2], ya que el volumen de los mismos hizo impráctico su manejo manual a principios de los años setenta, época en la que otros datos que actualmente son procesados con técnicas bioinformáticas no estaban disponibles o eran más bien escasos. En la actualidad, el principal problema al que se enfrenta la bioinformática como campo, es la integración y extracción de información combinando de diferentes fuentes de datos de origen heterogéneo[3].

A lo largo del presente trabajo, manejaremos diferentes tipos de datos (que servirán como insumo de cada uno de los desarrollos realizados) los cuales comparten las siguientes características:

- Son almacenables
- Tienen una sintaxis precisa
- Son legibles con una semántica no ambigua
- Describen objetos o entidades biológicas mediante alguna o algunas de sus características físico-químicas.

Los primeros tres elementos se refieren todos aquellos datos que pueden ser procesados por una computadora mediante técnicas informáticas. El último agrega el componente biológico. Esos datos sufrirán combinaciones y transformaciones que darán lugar a nuevo conocimiento, plausible de ser utilizado tanto para validar teorías, como para alimentar modelos predictivos.

En desarrollos bioinformáticos correctos y consistentes, los datos de salida deben poseer características recién enumeradas, para servir como insumo de futuros procesos, y que los métodos utilizados sean replicables, como muestra la figura 1.1.1.

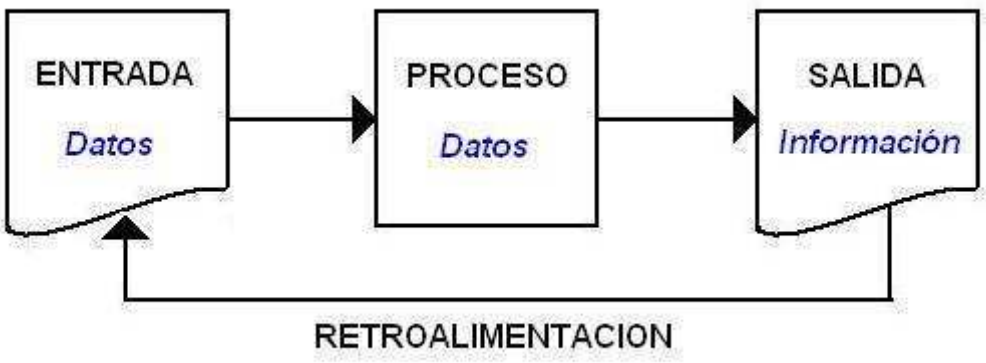


Figura 1.1.1: Diagrama de flujo del procesamiento computacional de la información como sistema replicable, que en esta tesis tendrá como datos e información la descripción fisicoquímica y contextual de objetos y entidades biológicas.

Esta tesis puede definirse como un trabajo en el campo de la bioinformática. A lo largo del mismo, se utilizará de manera sistemática la lógica descrita anteriormente para la resolución de tres problemas particulares de la biología molecular estructural.

1.2 Big Data Biológica

Son hitos remarcables en la historia de la bioinformática el descubrimiento del código genético[4] en la década de 1960 y el surgimiento de las técnicas de secuenciación de ADN[5]

en la década de 1970. El tratamiento de secuencias (tanto de ADN y ARN, como de proteínas) ha sido una de las áreas más prolíficas en cuanto al desarrollo de técnicas informáticas para usos aplicados a la biología. Entre ellos el alineamiento de secuencias (método extendido históricamente para compararlas) ha permitido inferir relaciones evolutivas[2], homología entre proteínas[6], asignar función a proteínas nuevas[7], inferir la estructura tridimensional[8,9] y plegado de proteínas[10,12], etcétera.

En particular, en el año 1982, con la aparición de GenBank[11], se establece el primer repositorio de secuencias de nucleótidos de acceso público, hecho motivado por la existencia de una cantidad creciente de información disponible y la necesidad de acceder a dicha información de una manera estandarizada. En los años posteriores, esta metodología se extendió (ver figura 1.2.1) y las bases de datos computacionales fueron utilizadas masivamente como herramienta de almacenamiento de todo tipo de datos biológicos: secuencias de ADN, ARN y proteínas[13], de estructuras[14], interacciones[15], etc.

Si bien el crecimiento en la cantidad de información almacenada fue significativa en las postrimerías del siglo XX, la aparición de nuevas técnicas para la secuenciación de ADN[16] a principios del siglo XXI, y la aparición de nuevas ómicas, dejó evidenciada la poca integración entre las diferentes bases de datos existentes[17]. El crecimiento exponencial de la capacidad de cómputo permitió plantearse la posibilidad de correlacionar estos datos heterogéneos provenientes de diversas fuentes para generar nuevo conocimiento, dando lugar a la necesidad de un cambio de paradigma y la incorporación de técnicas de Big Data ya existentes, aplicados a los problemas emergentes.

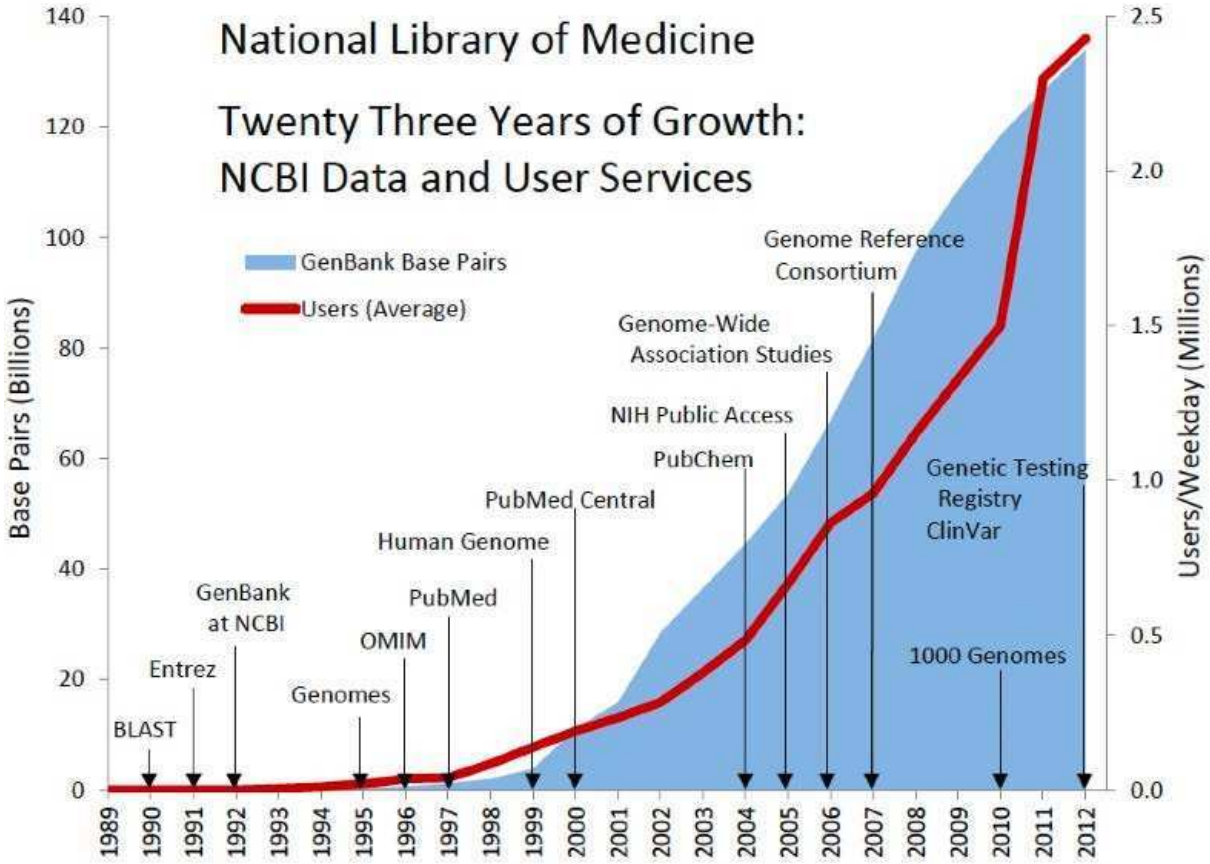


Figura 1.2.1: Gráfico publicado por el NIH donde se remarca la aparición de bases de datos biológicas que constituyen hitos para la bioinformática, mostrando tanto el crecimiento exponencial de la cantidad de información de secuencias disponibles como el crecimiento constante de usuarios dentro de la comunidad, lo que evidencia la utilidad y creciente dependencia de este tipo de recursos.

Big Data es un término que hace referencia al área de las ciencias de la computación que estudia el tratamiento de datos masivos[18]. Por un lado, manejar grandes volúmenes de datos plantea una serie de problemas técnicos para su acceso en forma eficiente. El crecimiento exponencial en la producción de datos hizo obligatoria su implementación en las bases de datos biológicas.

Los desarrollos realizados a lo largo de esta tesis están orientados a convertir grandes volúmenes de datos (por ejemplo: genomas enteros, incluyendo secuencias, estructuras y

anotaciones de distinta naturaleza, etcétera), en información, enmarcándose en el área de Big Data en biología

1.3 Procesamiento de datos biológicos

1.3.1 Comparación y alineamiento de secuencias

Uno de los principales procesos en el estudio de la biología, es el de la comparación. Las secuencias (de ADN, ARN y proteínas), por supuesto, no escapan a este proceso. La forma más básica en la que pueden compararse secuencias es de a pares. Un alineamiento de a pares es una forma de representar y comparar dos secuencias de proteínas (o de ADN, o de ARN) para resaltar sus zonas de similitud. Las mismas, podrían indicar relaciones funcionales y evolutivas entre las proteínas involucradas.

Cuando dos proteínas comparten un ancestro común, las posiciones del alineamiento que no coinciden pueden interpretarse como mutaciones puntuales (sustituciones), y los huecos como indels (mutaciones de inserción o deleción) introducidas en el proceso en el que se produjo la divergencia evolutiva. En el alineamiento de secuencias proteicas, el grado de similitud entre los aminoácidos que ocupan una posición concreta en la secuencia puede interpretarse como una medida aproximada de conservación en una región particular, o secuencia motivo, entre linajes.

La extensión natural de la comparación (alineamiento) de a pares, es la que se realiza para conjuntos de secuencias, lo que da lugar a alineamientos múltiples. El problema de alinear múltiples secuencias (MSA, por sus siglas en inglés) despertó tempranamente, en la década de 1970, el interés de la comunidad por las inferencias que era posible extraer, en la medida que la cantidad de secuencias disponibles fue siendo más voluminosa. La misma involucra el alineamiento de tres o más secuencias y busca encontrar posiciones equivalentes a lo largo de cada una de las secuencias alineadas (figura 1.3.1).

Figura 1.3.1: Visualización de un alineamiento de secuencias múltiple en el software Clustal[19], de secuencias relacionadas evolutivamente. Las posiciones están coloreadas por características fisicoquímicas similares, permitiendo interpretar amigablemente sustituciones que provocan pequeños o grandes cambios por pertenecer a grupos de aminoácidos de distintas características.

Un alineamiento múltiple puede proveer información de peso acerca de la relación estructural y/o funcional dentro de un conjunto de secuencias de proteínas (por ejemplo: la conservación evolutiva de aminoácidos importantes estructural o funcionalmente se corresponden con patrones en determinadas regiones de la secuencias). Pueden también ser útiles, por ejemplo, siendo almacenados como modelos ocultos de Markov[20] para la asignación de pertenencia de una nueva proteína a una familia de proteínas, permitiendo en muchos casos inferir, por ejemplo, su función o su dominio de plegado.

Los alineamientos múltiples son un intento de representar secuencias relacionadas evolutivamente en una forma consistente. Encontrar el alineamiento óptimo en base a un modelo evolutivo dado es equivalente a maximizar la probabilidad de que las secuencias hayan evolucionado de la forma en la que el alineamiento indica[21].

Existen rasgos críticos que definen la estructura y función de la proteína. El sitio activo de una enzima, por ejemplo, requiere que determinados residuos de aminoácidos tengan una

orientación tridimensional precisa. Así mismo, una interfaz de unión proteína-proteína puede constar de una amplia superficie con restricciones en la hidrofobicidad o polaridad de los residuos de aminoácidos. Las regiones funcionalmente restringidas de las proteínas evolucionan más lentamente que las regiones sin restricción, como bucles superficiales, dando lugar a bloques discernibles de secuencias conservadas cuando se compara las secuencias de una familia de proteínas. Esos bloques son habitualmente designados como "motivos" y proteínas que pertenecen a la misma familia muestran una alta conservación en estos motivos, presentando generalmente entonces una estructura tridimensional (plegado) similar, y una misma función.

En la sección de métodos de esta tesis entraremos en el detalle de los algoritmos que permiten alinear múltiples secuencias y cuál es la información que puede inferirse a partir de los mismos.

1.3.2 Determinación de la estructura proteica

La estructura tridimensional de una proteína es la disposición en el espacio de los átomos que la componen, provenientes de su secuencia de aminoácidos. Existen diferentes niveles de estructuración de una proteína, influyendo los niveles inferiores en la disposición de los niveles superiores.

La estructura primaria está definida por la secuencia de aminoácidos que componen a la proteína, unidos covalentemente mediante el enlace peptídico. El orden de los aminoácidos es consecuencia del material genético: cuando se traduce el RNA se obtiene el orden que va a dar lugar a la secuencia de la proteína.

La estructura secundaria es la disposición espacial local que ocupa la "columna vertebral" (backbone) de la proteína, y se determina mediante la conformación de enlaces tipo puente de hidrógeno. Existen estructuras secundarias bien definidas y ordenadas, como alfa-hélices, hojas-beta y regiones desestructuradas en cuanto a su estructura secundaria.

La estructura terciaria se define como la estructura que adopta la cadena polipeptídica en el espacio. El modo en que la secuencia de aminoácidos se pliega en el espacio (de forma globular, como fibra, etc) determinará el o los dominios de plegado que pueden ser asignados a la proteína (figura 1.3.2.1).

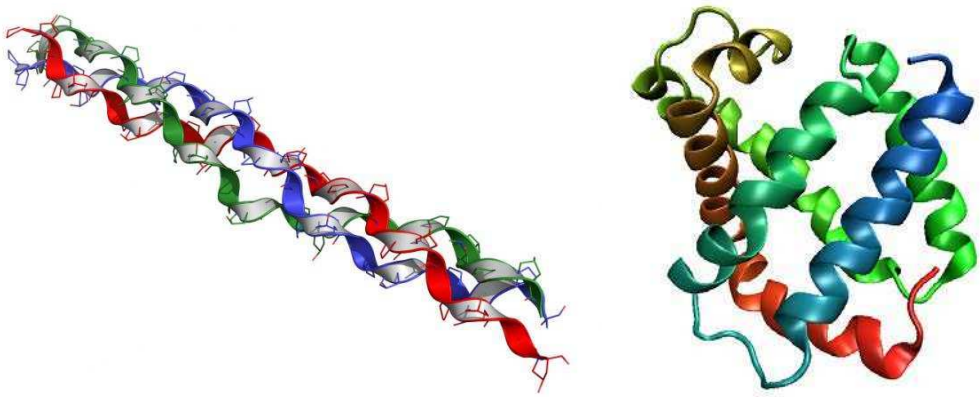


Figura 1.3.2.1: Visualización de dos estructuras de proteína. A la izquierda una proteína cuyo plegamiento es fibrilar y la la derecha una cuyo plegamiento es globular.

El plegamiento suele realizarse de manera tal que los aminoácidos apolares que componen la proteína estudiada se sitúan hacia el interior y los polares hacia el exterior, en medios acuosos. Esto provoca una estabilización por interacciones hidrofóbicas, de fuerzas de van der Waals y de puentes disulfuro (covalentes, entre aminoácidos de cisteína convenientemente orientados) y mediante enlaces iónicos (figura 1.3.2.2).

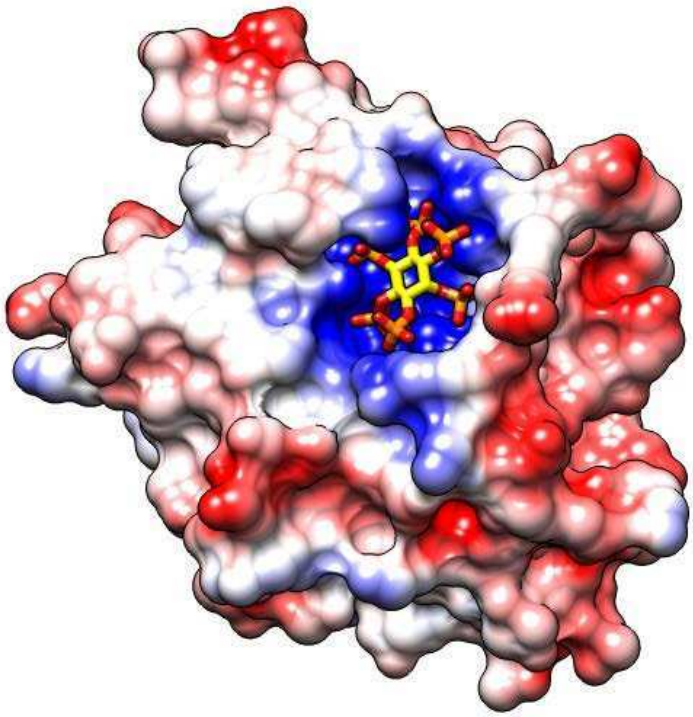


Figura 1.3.2.2: Visualización de la estructura de una proteína coloreando cada región en base a su densidad de carga, siendo las regiones más polares las de un rojo más fuerte y las regiones más apolares las de un azul más fuerte. Como puede observarse, los aminoácidos se organizan de manera tal que las zonas expuestas hacia el exterior son las más polares. Regiones apolares pueden constituir una zona expuesta al solvente con la consecuencia de constituir zonas de probable acoplamiento a compuestos que estén disponibles en el medio en el cual está inserta la proteína.

Se denomina dominio estructural a un elemento constitutivo (o unidad) de la estructura de las proteínas que estabiliza su plegado de manera independiente. Los dominios son, a menudo, seleccionados evolutivamente porque poseen una función característica de la biología de la proteína pertenecen (por ejemplo "dominio de unión a ADN").

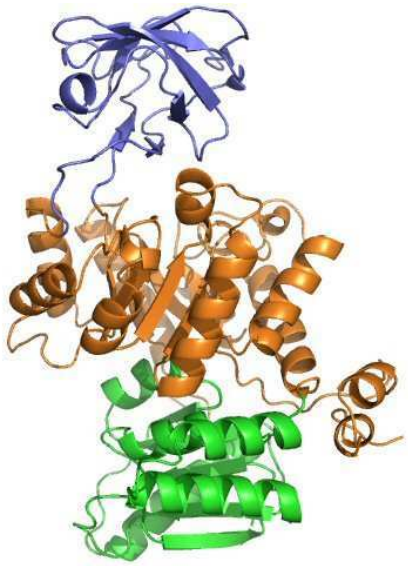


Figura 1.3.2.3: Visualización de la entrada del Protein Data Bank (PDB)[CITA] 1pkm, la cual está compuesto por tres dominios estructurales que han sido remarcados con distintos colores.

La estructura cuaternaria se encuentra definida por diferentes cadenas peptídicas que se pliegan formando dominios que interactúan de una manera particular formando multímeros (figura 1.3.2.3), el cual posee propiedades distintas a la de los monómeros que la conforman.

Alrededor del 90% de las estructuras de las proteínas conocidas actualmente han sido determinadas mediante cristalografía de rayos X[22]. Este método permite medir la densidad de distribución de los electrones de la proteína en las tres dimensiones del espacio, determinando de esta forma las coordenadas de los átomos relativas a las demás posiciones con certeza.

El Protein Data Bank (PDB) nace en el año 1971 en Brookhaven National Laboratory conteniendo solo 7 estructuras cristalográficas. Es actualmente el repositorio universalmente aceptado de estructuras tridimensionales de macromoléculas como proteínas, ácidos nucleicos y sus complejos: con lípidos, azúcares, como así también con

diversos ligandos, entre ellos compuestos tipo droga. En su última versión cuenta con ~115.000 estructuras depositadas.

La obtención de cristales y la resolución de estructuras a partir de experimentos de difracción de rayos X enfrentan una limitación práctica; en la medida en que la cantidad de datos de secuencias generados crecen de manera exponencial (figura 1.3.2.4), gracias a nuevas técnicas de secuenciación, la cantidad de estructuras crecen de manera lineal. Esta limitación, produce en la práctica una brecha que se amplía día a día, entre la cantidad de secuencias conocidas y aquellas para las que se conoce (de manera experimental) su estructura tridimensional.

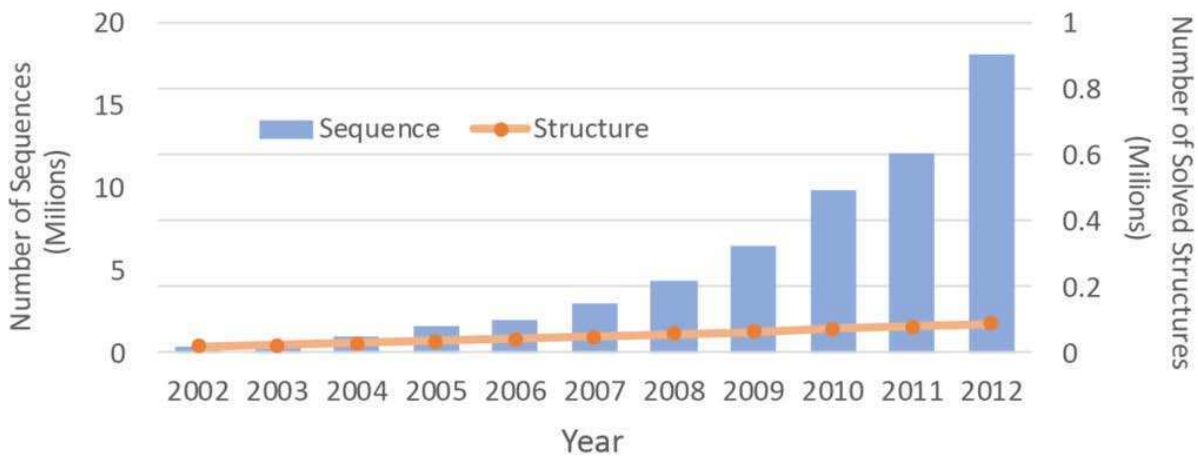


Figura 1.3.2.4: Gráfico que enfrenta el crecimiento exponencial de secuencias disponibles en bases de datos versus la cantidad de estructuras. Puede observarse el patrón exponencial de las secuencias y el crecimiento lineal de estructuras.

El uso de algoritmos para la determinación mediante métodos computacionales de estructuras de proteínas de la que sólo tenemos su información de secuencia es denominado modelado. Existen en la actualidad dos estrategias usadas mayoritariamente para modelar estructuras de proteína: modelado *ab-initio* y modelado por homología.

Los métodos *ab-initio*, que también pueden encontrarse bajo el nombre de métodos *de-novo,* comparten la estrategia de intentar generar la estructura de la proteína de interés basándose únicamente en principios físico-químicos y teniendo como dato de entrada únicamente la secuencia de aquella molécula que se intenta modelar[23]. Fundamentalmente lo que computa el algoritmo es, mediante el uso de algún potencial definido por el programa que estemos usando, las transiciones que ocurren del estado desplegado de la proteína hasta su estado plegado final mediante dinámica molecular o muestreo del tipo Monte Carlo. Estos métodos poseen la desventaja de requerir grandes tiempos de cómputo. Tal es así que existen, por ejemplo, iniciativas para modelar macromoléculas mediante estas técnicas de manera colaborativa como es Rosetta@Home donde el cómputo se distribuye entre diferentes usuarios.

Por otro lado, los métodos de modelado por homología[24] (o modelado comparativo) son computacionalmente mucho menos costosos. El aspecto fundamental de estos algoritmos es que incorporan información conocida sobre proteínas que ya tienen resuelta su estructura tridimensional. El modelado comparativo utiliza estructuras completas de proteínas conocidas como molde (usualmente denominadas *templados*) para construir a partir de éste el modelo de interés. Un esquema general se muestra en la figura 1.3.2.5 y en los métodos computacionales de esta tesis se desarrollará el concepto más en detalle.

Servirán como templado aquellas proteínas que puedan ser consideradas homólogas (que posean una identidad de secuencia tal, que en base a los criterios que cada algoritmo defina, la relación de homología pueda ser establecida). A medida que los datos aumentan, también aumenta el cubrimiento del espacio conformacional de estructuras y por ende el poder descriptivo de esta técnica.

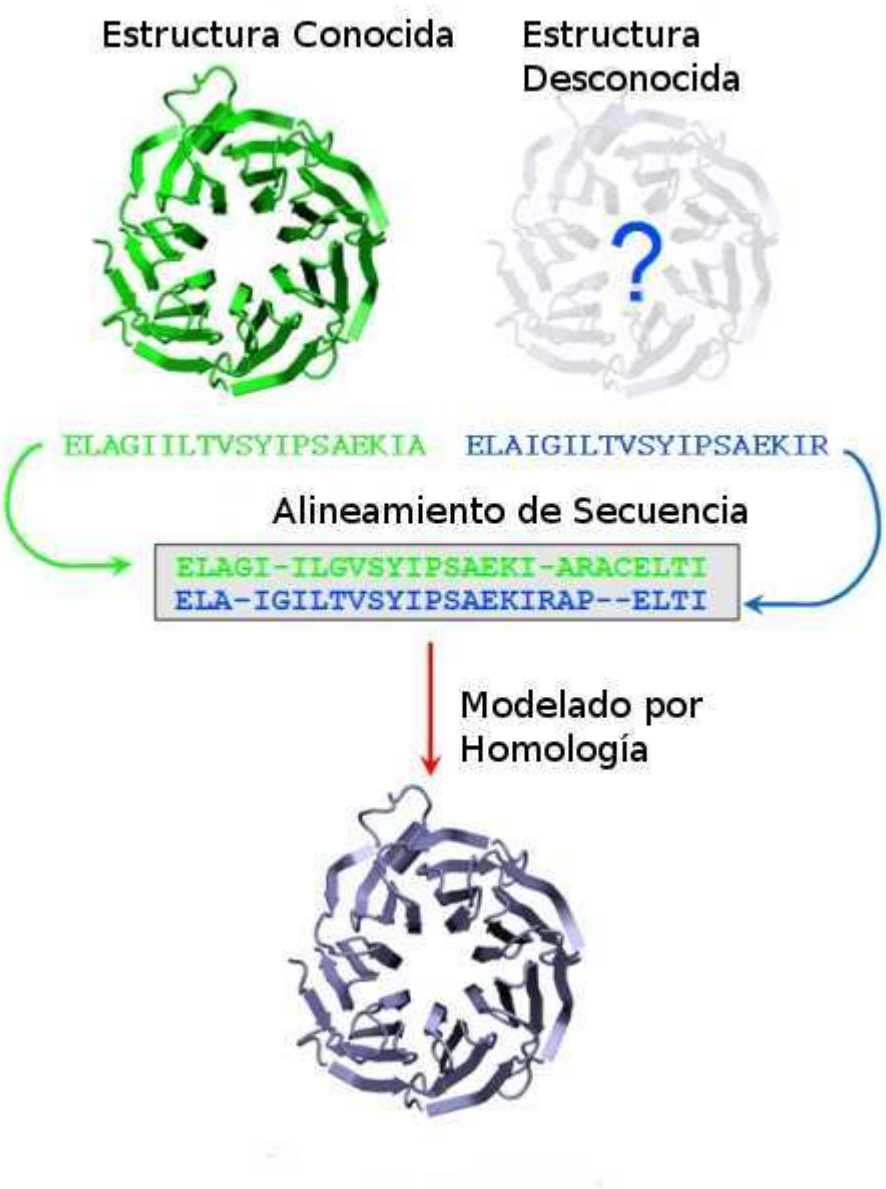


Figura 1.3.2.5: Esquema general del algoritmo de modelado por homología, Teniendo dos proteínas homólogas en secuencia, se usa como molde para generar la estructura desconocida aquella que está resuelta, intentando predecir los cambios estructurales generados por las diferencias de aminoácidos.

Cabe destacar que hay proteínas de las cuáles no se obtiene información alguna buscando en bases de datos, principalmente porque ninguna proteína que posea resuelta su estructura cumple con la condición de tener una identidad suficiente que permita que se

la use como templado, dejando como única posibilidad utilizar métodos *ab-initio*. La ventaja de estos métodos es que, a diferencia del modelado por homología, los modelos generados no se encuentran sesgados por las estructuras de proteínas actualmente resueltas, lo que nos da el potencial de generar modelos de calidad sobre moléculas para las cuales su plegado no se encuentra resuelto experimentalmente.

En esta tesis, la técnica de modelado por homología se ha ajustado a las necesidades presentadas a lo largo de los distintos desarrollos realizados: al tener que generar, por ejemplo, modelos para todas aquellas proteínas que no tienen su estructura resuelta en el Protein Data Bank, ninguna otra técnica nos hubiera permitido resolver un porcentaje importante de estructuras de un genoma en tiempos de cómputo razonables.

1.3.3 Determinación de propiedades derivadas de la estructura proteica

El insumo más importante de muchos de los métodos aplicados en el presente trabajo es la estructura en tres dimensiones de las proteínas, y ello se debe a las conclusiones que pueden obtenerse en función de las propiedades que podemos conocer y/o calcular a partir de ella.

Dada una estructura, una de las entidades que pueden ser calculadas sobre ella son los «bolsillos» que presenta. No existe una única manera de definir lo que es un bolsillo en una proteína (ver Métodos Computacionales), pero dicho de una manera coloquial, un bolsillo es una cavidad presente en la estructura, la cual presenta determinadas propiedades (volumen, polaridad, etc.) que lo hacen más o menos apto para que una molécula pequeña pueda "ingresar" en el mismo y unirse a la proteína (de manera no covalente), inhibiendo o modulando su función en algunos casos. En la figura 1.3.3.1 se muestra la estructura de una proteína unida a dos compuestos pequeños insertos en bolsillos presentes en la estructura.

Llamamos "drogabilidad estructural"[25,26] al factor o puntaje que determina esta capacidad de los bolsillos de unir compuestos tipo droga (definiremos qué es un compuesto tipo droga en la sección de métodos de esta tesis).

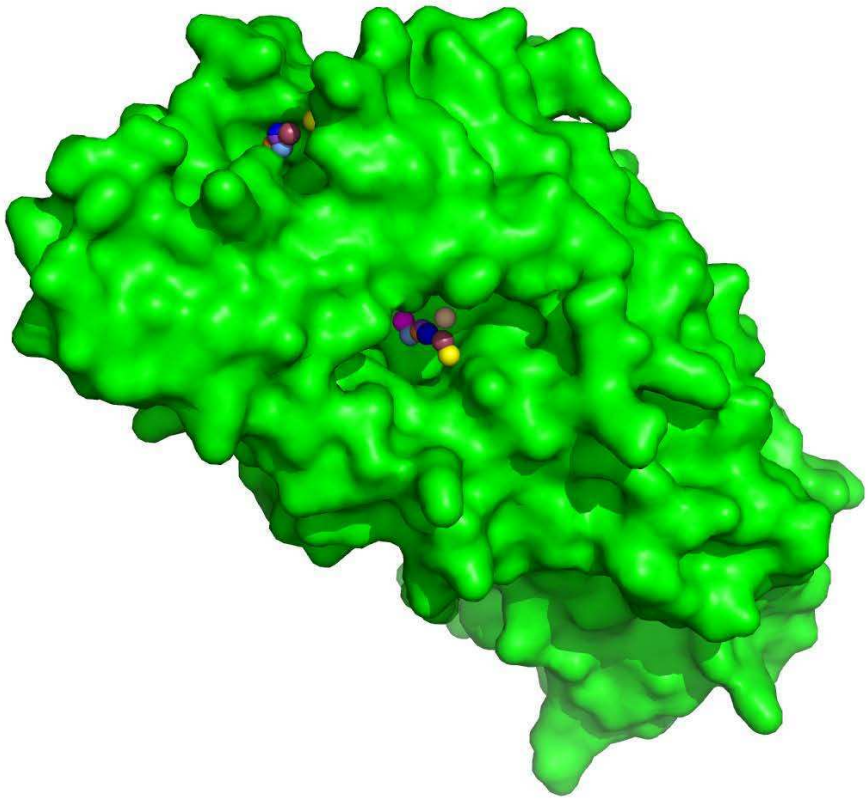


Figura 1.3.3.1: Estructura de una proteína visualizando en verse su superficie accesible al solvente. Puede visualizarse compuestos pequeños insertos en bolsillos unidos no covalentemente.

Una de las aplicaciones fundamentales y directas del procedimiento de encontrar bolsillos estructuralmente drogables es la de determinar el sitio activo de aquellas proteínas que presentan una actividad enzimática. Localizar el sitio activo y poder determinar sus propiedades estructurales resulta fundamental a la hora de diseñar compuestos que tengan buenas posibilidades de acoplarse a la proteína favoreciendo o inhibiendo la catálisis dependiendo del efecto que quiera lograrse.

Si bien existen diferentes estrategias para encontrar cavidades en la estructura proteica, la mayoría de los métodos computacionales existentes no relacionan esta información con las propiedades que debiera tener un compuesto para unirse a la cavidad encontrada. En resolver este problema se ha centrado uno de los desarrollos de esta tesis.

Otras técnicas que hemos utilizado en el presente trabajo son el uso de base de datos de sitios catalíticos[27] en conjunto con información de conservación de residuos, lo que permite extrapolar la ubicación de dichos sitios a proteínas que no lo tienen asignado.

Algunas de las entidades que resultan importantes de determinar conociendo la estructura proteica son las regiones que sirven de interfaz para la interacción proteína-proteína. Una interfaz de unión proteína-proteína puede constar de una amplia superficie con restricciones en la hidrofobicidad o polaridad de los aminoácidos que la componen. Entender dónde se encuentra localizada y con qué otras proteínas es capaz interactuar resulta muy importante para determinar si pueden afectarse estas interfaces y de qué forma (mediante una mutación u otra causa). Afectar la interfaz proteica puede generar efectos relacionados con potenciales cambios en la transducción de señales, la formación de complejos, fosforilación, etc. lo cual influye en el normal funcionamiento celular.

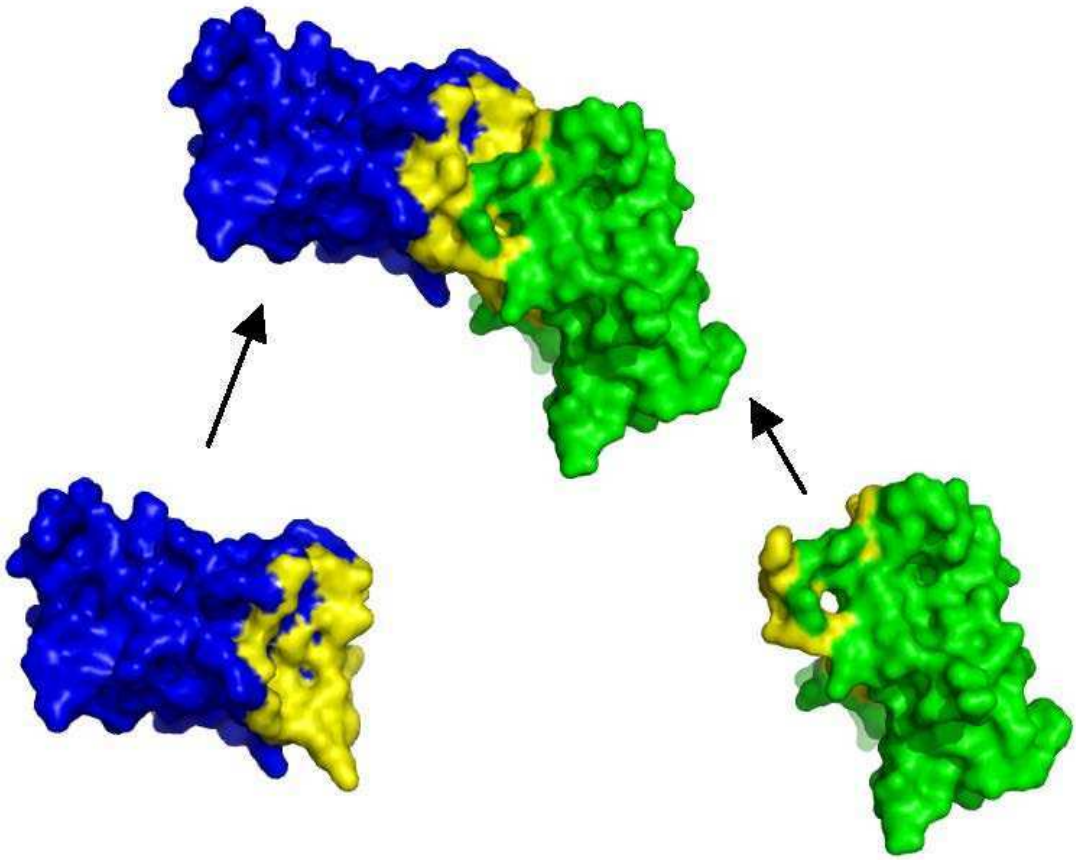


Figura 1.3.3.2: Estructura de dos proteínas que forman un complejo, visualizando en color amarillo la región de las mismas que constituyen la interfaz entre las mismas

Otras de las propiedades generales que resultarán de interés, pero que son particulares de un aminoácido dentro de la estructura son (entre otras):

→ El B-Factor, o factor de temperatura, que indica cuál es el desplazamiento promedio de un átomo con respecto a un valor promedio válido para el experimento mediante el cual se ha determinado la estructura. Un desplazamiento grande indica una mayor temperatura y por lo tanto la pertenencia a una zona potencialmente más móvil como puede ser un loop desordenado.

→ La el porcentaje de superficie accesible al solvente de la cadena lateral del aminoácido que está siendo analizado, valor que es útil para determinar si la posición en cuestión forma parte de la superficie de la estructura o del núcleo de la misma.

→ Las constantes de protonación y desprotonación de los aminoácidos con el fin de establecer si pueden o no, en determinadas condiciones de pH establecer enlaces no covalentes con determinadas moléculas.

→ La cantidad de átomos dadores y aceptores de puente de hidrógeno que un aminoácido puede establecer en la estructura, con el fin de establecer qué tipo de moléculas puede potencialmente interactuar con el mismo.

Cuando en la secuencia de una proteína se producen variaciones fruto de una modificación genética, los valores algunas o todas las propiedades que hemos mencionado pueden verse afectados, modificando o alterando de esta manera la función de la proteína (y consecuentemente su red de interacciones, etc.), lo que puede tener consecuencias a nivel celular y manifestarse en el organismo con algún síntoma, provocando potencialmente enfermedades.

En resumen, la estructura de una proteína no es solo un dato biológico, sino una fuente de datos sobre la cual se pueden obtener (calcular, leer o inferir) múltiples propiedades las cuales pueden vincularse con otras fuentes de datos y ser convertidas en valiosa información concerniente a diferentes aspectos (actividad enzimática, redes de interacciones, relación con enfermedades, etc.).

1.4 Bases de datos biológicas

Una base de datos, como concepto general, es un banco de información que contiene registros desglosados en propiedades y vinculados de una manera que permite relacionarlos como conjunto, ejercer sobre dicho conjunto búsquedas en función del valor de las propiedades, clasificarlos en función de filtros a las mismas, etc. En la actualidad, la

mayoría de las bases de datos se encuentran almacenadas de una manera digital porque esto permite su rápido procesamiento debido a la capacidad de cómputo de los dispositivos en la actualidad. Los programas que permiten interactuar de manera computacional con bases de datos digitales se denominan "gestores de bases de datos".

Podemos definir como base de datos biológica cualquier colección de información cuyos registros consistan en datos acerca de entidades biológicas. La misma puede provenir de experimentos científicos, literatura publicada, tecnología de experimentación de alto rendimiento, análisis computacional, etc.

Una base de datos biológica puede contener información de muy variadas áreas de investigación incluyendo genómica, proteómica, metabolómica, expresión génica mediante microarrays, filogenética y un largo etcétera. La información contenida en bases de datos biológicas incluye funciones, estructura y localización (tanto celular como cromosómica), efectos clínicos de mutaciones, así como similitudes de secuencias y estructuras tridimensionales de moléculas.

Suele hacerse en bioinformática la distinción entre bases de datos primarias y secundarias. Las primeras almacenan información que no ha sufrido procesamiento sino que es fruto directo del resultado de experimentos. Son ejemplos de bases de datos primarias GeneBank, UniProt[28], etc. Las bases de datos secundarias son, en cambio, el fruto del análisis y procesamiento de bases de datos primarias con el objetivo de generar nuevo conocimiento. Ejemplos de bases de datos secundarias son PFAM[29], CATH[30], etc. Un análisis más exhaustivo de diferentes bases de datos tanto primarias como secundarias puede encontrarse en el capítulo de Métodos Computacionales de esta tesis.

Una problemática común en las bases de datos biológicas es la integración de datos: los orígenes heterogéneos de los registros que pueblan las bases de datos biológicas y los problemas de heterogeneidad intrínsecos de la biología (por ejemplo: ¿cómo se numeran las posiciones de las secuencias de dos isoformas de una proteína?) plantean un problema

serio en el momento de vincular información depositada en distintas fuentes. Por ejemplo, uno de los recursos de integración que intenta vincular información de secuencia, estructura, función, taxonomía, etc. como es SIFT[31] es, al día de hoy, un recurso generado con una combinación de procesos automáticos y ayuda de la curación manual descripto esquemáticamente en la figura 1.4.1.

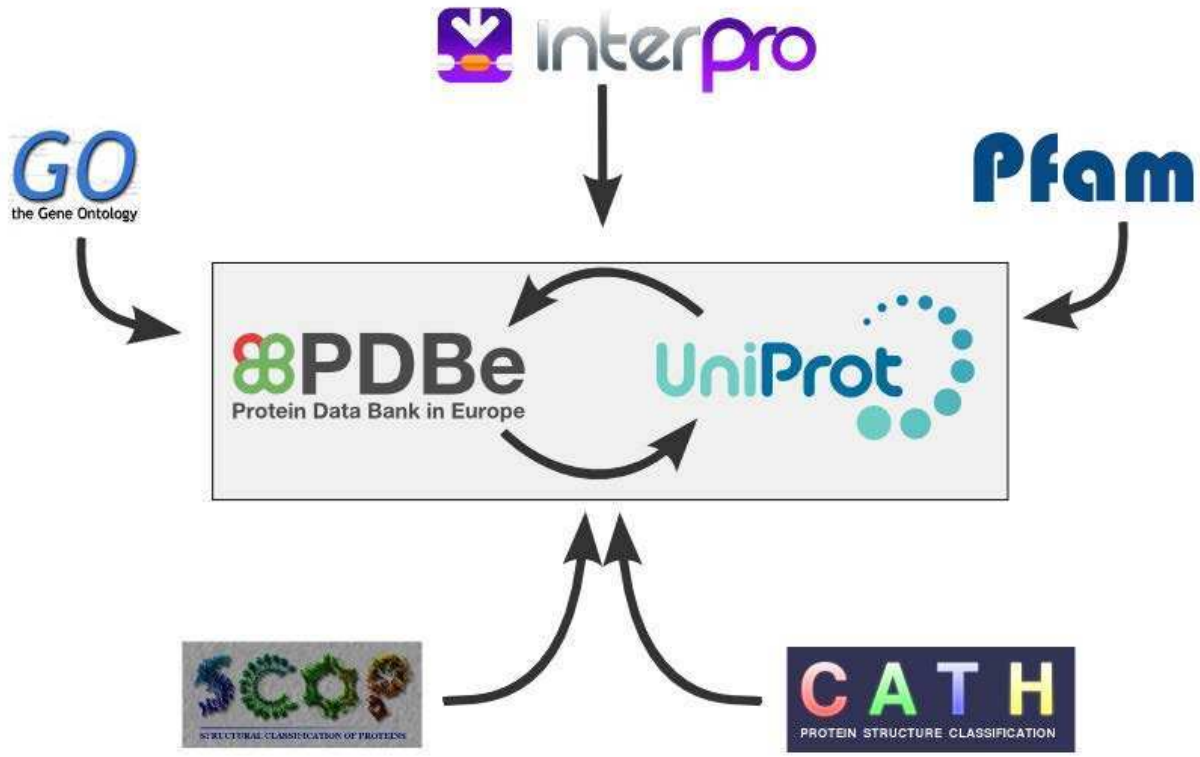


Figura 1.4.1: Esquema de procesamiento para la generación de la base de datos SIFT. En el centro, las bases de datos primarias a partir de las cuales se vinculan las numeraciones. Fuera del rectángulo central, las bases de datos secundarias cuyos registros son generados procesando la información primaria.

Extraer información vinculando distintas fuentes de datos es hoy en día uno de los problemas centrales en el campo de la bioinformática y es uno de los desafíos a los que nos hemos enfrentado en este trabajo.

Como ejemplo, la información proveniente de la estructura de una proteína, la cual aporta propiedades fisicoquímicas puntuales a nivel molecular, como puede ser las características de su sitio activo, puede ser combinada con información contextual como la esencialidad de la proteína para la supervivencia del organismo. Sin embargo, la heterogeneidad de las fuentes de datos, las diferencias de nomenclatura, etcétera, hacen que la integración (la cual puede aportar información muy valiosa), represente un desafío técnico. Inclusive, bases de datos de un mismo tipo de registros, como son las moléculas, pueden tener, debido a su motivación, formatos muy distintos. La base de datos ChEMBL[32], por ejemplo, para cada molécula almacena la información de sus ensayos, y posee una estructura bien distinta de la base de datos ZINC[33], en donde de cada compuesto interesa si puede ser adquirido o no y cuales son sus proveedores.

En los últimos años, debido al crecimiento exponencial en la cantidad de datos biológicos disponibles en repositorios de público acceso (secuencias genómicas y de proteínas, estructuras de proteínas, expresión génica, mutaciones, etc) los mismos han representado desafíos técnicos tanto desde el punto de vista del almacenamiento y acceso a los datos de una manera global y eficiente como una explosión en las posibilidades de extraer información y generar teorías y modelos a partir del uso de esos datos.

1.5 *Pipelines* de procesamiento y cálculo

Un *pipeline* es una palabra que sirve para designar un proceso que puede ser dividido en etapas independientes las cuales dependen unas de otras (figura 1.5.1). El concepto puede ser enmarcado en cualquier tipo de proceso, independientemente del campo en donde esta se utiliza (la ciencia, la industria, etc.). Racionalizar y estudiar la estructura de los procesos que conforman un *pipeline* tiene el sentido práctico de poder ordenar y optimizar el conjunto de tareas, hallar cuellos de botella focalizando esfuerzos en optimizar tareas críticas, etc.

Los *pipelines* computacionales son ya un lugar común en la investigación científica. Cualquier procesamiento de datos que involucre diferentes etapas dependientes unas de otras, definiendo un grafo dirigido y acíclico de ejecución, puede ser denominado *pipeline* (tubería en castellano).

Muchas de las herramientas desarrolladas en esta tesis son, en un sentido amplio, pipelines bioinformáticos, y un correcto desarrollo de los mismos, teniendo en cuenta cuestiones como la posibilidad de paralelizar cómputo y distribuirlo, hará de nuestros desarrollos herramientas más eficientes.

Figura 1.5.1: Representación mediante un grafo dirigido y acíclico de un *pipeline*, en este ejemplo uno de procesamiento de carne. El concepto de pipeline implica la ejecución de etapas identificables (procesos) e independientes que dependen unas de las otras formando un grafo dirigido y acíclico.

Un pipeline puede ser construído de diferentes maneras. Puede ser, por ejemplo, programado de forma ad-hoc, manejando el orden de ejecución desde el mismo programa, especialmente programado para ello. Esta estrategia puede ser útil si la complejidad del grafo que representa el pipeline no es muy grande, pero a medida que este crezca, crecerá también la necesidad de utilizar un marco de trabajo (framework) que permita manejar de

manera eficiente el flujo de los trabajos, su posible paralelización, la asignación de recursos y la fusión de resultados entre dos o más etapas para continuar con la ejecución etapas subsiguientes, etc.

Existen herramientas que permiten, mediante una interfaz gráfica, diagramar el procesamiento de un pipeline, pero tienen una orientación marcada a procesos de minería de datos en su gran mayoría. Un ejemplo de este tipo de herramientas es el framework Orange[34].

Otros frameworks permiten programar los pipelines de cómputo en lenguajes de programación que son los mismos en los que están programados cada uno de los pasos de cómputo mismo. Uno muy utilizado en el campo de la bioinformática es Ruffus[35], el cual tiene un diseño simple y versátil, que lo hace de útil aplicación para casos desde muy simples hasta muy complejos (algunos de los casos de éxito tienen más de 80 pasos de cómputo independiente). A lo largo de esta tesis hemos utilizado esta herramienta para etapas tanto de generación, recolección y digestión de datos para la creación de bases de datos como para etapas de cálculo.

Esta tesis, como hemos dicho antes, se vale del recurso de generar pipelines bioinformáticos para convertir y combinar datos y convertirlos en información de manera automática generando un recurso que ayuda en el análisis a diferentes tipos de especialistas que pueden ir desde un biólogo molecular buscando dilucidar mecanismos para tratar enfermedades bacterianas hasta un médico intentando diagnosticar los efectos de una mutación en la salud de un paciente.

1.6 Objetivos de este trabajo

El objetivo general del presente trabajo de tesis doctoral es el de aportar información que sea valiosa en el análisis a escala tanto genómica como proteica aplicada a dilucidar causas y plantear soluciones para el tratamiento de enfermedades (de origen bacteriano, genéticas,

etc.). Para lograr esto nos valdremos de herramientas informáticas que combinen, analicen y generen nuevos datos (cuando sea necesario), utilizando métodos existentes o desarrollando nuevos métodos computacionales.

Para cumplir con este objetivo general, planteamos los siguientes objetivos específicos:

1) Desarrollar una herramienta que asista en la elección de blancos proteicos en base a su drogabilidad estructural, la cual permita manejar datos a escala genómica (extraer blancos drogables dado el genoma de un organismo).

2) Desarrollar una herramienta que, dado un blanco proteico candidato, determine una lista de compuestos candidatos que posean buenas probabilidades de acoplamiento al mismo, afectando su función.

3) Desarrollar una herramienta que permita obtener de bases de datos de acceso público las mutaciones reportadas para una proteína de interés, y que permita analizar los efectos estructurales que tendrán tanto esas mutaciones como predecir el efecto de otras que resulten de interés en el momento del análisis y que no se encuentren descritas.

Un objetivo adicional e independiente en esta ha sido lograr que todos los desarrollos llevados a cabo sean reproducibles y que su grado de automatización sea el más grande posible, sin que esto afecte la calidad de los resultados obtenidos. Por eso mismo, nos hemos centrado en garantizar su accesibilidad mediante herramientas online, así como hemos puesto a disposición todo lo desarrollado en forma abierta para que cualquier usuario pueda tanto consultar como extender los desarrollos realizados.

2. Métodos Computacionales

2.1 Metodología Computacional

2.1.1 Bases de datos

Uno de los insumos fundamentales del que nos hemos valido en el presente trabajo para cumplir los objetivos planteados son las bases de datos. No es nuestro objetivo entrar en digresiones ni particularidades técnicas acerca de los distintos tipos de bases de datos existentes, pero sí remarcar características de distintas bases de datos creadas a lo largo de este trabajo y de aquellas con las que se ha interactuado.

Las bases de datos remotas con las que hemos interactuado en su totalidad tienen, además de una interfaz web, como la que se muestra en la figura 2.1.1.1, a partir de la cual se pueden consultar sus registros, una manera de acceder denominada "programática". Esto último quiere decir que, a través de un protocolo de red, se pueden consultar y descargar registros al sistema de archivos local.

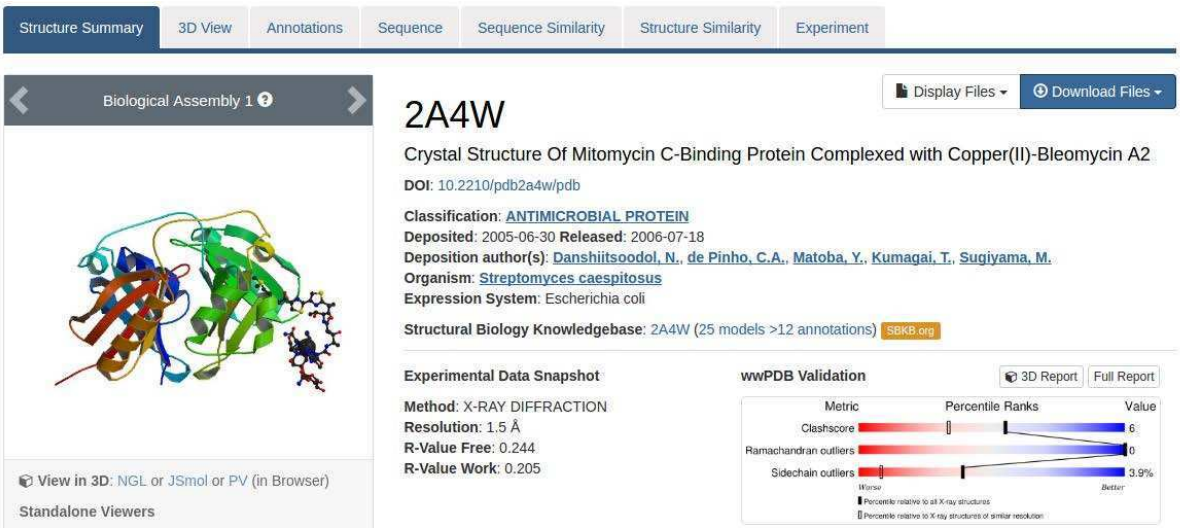


Figura 2.1.1.1: Vista de un registro del Protein Data Bank (2A4W) desde la interfaz web de la base de datos. La información puede ser visualizada de manera amigable pero su acceso no es práctico.

Poseer una manera estandarizada de acceder a los registros y también de realizar búsquedas en las bases de datos es un elemento fundamental a la hora de poder construir bases de datos locales y extraer información de manera masiva. En la figura 2.1.1.2 se visualiza un registro de la base de datos del PDB de manera "raw" (cruda), el cual puede ser descargado y procesado de forma programática.

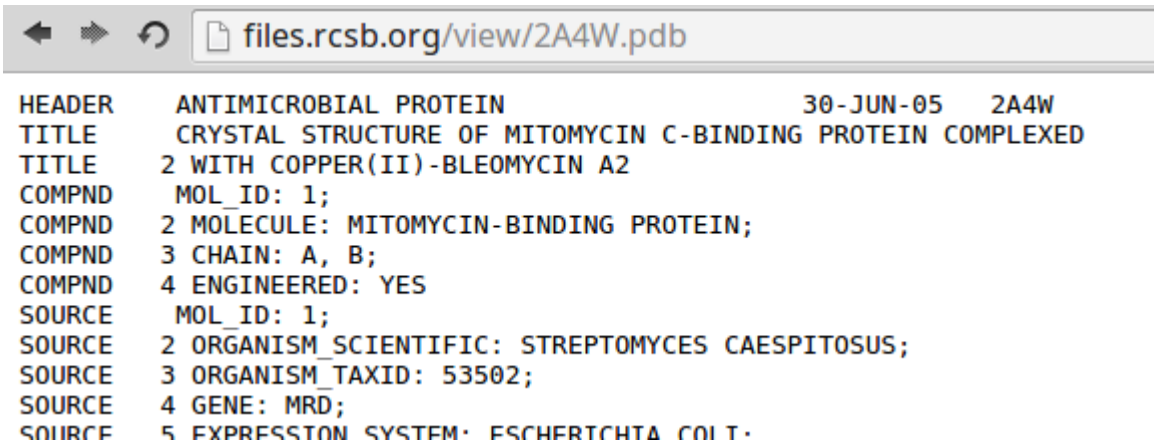


Figura 2.1.1.2: Vista de un registro del Protein Data Bank (2A4W) accediendo al sistema de archivos de la base de datos. La visualización del mismo en el caso de la figura se hace desde un navegador, pero ese archivo puede descargarse descargarse accediendo a la URL del mismo con comandos como **curl** o **wget**, permitiendo su almacenamiento de manera local y su tratamiento de manera sistemática.

Cada una de las entidades bioquímicas que analizamos en el presente trabajo tiene un formato de archivo particular que permite su adecuado almacenamiento y procesamiento de manera completa. Algunas bases de datos (Uniprot, PDB, etc.) poseen una interacción programática simple en la cual mediante la URL del registro particular obtenemos el archivo deseado y el procesamiento del mismo debe hacerse de manera local. En otras, como por ejemplo ZINC, las características avanzadas del acceso programático que ofrecen, nos permiten hablar de que interactuamos con una API (por sus siglas en inglés *Application Programming Interface*). Estas bases de datos poseen un lenguaje definido y un método de interacción que permite que parte del procesamiento se realice en el lado del servidor de la base de datos (para el caso de ZINC: búsquedas mediante distintos códigos, búsquedas por similaridad, acceso a compuestos en distintos formatos, etcétera).

Un ejemplo de protocolo de acceso programático a este tipo de bases se muestra el algoritmo 2.1.1.1 mediante pseudocódigo.

```
función ObtenerObjeto(códigoBD):

  si fechaRegistroEnWeb(códigoBD) > fechaRegistroEnDisco(códigoBD)
    o no existeRegistroEnDisco(códigoBD):
      archivoObjeto = ObtenerArchivoWeb(códigoBD)
      archivoObjeto.guardar()

  objetoBioquímico= CargarPropiedadesDeObjeto(archivoObjeto)

  devolver objetoBioquímico
```

Algoritmo 2.1.1.1: Pseudocódigo del algoritmo de obtencion de registros de las bases de datos remotas. Cada una de las funciones que son llamadas se explica con su nombre. La implementación se discutirá en la sección 2.1.2.

Una vez guardados los datos localmente, estos son utilizados para construir una base de datos local, la cual permite procesar y los datos de la manera que fuere necesario. Entre estas bases locales, podemos destacar dos tipos principales: las bases de datos relacionales, y las no relacionales.

Las bases de datos relacionales son la tecnología más extendida en el área de la computación. En ella se posee un conjunto de tablas, las cuales tienen campos que describen cada una de las propiedades de un objeto. Hay campos (por lo general uno por tabla) que funcionan como clave y su objetivo es representar unívocamente como identificador a un registro. Dos o más tablas se relacionan entre sí mediante los campos

clave eliminando de esta manera la redundancia. El esquema completo de las bases de datos suele realizarse mediante esquemas de representación denominados UML[36] (como en la figura 2.1.1.3) , los cuales vinculan las tablas relacionadas y muestran los campos de cada una de ellas.

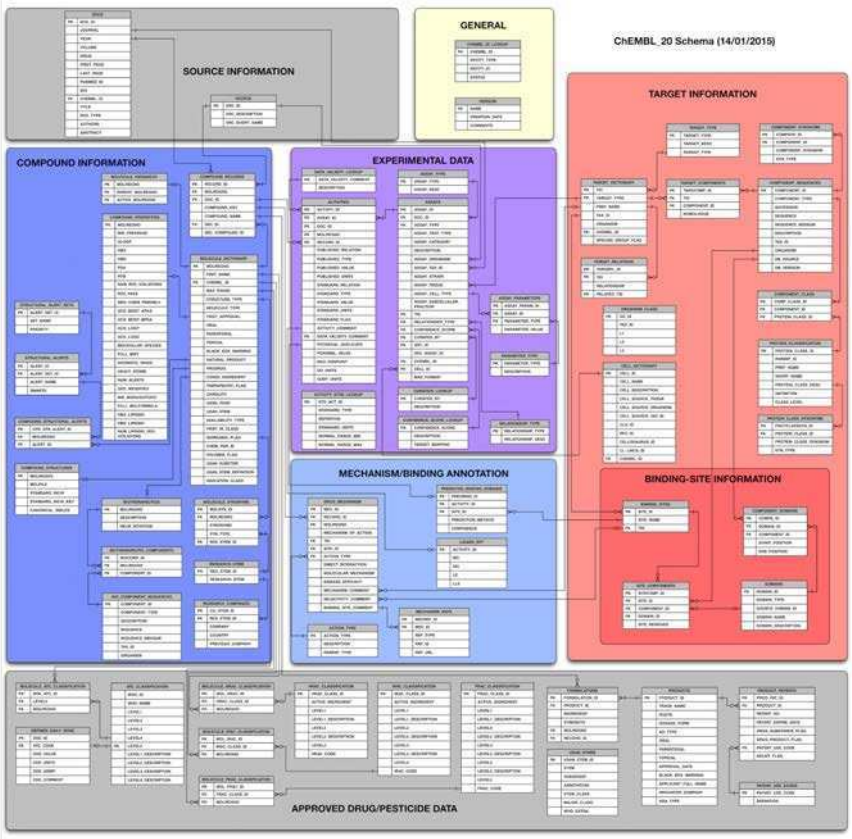


Figura 2.1.1.3: La base de datos de compuestos ChEMBL, además de ser consultada vía Web, puede ser descargada íntegramente como una base de datos relacional para ser usada con la tecnología MySQL[37]. Por ejemplo, una molécula (tabla compound) puede estar reportada como sitio de unión (binding site) de muchas proteínas (targets) y con solo almacenar la clave de las dos registros que forman un binding site habremos establecido la relación y eliminando la redundancia.

En cambio, las bases de datos no relacionales tienen la característica de almacenar colecciones de objetos de manera secuencial sin relacionar los registros entre sí. De esta

manera, cada una de las entradas de la base de datos contiene la información completa del registro que almacena. Por seguir el ejemplo de la figura 2.1.1.3, si lo que se tiene son colecciones de moléculas, cada una de ellas contendrá la descripción completa de las proteínas con las que interactúe. Si muchas moléculas interactúan con la misma proteína, cada vez se almacenarán todas sus propiedades, generando redundancia.

El beneficio de las bases de datos no relacionales es la posibilidad de paralelizar el tratamiento de los registros, ya que si no es necesario establecer relaciones entre distintas colecciones, el registro completo puede cargarse en un nodo (sea este un procesador, una computadora en un cluster, etcétera) y procesarse de manera independiente y paralela a los otros. En las bases de datos relacionales, si una consulta está vinculando dos tablas, se genera un bloqueo (lock) de las tablas consultadas, lo cual dependiendo del problema a tratar puede resultar en una pérdida de rendimiento significativa.

Un ejemplo de bases de datos relacional está descrito en la figura 2.1.1.3, en el que vemos entidades que se relacionan, como pueden ser sitios de unión y proteínas. Un típico ejemplo de base de dato no relacional es el Protein Data Bank, en donde cada una de las entradas (archivo .pdb) contiene toda la información de registro dentro del mismo, sin repartir la información heterogénea que se encuentra dentro del mismo en múltiples tablas. Si existiera, por ejemplo, una tabla de "átomos" y otra de "residuos" vinculadas por un campo identificador, estaríamos hablando de una base de datos relacional.

Las bases de datos relacionales y no relacionales no son ventajosas o inconvenientes en sí mismas sino que sirven para resolver situaciones de distinta naturaleza. En el presente trabajo nos hemos valido de ambas como herramienta para resolver problemas puntuales con la tecnología correspondiente de manera óptima.

2.1.2 Desarrollo de una librería bioinformática

El desarrollo fundamental que atraviesa toda esta tesis es la de una librería bioinformática que permite interactuar de manera transparente, desde el lenguaje de programación Python[38], con tres entidades fundamentales: objetos, bases de datos y algoritmos.

Un objeto es una entidad en un lenguaje de programación, la cual posee atributos (características, datos) y métodos asociados(comportamientos, algoritmos). En la figura 2.1.2.1, por ejemplo, manejamos un objeto de tipo "OnlineOrf", el cual pertenece a nuestra librería Python: el mismo, en este caso, representa un registro de la base de datos UniProt. Posee atributos, como por ejemplo su código, su nombre y su secuencia, y métodos como pueden ser "descargar todos los cristales que reresenten una estructura de esta proteína".

Los objetos "OnlineOrf", siguiendo el ejemplo, pueden ser creados por el usuario desde el lenguaje de programación con su código unívoco de identificación y la librería se encargará de conectarse con la base de datos correspondiente para verificar si tiene que descargar el archivo correspondiente al registro consultado, y posteriormente leer el archivo en el formato que corresponda para instanciar el objeto en memoria (en la figura 2.1.2.1, instanciar es definir los atributos del objeto "orf" con los datos correspondientes al registro P01111 de la base de datos UniProt).

```
1 # Agregamos al sistema de librerias la desarrollada en esta tesis
2 import sys
3 sys.path.append('/home/leandro/Dropbox/workspacesbg/sbg/')
4
5 # Cargamos el modulo de manejo de ORFs (interaccion con UniProt
6 from sbg.orf.Orf import OnlineOrf
7
8 orf = OnlineOrf("P01111")
9 print orf.name
```

```
<terminated> /home/leandro/Dropbox/workspacesbg/sbg/sbg/scripts/OTHERS/PruebaTesis.py
GTPase NRas
```

Algoritmo 2.1.2.1: Código y salida de consola fruto de su ejecución en el entorno de programación Eclipse[39] en el cual se instancia un objeto de tipo ORF y se consulta por su nombre para imprimirlo por pantalla.

Como puede observarse, los mecanismos de descarga del archivo de internet(cuando fuere pertinente), de lectura del código XML[40] conteniendo todas sus propiedades y su almacenamiento en memorias son transparentes para quien programa utilizando la librería.

En cuanto a las bases de datos locales y a los algoritmos, la idea fue la de programar "manejadores" (handlers) en los cuales se definen los parámetros de interacción (en el caso de las bases de datos lo que se está buscando y en el caso de los algoritmos los parámetros de entrada, y una vez definido se llama a la función "ejecutar" (run) lo cual habilita a leer la salida que, nuevamente, dependiendo del caso, es un tipo de objeto con las propiedades pertinentes dependiendo del caso. A modo de ejemplo, en el Algoritmo 2.1.2.2, el manejador del algoritmo FPocket[41], ejecuta la búsqueda de cavidades recibiendo como parámetro un objeto de tipo estructura de proteína sobre la cual se ejecutará el algoritmo.

```python
# Importamos los modulos necesarios
from sbg.structure.Structure import Structure
from sbg.pocket.FPocket import FPocket

# Creamos una estructura
st = Structure("2dnw")
# Instanciamos el algoritmo con su directorio de salida
fp = FPocket("/home/leandro/", st)
# coremos
fp.run(savePockets=True)

# para cada pocket
for pocket in fp.pockets:
    # para cada propiedad del pocket
    for property in fp.pockets[pocket].properties_dict:
        # Imprimimos la propiedad
        print pocket, property, fp.pockets[pocket].properties_dict[property]
```

```
0 Pocket Score 34.8826
0 Polarity Score 9
0 Real volume (approximation) 1377.1096
0 Volume Score 3.6667
0 Local hydrophobic density Score 60.6429
0 Mean alpha-sphere SA 0.5172
0 Proportion of apolar alpha sphere 0.5091
0 Number of apolar alpha sphere 84
0 Mean alpha-sphere radius 3.9246
0 Hydrophobicity Score 26.7619
0 Mean B-factor 0.0000
0 Charge Score 0
0 Drug Score 0.7557
0 Number of V. Vertices 165
1 Pocket Score 8.8515
1 Polarity Score 6
1 Real volume (approximation) 738.2051
1 Volume Score 3.6250
1 Local hydrophobic density Score 18.0000
1 Mean alpha-sphere SA 0.5977
```

Algoritmo 2.1.2.2: Código y salida de consola en donde se instancia un objeto del tipo estructura (Structure) y se lo pasa al manejador del algoritmo *fpocket*(ver Métodos Computacionales). Luego de ejecutar el algoritmo, se muestra número de pocket, propiedad y valor de la propiedad para cada pocket por consola. Nuevamente, la llamada de sistema al algoritmo, el almacenamiento de la salida en disco, su lectura y procesamiento son transparentes para el usuario de la librería.

En este caso, la salida del algoritmo es representada en nuestra librería con objetos del tipo "Pocket" los cuales pueden ser manipulados y/o utilizados a su vez como entrada de posteriores etapas de procesamiento.

La librería desarrollada ha sido utilizada en cada uno de los desarrollos de esta tesis y está disponible para su descarga en:

https://www.dropbox.com/sh/ek5d5sv4vh048o8/AAB0H1Na2CH08EFUnJrnTthja?dl=0.

Entre sus funcionalidades más importantes, descritas en la tabla 2.1.2.1, se encuentran las de interactuar de manera transparente para el usuario con todas las bases de datos que fueron de utilidad en este trabajo (UniProt, PDB, PFam, etc.) y también con los algoritmos de procesamiento realizados por terceros y utilizados en esta tesis (explicados más adelante).

Entidad	Interactúa con	Tipo de Entidad
Orf	UniProt	Objeto
ProteinList	archivos .fasta	
Structure	PDB	Objeto
Family	PFam	Objeto
Compound	PDB y ChEMBL	Objeto
Pocket	archivos .pdb	Objeto
BlastHandler	blast	Algoritmo
CD-HITHandler	CD-HIT[42]	Algoritmo
HMMerHandler	hmmer3[43]	Algoritmo
rDockHandler	rDock	Algoritmo
foldXHandler	foldx4[44]	Algoritmo
ModellerHandler	modeller	Algoritmo
DBHandler	MySQL, SQLite[45] y MongoDB[46]	Base de Datos

Tabla 2.1.2.1: Algunas de las entidades que pueden ser manejadas con la librería desarrollada manteniendo la coherencia de encapsulación descrita.

La librería desarrollada puede ser comparada en funcionalidad con otras de uso extendido como BioPython[47], BioJava[48], BioPerl[49], etc. las cuales podrían haber sido utilizadas, pero sin mantener una cohesión que permita su integración correcta a lo largo de todos los desarrollos de esta tesis.

2.1.3 Programación de *pipelines* bioinformáticos

Todos los desarrollos realizado en el presente trabajo constituyen herramientas *in silico*, las cuales a través del procesamiento de datos generan información de distintos tipos. En cada uno de estos desarrollos el cálculo requiere varias etapas, algunas de las cuales dependen de otras pero no de todas, permitiendo que algunos cálculos puedan ejecutarse paralela y asincrónicamente, definiendo un grafo dirigido y acíclico de ejecución.

Algunas de las características deseadas para la ejecución de estos grafos o pipelines de procesamiento son las siguientes:

- Que puedan ser ejecutadas en paralelo tareas independientes.
- Que si una tarea falla, las tareas independientes terminen de ejecutarse y que el grafo de ejecución pueda ser retomado a partir de la tarea que falló.
- Que la sintaxis para definir los pipelines sea simple y clara.

La librería Ruffus disponible para el lenguaje de programación Python nos ha sido de mucha utilidad para crear pipelines de procesamiento de datos. En los mismos, se crean funciones las cuales leen datos de entrada y definen datos de salida, que a su vez son leídos por uno o muchos procesos de dependan de esa tarea, creando de esta manera el grafo mencionado.

A modo de ejemplo, en la figura 2.1.3.1, la cual describe las distintas operaciones que pueden componer el grafo, el pipeline se inicia originado con los parámetros iniciales de la corrida. Los mismos generan una multiplicidad de datos que son procesados (transformados) por procesos, los cuales luego son combinados para generar datos finales en un último nodo de condensación de resultados.

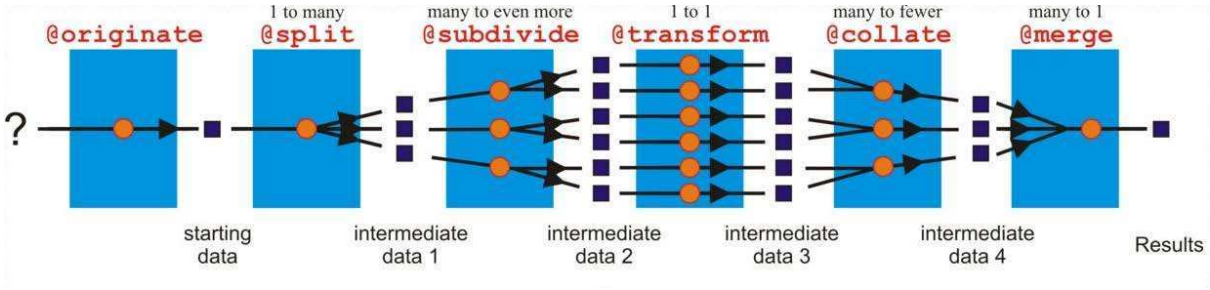


Figura 2.1.3.1: Tipos de funciones que pueden definirse como parte de un pipeline desarrollado con la librería Ruffus.

Por ejemplo, en el capítulo 3 de esta tesis, veremos cómo una mutación sobre una proteína debe ser analizada sobre todos los cristales donde ésta puede ser "mapeada". Diagnosticar su efecto requerirá entonces calcular propiedades que tienen que ser contextualizadas dentro del cristal para, a posteriori, conociendo los diferentes valores de las propiedades calculadas en los diferentes cristales, clasificarla a nivel proteína. En este sentido, la Figura 2.1.3.2, muestra para el ejemplo descrito la utilización de cada una de las operaciones provistas por la librería Ruffus.

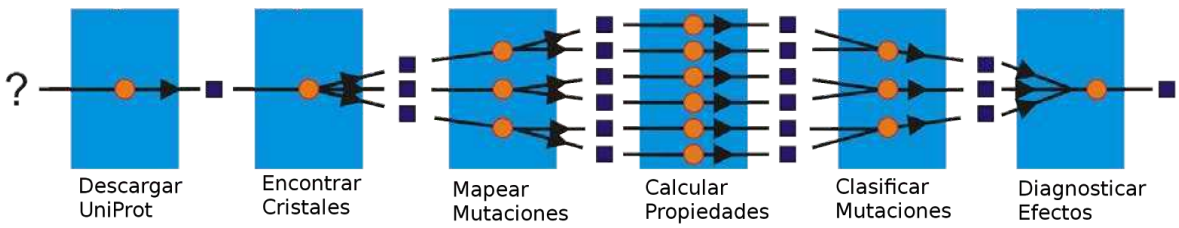


Figura 2.1.3.2: Aplicación de cada una de las operaciones provistas por Ruffus al problema de diagnosticar efectos de mutaciones en proteínas.

Esta librería se encarga de manera automática de monitorear la correcta finalización de cada una de las tareas, dando lugar a la ejecución de las subsiguientes, permitiendo que su ejecución sea distribuida entre múltiples nodos procesando registros y datos en paralelo.

Otra de las utilidades que la librería posee es la de poder dibujar los grafos de ejecución para entender de una manera visual cómo es que se ejecutó, y dónde falló si corresponde, el pipeline general. Un ejemplo se muestra en la figura 2.1.3.3, la cual posee distintos nodos que representan procesos en distintos estados de ejecución, mostrando de costado la descripción de qué significa la representación de cada estado. En el mismo se poseen dos tareas finales o terminales, y solo necesita recalcular una parte del proceso, la cual permite arribar a uno de esos estados, sin afectar (ni destinar tiempo de cálculo) a las tareas que permitieron arribar a la otra terminal.

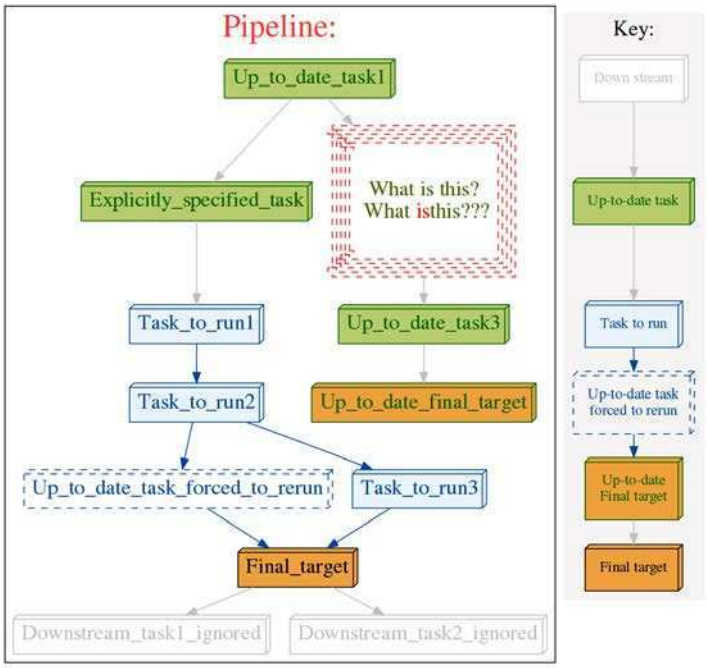


Figura 2.1.3.3: Ejemplo de un pipeline de ejecución que tiene dos funciones que funcionan como final generando resultados. Una de ellas está actualizada y la otra necesita volver a ser ejecutada para generar resultados actualizados.

En resumen, la librería Ruffus nos aporta un entorno de trabajo que garantiza la replicabilidad de cada uno de los procesos programados, su extensibilidad a nuevos datos de entrada con un esfuerzo bajo y su ejecución de manera eficiente.

2.1.4 Programación de servidores web

Todos los desarrollos presentados en este trabajo están disponibles para el uso de la comunidad a través de servidores Web. Esta estrategia de comunicación nos ha resultado particularmente útil para compartir los resultados obtenidos tanto dentro de nuestro grupo de trabajo, como con colaboradores y colegas.

En el capítulo tres de esta tesis, el desarrollo realizado consta de una base de datos que solo presenta la información una vez que esta fue calculada para un organismo entero, sin permitirle al usuario realizar ejecuciones. La misma fue desarrollada utilizando la tecnología Spring[50] y el lenguaje de programación Java. En los capítulos cuatro y cinco, en los cuales, además de consultar los datos existentes el usuario puede generar nuevos trabajos (jobs) que implican la ejecución de pipelines, hemos elegido la librería Bottle[51] de python.

Independientemente de la librería utilizada, la estructura subyacente en cada uno de los servidores web es la misma: el *backend* es el nombre que reciben los módulos programados para trabajar en directa conexión con la base de datos y lanzar los procesos que sean necesarios para cargar resultados. Estos son los módulos programados en Bottle y Spring. Por otro lado, el *frontend* es una respuesta que se produce en forma de página web, de archivo, o aquello que el usuario haya solicitado (y el servidor esté en condiciones de brindarle). La programación del *frontend* ha sido llevada a cabo utilizando HTML[52], JavaScript[53], CSS[54] y Bootstrap[55].

En la figura 2.1.4.1, se muestra cómo el usuario de un servidor web interactúa únicamente con la interfaz de usuario (frontend) la cual es programada como una entidad independiente de la capa de manejo de datos y procesos (backend).

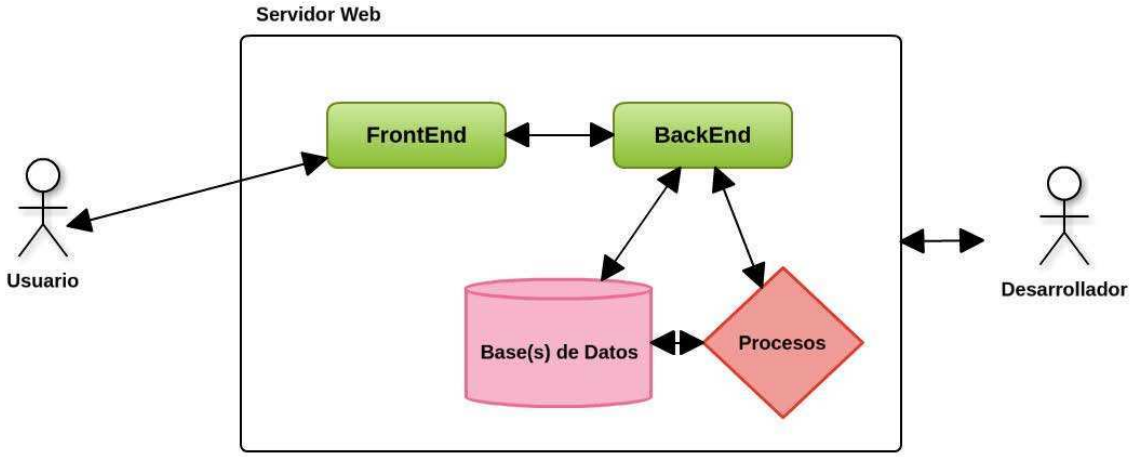


Figura 2.1.4.1: Esquema para la construcción de servidores web. Cada uno de los desarrollos producto de esta tesis tiene un servidor web mediante el cual se puede o bien consultar la información o bien procesar nuevos trabajos.

2.2 Alineamiento de secuencias

El alineamiento de secuencias es una de las herramientas fundamentales de las cuales nos hemos valido en este trabajo, y que tiene una variedad de aplicaciones que ya se han discutido en el capítulo introductorio. Previo a entrar en las particularidades de los algoritmos utilizados, es necesario introducir un par de elementos relacionados con esta técnica, como ser las matrices de sustitución y los dot plots.

El proceso de evolución, como es sabido, introduce mutaciones de una generación a la siguiente en el código genético de los organismos. No todas las mutaciones en el ADN se traducen en cambios aminoacídicos en las proteínas, las que sí los producen se denominan mutaciones no sinónimas. Estos cambios tendrán, de acuerdo a su naturaleza, diferente impacto en la función proteica, y por consiguiente en la biología del organismo. De acuerdo

a cómo los mismos afecten el fitness del organismo, estos estarán más o menos representados en la siguiente generación y, a la larga, se verán representados en mayor o menor medida en el conjunto de secuencias que representan el linaje. En resumen, cambiar un aminoácido por otro en la secuencia de una proteína tendrá distinto costo evolutivo dependiendo de qué aminoácido es reemplazado por qué otro y en dónde ocurre este cambio.

Para cuantificar ese costo de reemplazar un aminoácido por otro, es que existen las denominadas matrices de sustitución de las cuales se valen los algoritmos de alineamiento. Las mismas pueden considerarse tablas de doble entrada, donde la posición (i, j) de la misma representa el costo de sustituir el aminoácido i por el j. Existen en la actualidad básicamente dos tipos de matrices de sustitución utilizadas por los algoritmos de alineamiento. Las matrices BLOSUM y las matrices PAM[56]. Ambas matrices pueden ser consideradas de log-probabilidad, en donde las posiciones indican cuán probable es observar un cambio de aminoácido en función de lo que se observa empíricamente:

$$a_{ij} = log\ \frac{p_{ij}}{p_i * p_j} = log\ \frac{frecuencia\ observada}{frecuencia\ esperada}$$

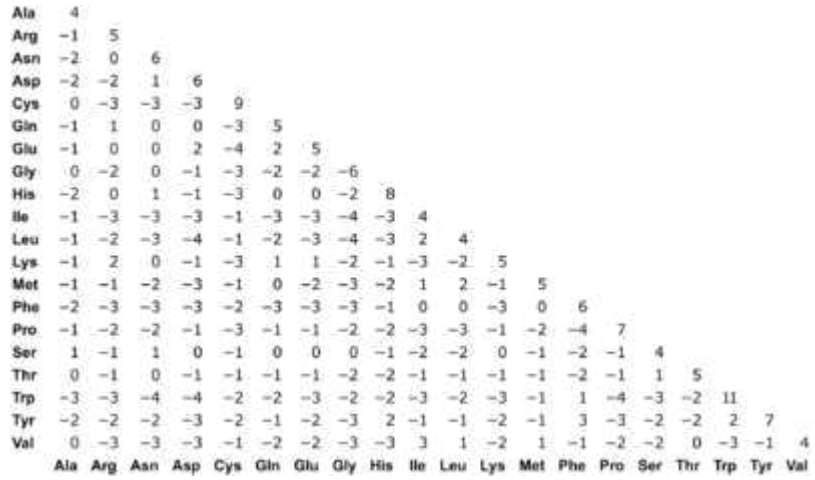
Aunque las dos matrices nombradas sean matrices de log-probabilidad, son distintas fundamentalmente en cuanto a su construcción.

Las matrices PAM (Point accepted mutation, o mutación puntual aceptada) se determinan observando las diferencias en proteínas relacionadas de manera cercana (similitud>85%). La matriz PAM1 expresa el radio de sustitución que debe esperarse si el 1% de los aminoácidos sufren sustituciones. Por propiedades algebraicas de las matrices y por cómo está constituída, se cumple además que:

$$PAM_N = (PAM_1)^N$$

En otras palabras, las matrices PAM superiores a 1 extrapolan la probabilidad original observada estadísticamente, en N pasos evolutivos, asumiendo que los mismos son independientes. Las matrices que suelen utilizarse en los algoritmos de alineamiento son PAM30, PAM70 y PAM250, las cuales permiten relacionar secuencias de proteínas con distintos niveles de divergencia evolutiva creciente.

Las matrices BLOSUM (BLOck Substitution Matrix), en cambio, calculan sus probabilidades basadas en alineamientos de proteínas evolutivamente divergentes, observando bloques de secuencias conservadas encontradas en múltiples alineamientos de proteínas. Se asume que estas secuencias conservadas tienen que tener una importancia funcional dentro de las proteínas relacionadas. Con el fin de reducir el sesgo provocado por secuencias cercanamente relacionadas, los segmentos de un bloque que posean una identidad secuencial mayor a un determinado umbral fueron agrupadas en un único bloque. De esta manera, por ejemplo, para la matriz BLOSUM62, este umbral de identidad de secuencia de los bloques se fijó en el 62%. Se consideraron, entonces, las sustituciones observadas dentro de estos grupos, para determinar los valores de la matriz de identidad. Las matrices BLOSUM de numeración alta (usualmente BLOSUM80) se utilizan entonces para alinear secuencias cercanamente relacionadas, mientras las de números más bajos son útiles para alinear secuencias más divergentes. La matriz BLOSUM62 suele utilizarse para detectar similitudes en secuencias distantes, y esta es la matriz usada por defecto aplicaciones de alineamiento, como *blast*.



	Ala	Arg	Asn	Asp	Cys	Gln	Glu	Gly	His	Ile	Leu	Lys	Met	Phe	Pro	Ser	Thr	Trp	Tyr	Val
Ala	4																			
Arg	−1	5																		
Asn	−2	0	6																	
Asp	−2	−2	1	6																
Cys	0	−3	−3	−3	9															
Gln	−1	1	0	0	−3	5														
Glu	−1	0	0	2	−4	2	5													
Gly	0	−2	0	−1	−3	−2	−2	−6												
His	−2	0	1	−1	−3	0	0	−2	8											
Ile	−1	−3	−3	−3	−1	−3	−3	−4	−3	4										
Leu	−1	−2	−3	−4	−1	−2	−3	−4	−3	2	4									
Lys	−1	2	0	−1	−3	1	1	−2	−1	−3	−2	5								
Met	−1	−1	−2	−3	−1	0	−2	−3	−2	1	2	−1	5							
Phe	−2	−3	−3	−3	−2	−3	−3	−3	−1	0	0	−3	0	6						
Pro	−1	−2	−2	−1	−3	−1	−1	−2	−2	−3	−3	−1	−2	−4	7					
Ser	1	−1	1	0	−1	0	0	0	−1	−2	−2	0	−1	−2	−1	4				
Thr	0	−1	0	−1	−1	−1	−1	−2	−2	−1	−1	−1	−1	−2	−1	1	5			
Trp	−3	−3	−4	−4	−2	−2	−3	−2	−2	−3	−2	−3	−1	1	−4	−3	−2	11		
Tyr	−2	−2	−2	−3	−2	−1	−2	−3	2	−1	−1	−2	−1	3	−3	−2	−2	2	7	
Val	0	−3	−3	−3	−1	−2	−2	−3	−3	3	1	−2	1	−1	−2	−2	0	−3	−1	4

Figura 2.2.1: Matriz de sustitución BLOSUM62

Números más altos en matrices BLOSUM expresan mayor identidad de secuencia, por lo tanto sirven para alinear secuencias evolutivamente más cercanas, a diferencia de las matrices PAM, en donde números más altos expresan mayor divergencia evolutiva permitida. Las matrices de sustitución permiten entonces, dado un alineamiento de a pares, asignar un puntaje a cada una de las columnas. La suma total de esos puntajes nos permite de esa manera evaluar los alineamientos.

Los *dot-plots* son un método gráfico que permite visualizar y a veces realizar un alineamiento, identificando las regiones que poseen una alta similaridad. Se construye de manera de poner una fila por cada posición en una de las secuencias y una columna por cada posición de la otra secuencia, como se muestra en la figura 2.2.2. Se colorean las posiciones que coinciden con el mismo aminoácido. De esta manera, si dos regiones son altamente homólogas se observarán porciones diagonales coloreadas.

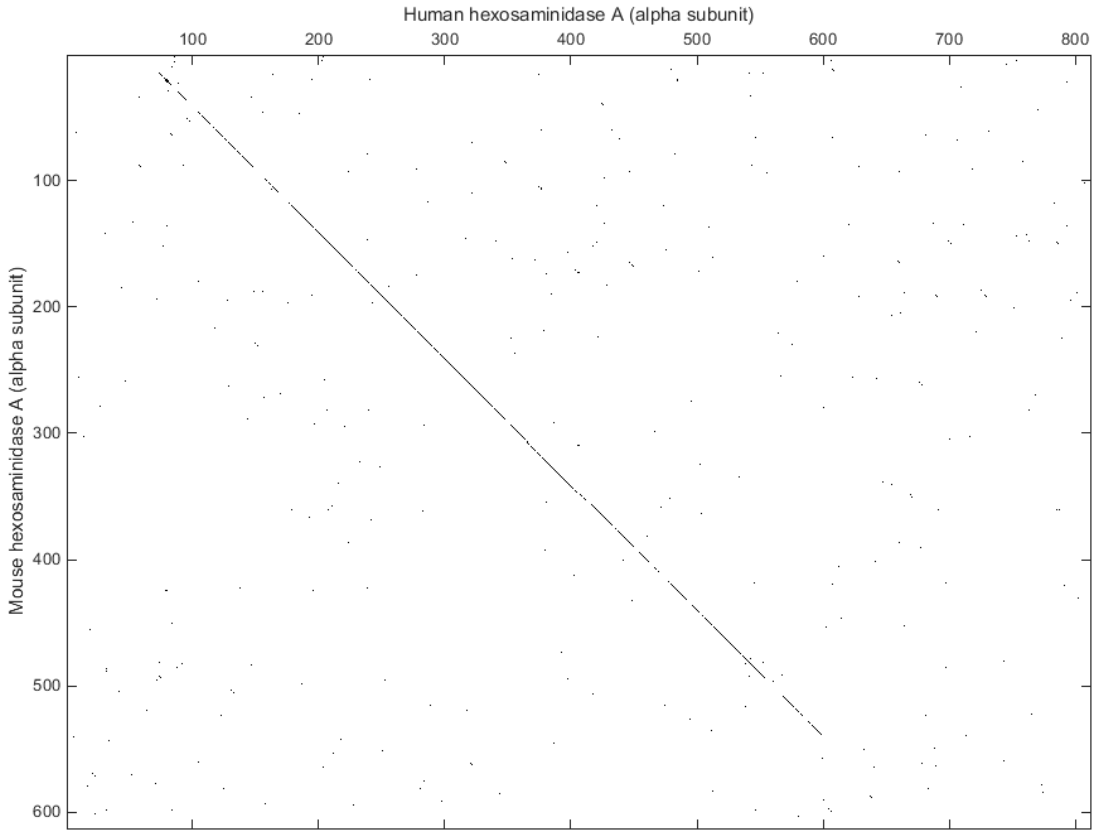


Figura 2.2.2: Dot-plot que muestra el alineamiento de la subunidad A de las hexoamidasas humana y de raton alineadas. Como puede observarse, hay una clara homología de secuencia con un corrimiento en la misma: el apareamiento óptimo comienza en la posición 10 de la secuencia de ratón con la posición 80 de la humana, terminando aproximadamente unos 500 aminoácidos después, siendo significativamente diferentes las secuencias a partir de ese punto.

2.2.1 blast

El algoritmo «blast» por su sigla en inglés "Basic Local Alignment Search Tool"[57] (herramienta de búsqueda básica de alineamientos locales) es actualmente la herramienta estandarizada en la comunidad para llevar a cabo alineamientos de a pares y búsqueda en bases de datos de secuencias, utilizando el alineamiento local como herramienta.

La primera distinción que necesitamos establecer a la hora de hablar de alineamientos es entre los que son «globales» y los que son «locales», los cuales están presentados gráficamente en la figura 2.2.1.1. La primera estrategia, a la hora de optimizar el alineamiento entre dos o más secuencias, buscará maximizar el alineamiento entre secuencias enteras, lo que puede llegar a incluir largos tramos de baja similaridad entre las secuencias alineadas. Por su parte, los alineamientos locales tienen como objetivo la búsqueda se subsecuencias relativamente conservadas, descartando regiones de las secuencias que no se encuentran conservadas (y que por lo tanto no pueden alinearse) que no contribuyen a la medida de similaridad.

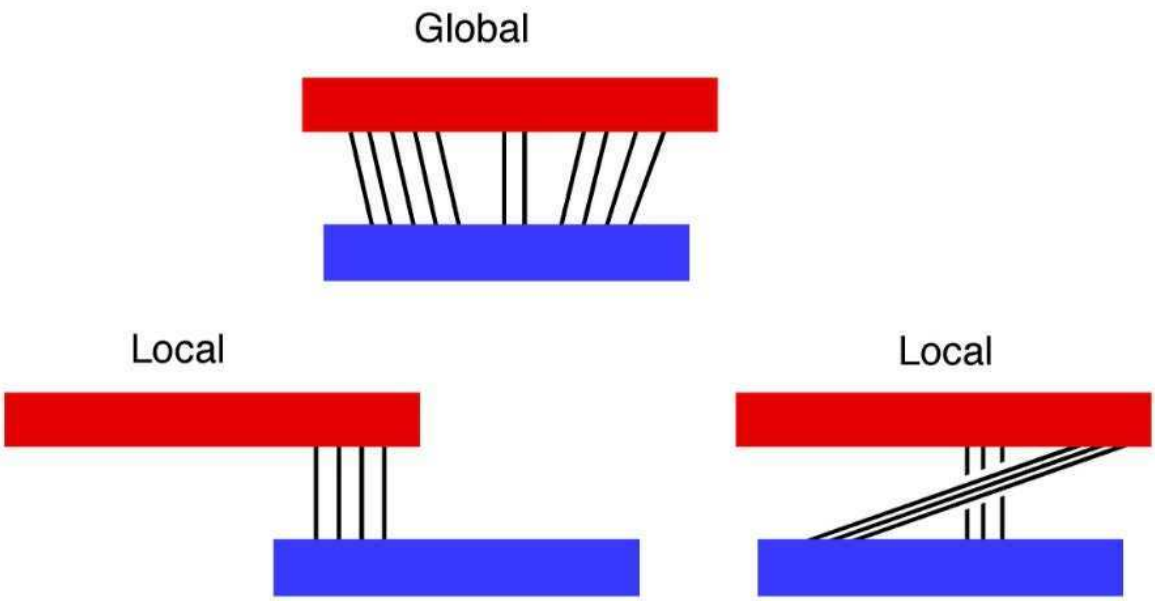


Figura 2.2.1.1: Diferencia entre un alineamiento global, en el que se busca beneficiar el mayor solapamiento posible entre las dos secuencias alineadas, y uno local, en el que se intenta localizar regiones que posean alto solapamiento, independientemente de la presencia de regiones sin solapamiento u orden de aparición de los fragmentos solapados.

La implementación de esta herramienta está basada en el algoritmo de Smith & Waterman[58] cuya finalidad, precisamente, es la de hacer alineamientos locales de secuencias de caracteres. El mismo es un algoritmo de programación dinámica que busca los alineamientos locales óptimos basado en una matriz de sustitución (descritas más arriba) y un esquema de penalización de gaps dado.

El funcionamiento básico del algoritmo, dicho coloquialmente, es el de buscar palabras pequeñas de la secuencia *query* que aparecen de manera exacta en la base de datos, para después extenderlas todo lo posible en función de la matriz de sustitución y finalmente evaluar la significación estadística de los alineamientos resultantes.

Dados los siguientes elementos:

- a, b: cadenas de caracteres sobre un alfabeto Σ (20 aminoácidos para proteínas, 4 bases para ácidos nucleicos, etc).
- $m = longitud(a)$, $n = longitud(b)$
- $s(a, b)$ función de similaridad expresada sobre el alfabeto en función de la matriz de sustitución.
- $H(i, j)$ es el puntaje de similaridad máxima entre los sufijos de $a[1..i]$ y $b[1..j]$
- W_i es el costo de inserción de un gap.

Se construye una matriz de H de m filas y n columnas, siguiendo las siguientes reglas:

- $H(i, 0) = 0,\ si\ 0 \leq i \leq m$
- $H(0, j) = 0,\ si\ 0 \leq j \leq n$
- $H(i, j)\ \ si\ 1 \leq i \leq m \bigwedge 1 \leq j \leq n\ es\ el\ máximo\ entre$:
    - 0
    - $H(i-1, j-1) + s(a_i, b_j)$ (estado match)
    - $max_{k \geq 1}\{H(i-k, j) + W_k\}$ (estado deleción)
    - $max_{l \geq 1}\{H(i, j-l) + W_l\}$ (estado inserción)

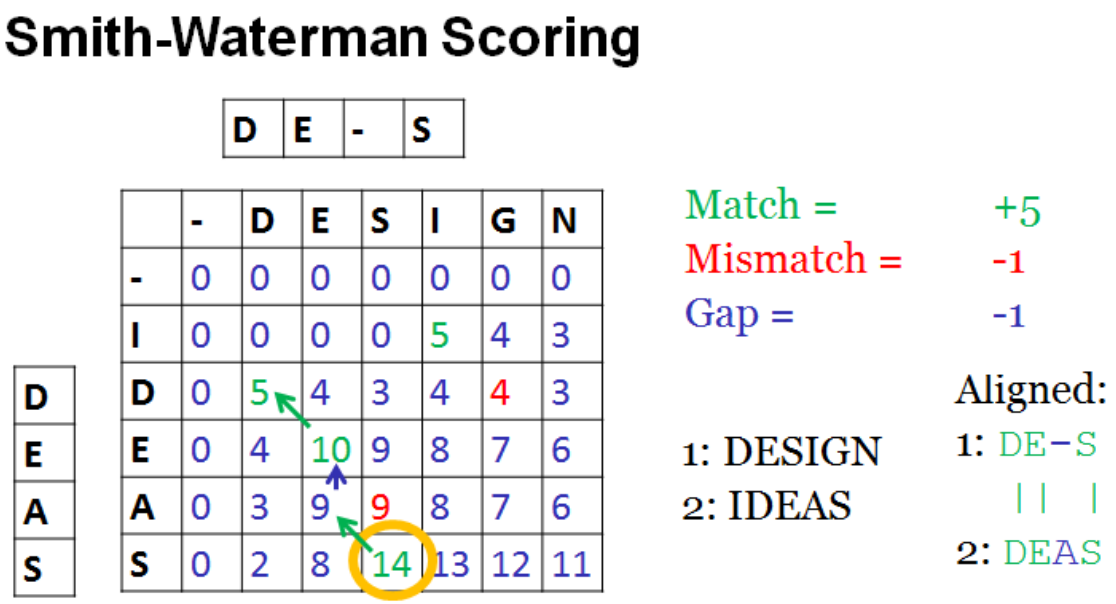


Figura 2.2.1.1: Ejemplo de matriz H siendo las cadenas a alinear "DESIGN" y "IDEAS" en donde se muestran los puntajes para cada tipo de operación y el camino que conduce al alineamiento de mayor puntaje.

En particular, el algoritmo específico de BLAST se compone de tres etapas. En la primera, denominada *seeding* se buscan todas las palabras de un largo determinado(w) presentes en la base de datos que se corresponda con alguna de las palabras de largo w en la secuencia que quiere alinearse, con la condición de que supere el alineamiento un puntaje t, también parámetro del sistema, basándose en la función s que es la que utiliza las matrices de sustitución, y que se encuentren al menos a una distancia a, también parámetro de la corrida, de otra palabra. Todas las palabras de largo w dentro de la base de datos son indexadas de manera de conocer su posición exacta en la misma de manera eficiente lo cual permite que el algoritmo pueda ejecutarse sobre grandes bases de datos insumiendo poco tiempo de cálculo.

En la segunda etapa se produce una extensión de todas las palabras que han sido vinculadas en el paso previo utilizando el algoritmo de Smith & Waterman previamente descrito. Se extiende el alineamiento todo lo posible mientras que el puntaje del mismo no decrezca en más de x puntos. El valor x también es un parámetro de la corrida.

Una vez terminada la extensión de todas las palabras, en la tercera etapa del algoritmo se procede a su evaluación: cada uno de los alineamientos realizados es evaluado para determinar su significación estadística. Para ello, el programa elimina los alineamientos inconsistentes (alineamientos que junten la misma parte de la secuencia que ha servido de input con distintas partes de una secuencia en la base de datos). Una vez realizado esto, se calcula la puntuación final de los alineamientos resultantes y se determina su significancia estadística, tomando en cuenta la probabilidad que tiene dicho alineamiento de haber sido obtenido por azar de acuerdo al tamaño de la base de datos. Al final se reportan sólo los alineamientos que hayan obtenido una probabilidad menor a E. El parámetro E es conocido como e-valor (*e-value*) de corte, y nos permite definir qué alineamientos queremos obtener de acuerdo a su significación estadística. Cuanto menor sea el valor de E, más significativo es un alineamiento.

De este modo, *blast* resulta de utilidad ya que tomando como punto de partida una secuencia *query* y un conjunto de secuencias en una base de datos, el mismo nos devolverá alineadas todas aquellas secuencias de la base de datos que alineadas de a pares con la secuencia *query* superen un umbral de significancia determinado.

Esta técnica nos ha servido para, por ejemplo mediante la determinación de homología entre proteínas, asignar función a proteínas en organismos recientemente secuenciados donde la función es desconocida; o establecer si una droga que actúa sobre una proteína en un organismo patógeno tiene chances de afectar también a su hospedador, etcétera.

También hemos usado el concepto de homología para crear modelos de estructuras tridimensionales en proteínas donde esta no se encuentra resuelta pero sí lo está en proteínas homólogas, como explicaremos más adelante.

2.2.2 hmmer

Un alineamiento múltiple de secuencias de proteínas aporta información acerca de la relación estructural y/o funcional dentro del conjunto que lo compone. Proteínas que poseen un mismo origen evolutivo conservarán inmutadas posiciones claves para su plegamiento o su función y esto permite agruparlas en familias, y a partir de ello inferir su posible función utilizando el concepto de "culpa por asociación".

El algoritmo «hmmer», también utilizado para evaluar alineamientos múltiples de secuencias, se vale de modelos ocultos de markov (HMMs por su sigla en inglés) para establecer si una secuencia de interés tiene buenas probabilidades de pertenecer o no a una familia de proteínas.

Una familia de proteínas está entonces representada por un HMM construído en base a un conjunto de secuencias *seed* o semilla que las representan[29]. Estos modelos son una construcción estadística en el que se asume que el modelo a generar sigue las reglas de un proceso de Markov (fenómeno aleatorio dependiente del tiempo para el cual se cumple la propiedad de Márkov: el valor futuro de una variable aleatoria depende únicamente de su valor presente, siendo independiente de la historia de dicha variable).

Un HMM se define de manera formal como una tupla (Q, V, π, A, B) en donde:

- $Q = \{1, 2...N\}$ representa el conjunto de estados del sistema.
- V representa el conjunto de posibles valores $\{v_1, v_2, ... v_M\}$ observados en cada estado. M es el número de palabras posibles y cada v_k hace referencia a una palabra diferente.
- $\pi = \{\pi_i\}$ es el valor que cada uno de los estados tiene de ser el estado inicial.

- $A = \{a_{ij}\}$ denota las probabilidades de transición entre los estados i y j para cada par de estados posibles, inclusive si $i = j$.
- $B = \{b_j(v_k)\}$ son las probabilidades de observar el símbolo de la posición k dentro de la palabra v estando en el estado j.

En la figura 2.2.2.1 se esquematiza mediante un grafo las transiciones entre estados marcadas por las probabilidades descritas más arriba.

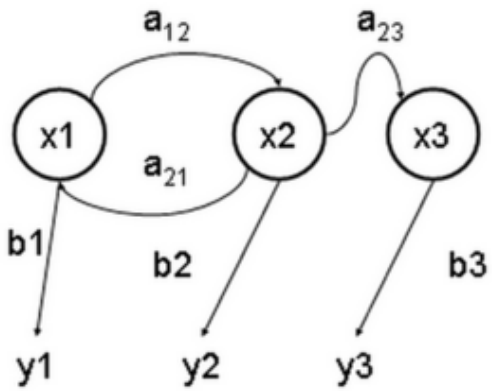


Figura 2.2.2.1: Representación gráfica de un Modelo Oculto de Markov, donde las *x* representan los estados ocultos, las *y* los símbolos emitidos, las *a* las probabilidades de transición entre estados, y las *b* las probabilidades de emisión.

De esta manera, teniendo un HMM construido, la probabilidad de observar una secuencia Y constituída por los símbolos $y(0), y(1) \ldots y(L-1)$, siendo L la longitud de la secuencia, está dada por la fórmula:

$$P(Y) = \sum_{X} P(Y|X)P(X)$$

En donde la sumatoria se extiende sobre todas las secuencias de nodos ocultos $X = x(0), x(1) \ldots x(L-1)$. El cálculo de $P(Y)$ por fuerza bruta resulta un problema que

consume un tiempo de cómputo exponencial y por eso es que el modelo permanece oculto, conociéndose las secuencias de caracteres emitidos pero no el grafo de estados ocultos.

De esta manera, el uso de modelos ocultos de Markov nos permite resolver de manera eficiente el problema de alineamientos múltiples de secuencias de proteínas. Aplicar este sistema nos permite además conocer cómo se alinea la proteína con las demás de su familia, en qué medida (mediante una función de puntaje) la proteína es similar a las demás de su familia (que probabilidad hay de que el HMM "emita" nuestra secuencia objetivo); y entrenar, generando nuevos modelos ocultos, el algoritmo para definir nuevas familias (cómo se definirán los parámetros de la misma).

En la figura 2.2.2.2 se observa un alineamiento múltiple de secuencias de globinas y en la parte inferior de la figura se expresa de manera esquemática la probabilidad de emisión de cada símbolo en cada una de las posiciones (nodos x en el gráfico de más arriba) del alineamiento múltiple. Vemos que para posiciones muy conservadas las probabilidades de emisión (altura de la columna) son más altas que en posiciones menos conservadas.

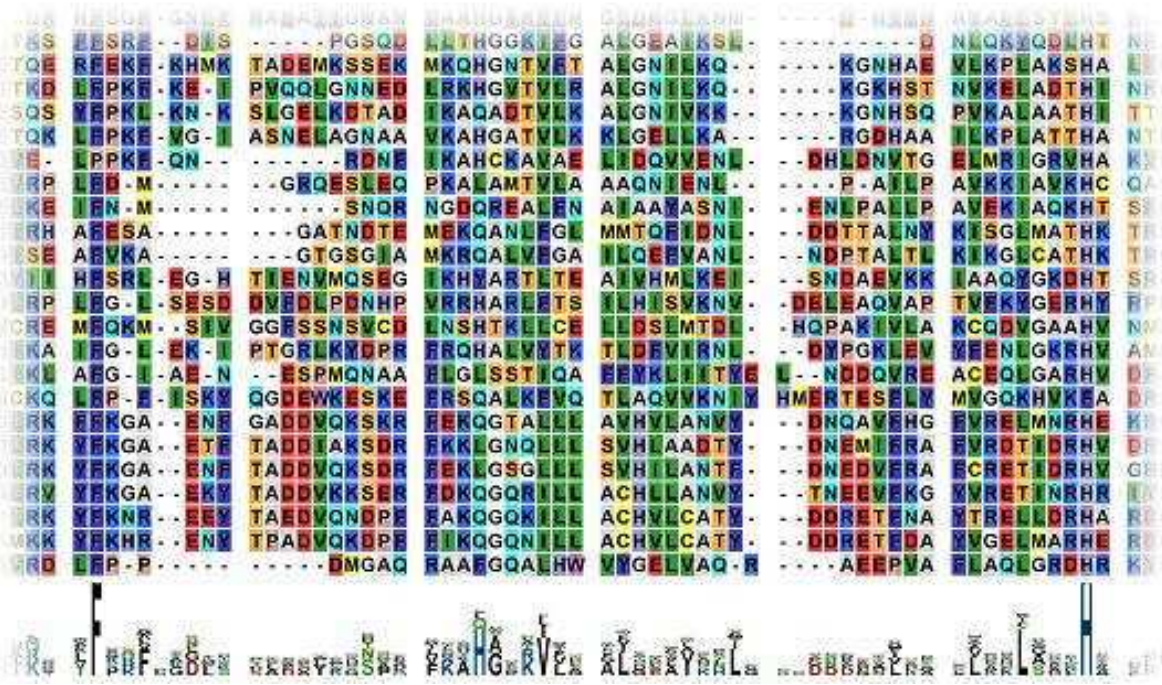


Figura 2.2.2.2: Porción del alineamiento de la familia de globinas mediante el algoritmo hmmer y visualización de la misma en base a su Logo de secuencia[59] en la parte inferior de la figura.

Los HMMs de cada una de las familias de la base de datos PFam (ver más abajo) se generan a partir de un alineamiento múltiple global (en versiones anteriores definido manualmente y en versiones actuales definido de manera automática) de secuencias que se conoce pertenecen a una misma familia, para luego hacer una búsqueda en una base de datos de proteínas extensiva (SwissProt[60]) que permita ir "entrenando" el modelo oculto de Markov, lo que coloquialmente significa determinar si la proteína que estoy evaluando me ayuda a discriminar mejor o peor a los miembros de la familia versus una secuencia tomada aleatoriamente de la base de datos.

En esta tesis, hemos utilizado HMMs y el algoritmo *hmmer* para identificar en proteínas la familia de plegado a la que pertenecen. Además, esta información nos ha sido útil también para extrapolar otras conclusiones: por ejemplo, acerca de la posible unión un mismo

ligando en diferentes integrantes de una misma familia o determinar aminoácidos importantes que podrían formar el sitio activo de proteínas enzimáticas.

2.3 Cálculos sobre la estructura proteica

Muchas de las propiedades que pueden obtenerse sobre una proteína, son extraídas a partir de su estructura. La estructura de una proteína se especifica en un archivo de formato específico denominado pdb. El formato de archivos .pdb es el que especifica las coordenadas en tres dimensiones de los átomos que componen a la molécula analizada, y varias de las propiedades sobre las que estamos interesados para este trabajo se obtienen de la lectura directa de este tipo de archivos.

Algunos ejemplos son:

- El b-factor de cada uno de los átomos, el cual expresa un valor que es función de la temperatura, leyendo valores altos para regiones más calientes, lo que indica zonas más móviles dentro de la estructura de la proteína y por lo tanto menos rígidas.
- El pKa o constante de disociación ácida de los aminoácidos que forman parte de la estructura.
- Las superficie accesible al solvente de los aminoácidos que forman parte de la estructura. Dato que representa el área que puede estar en contacto directo con el medio en el que la proteína puede estar expuesta interactuando con el solvente. De la misma se deriva el porcentaje de la cadena lateral expuesta de ese aminoácido, para saber si el mismo forma parte de la superficie o del core hidrofóbico de la estructura que está siendo evaluada.
- Las cavidades o bolsillos presentes en la estructura, las cuales serán explicadas en mayor detalle más adelante en este mismo capítulo.

2.3.1 Modelado por homología

Tal como hemos nombrado en la introducción de esta tesis, para algunas proteínas de las cuales no conocemos su estructura tridimensional, podemos intentar construir un modelo basado en estructuras que sí se encuentran resueltas y que son homólogas a la proteína objetivo y que poseen un alto porcentaje de identidad.

El algoritmo elegido a lo largo de los desarrollos de esta tesis para realizar modelos comparativos es MODELLER[24], el cual posee un *pipeline* general que, mediante el uso del algoritmo *blast* sobre una base de datos de secuencias con estructuras disponibles, busca aquellas que cumplan con un umbral de identidad de secuencia con respecto a aquella proteína que se desea modelar y la selecciona como "molde" o templado (se recomienda una identidad mayor al 50% en los templados para que los modelos sean de buena calidad). El funcionamiento de MODELLER se esquematiza en la figura 2.3.1.1.

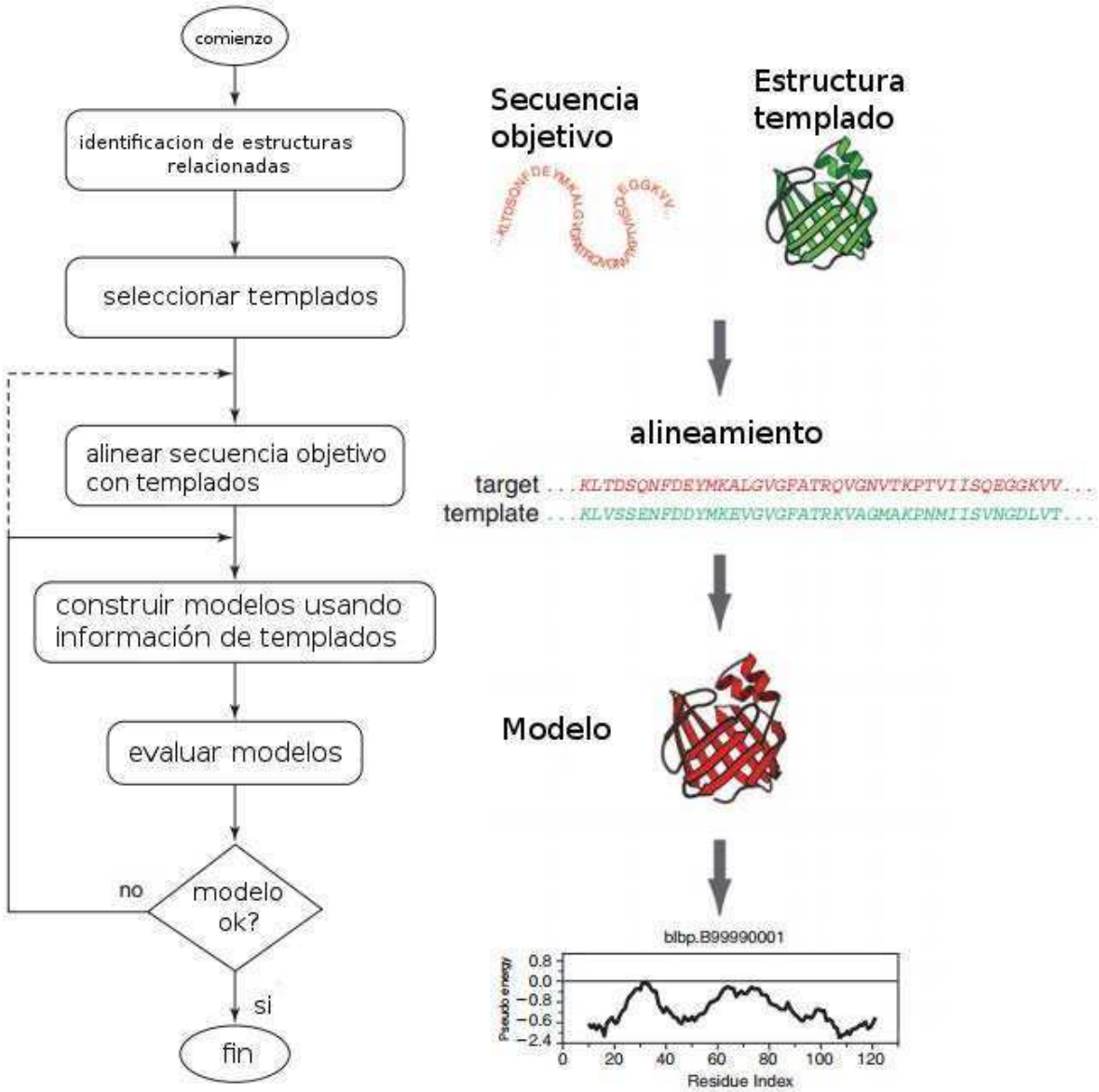


Figura 2.3.1.1: Diagrama de control del algoritmo de modelado comparativo utilizado por el software MODELLER. Los modelos son evaluados luego de ser construidos con un potencial clásico con el cual se minimiza la energía, aceptando el modelo si la misma no supera cierto umbral dependiente del largo de la secuencia.

MODELLER implementa un modelado comparativo de estructuras de proteínas que se basa en la satisfacción de restricciones espaciales como pueden ser:

- Restricciones en las distancias y ángulos diedro en la secuencia objetivo, extraídas de su alineamiento con las estructuras templado.
- Restricciones estereoquímicas como largo de uniones y preferencia en ángulos de unión, obtenidas del campo de fuerzas CHARMM-22[61].
- Preferencias determinadas estadísticamente para ángulos diedro y para distancias en enlaces de no unión, obtenidas de un conjunto representativo de estructuras resueltas de proteínas.

Estas restricciones espaciales, expresadas como funciones de densidad de probabilidad, son combinadas en una función objetivo que es optimizada por combinaciones de gradientes conjugados y mecánica molecular con simulated annealing -recocido simulado-. En términos prácticos, dadas las estructuras templado, se va "montando" la secuencia a modelar posición a posición satisfaciendo las restricciones ya enumeradas, teniendo en cuenta el alineamiento de la proteína a modelar con los templados.

Las restricciones que se toman en cuenta para el simulated annealing son evaluadas en base a un potencial denominado DOPE[62] (Discrete Optimized Protein Energy). Este potencial se define como el logaritmo negativo de la densidad de probabilidad conjunta de ocurrencia de las coordenadas cartesianas atómicas entre todos los pares de átomos. DOPE puede ser pensado como un potencial de interacciones de a pares, donde la energía se determina a partir de datos empíricos derivados de estructuras conocidas.

La densidad de probabilidad conjunta puede expresarse de la siguiente manera:

$$p(x_1, x_2, ..., x_n) \approx \prod_{i \neq j}^{N} p(x_i, x_j) \,/\, \left(\prod_{i}^{N} p(x_i) \right)^{N-2} \propto \prod_{i \neq j}^{N} p(x_i, x_j)$$

Donde p es la función de densidad de probabilidad, el vector x_i representa las coordenadas cartesianas del átomo i, N es la cantidad de átomos del templado y vemos que evoluciona proporcionalmente tratando los átomos de a pares. En otras palabras, se

verán beneficiadas energéticamente configuraciones observadas en estructuras conocidas y se verán penalizadas configuraciones atómicas no observadas.

De estos términos apareados es más simple conocer su expresión:

$$p_{mn}(r) = N_{mn}(r) \,/\, \sum_{r_i} N_{mn}(r_i)\Delta r$$

Donde m y n denotan tipos de átomos y $M_{mn}(r)$ es el número de pares de átomos (m, n) dentro de la distancia $(-r, r)$.

Calculado este potencial sobre las estructuras que sirven de templado se va generando la nueva estructura de manera de minimizar la energía en base a este potencial, como puede verse de manera esquemática en la figura 2.3.1.2, en donde se evalúa con el potencial generado la estructura a generar satisfaciendo las restricciones espaciales obtenidas.

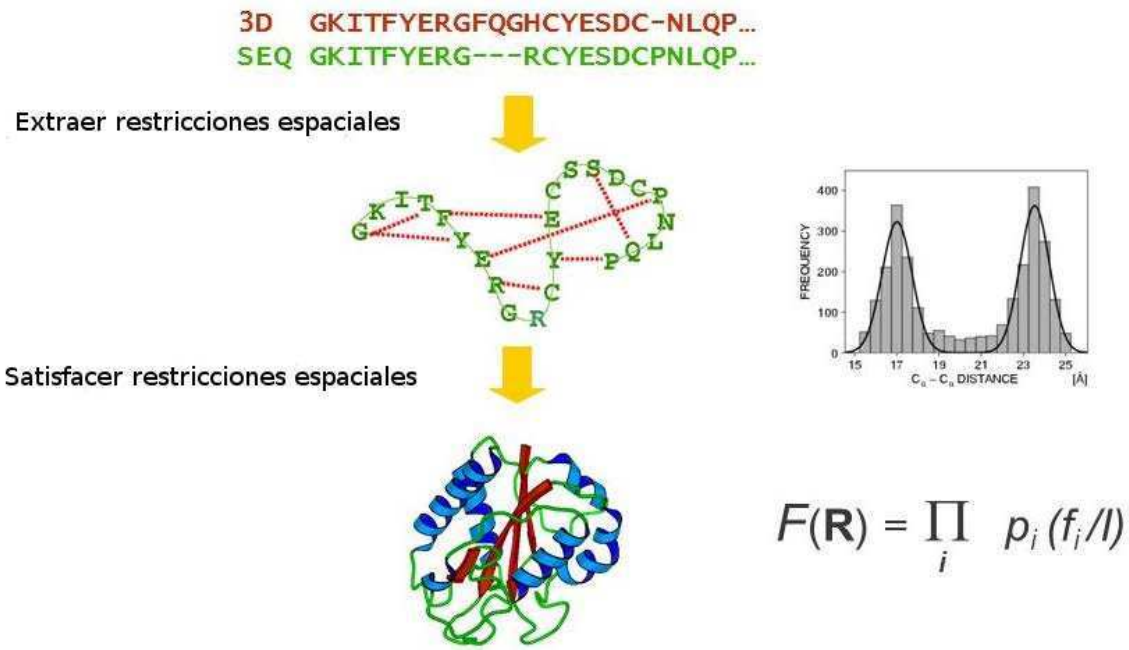


Figura 2.3.1.2: Esquema de evaluación de restricciones espaciales en base a densidades de probabilidades de distancias observadas en las estructuras que sirven de templado. Aquellas configuraciones de distancias más observadas en los templados se verán beneficiadas energéticamente a la hora de construir el modelo.

Usualmente, se generan muchos modelos en base a distintos alineamientos posibles de la proteína a modelar y a distintas conformaciones de distancias y ángulos que satisfacen ser mínimos dentro del potencial, y luego se evalúa para todos los modelos cuál es el de menor valor de energía. Finalmente, el modelo es minimizado con un campo de fuerzas clásico (CHARMM).

Queda evidenciado que el poder de modelado de este tipo de métodos se circunscribe a reproducir configuraciones de plegado de estructuras ya resueltas. Nuevas configuraciones de plegado quedarán perjudicadas energéticamente por no tener significancia estadística en el potencial, reduciendo el poder predictivo de esta herramienta en estas situaciones. Sin embargo, en líneas generales es posible obtener un buen modelo a partir de la base

completa de estructuras de proteínas, ya que una buena cantidad del universo de los dominios de plegado se encuentran actualmente resueltos estructuralmente.

2.3.2 Determinación de cavidades

Una vez determinada la estructura de la proteína (ya sea mediante un cristal, resonancia magnética nuclear, modelado por homología o modelado *ab-initio*) es importante determinar cuál es la zona en la región de la misma en donde se producirá, cuando corresponda, el acoplamiento con un compuesto que module su actividad, o sea su sitio activo.

La porción de estructuras cristalinas en las que la ubicación del sitio activo está determinada es bajo, y no se pueden hacer extrapolaciones a modelos de manera directa. Por esto, hemos encontrado necesario utilizar algoritmos de detección de bolsillos (a partir de aquí indistintamente podrán ser llamadas cavidades). Después de evaluar diferentes estrategias planteadas para resolver este problema, hemos elegido para los diferentes desarrollos de esta tesis el algoritmo *fpocket*, el cual está basado en una estrategia de α-spheres y la triangulación de Delaunay.

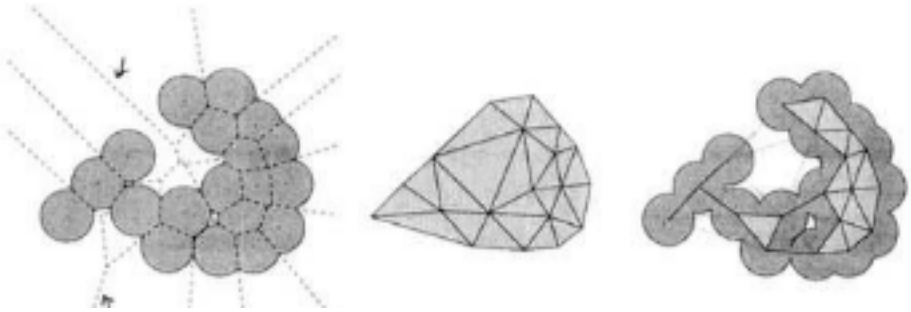


Figura 2.3.2.1: Ejemplo de triangulación en un modelo bidimensional. Se colocan α-spheres sobre la superficie de la proteína Luego se generan los triángulos que unen los centros de cada una de estas esferas colocadas. Se consideran cavidades aquellas porciones del espacio formada por triángulos que no encierran ninguna región que contenga un átomo de la estructura.

Una α-sphere, resumidamente, es una esfera que contacta cuatro átomos de la estructura de la proteína en sus límites y que no contiene (encierra) átomos dentro de la misma. Los cuatro átomos, por definición, se encuentran a la misma distancia (el radio) del centro de la esfera. Para el caso de las proteínas, pequeñas esferas pueden ser colocadas dentro de la estructura, y esferas de radio mayor pueden ser colocadas en su superficie. Los valores permitidos para el radio de las esferas son un parámetro del algoritmo. Valores más pequeños se utilizan, por ejemplo, para detectar túneles en la estructura, cuatro átomos en un plano definen técnicamente una esfera de radio infinito, los tamaños de esfera óptimos para detectar cavidades que alojan ligandos tienen un rango definido y validado empíricamente. La validación de los parámetros del algoritmo fue definida de manera de maximizar la detección de cavidades en el PDB que se encuentran unidos a ligandos experimentalmente con la menor cantidad posible de falsos positivos.

Luego de colocar todas las esferas posibles del rango de radios determinados como parámetro del algoritmo, aquellas zonas que poseen una alta densidad de esferas superpuestas son las que poseen mayores probabilidades de alojar ligandos. Cada bolsillo se define entonces mediante las esferas que lo conforman. Se define un parámetro en el algoritmo para determinar la mínima cantidad de esferas que son necesarias para constituir un pocket, evitando de esta manera definir cavidades muy pequeñas.

Por otro lado, las propiedades de cada uno de los bolsillos vienen dadas por las características de las esferas que lo componen (volumen, superficie, etc.) y por los átomos de la proteína que contribuyeron a definir sus esferas (carga, dadores y aceptores de puente de hidrógeno, etc.). Maximizados los parámetros de corrida del algoritmo para la detección de verdaderas cavidades que alojan ligandos, se define el Druggability Score (puntaje de drogabilidad estructural) como una combinación lineal de las propiedades calculadas normalizadas entre cero y uno, de forma de representar valores más altos para

bolsillos con mayores chances de alojar ligandos y valores menores cuando las chances de acoplamiento bajan.

2.4 Tratamiento computacional de moléculas orgánicas pequeñas

2.4.1 Almacenamiento de compuestos

En el capítulo 3.2 de este trabajo, uno de los elementos con los que trabajaremos son las moléculas pequeñas: a diferencia de las macromoléculas cuya forma de almacenamiento son sus coordenadas tridimensionales producto de su determinación experimental, las moléculas pequeñas pueden ser almacenadas en diferentes formatos dependiendo de la aplicación que necesite procesarlas. De los tipos de almacenamiento existentes, la primera gran distinción que podemos hacer es entre uni o bidimensionales, los cuales no almacenan las coordenadas sino una descripción de las moléculas relacionadas con sus enlaces covalentes, como si se tratara de un grafo, y las formas tridimensionales, que poseen información tridimensional explícita.

2.4.1.1 SMILES, SMARSTS y *fingerprints* moleculares

Una de las formas más simples de representar compuestos es mediante SMILES[63] (por su acrónimo en inglés *Simplified Molecular Input Line Entry System*) el cual puede denominarse como un sistema de anotación "en una línea": es un método tipográfico que utiliza solo caracteres para describir a los compuestos.

Los SMILES son una construcción lingüística que almacena de una molécula cuáles son los átomos que la componen y cómo están conectados(tipos de enlace). Parte del poder de este tipo de anotación es que la misma es única: con una manera estandarizada de

anotación. De este modo, cada SMILES es unívoco con su estructura y viceversa. De esta manera la anotación de una molécula es única y permite que esta sea almacenada de una manera comprimida y útil para realizar determinados tipos de búsquedas en bases de datos. Otra propiedad importante de este tipo de anotación es que es compacta. Comparada con una tabla de conexión entre átomos, está calculado que un SMILES almacenará la misma información utilizando entre un 50% y un 70% menos de espacio de almacenamiento. Comparado a un método de anotación en tres dimensiones (que contiene más información almacenada, pero que puede ser inferida a través del SMILES con algoritmos que lo procesen) el espacio ocupado es alrededor de 1%, funcionando como un buen método de compresión de la información molecular.

En los SMILES, cada átomo es representado con su símbolo atómico (C: carbolo, Cl: cloro, etc. El caso especial de carbono aromático se denota con c minúscula) y esas son los únicos caracteres que utilizan letras del abecedario. Los hidrógenos nos se explicitan a menos que sea necesario para indicar propiedades particulares. Entre corchetes se pueden agrupar partes de moléculas que luego están conectadas con un átomo en particular que lo precede en la secuencia de caracteres. Estados especiales de protonación se denotan con los símbolos "+" y "-". Enlaces simples no se denotan sino que siguen la secuencialidad de la cadena de caracteres (un átomo, o una sub molécula encerrada entre corchetes está unida mediante un enlace simple a la anterior) mientras que los enlaces dobles o triples se anotan mediante "=" y "#" respectivamente. Los anillos de átomos se denotan mediante el nombramiento de uno de los átomos con un 1 que permite volver a "conectar" un átomo al final de la escritura del ciclo con ese átomo del comienzo. La quiralidad se maneja mediante los símbolos "/" y "\". Los isótopos se manejan mediante la anotación en números de la especie que se está anotando antes del símbolo atómico, encerrando al átomo entre corchetes. Existen más reglas de anotación con este sistema pero no es nuestra intención escribir aquí un manual de esta metodología.

A modo de ejemplo, en la tabla 2.4.1.1, se muestran una serie de compuestos con sus respectivos SMILES en las que pueden deducirse de manera intuitiva las principales características de la notación, y donde queda evidenciado que el formato es apto de ser leído por humanos de una manera analítica.

SMILES	**Name**	**SMILES**	**Name**
CC	etano	[OH3+]	ion hidronio
O=C=O	dióxido de carbono	[2H]O[2H]	óxido de deuterio
C#N	hydrogen cyanide	[235U]	uranio-235
CCN(CC)CC	trietilamina	F/C=C/F	E-difluoroeteno
CC(=O)O	ácido acético	F/C=C\F	Z-difluoroeteno
C1CCCCC1	ciclohexano	N[C@@H](C)C(=O)O	L-alanina
c1ccccc1	benzeno	N[C@H](C)C(=O)O	D-alanina

Tabla 2.4.1.1: Ejemplos de representación de compuestos de manera bidimensional mediante SMILES.

Por ejemplo, en la citada tabla, puede observarse en el ejemplo de etano y dióxido de carbono como se evidencian los enlaces dobles pero no los simples, o como se manejan los ciclos dándole nombre a uno de los átomos del mismo (C1 en ciclohexano) para volver a referenciarlo al final del ciclo.

En conjunto con la denotación SMILES, sus desarrolladores han creado un lenguaje de expresiones regulares llamado SMARTS que permite de una manera analítica realizar operaciones con las moléculas anotadas. La librería Babel[64] posee en su implementación el

manejo de SMILES y las operaciones que describiremos se encuentran implementadas de una manera eficiente.

Los SMARTS puede considerarse un lenguaje para la especificación de subestructuras de moléculas y de patrones de las mismas. Tiene, por ejemplo, operadores lógicos: "&" para el operador "y", "," para el operador "o", "!" para el operador "no". El carácter "*" puede representar cualquier átomo, el "R" representa anillos, etc. Nuevamente, al no ser nuestra intención escribir un manual de esta tecnología, no pondremos cada una de las reglas existentes.

En la tabla 2.4.1.2 pueden observarse algunas reglas de construcción de smarts. Por ejemplo, si utilizamos como patrón de búsqueda "cccc" en una base de datos, todos los compuestos que tengan como subestructura cuatro átomos de carbono unidos con enlace simple (butano, pentano, hexano, etc.) serán matches positivos.

SMART	**Interpretación**
cc	cualquier par de carbonos aromáticos unidos
c:c	carbonos aromáticos unidos por un enlace aromático
c-c	carbonos aromáticos unidos por un enlace simple (ej: bifenil)
[C,c]	cualquier carbono

Tabla 2.4.1.2: Ejemplos simples de búsquedas de moléculas que contengan las subestructuras especificadas mediante el lenguaje SMARTS..

Todas las expresiones de tipo SMILES son expresiones SMARTS válidas, pero la semántica cambia pues las primeras denotan moléculas mientras que las segundas denotan

patrones. Una molécula representada mediante un SMILES, salvo casos especiales, es el patrón que identifica esa molécula cuando se la busca mediante SMARTS.

El principal uso de estas tecnologías es el de poder realizar búsquedas en grandes bases de datos de compuestos permitiendo buscar aquellos que cumplan con determinadas medidas de *similaridad química*. Para acelerar el proceso de filtrado de moléculas a partir del SMILES de cada una de ellas se construye un *fingerprint* (en castellano, la huella digital de la molécula), la cual es una representación abstracta de algunas características constitutivas de la la misma.

La búsqueda de compuestos mediante una subestructura es un problema de tipo NP-completo. No es nuestro objetivo entrar en detalles de la implicancia de ser un problema de este tipo, nos bastará con decir que resolver este problema insume una cantidad de tiempo que escala de manera exponencial con respecto al número de átomos de las moléculas implicadas en la función de búsqueda de subestructura. Sin embargo, este es un costo que se denomina de "peor caso", lo que significa que el costo exponencial de esta operación depende de ciertas condiciones en las moléculas que están siendo comparadas. Afortunadamente, existe una salvedad y es que, si bien no podemos detectar la presencia de una subestructura en tiempo polinomial, sí podemos detectar la ausencia de una subestructura en una molécula de interés (en la mayoría de los casos en un tiempo lineal). La metodología que suele utilizarse entonces al momento de buscar una subestructura en una base de datos es un *screen* (muestreo) de todas las moléculas filtrando aquellas en donde no se encuentra la subestructura buscada.

Para construir el fingerprint de una molécula lo que suele hacerse es buscar la presencia de determinadas "claves estructurales" las cuales se representarán mediante *bits* en el fingerprint que estarán "prendidos" (tendrán un valor de 1, indicando que se cumple con esa clave estructural) o "apagados" (tendrán un valor de 0).

Existen diferentes maneras de especificar claves estructurales, las cuales serán útiles para representar similaridad entre moléculas con distintas metodologías. Generar estas claves estructurales que luego deberán ser calculadas para cada molécula cuando se construya el fingerprint es, por supuesto, algo que demanda tiempo. Lo mismo sucederá al evaluarlos. Si uno tiene una base de datos realmente grande (decenas de millones de compuestos) de la cual querrá hacer filtros muy simples, será una mala idea crear una gran cantidad de claves estructurales que insuman tiempo de construcción/evaluación.

Algunas de las claves estructurales más comunes a la hora de crear fingerprints de moléculas son:

- La presencia/ausencia de un determinado elemento o repeticiones del mismo (por ejemplo: "al menos un nitrógeno", "al menos cuatro oxígenos", etc).
- Configuraciones electrónicas inusuales o importantes (por ejemplo: "carbono sp3", "nitrógeno triplemente enlazado", etc).
- Presencia de anillos o sistemas de anillos.
- Presencia de grupos funcionales(alcoholes, aminas, etc.)
- Presencia de grupos funcionales de especial importancia dependiendo de la base datos (por ejemplo, en una base de datos de compuestos organometálicos pueden crearse bits especiales para denotar la presencia de grupos funcionales que contengan metales, en una base de datos de drogas para la presencia de esteroides, etc.)

Resulta bastante evidente la rapidez que puede obtenerse en los algoritmos de búsqueda que apliquen fingerprints, los cuales, dado el compuesto de interés, verificarán qué claves estructurales estarán prendidas y apagadas en cada molécula de la base de datos, y de esa manera compararán de una manera muy simple y rápida (de costo lineal en función del largo del fingerprint) cuales son las coincidencias en las claves estructurales.

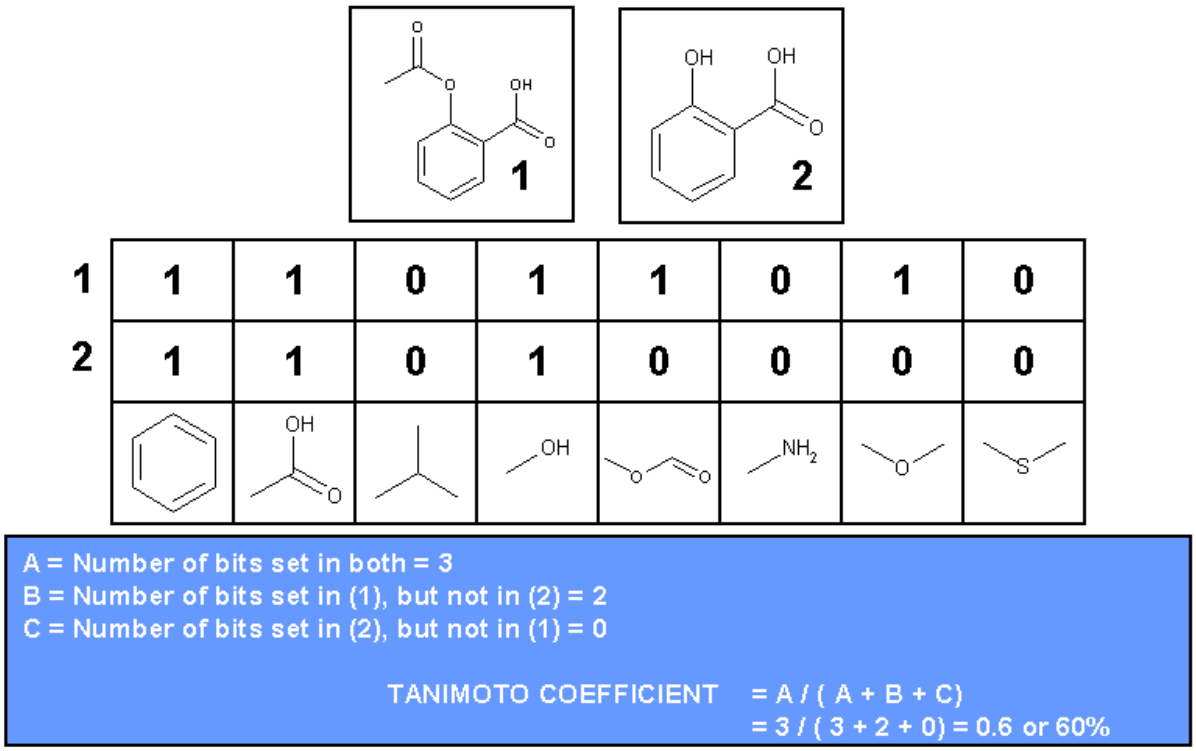
Un punto adicional de gran relevancia en relación con los fingerprints moleculares es el siguiente: habría que generar una gran cantidad de estas "claves estructurales" o "patrones" para representar correctamente la gran variabilidad química que puede haber en una base de datos de compuestos. Además, la cantidad de ceros en los bits del fingerprint sería muy grande (la gran mayoría de las moléculas cumpliría sólo unas pocas claves) haciendo esta denotación muy "esparza". Por eso mismo, lo que se hace en la práctica, es tomar tiras lineales de átomos enlazados (en la práctica 7) y para cada uno de ellos computar las claves estructurales presentes en los mismos. Para este número acotado de átomos, la cantidad de claves a generar será mucho menor, permitiendo que la computación sea más eficiente.

Para medir entonces la similitud química entre dos moléculas (siempre con respecto a una forma de construir un fingerprint) suele usarse, y nosotros lo tomaremos como medida estándar de similaridad el coeficiente de Tanimoto[65]. El mismo se define de la siguiente manera: dadas dos moléculas A y B, su índice de similitud TI es

$$TI(A,B) = \frac{c}{(a+b+c)}$$

Donde a es la cantidad de bits "prendidos" en el fingerprint de la molécula A y no en la molécula B, b es la cantidad de bits "prendidos" en el fingerprint de la molécula B y no en la molécula A y c es la cantidad de bits "prendidos" en las dos moléculas.

Un ejemplo de índice puede observarse en la figura 2.4.1.1 en donde se ha construido un fingerprint *ad-hoc* para marcar si están presentes algunos grupos funcionales que aparecen en dos moléculas de ejemplo.



1	1	1	0	1	1	0	1	0
2	1	1	0	1	0	0	0	0

Figura 2.4.1.1: Calculo del coeficiente de tanimoto utilizando fingerprint de molécula completa, donde cada bit está prendido o apagado en función de ciertos elementos estructurales arbitrarios definidos para el ejemplo.

De esta manera, cuando dos moléculas sean iguales, su índice valdrá 1, y cuando sean completamente distintas (siempre de acuerdo a las claves estructurales definidas por el fingerprint) el índice valdrá cero.

2.4.1.2 InChI e InChIKey

La notación InChI[66] (International Chemical Identifier) es la forma estándar definida por la IUPAC (Asociación internacional de Química Pura y Aplicada) para proveer una forma legible por humanos de codificar compuestos y para facilitar la búsqueda de esa información en bases de datos y en la web. Una de las características que posee es que los algoritmos de codificación están disponible bajo licencia libre LGPL.

La forma en la que se construye el InChI de una molécula es mediante *capas* de información: los átomos y su conectividad, su información tautomérica, su información isotópica, su estereoquímica, y, finalmente, su información electrónica. No todas las capas tienen que ser provistas de manera obligatoria (por supuesto, la única obligatoria es la primer capa).

Cada InChI comienza con los caracteres "InChI=" seguidos de la versión de la misma que se esté utilizando (actualmente, la última versión es la 1). Cada capa y subcapa de información es separada con el carácter "/" y comienza con una letra que funciona a manera de prefijo identificatorio. Algunas de las capas son:

- Capa principal:
    - Fórmula: contiene la fórmula del compuesto y no tiene prefijo. Es la única capa obligatoria.
    - Conexión: prefijo "c", los átomos de la fórmula son numerados (exceptuando los hidrógenos) y se especifica cuales tienen enlaces con cuales.
    - Hidrógenos: prefijo "h", describe cuántos átomos de hidrógeno están conectados a cada uno de los demás átomos.
- Capa de carga: tiene una subcapa de protonación (prefijo "p") y una subcapa de carga ("q").
- Capa de estereoquímica: tiene subcapas de dobles enlaces y cumulenos, de estereoquímica tetraédrica y alenos, isotopía, etc.

A modo de ejemplo en la figura 2.4.1.2 se muestra la anotación InChI de una molécula a nivel de capa principal con cada una de sus subcapas.

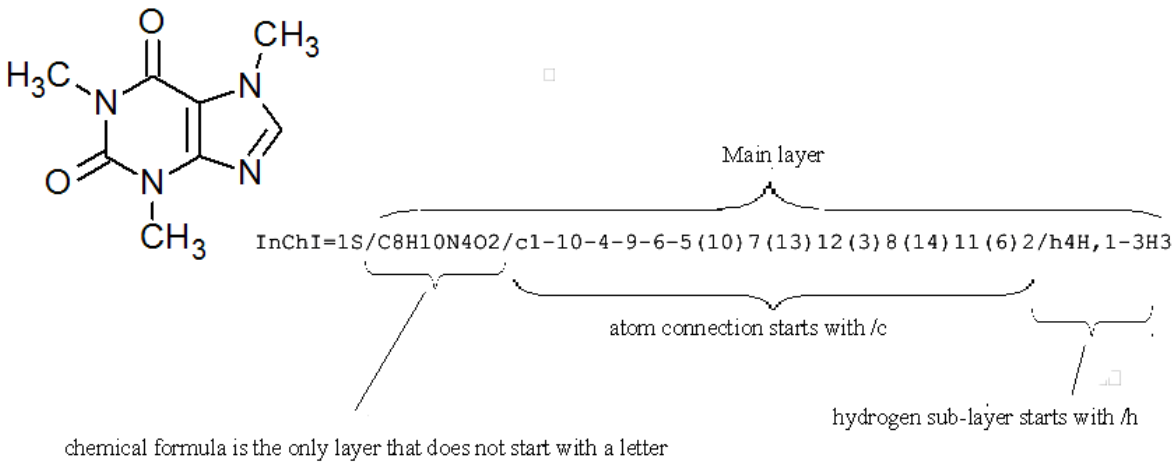


Figura 2.4.1.2: Representación de una molécula mediante un InChI señalando cada una de las capas que componen la capa principal. En la fórmula se anotan la cantidad de átomos de cada tipo presentes en el compuesto. La capa de conexión tomas los átomos de la fórmula numerados en el orden indicado (los átomos 1 a 8 son de carbono, del 9 al 12 de nitrógeno y el 13 y 14 de oxígeno, los átomos de hidrógeno se especifican aparte en la subcapa que comienza no **h**.

La InChIKey de un compuesto (a veces llamada InChI *hasheado*) es una representación condensada (25 caracteres) producto de la aplicación de una función de hash al InChI de un compuesto, con el fin de facilitar las búsquedas en grandes bases de datos de compuestos de manera específica, y permitir un almacenamiento de los mismos en campos de largo fijo (lo que permite su eficiente almacenamiento en bases de datos relacionales). Un campo de largo fijo con alfabeto definido es fácilmente *indexable*, permitiendo que una búsqueda pueda realizarse con costo lineal en función de la cantidad de símbolos.

Una función de hash (h) es aquella que cumple la propiedad de poder ser computada por un algoritmo de forma tal que:

$$H : U \rightarrow M \,/\, x \rightarrow h(x)$$

Por decirlo de una manera coloquial, la función h no es otra cosa que la proyección de un conjunto U (que en el caso de los InChI es el conjunto de las secuencias de caracteres con las cuales pueden construirse) en un conjunto M (que en los InChI es el conjunto de secuencias de caracteres alfanuméricos de largo 25).

Al tratarse de un dominio de cardinalidad infinita (numerable) sobre un conjunto finito, queda claro que pueden producirse colisiones (dos moléculas pueden compartir un mismo hash): no es nuestro objetivo entrar en detalles técnicos de los algoritmos de hashing, pero existen algunas estrategias para evitar que esto suceda, en la medida de lo posible. La cantidad de moléculas que pueden ser representadas mediante InChIKey son 23^{25} (cantidad de letras elevado al largo de la representación). De forma teórica, debiera producirse una colisión cada 75 de millones de moléculas. Estudios de colisiones sobre moléculas en bases de datos existentes[67] muestran que en la práctica se cumple con estas proyecciones teóricas.

2.4.1.3 Formatos en tres dimensiones

Las notaciones anteriormente mencionadas pueden considerarse formatos comprimidos y lineales para la anotación de moléculas. La forma no comprimida de anotar la estructura de moléculas tiene una correspondencia con el formato PDB ya explicado, en el cual se anotan las coordenadas tridimensionales de manera directa de todos sus átomos.

Existen básicamente dos maneras de anotar moléculas en base a sus coordenadas: de manera cartesiana, en donde las coordenadas tienen un sistema de referencia arbitrario, o coordenadas internas (simil polares), en las cuales el primer átomo ocupa la coordenada de referencia, y en función de ella se van definiendo las posiciones y ángulos relativos de los demás átomos. Además de las coordenadas, varios de los formatos tridimensionales tienen

una sección dedicada a la anotación de propiedades sobre el compuesto que se está anotando, como puede observarse en la figura 2.4.1.3 para el formato SDF en particular.

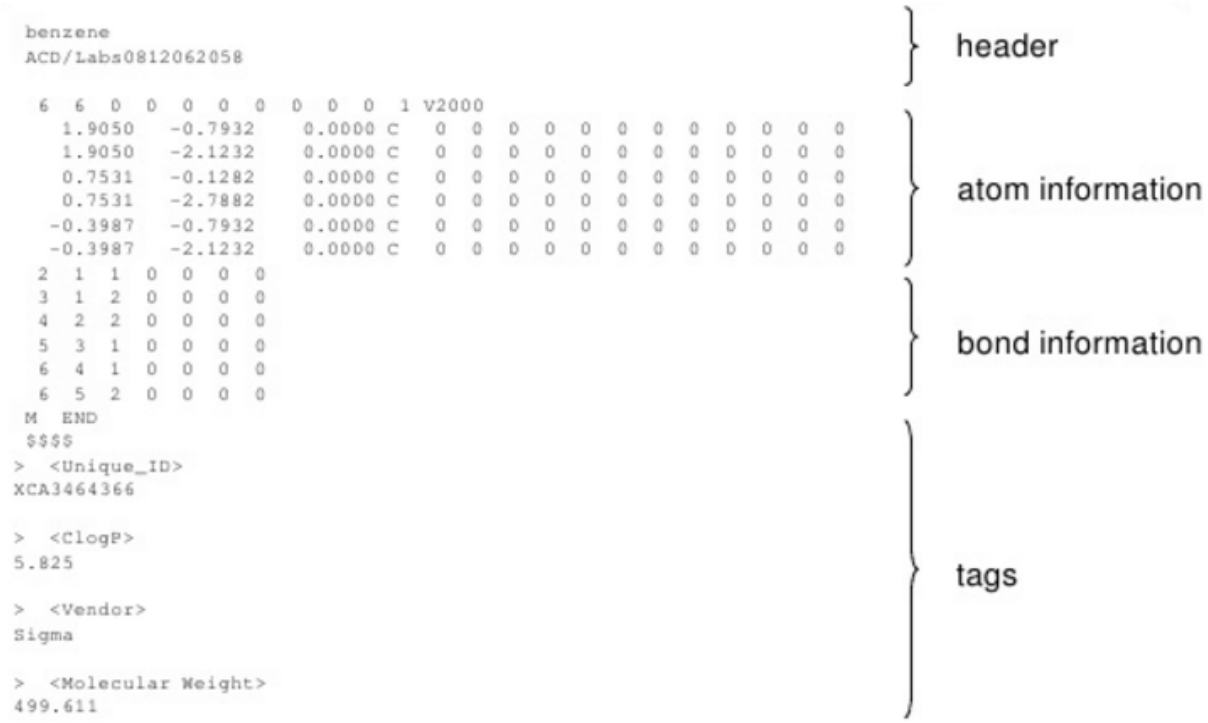


Figura 2.4.1.3: Archivo SDF para la molécula benceno.

Existe una variedad muy grande de formatos los cuales no es nuestra intención describir en detalle, algunos de los más utilizados son SDF, MOL, MOL2, CIF, PDB, COM, etc. En esta tesis, por tratarse del formato que sirve como input en varios algoritmos que son de nuestro interés (algoritmos de docking como rDock[68], por ejemplo), utilizaremos el formato SDF y, en segunda instancia, el ya explicado formato PDB, que además de usarse para moléculas, puede usarse para moléculas pequeñas.

2.4.2 Procesamiento y operaciones sobre compuestos

Una de las principales herramientas disponibles para el tratamiento computacional de moléculas, el cual ha sido utilizado en el frecuente trabajo en muchas de sus

funcionalidades, es Babel. El mismo tiene su origen, en sus primeras implementaciones, en la necesidad de poder interconvertir entre diferentes formatos de descripción y anotación de moléculas, que como ya hemos visto son muy variados. Esta herramienta provee una manera muy simple para, mediante una simple instrucción en línea de comando, proveer un archivo en un formato de entrada y especificar cuál es el formato de salida deseado.

En la figura 2.4.2.1 se muestra la arquitectura y las relaciones entre los distintos módulos de esta herramienta y cómo fluye la información en la misma.

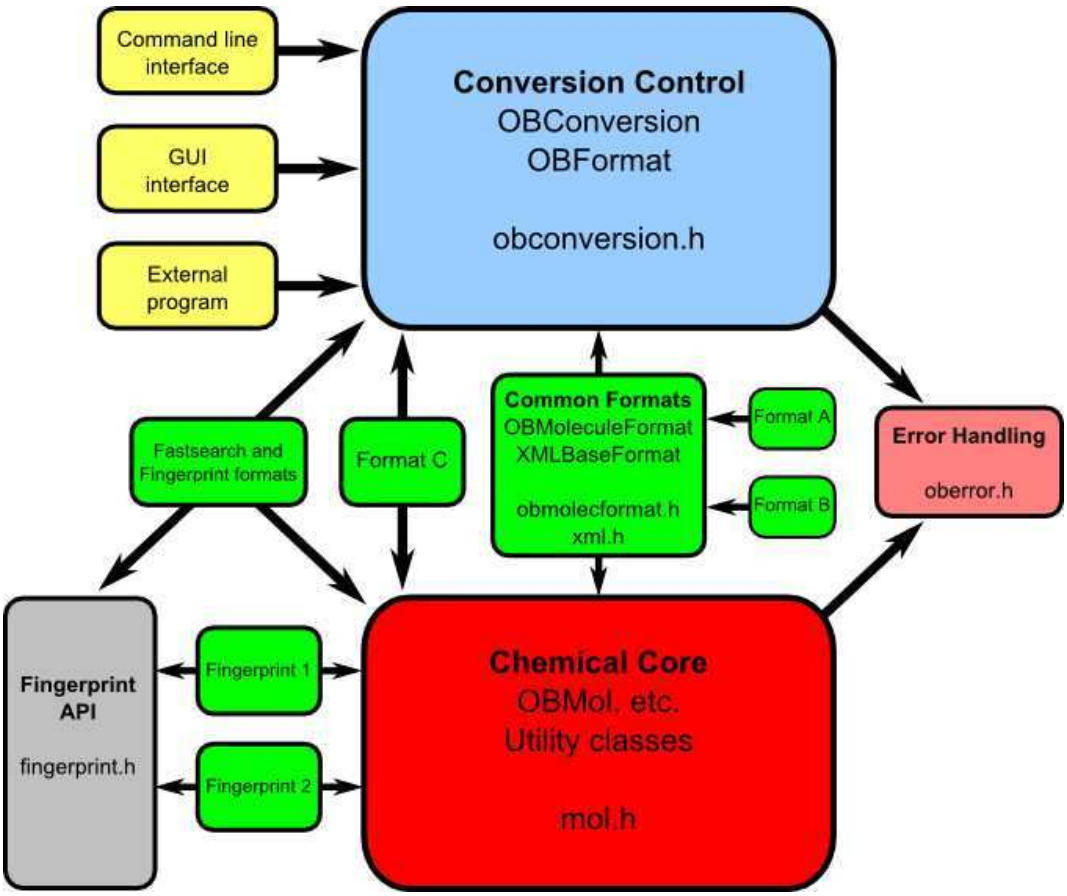


Figura 2.4.2.1: Arquitectura del software Babel, además del módulo de interconversión entre distintos formatos de moléculas, el software provee un motor de generación y búsqueda de fingerprint a través del sistema FastSearch. Otros módulos, no usados en esta tesis, calculan estructura tridimensional, tautomería, etc.

Además de la mencionada funcionalidad de interconversión entre formatos, en este trabajo usamos el software Babel para el cálculo, almacenamiento y búsqueda utilizando fingerprints de moléculas, previamente nombrados en los métodos de esta tesis. El tipo de fingerprint generado para cada molécula (llamado FP2) identifica todos las subestructuras lineales y de anillos que tengan largos de 1 a 7 (excluyendo las subestructuras de un átomo de carbono o nitrógeno). Al fingerprint generado se le aplica una función de hash para generar una cadena de bits de largo 1024.

Si una molécula se usa como búsqueda de tipo "subestructura" contra una base de datos de compuestos, entonces todos los bits "prendidos" en la molécula de entrada deben estar prendidos en la molécula objetivo en la base de datos para que esto se cumpla. Los fingerprints, como ya hemos explicado, también pueden utilizarse para filtrar por similaridad en base al coeficiente de Tanimoto entre moléculas.

La búsqueda de manera repetida sobre la misma base de datos de moléculas, en su implementación *naive* requerirá el chequeo repetido de cada fingerprint, e insumirá una cantidad de cómputo lineal en función de la cantidad de moléculas en la base de datos. Para optimizar esto, Babel incorpora un índice (llamado FastIndex) el cual es almacenado mediante un árbol binario en donde cada par de hijos de un nodo representa que el siguiente bit dentro del fingerprint que se va construyendo al recorrer el árbol está "prendido" o "apagado". Además de eso, cada archivo que almacena el FastIndex, guarda la posición que la molécula que representa el fingerprint que se está almacenando, ocupa en el archivo original: de esta manera, por poner un ejemplo, el acceso a una molécula de las características deseadas, teniendo una base de datos en formato SDF (que para millones de moléculas puede representar una gran cantidad de información), se realiza con un puntero directo a su posición en la base, sin que tener que leer la base de datos completa de manera lineal. Este software es de código abierto, tiene una gran cantidad de

colaboradores que trabajan sobre él, y se encuentra disponible para que cualquier usuario que quiera extenderlo pueda hacerlo.

Otra librería que hemos usado de manera extensiva en el presente trabajo es JChem[69], desarrollada por la compañía ChemAxon. La misma es de uso gratuito para investigación y paga para usos comerciales. La misma nos ha servido para, teniendo una molécula de interés en algún formato, ya sea bidimensional o tridimensional, calcular propiedades y estructuras. La compañía ofrece integrado a la librería una variedad de recursos relacionados con la visualización, almacenamiento y administración de compuestos de las que no hemos hecho uso, prefiriendo en estos casos alternativas de uso libre.

El software cuenta con distintos módulos, de los cuales haremos hincapié en los denominados "Calculator plugins" (módulos de cálculo). Los mismos funcionan de una manera homogénea: se instancia un objeto (en memoria) del tipo Molécula, y se lo pasa como entrada al Plugin, al cual se le ejecuta el método "Run"(correr) y el mismo instancia en el objeto propiedades que a partir de ese momento pueden ser leídas.

Por ejemplo el "Tautomer Generation Plugin", al ser ejecutado sobre una molécula, le instancia una propiedad que es una lista de todos sus tautómeros (que son a la vez moléculas sobre las que pueden realizarse operaciones como con cualquier otra).

En la figura 2.4.2.1 se presenta un listado de módulos de cálculo provistos por la herramienta JChem, entre los que podemos mencionar, por ejemplo, los de generación de tautómeros, estereoisómeros y geometrías, de los que nos valdremos más adelante, o los de cálculo de propiedades como son logP o superficie.

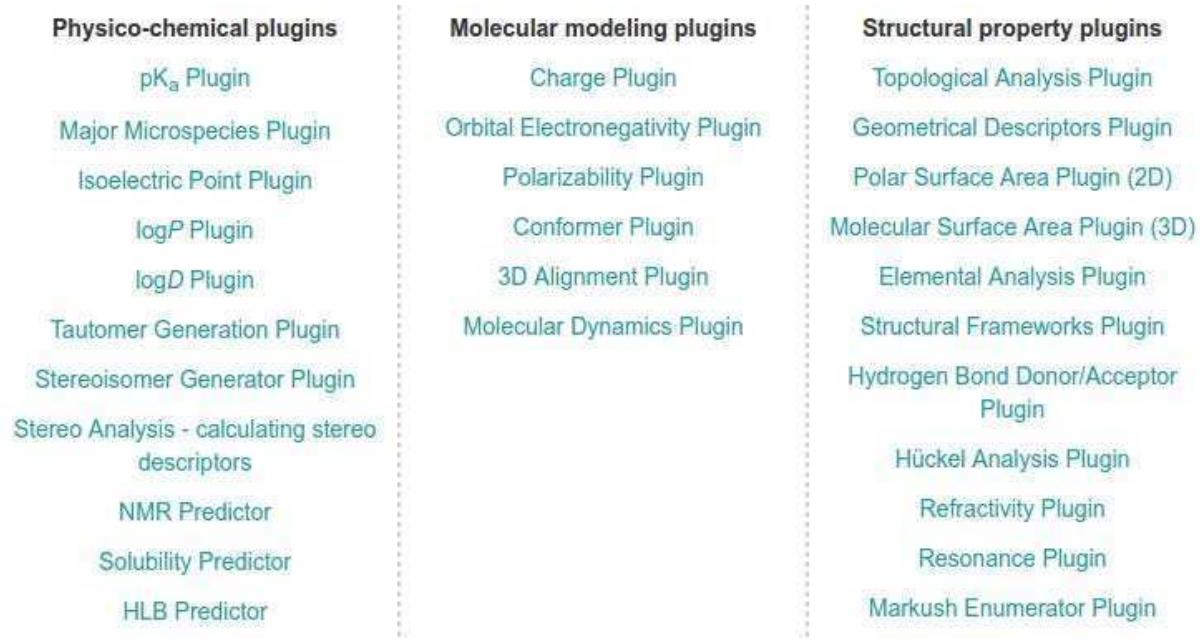


Figura 2.4.2.2: Listado de los plugins de cálculo provistos por el software JChem desarrollado por ChemAxon.

De cada uno de los Plugins de cálculo se provee una documentación pero no el código fuente original para realizar correcciones o extensiones al software. Haremos una descripción aquí de los basamentos teóricos provistos por la herramienta para las propiedades más importantes que hemos calculado utilizando esta herramienta.

El generador de estereoisómeros produce de manera analítica todos los posibles (estereoisómeros) para un compuesto, teniendo en cuenta incluso tetraedros y dobles enlaces como centros estereogénicos. Durante la generación, las conformaciones improbables desde el punto de vista energético son filtrados (y enumerados desde el mejor rankeado energéticamente al peor). Cada uno de los compuestos generados es evaluado con el potencial DREIDING[70] el cual es un potencial de uso general que también se usa para evaluar las geometrías generadas (ver más abajo). De forma análoga, el generador de microespecies realiza un muestreo a diferentes pHs de todas las especies de protonación posible, generando un muestreo de probabilidades.

A modo de ejemplo, en la figura 2.4.2.3, se muestra la distribución de microespecies para una molécula evaluada en diferentes valores de pH. En valores menores a 1.6 o mayores a 6 las poblaciones son totales para cada una de las especies, mientras que para valores intermedios la población está repartida.

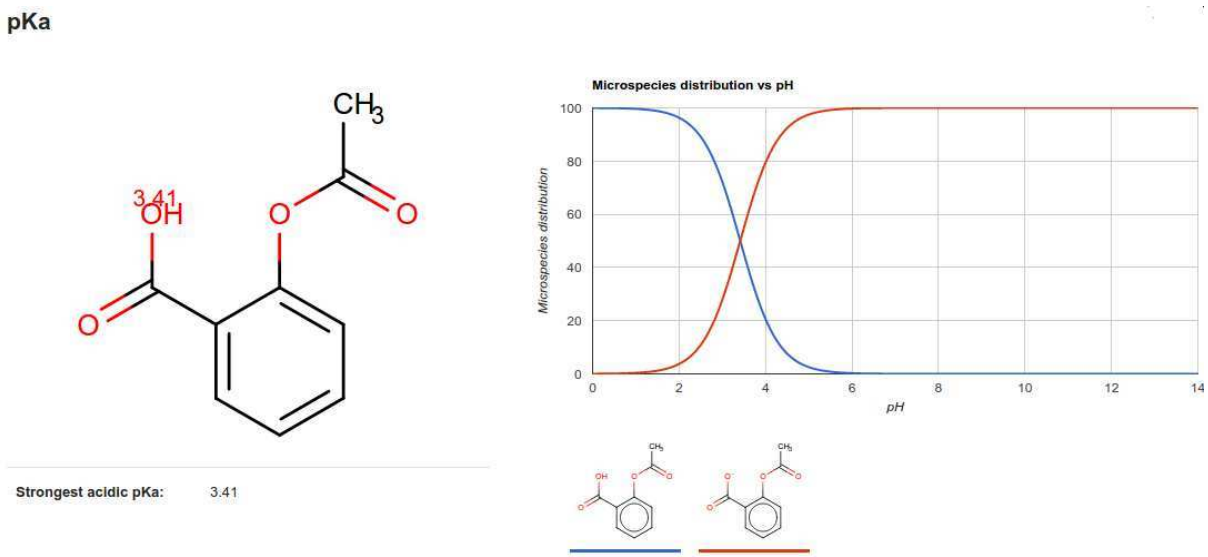


Figura 2.4.2.3: Visualización de la distribución de microespecies en función del pH provisto por la página chemicalize.org.

Una vez determinadas todas las subespecies de un dado compuesto, el método utilizado por la librería para generar geometrías, determina las coordenadas atómicas con un algoritmo paso a paso basado en grafos, mediante una estrategia de divide & conquer. Cada fragmento es una subestructura de la molécula original, con hidrógenos sustituyentes en aquellos enlaces que se cortan para generar fragmentos. El proceso de construcción se representa como un árbol en donde cada nodo representa una conformación parcial del fragmento (subestructura con H sustituyentes, ver figura 2.4.2.4) y un método (fusión de fragmentos o construcción de fragmento) que genera el análisis. El análisis conformacional (incluso generando un único confórmero) es hecho mediante un modelo manejado bajo demanda: un "requerimiento de construcción" es pasado a la raíz del árbol de construcción. Cada fusión fragmento-fragmento intenta cumplir el requerimiento usando confórmeros

generados por los subárboles asociados. Una vez que las posibilidades de fusión son exploradas, un requerimiento adicional se aplica, recursivamente, en cada uno de los fragmentos involucrados.

La descomposición en fragmentos adecuados es hecha mediante múltiples heurísticas (la figura 2.4.2.4 muestra una descomposición a nivel átomo que es solo ilustrativa). Las equivalencias geométricas y topológicas son tratadas mediante las sucesivas capas de fusión (identificando ramas del árbol que representan el mismo compuesto llevando a cabo una superimposición) lo cual acelera el proceso y elimina redundancia. De esta manera, las geometrías resultantes están listas para ser evaluadas energéticamente.

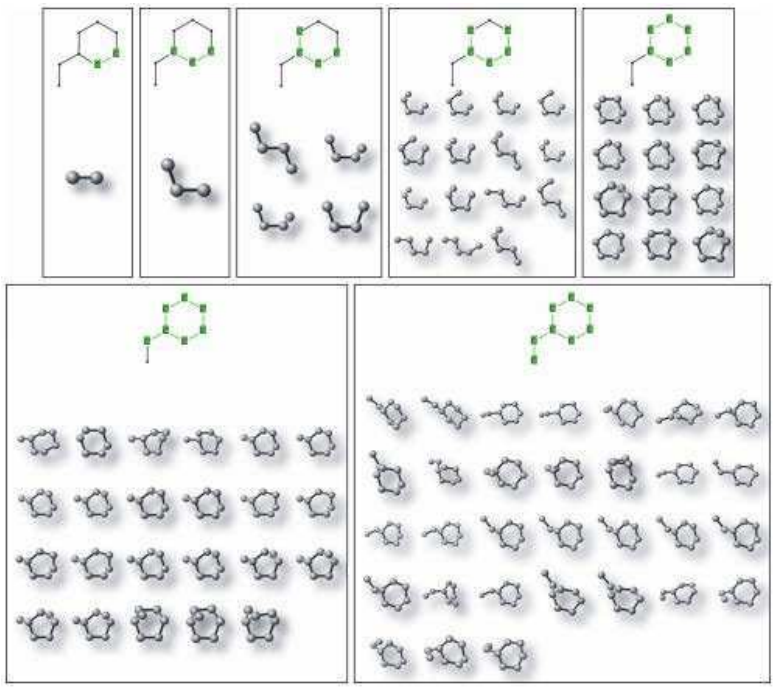


Figura 2.4.2.4: Vizualización del paso a paso de generación de geometrías para un compuesto. La misma funciona a manera ilustrativa, ya que se va agregando átomo a átomo, mientras que en la práctica grupos funcionales de los que se conocen sus ángulos o una serie de conformaciones (por ejemplo anillos) son agregados por el algoritmo de divide and conquer de manera íntegra.

Finalmente, como ya adelantamos, los modelos generados son evaluados con el campo de fuerzas DREIDING, el cual tiene dos componentes, la energía de valencia y la energía de no unión:

$$E = E_{val} + E_{nb}$$

Expresándose cada término de la siguiente manera:

$$E_{val} = E_{estiramiento\ de\ enlace}(2 - \text{á}tomos) + E_{\text{á}ngulo\ de\ enlace}(3 - \text{á}tomos)$$

$$+ E_{torsión\ de\ enlaces}(4 - \text{á}tomos) + E_{inversión}(4 - \text{á}tomos)$$

$$E_{nb} = E_{vdw} + E_{electrostática} + E_{hydrogen\ bonds}$$

La forma funcional de $E_{estiramiento\ de\ enlace}$ es la de un oscilador armónico en torno a un valor de enlace paramétrico, de la misma forma, las $E_{ángulo\ de\ enlace}$ y $E_{torsión\ de\ enlace}$ es la de un oscilador armónico donde los términos del oscilador son los cosenos de los ángulos en torno a valores parametrizados.

En cuanto al término $E_{inversión}$, dado un átomo unido exactamente unido a tres átomos, a menudo es necesario incluir un término que defina cuán difícil es forzar a los tres enlaces en el mismo plano o cuan favorable es mantenerlos en el mismo plano. El término se define de la forma:

$$E_{inversión} = 1/2 * k_{inv}(\theta)^2$$

Siendo k_{inv} una constante empírica y es un término que delata el ángulo de torsión impropia, siendo θ ángulos parametrizados por el campo de fuerza.

Todos los términos de no unión son de la forma Lennard Jones, con diferentes parametrizaciones dependiendo del tipo de fuerza que se esté expresando.

Cada una de las estructuras obtenidas se evalúa en base a la energía calculada con este método y se filtran aquellas de baja energía o se mantienen las mejores en base a la cantidad que pida el usuario.

Otros Plugins de cálculo han sido utilizados para calcular a compuestos desconocidos o que no figuraban en nuestras base de datos todas las propiedades necesarias para agregarlas como un nuevo registro sobre los cuales se puedan hacer búsquedas de compuestos que satisfagan dichas propiedades (logP, HBD, HBA, etc.)

En la página chemicalize.org pueden probarse cada una de estas funcionalidades de manera interactiva usando la librería desarrollada. En este trabajo, utilizamos la API generada en el lenguaje de programación Java, para realizar los cálculos de manera programática y de esta manera ensamblarlo dentro de los pipelines generados.

2.5 Bases de datos biológicas

Uno de los insumos fundamentales de esta tesis es la información contenida en diferentes bases de datos, de las cuales pudimos valernos para extraer información que ha resultado útil en cada uno de los objetivos que nos hemos propuesto cumplir. Las mismas poseen distinta información, sus registros están almacenados en formatos heterogéneos y poseen distintos niveles de anotación. Aquí explicaremos la metodología utilizada para acceder a algunas de ellas.

2.5.1 Bases de datos de secuencias de proteínas: UniProt

La base de datos UniProt (Universal Protein Resource) funciona como un recurso centralizado de secuencias de proteínas. La gran cantidad de datos obtenidos a través de experimentos de secuenciación, discutida más arriba, dejan en evidencia la necesidad de mantener un repositorio de registros unívocos y no redundantes, que permitan acceder a la información actualizada de estas secuencias.

Esta base de datos tiene como registros a las secuencias de cada una de las proteínas conocidas. En ese sentido, podemos considerar a esta base de datos como una base de datos primaria. Diferentes proteínas pertenecientes a distintas cepas de un mismo organismo constituyen registros independientes. Además, este recurso provee una gran cantidad de anotaciones y datos vinculando cada uno de sus registros con muchos otros recursos (generalmente bases de datos) disponibles. Algunas de las propiedades que UniProt provee sobre cada uno de sus registros son (entre otras):

- Anotación funcional general y vinculada a regiones particulares de la secuencia.
- Ontologías
- Contextualización en caminos metabólicos.
- Taxonomía.
- Localización Celular,
- Relación con patogenicidad.
- Mutagénesis.
- Interacciones.
- Estructuras tridimensionales.
- Dominios y familias.
- Splicing alternativo, isoformas.

Toda esta información está disponible para ser consultada a través de la página web www.uniprot.org y puede ser accedida en distintos formatos: directamente a través de la página, en formato FASTA (solo la secuencia y algunos datos generales puestos en la cabecera), XML (con la anotación completa del registro), etc. Cada una de las propiedades, cuando corresponden, tienen además la vinculación a la bibliografía que lo sustenta.

Los registros tienen además distintos niveles de anotación manual, y la base de datos clasifica esta información en registros "Reviewed" (revisados) y "Unreviewed" (no revisados, generados de manera automática extrayendo información de secuenciación). En la figura 2.5.1.1 se muestra el resultado de una búsqueda en la página web de UniProt en la que se listan en primer lugar los registros revisados y luego los no revisados que coinciden con la búsqueda de la "p53".

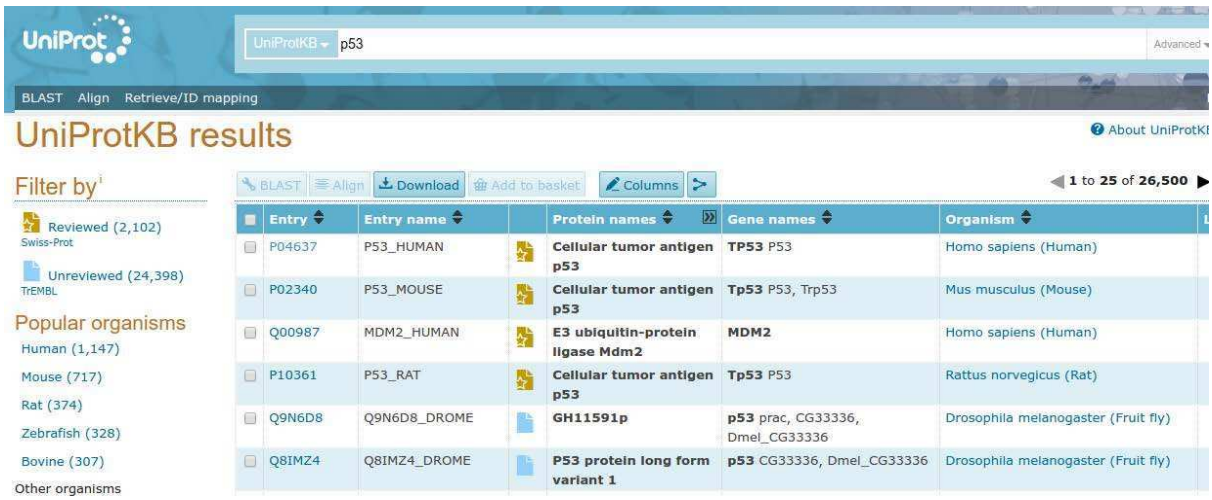


Figura 2.5.1.1: Captura de pantalla de la búsqueda de la palabra clave p53 en el sitio web del la base de datos UniProt. Los registros revisados son beneficiados en su posición de aparición en los primeros puestos del listado resultante.

La base de datos es mantenida por el European Bioinformatics Institute (EBI), el Protein Information Resource (PIR) y el Swiss Institute of Bioinformatics (SIB).

Existen otras bases de datos disponibles de secuencias de proteínas, como puede ser la base de datos 'Protein' incluida en RefSeq, que aunque son recursos valiosos, no poseen el nivel de anotación que sí tiene UniProt, el cual nos resultará particularmente útil a lo largo de toda esta tesis donde, por ejemplo, la usaremos para obtener el conjunto de proteínas resueltas en un organismo, o para conocer qué variantes se encuentran anotadas para una proteína en particular.

2.5.2 Bases de datos de familias de proteínas: PFam

Como ya hemos mencionado, los alineamientos múltiples de secuencia, han demostrado ser herramientas fundamentales en el momento de entender características de la estructura y la función de las proteínas, permitiendo dilucidar la predicción de estructura secundaria[9], el dominio de plegado al que pertenece la proteína evaluada, entre otras utilidades.

Cada registro de la base de datos Pfam[29] representa una familia o dominio de plegado, la cual está constituida por el HMM que la define mediante dos alineamientos: uno denominado 'alineamiento semilla' y otro denominado 'alineamiento completo'.

La denominación de familia tiene relación con la pertenencia en un mismo origen evolutivo mientras que dominio de plegado se refiere a la característica estructural de adquirir una topología similar en el espacio. Pese a esta diferencia etimológica ambos términos pueden utilizarse indistintamente para vincular similares conjuntos de proteínas[71].

Cada familia de la base de datos PFam es construída llevando a cabo los cuatro pasos siguientes:

- Construir un alineamiento semilla de alta calidad.
- Construir un perfil HMM utilizando el software *hmmer3* basado en el alineamiento semilla.

- Buscar en toda la base de datos UniprotKB versus el perfil que acaba de crearse.
- Setear umbrales específicos (GAs, gathering thresholds) para la familia sobre la que se está construyendo la entrada (el HMM) que permitan determinar la pertenencia o no a cada familia de una secuencia de manera óptima. Todas las regiones de secuencia que superen el GA de la familia son incluídos en el alineamiento completo de la misma.

La mayoría de las secuencias de proteína conocidas (76.1% de la base de datos UniProt), pueden ser agrupadas en alguna de las 16295 familias definidas en la base.

Los GAs de cada familia son elegidos con el objetivo de maximizar el cubrimiento y a la vez excluir cualquier falso positivo en la recolección de porciones de secuencias. Si bien el número de falsos positivos para un umbral dado es generalmente desconocido, una forma de monitorear la proporción de falsos positivos indirectamente es chequeando los solapamientos entre una familia y otra: si la misma región de una secuencia es atrapada por dos familias distintas, esto debe considerarse como un falso positivo para alguna de las dos (con la salvedad que las familias pertenezcan a un mismo clan).

Una de las características que las últimas versiones de esta base de datos incorporan, es la definición de familias utilizando únicamente registros de secuencias pertenecientes a proteomas de referencia. En versiones anteriores, los alineamientos semilla y completos se generaban con respecto a la totalidad de la base de datos UniProtKB.

Los proteomas de referencia abarcan secciones representativas de la diversidad taxonómica de los proteomas completos que pueden encontrarse dentro de UniProtKB. Incluyen proteomas bien definidos de organismos que son de interés en biomedicina y biotecnología.

La condición de que las secuencias que forman parte de los alineamientos semilla deban pertenecer a proteomas de referencia, sumadas a la condición de que ninguna secuencia puede tener una identidad mayor al 80% con ninguna de las otras que forman parte del

alineamiento semilla, permiten no sobrerrepresentar secuencias de manera errónea dentro de una familia.

La versión 29 de esta base de datos es la que incorpora estas condiciones, y después de una revisión de la versión 28, fueron encontradas secuencias redundantes en un 96% de las familias previamente definidas, cambiando entonces significativamente los registros de esta base de datos.

Los clanes son elementos agrupados de esta base de datos que comparten un mismo origen evolutivo (osea, grupos de familias). Son establecidos en base a cuatro elementos que son analizados en el momento de establecer las relaciones entre las familias candidatas a ser agrupadas: estructuras relacionadas, función relacionada, solapamiento significativo de secuencias en perfiles HMM y comparaciones entre perfiles.

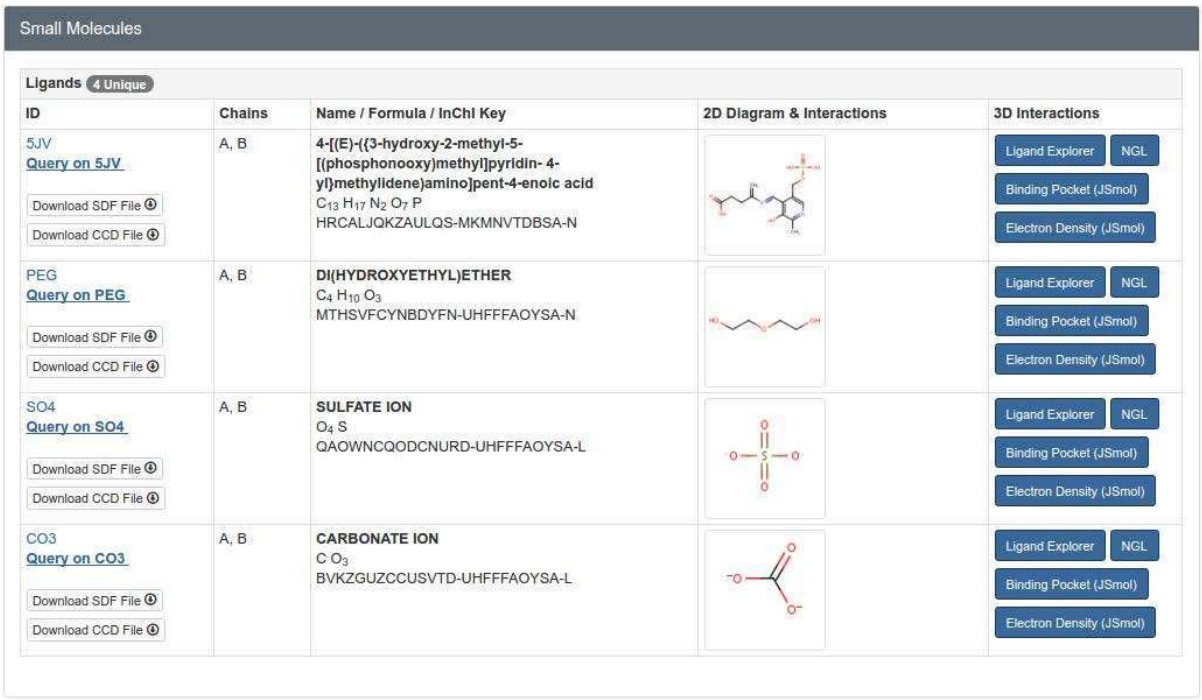
2.5.3 Bases de datos de estructuras: Protein Data Bank

El wwPDB (worldwide Protein Data Bank) es un consorcio internacional que maneja el depósito, procesamiento y publicación de la base de datos unificada de estructuras de macromoléculas determinadas experimentalmente a nivel global. Es manejada por el RCSB (Estados Unidos), EBI (Europa) y PDBj (Japón) y es de libre y público acceso.

Los registros del PDB contienen información de coordenadas 3D a nivel atómico, información acerca del contenido químico tales como la secuencia del polímero y la química de los ligandos, cofactores y solventes presentes en el experimento donde se resolvió la estructura de la macromolécula que representa la entrada en la base de datos. El archivo contiene también información general acerca del experimento usado para obtener cada estructura y parámetros de calidad acerca de las mismas.

Otra información útil es la vinculada con otras bases de datos, como por ejemplo qué parte de la proteína o proteínas han sido cristalizadas en el experimento, cuáles son los residuos que han sido modificados en caso de que existan, cuales son las moléculas que

han sido co-cristalizadas con la macromolécula objetivo (incluso conociendo información relacionada con la afinidad en caso de que exista el dato[72]), etcétera.



Small Molecules

Ligands 4 Unique

ID	Chains	Name / Formula / InChI Key	2D Diagram & Interactions	3D Interactions
5JV Query on 5JV Download SDF File Download CCD File	A, B	**4-[(E)-({3-hydroxy-2-methyl-5-[(phosphonooxy)methyl]pyridin- 4-yl}methylidene)amino]pent-4-enoic acid** C_{13} H_{17} N_2 O_7 P HRCALJQKZAULQS-MKMNVTDBSA-N		Ligand Explorer NGL Binding Pocket (JSmol) Electron Density (JSmol)
PEG Query on PEG Download SDF File Download CCD File	A, B	**DI(HYDROXYETHYL)ETHER** C_4 H_{10} O_3 MTHSVFCYNBDYFN-UHFFFAOYSA-N		Ligand Explorer NGL Binding Pocket (JSmol) Electron Density (JSmol)
SO4 Query on SO4 Download SDF File Download CCD File	A, B	**SULFATE ION** O_4 S QAOWNCQODCNURD-UHFFFAOYSA-L		Ligand Explorer NGL Binding Pocket (JSmol) Electron Density (JSmol)
CO3 Query on CO3 Download SDF File Download CCD File	A, B	**CARBONATE ION** C O_3 BVKZGUZCCUSVTD-UHFFFAOYSA-L		Ligand Explorer NGL Binding Pocket (JSmol) Electron Density (JSmol)

Figura 2.5.2.1: Moléculas unidas al registro 5E3K. Además de vincularse los registros, como podemos observar la base de datos permite obtener la molécula en otros formatos como SDF y CCD, conocer cuál es el bolsillo unido a cada uno de los compuestos, visualizarla en tres dimensiones, etcétera.

Cada una de las estructuras depositadas en la base de datos es sometida a un proceso de validación que incluye el chequeo de distancias y ángulos de enlaces covalentes, validación estereoquímica, correcto uso de nomenclatura, comparación de secuencia contra SeqRes (base de datos que contiene las secuencias de referencia de los registros incluidos en la base), distancias de solvente, etc. Aquellas estructuras que superan estos chequeos pasan a constituir un nuevo registro.

2.5.4 Bases de datos de variantes

Un aspecto clave de cualquier investigación en el campo de la genética es la asociación de variantes genéticas con fenotipos. Se estima que las variaciones entre un par de cromosomas humanos cualesquiera ocurren cada aproximadamente 1200 pares de bases, siendo los polimorfismos de nucleótido simples (SNPs por sus siglas en inglés) las variaciones genéticas más comunes. El estudio de estas variaciones tiene gran interés en el campo tanto de la genética[73], como de la fármaco-genómica[74,75], la genética poblacional[76,77], etc. Para cubrir la necesidad de tener propiamente catalogadas estas variaciones y sus reportes el NCBI estableció en 1998 la dbSNP[78] (Single Nucleotide Polymorphism Database). El mismo funciona como un repositorio central y público a nivel global de variaciones genéticas.

Una vez que las variaciones son identificadas y catalogadas en la base de datos, las mismas pueden ser anotadas y clasificadas en las siguientes categorías: (i) sustituciones de nucleótido simple (99.77% de la base); pequeñas inserciones o deleciones (0.21%); y regiones invariantes de secuencia, repeticiones microsatélites, etc. las cuales constituyen una porción insignificante de la base que no es de utilidad en el presente trabajo.

En dbSNP no hay ningún requerimiento si asuncion acerca de frecuencias alélicas mínimas o neutralidad funcional para que un SNP constituya un registro, funcionando como un repositorio general de las mismas. Por su alcance, esta base contendrá tanto las mutaciones de interés clínico (las cuales serán de especial utilidad en el desarrollo de esta tesis) como aquellas de efecto neutral. Las entradas de esta base de datos tienen anotado como datos de registro las condiciones específicas experimentales en las que se obtuvo la secuencia que contienen la variación, la descripción de la población conteniendo la variación y la información de frecuencia por población o por genotipo individual.

En dbSNP la gran mayoría de los registros son de *Homo sapiens*, sin embargo existen entradas para *Mus musculus* y en mucha menor medida para otros organismos. En principio, la base acepta variaciones reportadas de cualquier organismo.

Para los desarrollos realizados en esta tesis, la información de dbSNP es accedida a través de su anotación en UniProt, donde cada registro, que representa una variación en el genoma, puede ser (cuando corresponda) mapeada en una variación en la secuencia de una proteína a la que accedemos de manera programática.

Otra de las bases de datos de variaciones que hemos usado es este trabajo es ClinVar[79], la cual ha sido creada para garantizar un acceso programático a variaciones que sean de importancia médica en el genoma humano. Así como dbSNP es una base de datos que puede ser considerada primaria, ClinVar es una base de datos secundaria construida a partir de esta y de dbVar[80] (base de datos de NCBI que anota variaciones a nivel de estructura del genoma), aunque también se aceptan en la misma entradas de nuevos registros de manera independiente.

La información contenida en cada registro tiene anotaciones relacionadas con el fenotipo, interpretación de efectos funcionales y significancia clínica, además de la metodología utilizada para obtener las variantes y la evidencia que sostiene su existencia (incluyendo citas y número de casos).

Los diferentes tipos de significación clínica anotados por ClinVar para cada variante son: benignas, probablemente benignas, efecto desconocido, probablemente patogénicas, patogénicas, afecta respuesta a drogas, etc.

Las variaciones, desde el punto de vista técnico, son accedidas a través de la API provista por el recurso SwissVar[81], cuyo acceso de manera programática es mucho más intuitivo. En la figura 2.4.5.1 se muestran los registros de variaciones mapeados a proteína para el registro del gen HRAS humano que provee SwissVar y que son inferidas de clinvar.

Accession	Entry name	Disease	Variants	3D mapping (variant position)
P01112	RASH_HUMAN	costello syndrome	p.Gly12Ala p.Gly12Ser p.Gly12Val p.Gly12Glu p.Gly12Asp p.Gly12Cys p.Gly13Cys p.Gly13Asp p.Thr58Ile p.Lys117Arg p.Ala146Val p.Ala146Thr	
P01112	RASH_HUMAN	congenital myopathy with excess of muscle spindles	p.Gly12Ser p.Gly12Val p.Gln22Lys p.Glu63Lys	
P01112	RASH_HUMAN	thyroid cancer, non-medullary, 2	p.Gln61Lys	
P01112	RASH_HUMAN	variety of human tumors		
P01112	RASH_HUMAN	bladder cancer		
P01112	RASH_HUMAN	schimmelpenning-feuerstein-mims syndrome	p.Gly13Arg	

Figura 2.5.4.1: Listado de mutaciones reportadas para el código UniProt P01112 (HRAS) en la página web de SwissVar agrupadas por la enfermedad a la que están vinculadas. Además de la visualización web, existe un api para su acceso programático como ya hemos explicado para otras bases de datos.

Otra de las fuentes de SNPs que hemos explorado en este trabajo ha sido ExAC (por Exome Aggregation Consortium, sus siglas en inglés). El mismo, es una asociación de investigadores e institutos que han aunado esfuerzos con el fin de homogeneizar la información proveniente de experimentos de secuenciación de exomas y presentar la información de una manera sumarizada.

La base de datos está compuesta en la actualidad por la información de ~60000 individuos no relacionados secuenciados en diferentes proyectos relacionados con genómica poblacional. El objetivo de EXAC es dotar a cada una de las mutaciones de su frecuencia alélica.

2.5.5 Bases de datos de compuestos

En el desarrollo de esta tesis hemos usado como insumo diferentes bases de datos de moléculas, los cuales proveen información similar acerca de los compuestos pero que tienen su origen en distintos tipos de información.

Una de ellas es ChEMBL, la cual ha sido construída en base a la información producto de ensayos de bioactividad de compuestos extraídas de publicaciones científicas. Para cada uno de los ensayos cargados en la base, se anotan qué tipo de ensayo se está cargando (actividad, ADME, kD, etc), cuáles son los compuestos y cuáles son los targets (proteínas, células, organismos, etc), así como los resultados del mismo en cuanto a información de actividad, especificidad, etc. Particularmente, en esta tesis hemos estado interesados en los compuestos que tengan al menos un ensayo que los marque como activos para algún target del tipo proteína. Además de la información provista por literatura, ChEMBL incorpora la información provista sobre actividad de todos los compuestos aprobados por la FDA (Food & Drug Administration).

Los datos de esta base están disponibles para ser montados de manera local mediante una base de datos MySQL. De cada uno de los compuestos, además de saber contra qué targets ha sido ensayado, se proveen sus estructuras tanto bidimensionales (SMILES, InChI) como tridimensionales (SDF). Además se proveen una serie de propiedades que hemos usado a manera de filtro y para relacionar con los sitios activos que hemos postulado que poseen las proteínas que presumimos son sus blancos: logP, logS, logD, volumen, carga, dadores/aceptores de puente de hidrógeno, cantidad de enlaces de anillo, etc.

Otro aspecto que nos interesa de los compuestos es la capacidad de los mismos de ser adquiridos. La base de datos ZINC ha sido creada con el objetivo de facilitar a los investigadores la superación de esa barrera que representa no tener fácilmente disponibles

los compuestos para llevar a cabo experimentos de screening. En la misma, además de las propiedades básicas que hemos mencionado nos interesaban extraer en la base de datos ChEMBL, se tiene información acerca de los proveedores que pueden brindar al investigador el compuesto de interés.

Como ya hemos mencionado en la introducción de este trabajo, el acceso a esta base de datos ha sido diseñado de manera de poder realizar búsquedas de manera programática para acceder a los compuestos en el formato deseado (SMILES, SDF, etc) permitiendo realizar búsquedas por similitud química, subestructura y otras en el lado del servidor.

Además de servir para realizar búsquedas, la misma funciona también como catálogo para los vendedores de compuestos y también se han creado diferentes datasets que pueden ser de gran utilidad para diferentes aplicaciones: subconjunto de drogas, subconjunto de productos naturales, building blocks[82], compuestos biogénicos, etc. De las mismas se ofrecen además de poder descargarlas en diferentes formatos, distribuciones analizadas que permiten realizar estadística e inferencia estadística sobre los mismos.

A modo de ejemplo, en la figura 2.5.5.1 se muestra para un conjunto de moléculas los valores de carga neta versus los valores de desolvatación polar.

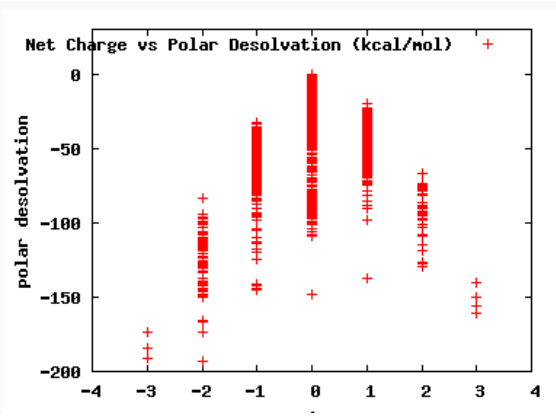


Figura 2.5.5.1: A manera de ejemplo, dos de las propiedades que pueden enfrentarse para hacer estadística, carga neta versus desolvatación polar aplicado al subset de compuestos etiquetados como droga en ZINC.

Un mismo compuesto, es claro, puede ser adquirido en más de un proveedor y sobre el espacio de compuestos pueden realizarse clusterizaciones a diferentes niveles de similaridad para tomar compuestos que funcionen como representantes de clusters.

Como ya hemos dicho, una de las principales aplicaciones que se le da a los compuestos adquiribles es someterlos a experimentos de screening con el objetivo de identificar moléculas con actividad biológica sin la necesidad de realizar el proceso de síntesis orgánica el cual es costoso tanto en recursos como en tiempo. En función de las propiedades de los compuestos sometidos a un proceso de clasificación, se ha caracterizado el espacio de compuestos adquiribles[83] de manera de cuantificar la cantidad de compuestos disponibles con aplicaciones determinadas, como se muestra en la figura 2.5.5.2.

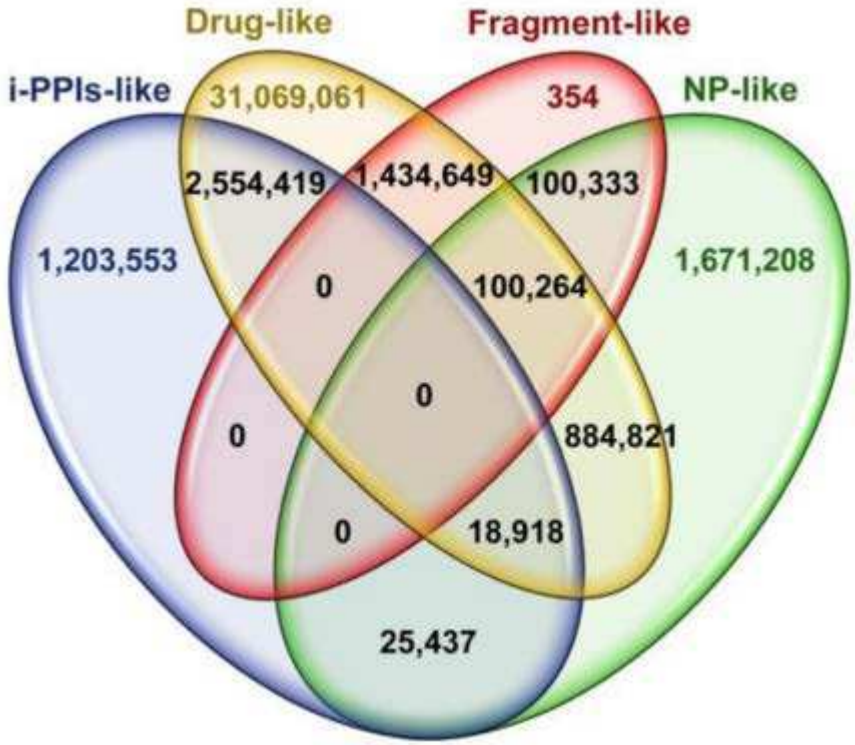


Figura 2.5.5.2: Análisis mediante un diagrama de Venn del espacio de compuestos adquiribles. PPI es la sigla de "protein protein inhibitor", osea inhibidores de la actividad proteína proteína. En este gráfico, además de los compuestos presentes en la base de datos ZINC, se han analizado los compuestos de la base de datos PurchasableBox.

En esta tesis, hemos construído una base de datos local de compuestos la cual es resultado del procesamiento de las bases de datos ZINC y ChEMBL, en la que hemos incluido los compuestos que son adquiribles y que tienen un ensayo marcado como activo para un blanco de tipo proteico. El mismo es de especial interés ya que en nuestra herramienta LigQ buscamos moléculas pequeñas tipo drogas que tengan buenas chances de modular la actividad proteica.

De cada uno de los compuestos guardamos las propiedades mencionadas más arriba y sus estructuras bidimensionales y tridimensionales. Con las mismas hemos también creado un FastIndex que permite su fácil acceso, usado por nuestra herramienta.

3. Desarrollos

3.1. Selección de potenciales blancos proteicos drogables en genomas (TuberQ)

3.1.1 Introducción

De acuerdo a reportes de la Organización Mundial de la salud, alrededor de ocho millones de personas por año desarrollan Tuberculosis (TB) y cerca de 1.3 millones de personas mueren al año por motivos relacionados con esta enfermedad. Los tratamientos comunes para TB involucran tratamientos de largo plazo con los denominados fármacos de primera línea[84] (Isoniazida, Rifampicina, etc.). Sin embargo, la aparición de cepas de *Mycobacterium Tuberculosis* (Mtb) resistentes a múltiples drogas y las interacciones negativas entre drogas con algunos medicamentos para el HIV (y otras enfermedades), ponen en evidencia la necesidad de nuevas alternativas para el tratamiento de esta enfermedad.

El genoma de Mtb abarca alrededor de 4000 genes y su conocimiento ha abierto las puertas a la búsqueda de nuevas aproximaciones terapéuticas[85,86]. En particular, el análisis a nivel del genoma tiene el potencial de extraer información valiosa para el desarrollo de nuevas drogas, al permitir identificar nuevos blancos antibacterianos. En los últimos años diferentes bases de datos han aparecido integrando datos a nivel de variaciones, información proteica, transcriptoma[87], etc. En esta tesis, hemos llevado a cabo la construcción de una base de datos centrada en la predicción de la drogabilidad estructural con miras a proyectos de diseño de drogas, con el objetivo de seleccionar a nivel genómico aquellos blancos proteicos que tengan las características necesarias para generar nuevas

terapias, basándonos en el concepto en el mencionado concepto de drogabilidad que ya ha sido explicado en la sección de métodos general de esta tesis.

Las drogas antibacterianas tienen como objetivo, en relación con su efecto biológico, matar a la bacteria (o evitar su replicación), bloqueando la función de una o varias proteínas blanco. Para incluir esta información en nuestro desarrollo, incorporamos a nuestra base de datos información relacionada con la esencialidad de los genes analizados, cuya inhibición puede redundar en efectos bacteriostáticos o bactericidas. La determinación de esencialidad de genes en Mtb fue establecida en base a la incorporación de datos de experimentos de mutagénesis[88], y en estudios *in-silico* basados en el balance de flujo metabólico[89] , que explicaremos más adelante.

Para llevar a cabo su efecto inhibitorio, los compuestos tipo droga se acoplan al sitio activo de las proteínas objetivo. Esta propiedad, la hemos inferido a partir del Druggability Score(DS de aquí en adelante, puntaje de drogabilidad estructural) ya explicado en los métodos de esta tesis, combinado con los demás factores que venimos enumerando.

Con respecto a la relevancia de los potenciales blancos proteicos en el estado patológico, muchos trabajos han puesto su atención en genes relacionados con la patogenicidad de Mtb usando sobre todo datos extraídos de experimentos de expresión en microarrays en diferentes condiciones, las cuales se suponen que reproducen algunos aspectos del entorno del bacilo dentro del macrófago[90,91]. Nuestro desarrollo ha incorporado e integrado toda esta información con el objetivo de tener en cuenta la sobreexpresión de genes en condiciones de estrés nitro-oxidativo. Para ello ha sido necesario la curación de datos de literatura, en ocasiones de forma manual.

Entonces, para contribuir a la búsqueda de nuevas drogas anti tuberculínicas desde un punto de vista centrado en los blancos proteicos, en este desarrollo generamos una base de datos del proteoma completo de Mtb, el cual hemos llamado TuberQ[92]. En esta base, relacionamos la drogabilidad estructural de toda aquellas proteínas que tienen resuelta su

estructura tridimensional sumada a aquellas para las que ha sido posible generar modelos por homología de buena calidad. Se ha calculado y recabado información sobre pockets, sitios activos, esencialidad, expresión bajo estrés, conservación, *off-targeting*, etc. Toda esta información integrada, es útil para descartar genes que aparentan ser buenos targets en base a su relevancia biológica, pero que no poseen buenas probabilidades de afinidad a un compuesto tipo droga, o para el descubrimiento de nuevos bolsillos drogables, incluyendo sitios alostéricos, en blancos actualmente conocidos, además de la aplicación directa que es la de encontrar blancos drogables novedosos.

3.1.2 Materiales y Métodos

El *pipeline* desarrollado consiste en la creación de una base de datos siguiendo los pasos descrito en la figura 3.1.2.1. Las secuencias de todos los Open Reading Frames (ORFs) objetivo y sus datos asociados fueron descargados de la base de datos UniProt. Todos los ORF fueron analizados con el software *hmmer3* asociando de esa manera sus dominios estructurales. Luego, cada ORF es sometido a una búsqueda con el software *blast* contra la base de datos de secuencias del PDB completo, para determinar cuándo un ORF (o una porción del mismo asignada a una familia) tiene una estructura tridimensional resuelta.

Basados en los resultados de los procesos ejecutados, cada ORF (o dominio asignado dentro del mismo) es etiquetado como "resuelto" o "no resuelto" con respecto a su estructura tridimensional. La estructura de los ORFs (o dominios) no resueltos es modelada, siempre que sea sea posible obtener un modelo de buena calidad, de acuerdo al pipeline de modelado descrito en la sección de métodos de esta tesis.

Para cada estructura (tanto resuelta como modelada) se han calculado luego las siguientes propiedades: sus bolsillos estructurales y su Drogabilidad Estructural (DS), su similaridad con proteínas humanas (para evaluar sus potenciales efectos de *off-targeting*,

dado que cualquier diseño de droga anti-TB debe ser específico para la bacteria y no interferir con las proteínas del hospedador), los sitios activos cuando estos puedan ser conocidos o inferidos, los residuos que tienen un alto índice de conservación (en bits), su potencial sensibilidad a especies reactivas de nitrógeno/oxígeno debidas a la presencia de residuos o cofactores específicos en el sitio activo y su esencialidad para la supervivencia del organismo.

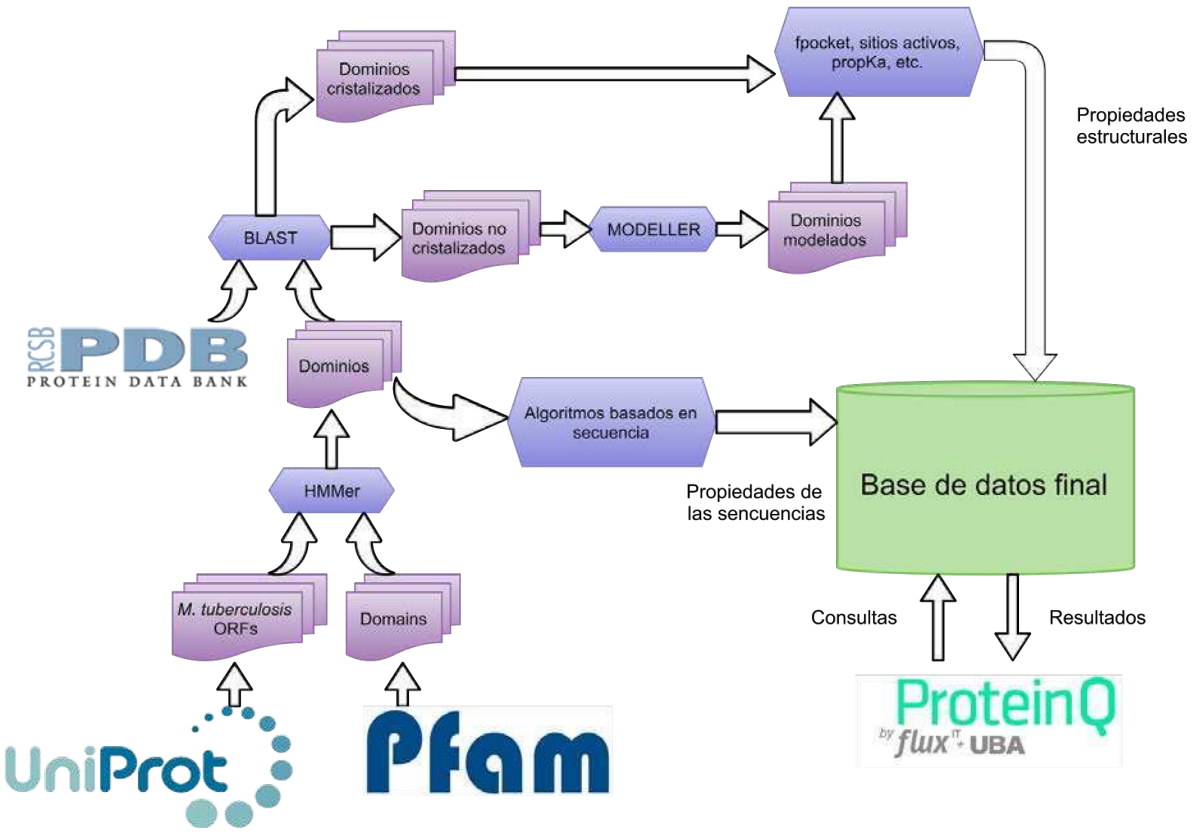


Imagen 3.1.2.1: El pipeline dearrollado para poblar la base de datos (ProteinQ), en primera instancia con la información sobre Mtb (TuberQ) y luego extendido a otros organismos.

Los ORFs con los que se ha poblado la base de datos TuberQ son los del genoma de referencia de Mtb: H37Rv. La cantidad de registros es 3982 ORFs. Luego de ser analizados estos registros, se asignaron 5822 dominios de la base de datos de PFam a algún ORF,

mientras que 1255 ORFs no tuvieron ninguna asignación de dominio. Un mismo dominio puede ser asignado a más de un ORF, en ese sentido, un total de 1658 dominios únicos fueron asignados a algún registro dentro del genoma.

Para determinar cuáles de los blancos proteicos podrían ser relevantes bajo condiciones de estrés nitro oxidativo, llevamos a cabo un análisis extensivo de la literatura derivada de experimentos de microarrays realizados en una variedad de condiciones de las cuales que se sabe o se teoriza que reproducen el entorno deseado. Los diferentes modelos con los que se han llevado a cabo estos experimentos son condiciones de hipoxia, falta de nutrientes, y reactividad al estrés nitro-oxidativo. Se han incluido también en el análisis cuatro criterios de esencialidad provenientes de análisis a nivel de genoma completo en Mtb[90,91,93,94].

En el momento de la creación de la base de datos, se encontraban disponibles 441 estructuras tridimensionales de proteínas de Mtb en PDB. Para el resto de los ORFs cuya estructura no estaba resuelta, se intentó generar un modelo por homología, pudiéndose generar modelos de alta calidad para 903 ORFs, lo cual representa el 34% del proteoma.

A todas las estructuras se las ha evaluado con el software *fpocket* con el fin de encontrar sus bolsillos, con interés en aquellos que puedan clasificarse como drogables. Basado en un análisis preliminar de la distribución de DS, hecho para todos los pockets que alojen un compuesto tipo droga en cualquier proteína co-cristalizadas con un compuesto de este tipo en el PDB, se ajustó la distribución en base a una distribución normal (figura 3.1.2.2). El centro de la distribución es un valor muy cercano al 0.7 de DS y el desvío estándar es de ~0.2. Por ello, hemos clasificado como no drogables aquellos bolsillos alejados en más de 2.6 desviaciones estándar por debajo de la media (DS < 0.2), como pobremente drogables los por debajo en más de una desviación estándar (DS < 0.5), como drogables a los que están a menos de una desviación estándar de la media (DS < 0.7) y como altamente drogables a los que están por encima de la media.

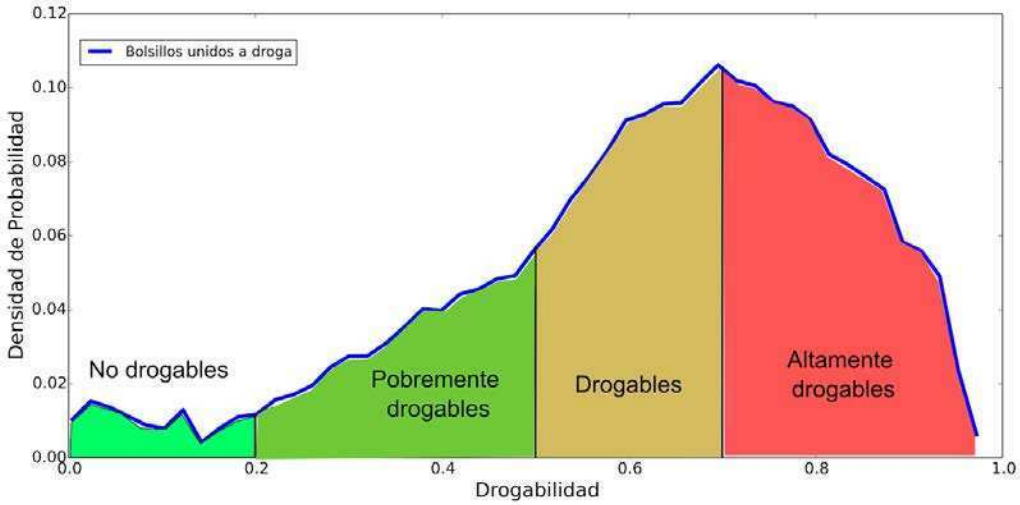


Figura 3.1.2.2: Distribución de drogabilidad de todos los bolsillos que contienen un compuesto tipo droga alojado en cualquier cristal del PDB. Para evitar sobrerrepresentar bolsillos de proteínas que tienen muchos cristales, si un mismo compuesto tipo droga se encuentra en más de un cristal de la misma proteína, se toma para ese caso el promedio de DS.

Para tener en cuenta los posibles estados de de oligomerización, para cada proteína a la cual le hemos asignado una estructura que forma parte de un complejo, se han añadido en el análisis la información relacionada con drogabilidad estructural relativa a todas las subunidades del complejo cristalizado y al complejo como una unidad. De esta manera, se pueden tener en cuenta en el análisis bolsillos drogables pertenecientes a la interfaz proteína-proteína con miras a desarrollar compuestos tipo droga que actúen sobre estas regiones afectando la interfaz. Otro de los factores a tener en cuenta para el análisis de bolsillos es el de la flexibilidad que la proteína pudiera presentar: si la proteína analizada presenta más de una estructura tridimensional resuelta, la base de datos presenta la información de drogabilidad para cada una de las estructuras disponibles.

Para etiquetar a un bolsillo como "sitio activo" de una proteína, nuestro *pipeline* se basa en dos análisis. El primero se basa en la información disponible en la base de datos

Catalytic Site Atlas[27]: cuando un residuo catalítico forma parte de un bolsillo drogable ese bolsillo es marcado como sitio activo de la proteína. El segundo análisis se basa en el estudio de la conservación que presenta cada uno de los residuos que forman parte de la secuencia de la proteína en la familia PFam que fue asignada a la misma (cuando la asignación se hubiere producido). Cuando una posición tiene un alto valor en bits y la misma posición coincide con el aminoácido conservado en la secuencia de la proteína interpretamos que dicha posición está altamente conservada. La misma puede estar altamente conservada porque es importante estructuralmente o porque es importante para la actividad enzimática. En el primero de los casos, estas posiciones no deberían pertenecer en principio a un bolsillo ya que estos se encuentran en la superficie de la estructura. En el segundo de los casos, cuando forma parte de un bolsillo drogable, podemos decir que el mismo posee una posición importante para la actividad enzimática y por lo tanto etiquetamos al mismo como parte del sitio activo.

3.1.3 Resultados

El *pipeline* para la generación de la base de datos puede ser aplicado a cualquier organismo o proteoma que quiera analizarse de manera completamente automática. La herramienta ha sido nombrada ProteinQ y su primera aplicación ha sido, como hemos estado describiendo sobre el proteoma de Mtb. La base de datos puede ser consultada en http://tuberq.proteinq.com.ar con una interfaz de usuario que ofrece muchas facilidades para distintos tipos de usuario y objetivos.

Las búsquedas con las que se puede acceder a los registros son a través de código de UniProt, PFam, PDB o cualquier palabra que pertenezca al nombre de cualquiera de esos tres registros, como muestra la figura 3.1.3.1.

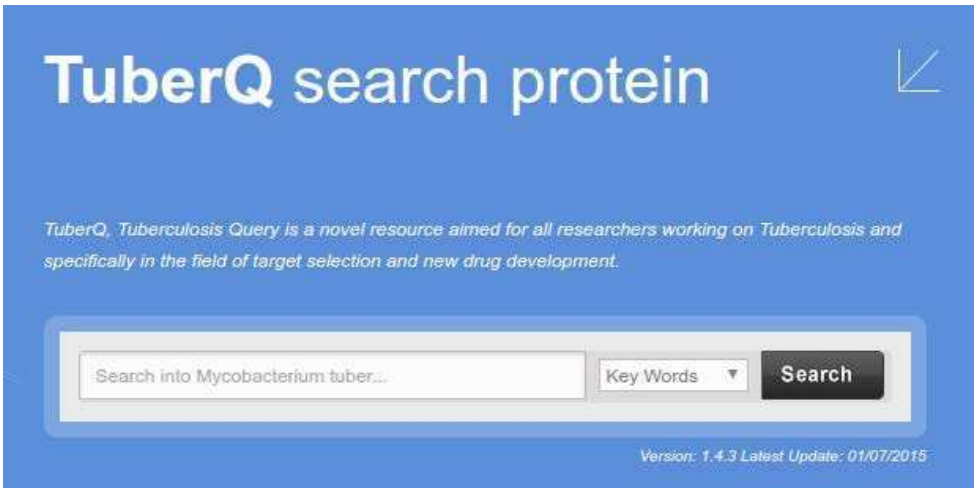


Figura 3.1.3.1: Pantalla inicial del sitio web de la base de datos TuberQ, el cual ofrece realizar búsquedas mediante diferentes criterios.

Los registros encontrados pueden agruparse en la página de resultados mediante el código UniProt o pueden desglosarse mediante la identificación UniProt-PFam-PDB, como se muestra en la figura 3.1.3.2.

Cada registro tiene tres solapas que presentan informaciones de distintos tipos. La primer solapa presenta información relacionada con la secuencia del registro que se está consultando: el resultado de asignar mediante *hmmer* una o más familias al ORF, el resultado de *blast* tanto contra PDB para asignar estructura tridimensional, cuando corresponde, como así también el match más significativo contra el proteoma humano, para sacar conclusiones relacionadas con *off-targeting*. También se remarcan aquellos residuos importantes con los criterios previamente especificados.

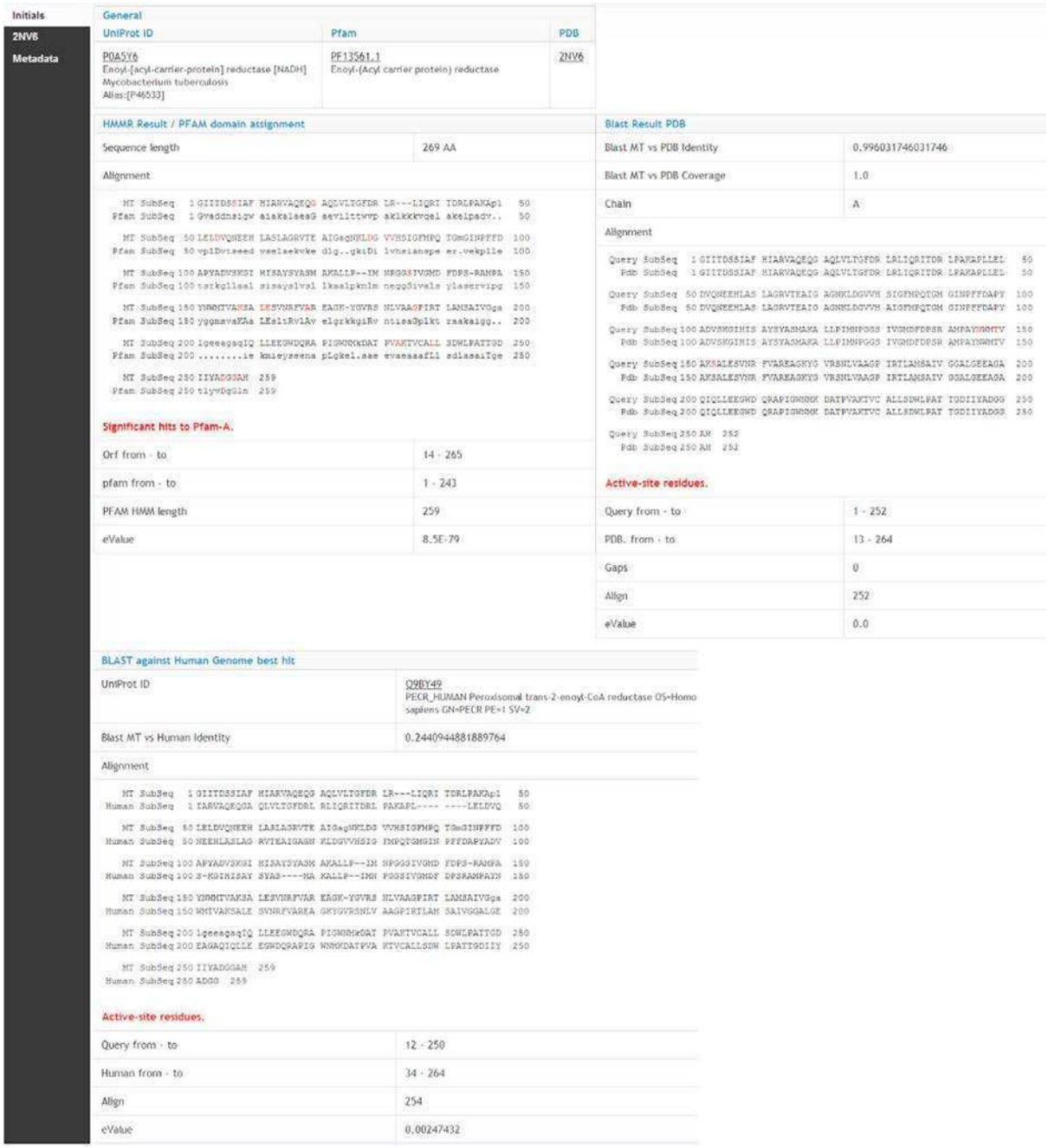


Figura 3.1.3.2: Página de información secuencial para un registro de la base de datos tuberQ. Pueden observarse los resultados de hmmer, blast y off-targeting, en donde se remarcan los aminoácidos relevantes.

La segunda solapa presenta información estructural, tanto si se trata de una proteína que tiene resuelta su estructura cristalográfica, como si tenemos un modelo por homología. En el caso de modelos, también está disponible para ser consultada la estructura que ha

servido como molde para construir el modelo en una solapa adicional. De las estructuras pueden visualizarse los residuos catalíticos provenientes de la base de datos CSA, los residuos importantes en cuanto a la información de los mismos en PFam, los bolsillos drogables, los heteroátomos, entro otras propiedades.

La visualización de la estructura se realiza con la tecnología GLMol[95], como puede verse en la figura 3.1.3.3, pero pueden descargarse los archivos para visualizarla de manera local en otros programas de visualización como PyMol[96] o VMD[97].

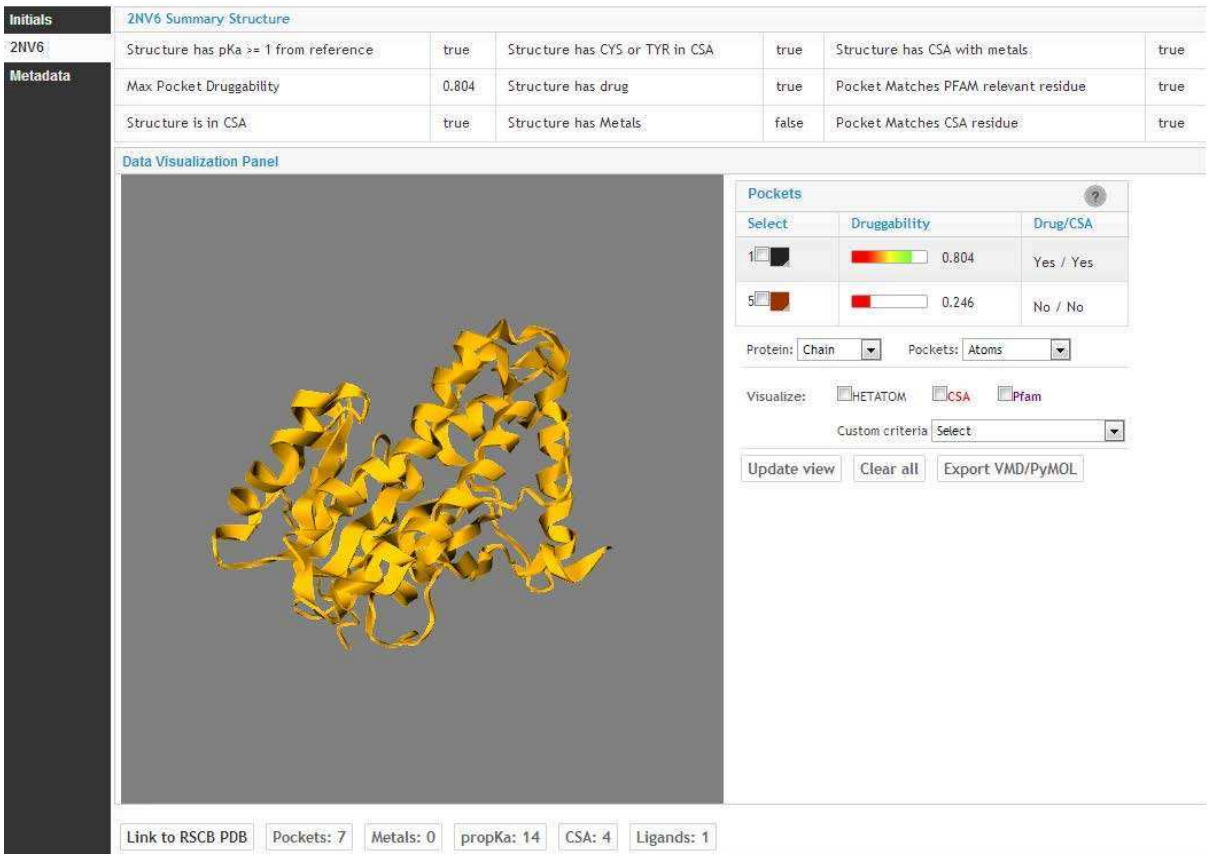
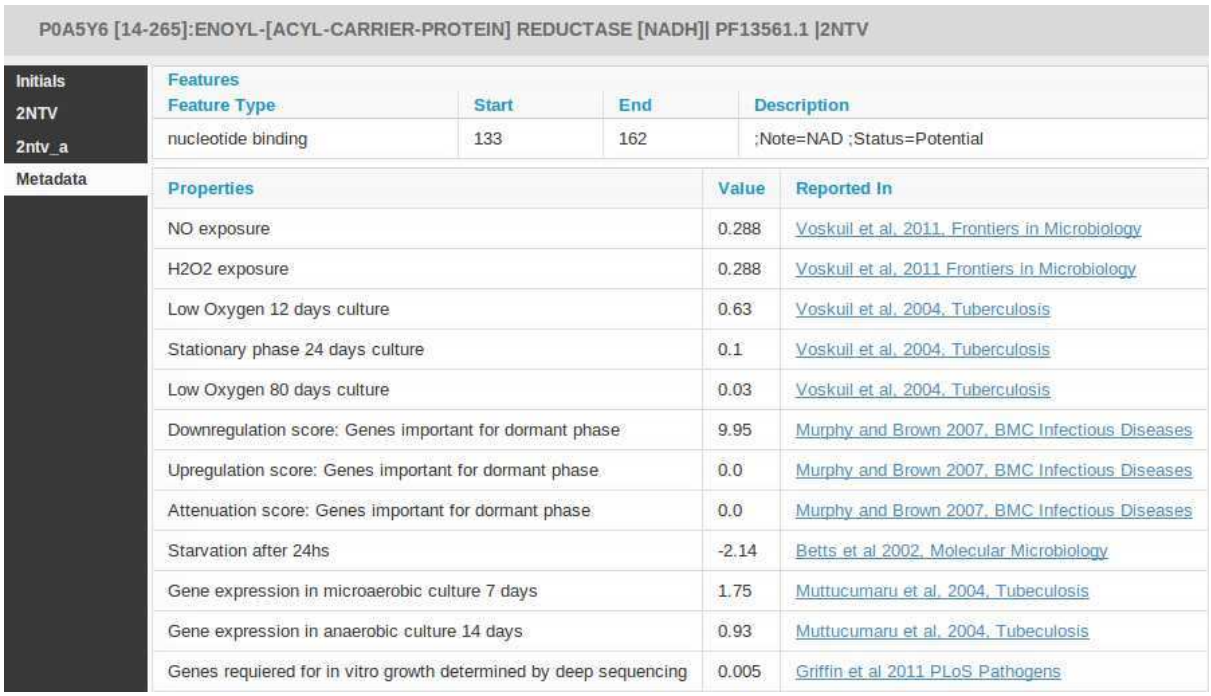


Figura 3.1.3.3: Solapa de visualización de estructura de proteínas en el sitio web de TuberQ. En el mismo se pueden visualizar los bolsillos con sus propiedades, compuestos pequeños unidos, etc. Los resultados están disponibles para ser descargados y visualizados de manera local.

La última solapa provee información adicional (o metadata, como puede verse en la figura 3.1.3.4) con información relacionada con etiquetas provistas por UniProt, como pueden ser regiones de unión, mutagénesis, etc. Además, en esta solapa es donde se reporta toda la información extraída de literatura relacionada con el nivel de expresión del gen bajo condiciones de stress y el criterio de esencialidad que ya hemos explicado.



P0A5Y6 [14-265]:ENOYL-[ACYL-CARRIER-PROTEIN] REDUCTASE [NADH]| PF13561.1 |2NTV

Initials
2NTV
2ntv_a
Metadata

Features

Feature Type	Start	End	Description
nucleotide binding	133	162	;Note=NAD ;Status=Potential

Properties	Value	Reported In
NO exposure	0.288	Voskuil et al, 2011, Frontiers in Microbiology
H2O2 exposure	0.288	Voskuil et al, 2011 Frontiers in Microbiology
Low Oxygen 12 days culture	0.63	Voskuil et al, 2004, Tuberculosis
Stationary phase 24 days culture	0.1	Voskuil et al, 2004, Tuberculosis
Low Oxygen 80 days culture	0.03	Voskuil et al, 2004, Tuberculosis
Downregulation score: Genes important for dormant phase	9.95	Murphy and Brown 2007, BMC Infectious Diseases
Upregulation score: Genes important for dormant phase	0.0	Murphy and Brown 2007, BMC Infectious Diseases
Attenuation score: Genes important for dormant phase	0.0	Murphy and Brown 2007, BMC Infectious Diseases
Starvation after 24hs	-2.14	Betts et al 2002, Molecular Microbiology
Gene expression in microaerobic culture 7 days	1.75	Muttucumaru et al, 2004, Tubeculosis
Gene expression in anaerobic culture 14 days	0.93	Muttucumaru et al, 2004, Tubeculosis
Genes requiered for in vitro growth determined by deep sequencing	0.005	Griffin et al 2011 PLoS Pathogens

Figura 3.1.3.4: La solapa de metadatos prove la anotación funcional provista por la literatura en relación a su esencialidad, su expresión bajo condiciones de estrés, falta de nutrientes, etc.

Un análisis estadístico de la base de datos generada para Mtb nos arroja el dato de que un 34% de las proteínas (ORFs) presentes en el genoma de referencia tienen una estructura definida. El 82% de ellas presentan un bolsillo con alta drogabilidad, dato que es alentador de cara al desarrollo de proyectos relacionado con descubrimiento de drogas, pero que también puede ser el resultado del desvío inherente hacia la cristalización de proteínas que se unan a compuestos en la base de datos del PDB.

En la figura 3.1.3.5 se compara la distribución de drogabilidad de bolsillos unidos a compuestos tipo droga en todo el PDB (mostrada previamente) y en Mtb, observándose que tienen una distribución comparable. También se observa que los bolsillos que se sabe que alojan a las drogas que actualmente sirven como tratamiento antibacteriano, son drogables o altamente drogables.

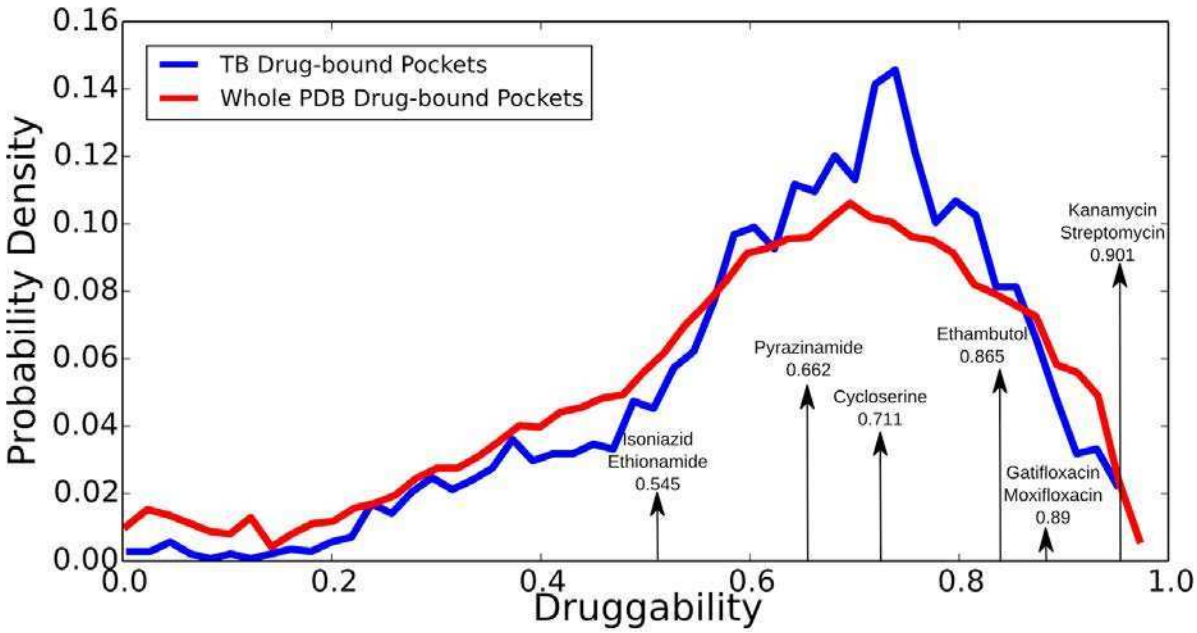


Figura 3.1.3.5: En azul, la distribución de pockets unidos a compuestos tipo drogas descrita en la figura 3.1.2.2, en rojo, los pockets unidos a droga dentro del conjunto de los cristales resueltos para proteínas de Mtb. Se observa que la distribución de drogabilidad sigue una misma tendencia. Se señala para todos los casos que se encuentran cristalizada la proteína sobre la que actúa una droga cuál es el pocket de máxima drogablidad para este blanco proteico: se observa que todos son drogables o altamente drogables.

Es importante aclarar que la presencia de un bolsillo altamente drogable es una condición necesaria pero no suficiente para que la proteína estudiada sea viable para proyectos de

desarrollo de drogas. Además de una inspección cuidadosa de la estructura de la proteína y del bolsillo y su presencia en redes metabólicas también hay que prestar atención con respecto a su esencialidad y la importancia de la misma en diferentes condiciones a las que puede ser expuesto el patógeno en distintos tratamientos.

En la tabla 3.1.3.1 se presentan estadísticas generales extraídas de la base de datos generada, indicando cardinalidades de conjuntos que cumplen criterios que hemos utilizado posteriormente para seleccionar blancos como candidatos para generar ligandos que modulen su actividad, sirviendo esto como tratamiento antibacteriano.

Propiedad	**Cantidad**
Cantidad de ORFs	3982
Cristales de proteínas	1319
Proteínas cristalizadas	441
Cristales drogables	660
Proteínas cristalizadas drogables	296
Proteínas modeladas	903
Proteínas modeladas drogables	690
Proteínas con estructura	1344
Cobertura del genoma	34%
Proteínas esenciales	379
Proteínas esenciales cristalizadas	113

Proteínas esenciales modeladas	137
Proteínas esenciales y drogables	184
Proteínas sobreexpresadas bajo stress ERON	713
Proteínas sobreexpresadas bajo stress ERON y drogables	475
Proteínas sobreexpresadas bajo stress ERON y esenciales	145
Proteínas sobreexpresadas bajo stress ERON, esenciales y drogables	111

Tabla 3.1.3.1: Estadísticas generales de la base de datos generada para Mtb.

Una de las estrategias para buscar blancos interesantes dentro del genoma de Mtb, como adelantamos en la introducción de este capítulo, fue la de identificar proteínas que ya son blancos de especies reactivas de oxígeno y nitrógeno (ERON) producidas por el sistema inmune e intentar inhibirlas también de forma farmacológica. Por lo tanto, además del análisis de expresión, utilizamos la información estructura-secuencia combinada con el conocimiento de la reactividad química para predecir la sensibilidad de las mismas frente a las ERON.

El principal blanco de estas especies son los centros metálicos de las proteínas, como son el grupo hemo y los residuos de cisteína y tirosina que pueden ser nitrados/oxidados. Usualmente, la modificación del estado de oxidación/coordinación de los centros metálicos de las metalo proteínas resulta en una pérdida parcial o total de su función, como ha sido descrito en las P450 (citocromos) de Mtb[98]. En el caso de cisteínas o tirosinas, es una asunción razonable que si estos residuos se encuentran presentes en el sitio activo (o bolsillo más drogable), su modificación química puede derivar en una actividad disminuida. Este es el caso de las cistein-proteasas que se transforman en inactivas al oxidarse la

cisteína del sitio activo[99] o en la MnSOD donde la nitración de la tirosina bloquea el sitio de unión del sustrato[100].

Con esto en mente, asignamos como potencialmente sensibles a estrés de ERON a todas las proteínas que tienen un centro metálico (cobre, hierro y zinc) adyacente al bolsillo del sitio activo, o un residuo de cisteína, tirosina en el sitio activo o bolsillo más drogable.

	Cristal (Modelo)	**Esencial (E)**	**Altamente Drogable(HD) y E**	**Sobreexpresada, HD y E**
Metal	149	86	57	41
Cisteína	130 (164)	64 (49)	37 (37)	30 (28)
Tirosina	269 (274)	135 (84)	82 (69)	58 (42)

Tabla 3.1.3.2: Proteínas de Mtb predichas como sensibles a ERON desglosadas por sitio activo metálico o con presencia de cisteína o tirosina, sumarizadas en función de distintas propiedades de interés. Las proteínas tienen una etiqueta que las marca como sobreexpresadas en condiciones de stress basadas en función del experimento cargado como metadata en nuestra base[CITA].

La información presentada en la tabla 3.1.3.2 muestra que hay cerca de 800 proteínas que son potencialmente sensibles a ERON. Cerca de 200 proteínas cumplen además con ser altamente drogables, esenciales y sobreexpresarse.

Para mostrar el potencial de TuberQ, ahora analizaremos en detalle la proteína *Inositol-3-phosphate syntase* (Uniprot P71703, gen Ino1) como ejemplo ilustrativo de los resultados que es posible de obtener con este tipo de análisis integrados.

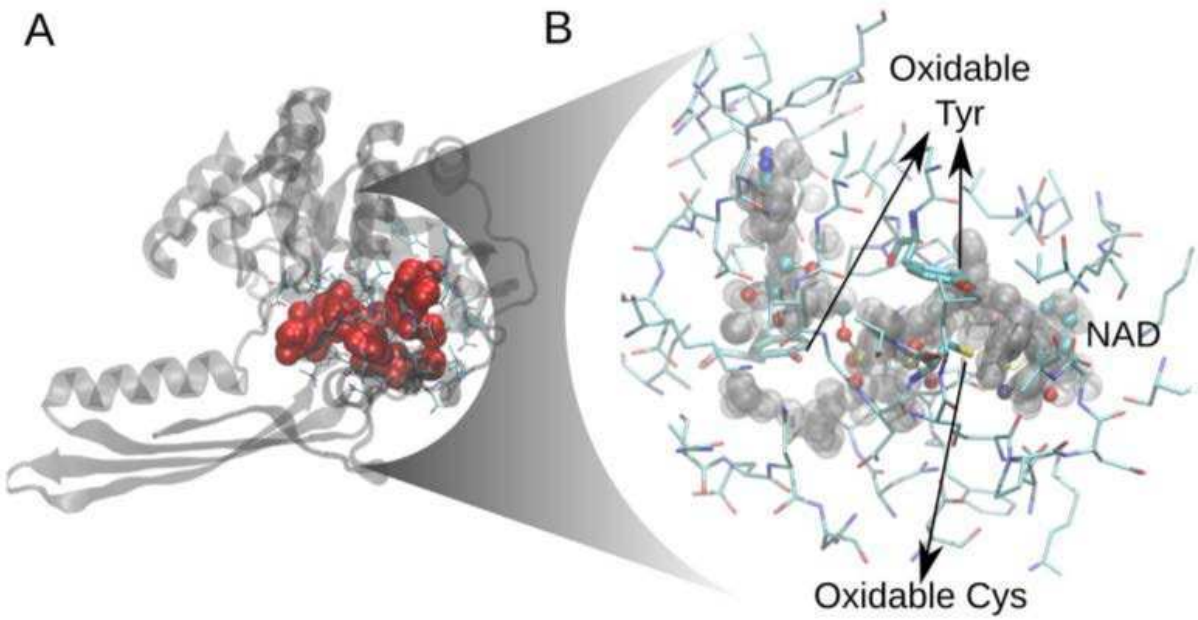


Figura 3.1.3.6: Análisis estructural de la proteína Ino1 en Mtb, tomado de Defelipe et.al., 2016[101]. A) Vista de la estructura resuelta (PDB 1GR0) con su bolsillo más drogable marcado con esferas rojas. B) Vista del bolsillo más drogable superpuesto con el sitio de unión a NAD. Los residuos de cisteína y tirosina oxidables se encuentran resaltados (más gruesos).

La proteína Ino1 es un miembro de la vía de síntesis del micotiol y ha sido descrita como esencial tanto por experimentos masivos como de mutación de la misma[102]. Forma parte del regulón DosR y está sobre expresada en condiciones de falta de nutrientes. Como puede observarse en la figura 3.1.3.6 su estructura presenta un bolsillo drogable (DS=0.719) que se solapa con el sitio de unión a NAD, un sitio conocido por poder albergar compuestos tipo droga en otras proteínas[103]. Como se destaca en la citada figura, Ino1 posee dos residuos sensibles a estrés como también un átomo de zinc estructural/catalítico (su rol no es bien comprendido). Este análisis combinado convierte a esta proteína en un blanco ideal para el desarrollo de compuestos tipo droga que inhiban su actividad.

3.1.3 Discusión

En este capítulo hemos combinado información sobre relevancia de sensibilidad, esencialidad, *off-targeting* con drogabilidad estructural para ayudar en la determinación de buenos candidatos a blancos proteicos con aplicaciones a organismos bacterianos y hemos hecho foco en Mtb presentando estadísticas y casos particulares.

El método desarrollado también ha sido aplicado en otros organismos como son *Corynebacterium Pseudotuberculosis*[104] y *Corynebacterium Diphteriae*, la replicabilidad del método asegura su extensión con un bajo esfuerzo. Por otro lado, el método ha servido para seleccionar blancos que actualmente están siendo ensayados como candidatos para el desarrollo de nuevos tratamientos en Mtb.

El desarrollo realizado ha sido expuesto a la comunidad en un sitio web que permite realizar consultas sobre la base de datos presentando la información de una manera amigable que permite a los usuarios acceder a la información necesaria para la determinación de la relevancia con respecto a las probabilidades de unión a compuestos tipo droga y su importancia como blanco potencial de cara al desarrollo de nuevos tratamientos.

Se ha presentado en detalle un ejemplo en donde se evalúa un blanco con características ideales para el desarrollo de inhibidores con vistas a generar nuevas drogas antibacterianas que evidencian el poder de análisis que otorga el desarrollo realizado.

3.2. Selección de ligandos para el mejoramiento de conjuntos de Virtual Screening(LigQ)

3.2.1 Introducción

El control de la actividad proteica a través de pequeñas moléculas orgánicas -tipo droga[105]- es no solo uno de los objetivos fundamentales de la farmacología, sino también una herramienta valiosa para el estudio de la función proteica en su contexto biológico.

El screening virtual(VS) *in-silico*[106] es una de las herramientas más usadas para la búsqueda de estas moléculas. Llevar a cabo un VS requiere, por lo general, un conjunto muy grande (del orden de los millones) de moléculas disponibles de manera comercial, las cuales son dockeadas (sometidas a un algoritmo de docking) en el blanco (la proteína de interés), conservando para análisis experimentales o computacionales ulteriores aquellos que éste algoritmo evalúe como más favorables a unirse desde un punto de vista energético.

La estructura proteica y el sitio de unión tiene que ser definidos de manera apropiada para que el análisis de VS pueda ser considerado correcto. La cantidad de compuestos que van a ser testeados en el experimento de VS definirán el costo computacional del proceso, y ya que solo un número relativamente bajo de compuestos (del orden de los cientos) pueden ser testeados experimentalmente, mejorar los sets para hacer VS significa encontrar la mayor cantidad posible de moléculas que se acoplen en la práctica con el target, reduciendo en el mayor grado posible el conjunto sobre el cual hacer VS (enriquecer el conjunto).

Los compuestos son obtenidos usualmente de base de datos en algún tipo de formato 2D (usualmente SMILES o InChI) y distintas técnicas computacionales son requeridas para convertirlos en una (o varias) estructuras 3D adecuadas, particularmente si todos los enantiómeros y tautómeros relevantes a un pH dado van a ser considerados para el

análisis. La base de datos que sirve como fuente para este desarrollo ha sido explicada en los métodos computacionales, y la base de datos para validación es DUD, también introducida en los métodos.

En este capítulo, presentamos el desarrollo de la herramienta LigQ (disponible en http://ligq.qb.fcen.uba.ar/), la cual mediante un pipeline quimio-informático intenta precisamente enriquecer en la mayor medida posible los conjuntos de compuestos para hacer VS dada como input una proteína de interés. Además, LigQ también es capaz, si no se conoce, de determinar el mejor sitio de unión de una proteína y de generar los modelos tridimensionales de las moléculas candidatas en el formato adecuado.

3.2.2 Materiales y Métodos

Con el fin de hallar soluciones al problema planteado, se ha diseñado un pipeline de procesamiento (figura 3.2.2.1) que permite postular para una proteína de entrada, un conjunto de estructuras de moléculas pequeñas tipo droga que tienen buenas probabilidades de unirse al sitio activo de la misma.

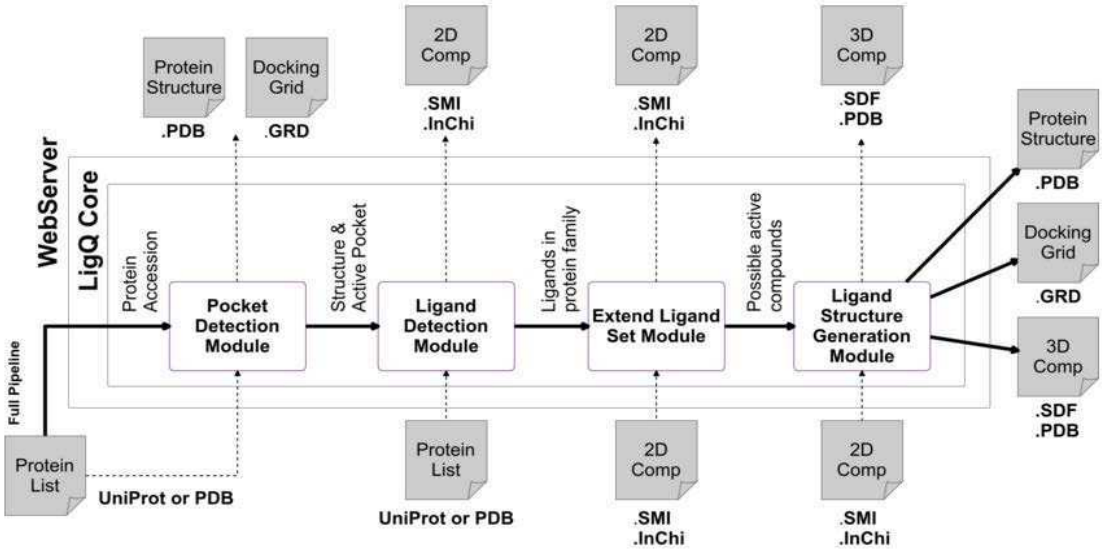


Figura 3.2.2.1: Pipeline de procesamiento propuesto para la herramienta LigQ. En el mismo, como única entrada se tiene un código de UniProt o PDB para una proteína. El producto final del procesamiento es una lista de estructuras de moléculas pequeñas y la estructura de la proteína con su correspondiente grilla delimitando el sitio activo para realizar experimentos de docking.

La herramienta está compuesta por cuatro módulos que pueden ser ejecutados de manera secuencial e independiente, cada uno con sus entradas y salidas específicas, como se muestra en la figura 3.2.2.1, y un servidor web que sirve de interfaz de usuario amigable para la ejecución de estas herramientas (las cuales también pueden ser ejecutadas por línea de comandos como cualquier otro programa).

3.2.2.1 Módulo de detección de bolsillos

El módulo de detección de bolsillos (PDM, por su sigla en inglés: Pocket Detection Module) toma como entrada una lista de proteínas ya sea mediante su código UniProt o su código PDB. Si la entrada es un código UniProt se selecciona la de mayor cubrimiento sobre la secuencia de la proteína objetivo, seleccionando aquella de mayor resolución en el caso de que fuera necesario un desempate. Si no existen estructuras disponibles entonces

se intenta generar un modelo por homología, tal como fuera explicado en capítulos anteriores.

De la estructura finalmente obtenida (la cual será uno de los resultados finales del pipeline del procesamiento) se buscarán todos los bolsillos y se generará, para aquél de mayor drogabilidad (DS), una grilla para realizar experimentos de docking (explicado más abajo).

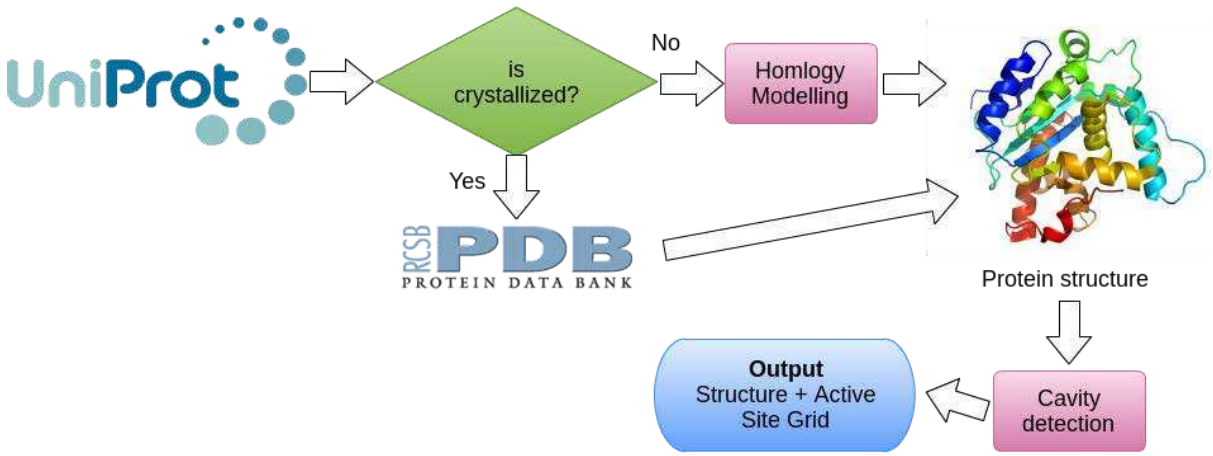


Figura 3.2.2.2: Esquema del pipeline de generacion estructuras y búsqueda de sitio activo PDM que constituye el primer paso de procesamiento de LigQ.

Existen dos maneras de generar grillas provistas por el algoritmo rDock[68], una similar al método de alpha-spheres ya explicado, y otro que es el que nosotros utilizamos, en donde se pone una esfera de un radio grande (por lo general 15A) en cada uno de los átomos de un "ligando referencia". El algoritmo de rDock realiza docking molecular intentando "acomodar" moléculas de entrada en una grilla inserta en la proteína, evaluando cada configuración posible en base a un potencial clásico. En nuestro caso, no tenemos ligando referencia, pero sí las alpha-spheres ubicadas dentro de la cantidad de interés, las cuales nos sirven como constitutivas de este "ligando dummy" con el cual generar la grilla. El espacio obtenido por estas esferas grandes es luego llenado por esferas pequeñas para

detectar aquellas porciones que delimitan la macromolécula (para evitar clashes) dando lugar a la grilla final.

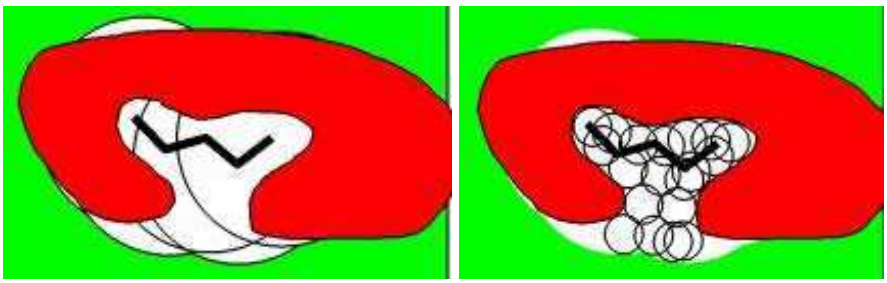


Figura 3.2.2.3: Pasos para generar grillas de dockeo del algoritmo rDock mediante el método de "reference ligand" el cual toma como zona para dockear compuestos aquella donde esté ubicado, gracias a nuestro pipeline, el pocket de mayor drogabilidad.

Las grillas para realizar experimentos de docking son las porciones del espacio en donde estos algoritmos intentarán acoplar ligandos en la proteína y evaluar energéticamente la afinidad de esa configuración tridimensional (hacerlo sobre toda la superficie de la proteína es prohibitivo computacionalmente, ya que por lo general se evalúan grandes librerías de los mismos) con la porción de la estructura de la proteína en donde se está intentando acoplarla.

3.2.2.2 Módulo de detección de Ligandos

En el segundo módulo, el módulo de detección de ligandos (LDM, por su sigla en inglés: Ligand Detection Module) tiene como objetivo, para la proteína sobre la que se está ejecutando el pipeline, realizar dos búsquedas. La primera recaba todos los ligandos que estén co-cristalizados con cualquier proteína de la misma familia PFam a la que pertenezca la proteína objetivo. La segunda, recaba todos los ligandos que tengan algún ensayo marcado como activo en la base de datos ChEMBL en cualquier proteína de la familia PFam

de la proteína objetivo. A este conjunto de compuestos lo denominamos "conjunto semilla" (seed set).

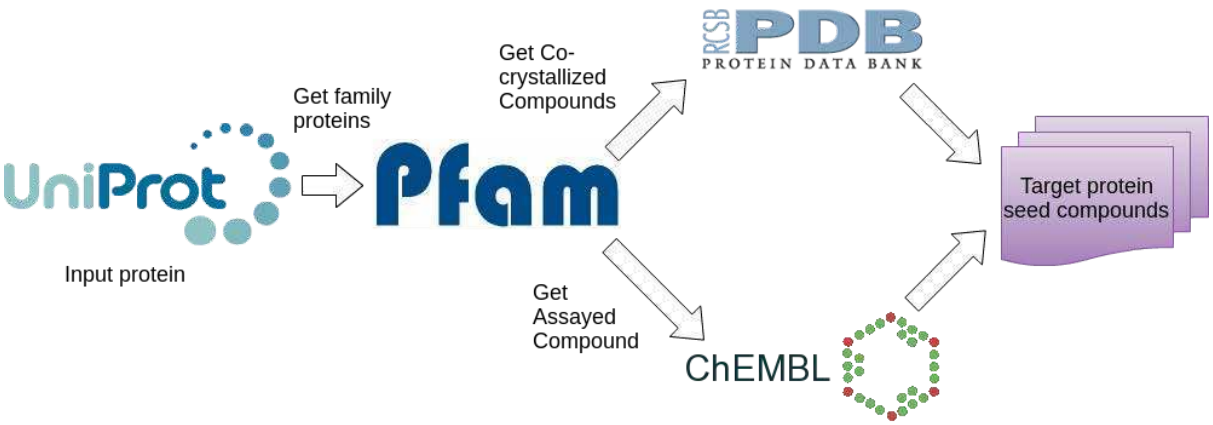


Figura 3.2.2.4: Esquema del funcionamiento del módulo de detección de ligandos de la herramienta LigQ.

3.2.2.3 Módulo de extensión de Ligandos

El tercer módulo, de extensión del conjunto de ligandos (LEM, por su sigla en inglés: Ligand Extension Module), toma cada uno de los elementos del conjunto semilla, de los cuales consideramos que tienen buenas probabilidades de unión al blanco objetivo, y para cada uno de ellos busca en la base de datos de compuestos que hemos descrito en la sección de métodos de esta tesis, todos aquellos que se encuentren a una distancia *d* de similaridad química determinada, o una lista de los *x* compuestos más parecidos en la base (siendo *d* o *x* definidos por el usuario en el momento de ejecutar este módulo).

Este módulo ofrece además la posibilidad de definir filtros sobre las propiedades que deben cumplir aquellos compuestos que lo conformen. Las propiedades son todas aquellas que hemos venido enumerando y pueden definirse valores máximos y mínimos. De esta forma, por poner un ejemplo que resulta obvio, no seleccionaremos compuestos que tengan un volumen mayor que la cavidad sobre la que intentaremos posteriormente dockearlo.

Este nuevo conjunto contiene compuestos que a diferencia del conjunto semilla, representan potenciales nuevos ligandos ya que no se tienen evidencias de acoplamiento con la proteína de interés ni con ninguna de la misma familia, pero que a la vez poseen características que los convierten en buenos candidatos: son similares a otros que sí cumplen con esto, y sus propiedades están definidas de manera tal de favorecer las chances de un buen acoplamiento.

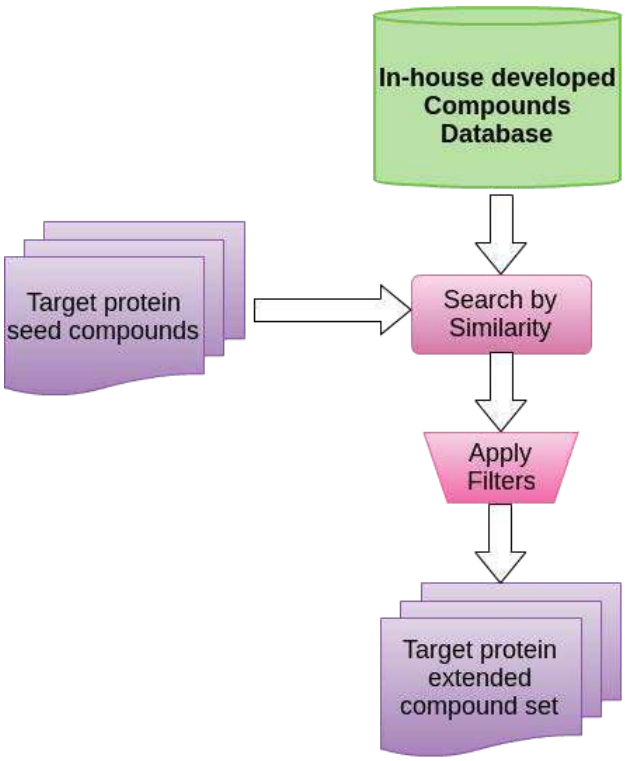


Figura 3.2.2.5: Esquema de funcionamiento del módulo de extensión de ligandos de la herramienta LigQ.

3.2.2.4 Módulo de generación de estructuras

El último módulo de nuestra herramienta LigQ es el de generación de estructuras (SGM, por su sigla en inglés: Structure Generation Module) de compuestos. En la misma, nos hemos valido de los algoritmos de cálculo provistos por la librería JChem[69] previamente

mencionados en los métodos de esta tesis para, de cada uno de los compuestos que componen el conjunto extendido de ligandos (o el subconjunto del mismo que seleccione el usuario), los cuales están almacenados en formatos de dos dimensiones (InChI) en la base de datos que hemos construído, obtener las estructuras en tres dimensiones más probables, considerando tautómeros, isómeros, estados de protonación etc.

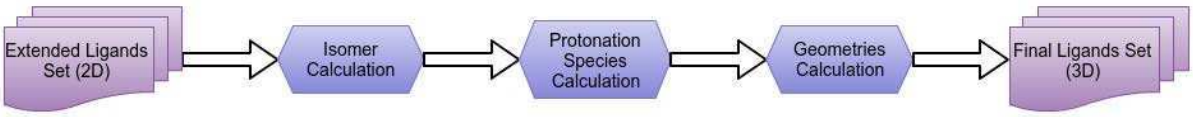


Figura 3.2.2.6: Etapas del procesamiento del módulo de generación de estructuras de moléculas pequeñas.

3.2.2.5 Base de datos de compuestos en LigQ

La base de datos en donde LigQ busca y extiende los conjuntos de compuestos está formada por todas aquellas moléculas pertenecientes a ZINC con la etiqueta "purchasable" (adquirible), clusterizando los mismos con un índice de similaridad de 0.95 y tomando de estos clusters representantes al azar (para eliminar la redundancia), más todos aquellos compuestos de la base de datos ChEMBL que han sido etiquetados como "activos" en cualquier bioensayo donde el target del mismo sea una proteína, más todos aquellos compuestos que hayan sido co-cristalizados con una proteína en el PDB. De todos estos compuestos se espera que sean plausibles de ser adquiridos y en los módulos en los que se devuelve una lista de compuestos se ha agregado un link a la base de datos ZINC para tener información acerca de cómo pueden adquirirse ese compuesto y compuestos similares (que complan la condicion de TI > 0.95 y ser "purchasables"). El tamaño de la base de datos es cercano a 1.5 millones de compuestos.

3.2.2.6 Docking molecular

Uno de los usos directos que pueden realizarse con la salida de la ejecución de nuestra herramienta, es la aplicación de algoritmos de docking en la cavidad seleccionada de la proteína objetivo sobre los compuestos encontrados. Los algoritmos de docking molecular juegan un rol importante en la optimización de compuestos, ya que mediante los mismos puede explorarse no solo la preferencia desde el punto energético de una cavidad de acoplarse con una molécula sobre otra, sino también la exploración de cuáles son las conformaciones o poses que la molécula puede adoptar dentro de la cavidad.

El algoritmo de docking molecular utilizado ha sido rDock[68] el cual tiene una función de puntaje que permite evaluar para cada molécula de entrada cuál es la afinidad que tendrá al momento de acoplarse en la cavidad objetivo. La función de puntaje está determinada por los siguientes términos:

$$S_{total} = W_{intermolecular} \cdot S_{intermolecular} + W_{intramolecular\ ligando} \cdot S_{intramolecular\ ligando}$$

$$+ W_{intramolecular\ sitio\ activo} \cdot S_{intramolecular\ sitio\ activo} + W_{restricciones} \cdot S_{restricciones}$$

El término intermolecular se divide a su vez en contribuciones de Van Der Waals, polares, de repulsión, aromáticas, de solvente y de rotación. El término intramolecular del ligando y de sitio activo tienen contribuciones de Van Der Waals, polares, de repulsión y diédricos, evaluados cada uno en función de cuáles son los átomos que aportan para ese score. El término de restricciones tiene contribuciones de cavidad, de "atadura", y de farmacóforos.

Los valores de los términos W (que a su vez se replican en cada sub término mencionado en el párrafo anterior, son obtenidos a partir de una regresión de cuadrados mínimos realizada para un conjunto de entrenamiento con el que se entrenó el algoritmo, de manera de reproducir de la la forma más fidedigna posible datos de sitios unidos ya

conocidos. Los términos de score S tienen diferentes formas funcionales, todas tomadas del potencial clásico de la herramienta RiboDock[107]. Una explicación en detalle de cada uno de los términos puede ser leída en el manual online de rDock.

Con respecto a la generación de diferentes poses de cada una de las moléculas que se están dockeando, con el fin de obtener aquellas más favorables, rDock utiliza una combinación de métodos estocásticos y determinísticos que permiten muestrear configuraciones tridimensionales de moléculas de baja energía. Básicamente, mediante técnicas de Montecarlo se generan conformaciones variando diferentes posiciones de ángulos de enlace y luego minimizando las configuraciones mediante un método de minimización Simplex[108].

3.2.3 Resultados

Sobre la herramienta desarrollada se realizaron dos validaciones independientes: la primera tendiente a medir cual es el enriquecimiento de los conjuntos para Virtual Screening generados, y, la segunda, la ejecución de algoritmos de docking molecular del conjunto de compuestos generado sobre el blanco proteicos sobre el que se ejecutó el pipeline. Para ambas validaciones el conjunto de blancos proteicos que hemos utilizado son los de la base de datos DUD[109], descrita en la sección de métodos de esta tesis. En particular, eliminamos del conjunto de blancos proteicos aquellos que pertenecieran a una misma familia, dejando únicamente un representante, para eliminar redundancia en los resultados, y finalmente permanecieron para el análisis los nueve compuestos que pueden observarse en la tabla 3.2.3.1 y otras de este capítulo.

El primero de los módulos, de detección de cavidades, no ha sido formalmente validado para esta herramienta ya que consideramos que un protocolo similar ya ha sido implementado exitosamente en la herramienta ProteinQ descrito en el capítulo anterior para el genoma de Mtb.

Para validar el segundo módulo, hemos utilizado nuestra herramienta para generar el conjunto semilla de ligandos de varios de los blancos proteicos de DUD. En los mismos, hemos podido observar que el aporte que realiza el hecho de buscar compuestos en la familia PFam (ver tabla 3.2.3.1) de la proteína objetivo es significativo, aportando numerosos compuestos al conjunto, los únicos en algunos casos, lo cual, cuando no se poseen ensayos o cristales, es de vital importancia ya que de otra forma no habría conjunto para extender.

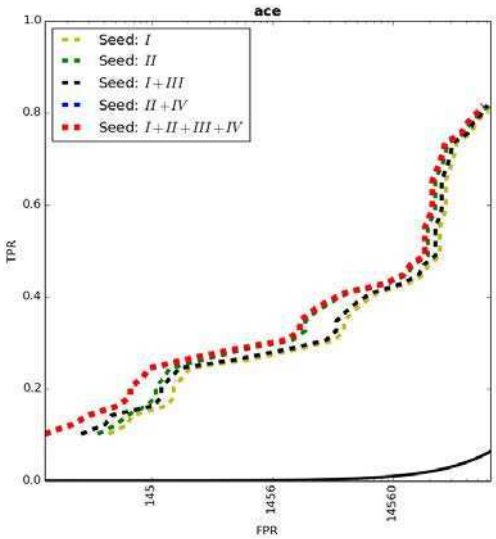
Proteína	Ligandos derivados de PDB	Ligandos derivados de PDB en PFam	Ligandos derivados bioensayos	Ligandos derivados de bioensayos en PFam
	Seed I	**Seed II**	**Seed III**	**Seed IV**
ace: Angiotensin-conv enzyme	16	23	0	21
ada: Adenosine deaminase	0	32	0	6
ampc: AmpC beta lactamase	52	80	156	157
Dhfr: Dihydrofolate reductase	5	120	1	13
gart: glycinamide ribonucleotide transformylase	6	34	0	4
Gbp: Glycogen phosphorylase beta	128	144	0	0
na: Neuraminidase	4	37	0	8
pnp: Purine nucleoside phosphorylase	19	89	0	6
tk: Thymidine kinase	27	29	0	1

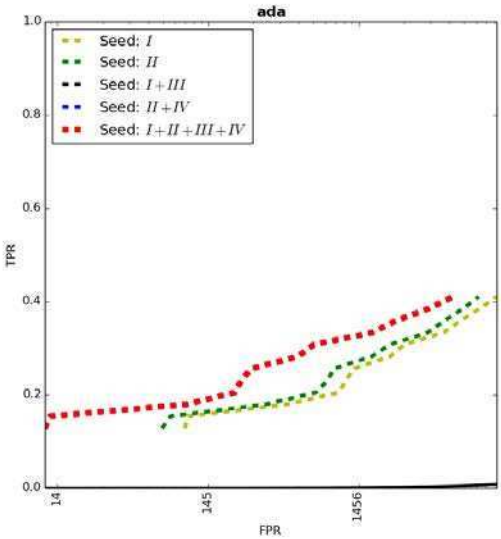
Tabla 3.2.3.1: Número de compuestos semilla derivados de cada uno de las bases de datos fuente para cada proteína analizada del DUD. Proteínas de la misma familia que formaban parte de la base de datos fueron agrupadas tomando un único representante. En los conjuntos que agregan PFam se tiene en cuenta los compuestos considerados unidos (co cristalizados o ensayados) sobre cualquier proteína de la familia de la de interés.

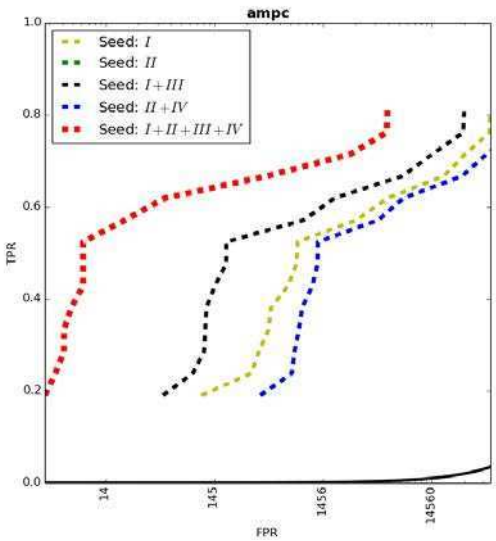
A continuación, procedimos a validar el Módulo de Extensión de Ligandos. Para ello, definimos diferentes conjuntos semilla de acuerdo a cada uno de los posibles orígenes definidos en la tabla 3.2.3.1. Luego, sobre la base de datos de ligandos generada para esta herramienta, explicada en los métodos de este capítulo, a la cual le añadimos explícitamente los ligandos conocidos para las proteínas objetivos que aparecen en la base de datos DUD, ejecutamos el módulo de extensión de ligandos. Estos ligandos de DUD constituyen entonces nuestros "verdaderos positivos".

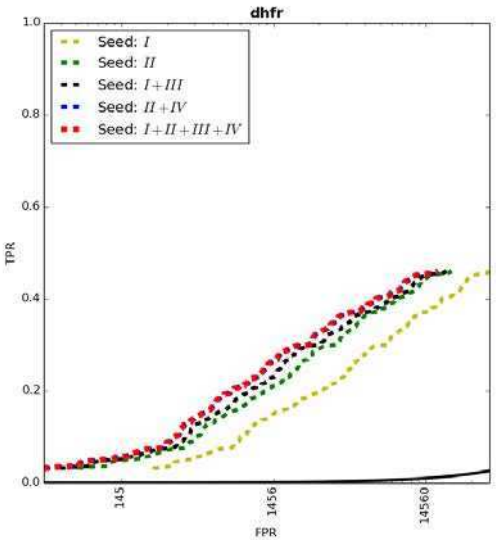
Para evaluar cada proteína y conjunto semilla, generamos curvas ROC semilogarítmicas en las cuales comparamos la porción de verdaderos ligandos para diferentes valores de corte de coeficiente de similaridad de tanimoto. Los resultados se muestran en la figura 3.2.3.1 y en la tabla 3.2.3.2.

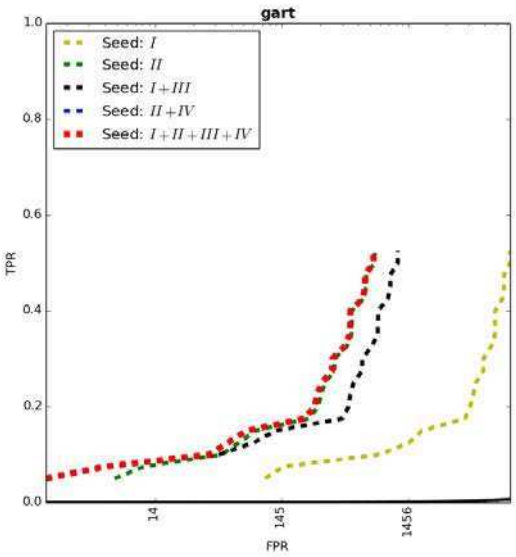
El TPR o True Positive Rate del gráfico es, de todos los ligandos presentes en DUD, cuántos fueron recuperados de la base. El FPR o False Positive Rate, es la cantidad de compuestos recuperados de la base de datos que no son ligandos conocidos (aunque sí potenciales ligandos). Por motivos de claridad, en lugar de expresarlo como proporción, en el gráfico se pone en el eje "x" la cantidad de compuestos, lo que resulta de mayor utilidad a la hora de comprender cuál será el costo posterior de experimentos de VS.

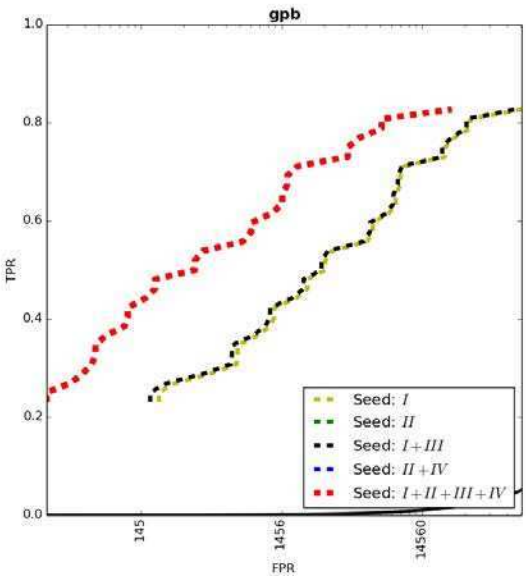

ace
Seed: I
Seed: II
Seed: $I+III$
Seed: $II+IV$
Seed: $I+II+III+IV$
TPR
FPR
1.0
0.8
0.6
0.4
0.2
0.0
145
1456
14560

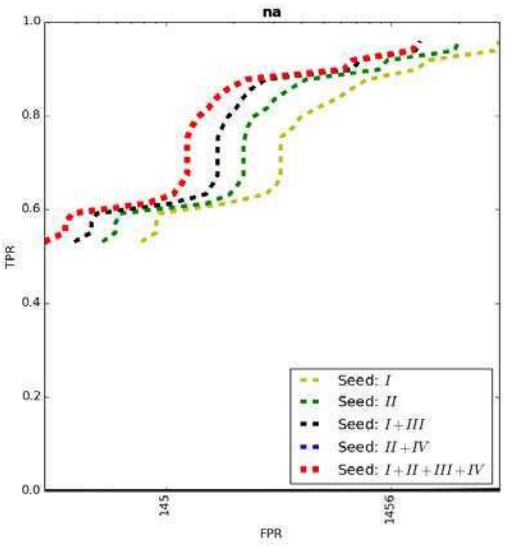

ada
Seed: I
Seed: II
Seed: $I+III$
Seed: $II+IV$
Seed: $I+II+III+IV$
TPR
FPR
1.0
0.8
0.6
0.4
0.2
0.0
14
145
1456

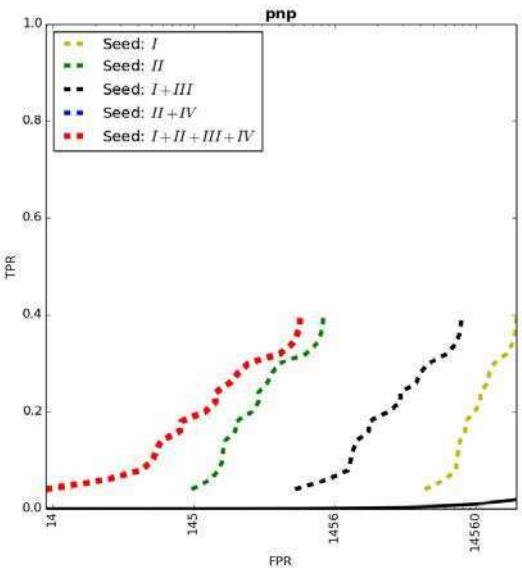

ampc
Seed: I
Seed: II
Seed: $I+III$
Seed: $II+IV$
Seed: $I+II+III+IV$
TPR
FPR
1.0
0.8
0.6
0.4
0.2
0.0
14
145
1456
14560


dhfr
Seed: I
Seed: II
Seed: $I+III$
Seed: $II+IV$
Seed: $I+II+III+IV$
TPR
FPR
1.0
0.8
0.6
0.4
0.2
0.0
145
1456
14560


gart
Seed: I
Seed: II
Seed: $I+III$
Seed: $II+IV$
Seed: $I+II+III+IV$
TPR
FPR
0.0
0.2
0.4
0.6
0.8
1.0
14
145
1456


gpb
Seed: I
Seed: II
Seed: $I+III$
Seed: $II+IV$
Seed: $I+II+III+IV$
TPR
FPR
0.0
0.2
0.4
0.6
0.8
1.0
145
1456
14560


na
Seed: I
Seed: II
Seed: $I+III$
Seed: $II+IV$
Seed: $I+II+III+IV$
TPR
FPR
0.0
0.2
0.4
0.6
0.8
1.0
145
1456


pnp
Seed: I
Seed: II
Seed: $I+III$
Seed: $II+IV$
Seed: $I+II+III+IV$
TPR
FPR
0.0
0.2
0.4
0.6
0.8
1.0
14
145
1456
14560

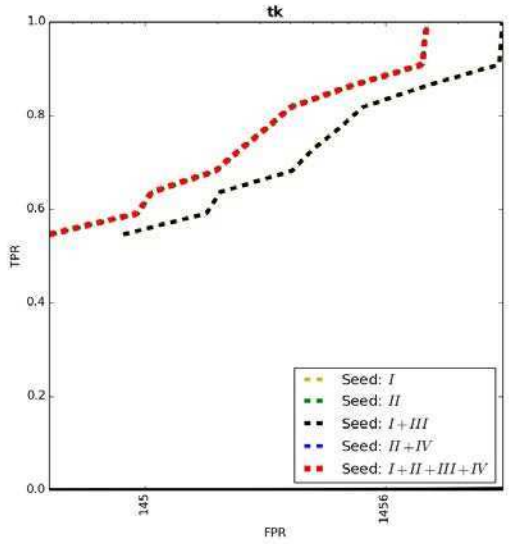


Figura 3.2.3.1: Curvas ROC semilogarítmicas para cada uno de los nueve blancos analizados de DUD. Se varía el índice de tanimoto para comparar cuántos ligandos en total se recuperan de la base de datos versus proporción de compuestos analizados (expresado cantidad de ligandos por claridad). Índices de similitud menores a 0.5 no han sido analizados, siendo importante el enriquecimiento cuando los valores de tanimoto son altos. Cada conjunto "seed" corresponde con el de la tabla 3.2.3.1.

Para cada caso, podemos analizar cuál es el factor de enriquecimiento aportado por la herramienta (EF) y para poder comparar distintos cortes o blancos proteicos entre sí, cuál es el enriquecimiento relativo a la cantidad de ligandos verdaderos que posee (Relative EF). El REF, indica cuánto más probable es tomar al azar un verdadero ligando en un conjunto enriquecido versus cuán probable es tomarlo en la base de datos original.

$$REF = P(ligando|conjunto\ enriquecido) / P(ligando|base\ de\ datos)$$

$$REF = \frac{ligandos\ obtenidos}{compuestos\ obtenidos} / \frac{ligandos\ totales}{compuestos\ totales}$$

Mientras que el EF indica el porcentaje de compuestos obtenidos que son verdaderos ligandos:

$$EF(\%) = \frac{ligandos\ obtenidos}{compuestos\ obtenidos} * 100$$

En la tabla 3.2.3.2 hemos analizado el factor de enriquecimiento obtenido para diferentes cantidades de compuestos totales recuperados de la base de datos en función de cuántos de ellos son verdaderos ligandos.

		Seed I			**Seed III**			**All seed Sets**		
	# true ligands	# of total Retrieved ligands: EF(%)-**REF**			# of total Retrieved ligands: EF(%) - **REF**			# of total Retrieved ligands: EF(%) - **REF**		
Proteína		100	1K	10K	100	1K	10K	100	1K	10K
ace: Angiotensin-converting enzyme	**49**	7 **2080**	1.3 **386**	0.2 **59**	8 **2377**	1.5 **445**	0.21 **62**	11 **3268**	1.5 **445**	0.21 **62**
ada: Adenosine deaminase	**39**	5 **1866**	0.8 **298**	0.16 **59**	7 **2613**	1.1 **410**	0.16 **59**	7 **2613**	1.3 **485**	0.16 **59**
ampc: AmpC beta lactamase	**21**	4 **2773**	1.2 **832**	0.14 **97**	4 **2773**	0.9 **624**	0.14 **97**	14 **9706**	1.5 **1040**	0.17 **117**
Dhfr: Dihydrofolate reductase	**410**	12 **426**	4.9 **174**	1.3 **48**	15 **532**	7.5 **266**	1.6 **58**	21 **745**	9 **319**	1.7 **61**
gart: glycinamide ribonucleotide transformylase	**40**	2 **728**	0.5 **182**	0.21 **76**	7 **2548**	2.1 **764**	0.21 **76**	7 **2548**	2.1 **764**	0.21 **76**
Gbp: Glycogen phosphorylase beta	**52**	12 **3360**	2 **560**	0.35 **98**	20 **5600**	3.2 **896**	0.43 **120**	20 **5600**	3.2 **896**	0.43 **120**

na: Neuraminidase	**49**	26 **7725**	4.3 **1277**	0.47 **139**	30 **8914**	4.4 **1307**	0.47 **139**	30 **8914**	4.6 **1366**	0.47 **139**
pnp: Purine nucleoside phosphorylase	**50**	2 **582**	0.2 **58**	0.04 **11**	6 **1582**	1.8 **495**	0.2 **58**	8 **2329**	2 **582**	0.2 **58**
tk: Thymidine kinase	**22**	12 **7941**	1.8 **1191**	0.22 **145**	13 **8603**	1.9 **1257**	0.22 **145**	13 **8603**	1.9 **1257**	0.22 **145**

Tabla 3.2.3.2: Factor de enriquecimiento para cada uno de los ligandos de la base de datos DUD analizados, medida para distintos conjuntos semilla y para distintas cantidades de compuestos totales recuperados de la base de datos total de ligandos.

Como puede observarse, en todos los registros de la tabla se logra un enriquecimiento real con los conjuntos de compuestos extendidos obtenidos. Por supuesto que ser más laxo en la extensión del conjunto semilla, redundará en un menor enriquecimiento pero permitirá ampliar el universo de exploración de nuevos ligandos.

Analizaremos ahora dos casos en detalle a modo de ejemplo. El caso de AMPc beta lactamasa en la figura 3.2.3.1, vemos que representa un caso ideal. Para el valor de similaridad más bajo que ha sido evaluado (0.5) se han recuperado de la base de datos 17 de los 21 ligandos considerados verdaderos, usando como conjunto semilla la unión todos los subconjuntos semilla generados (línea roja), trayendo de la base de datos menos de diez mil compuestos en total, lo cual es una proporción muy pequeña de la base de datos de ~1.5 millones de compuestos en caso de que esta debiera ser analizada. Incluso, cuando recuperamos 10 ligandos verdaderos, únicamente 100 compuestos son devueltos por la base, que es un número pequeño y fácilmente manejable. Por supuesto, valores altos de corte de similaridad atentan contra la novedad que puedan presentar los compuestos

extraídos de la base de datos. Los valores deben ser puestos en función de cuál es el objetivo con el que se ejecuta el módulo.

Un caso no tan favorable es el del blanco proteico dhfr (Dihydrofolate reductase), el cual requiere de un valor de corte de similaridad más bajo para obtener una porción considerable de ligandos verdaderos, lo que redunda en una mayor cantidad de ligandos totales devueltos para analizar, ya sea con experimentos de docking o con otro tipo de experimentos. De todas maneras, se observa que para recuperar la mitad de los ligandos, solo necesitamos el 1% del tamaño total de la base de datos de ligandos, lo cual habla de que, incluso en casos como estos, el uso de nuestra herramienta puede ser útil.

Para analizar la composición de los conjuntos extendidos (figura 3.2.3.2) que hemos obtenido, en relación con la diversidad del conjunto de partida, se han calculado clusters en base a la matriz de distancias de Tanimoto calculadas para todos los compuestos semilla (azul) sumados a todos los ligandos verdaderos (true positives) aportados por la base de datos DUD (rojo). Como puede observarse, en el caso de AmpC todos los ligandos se encuentran en el mismo cluster que alguno de los ligandos del conjunto semilla. En cambio en dhfr, sucede que hay algunos ligandos que se encuentran en clusters lejanos en distancia a los compuestos del conjunto semilla, lo que implica que habría que generar un conjunto extendido demasiado grande (un coeficiente de Tanimoto muy permisivo) para poder obtener todos los verdaderos ligandos.

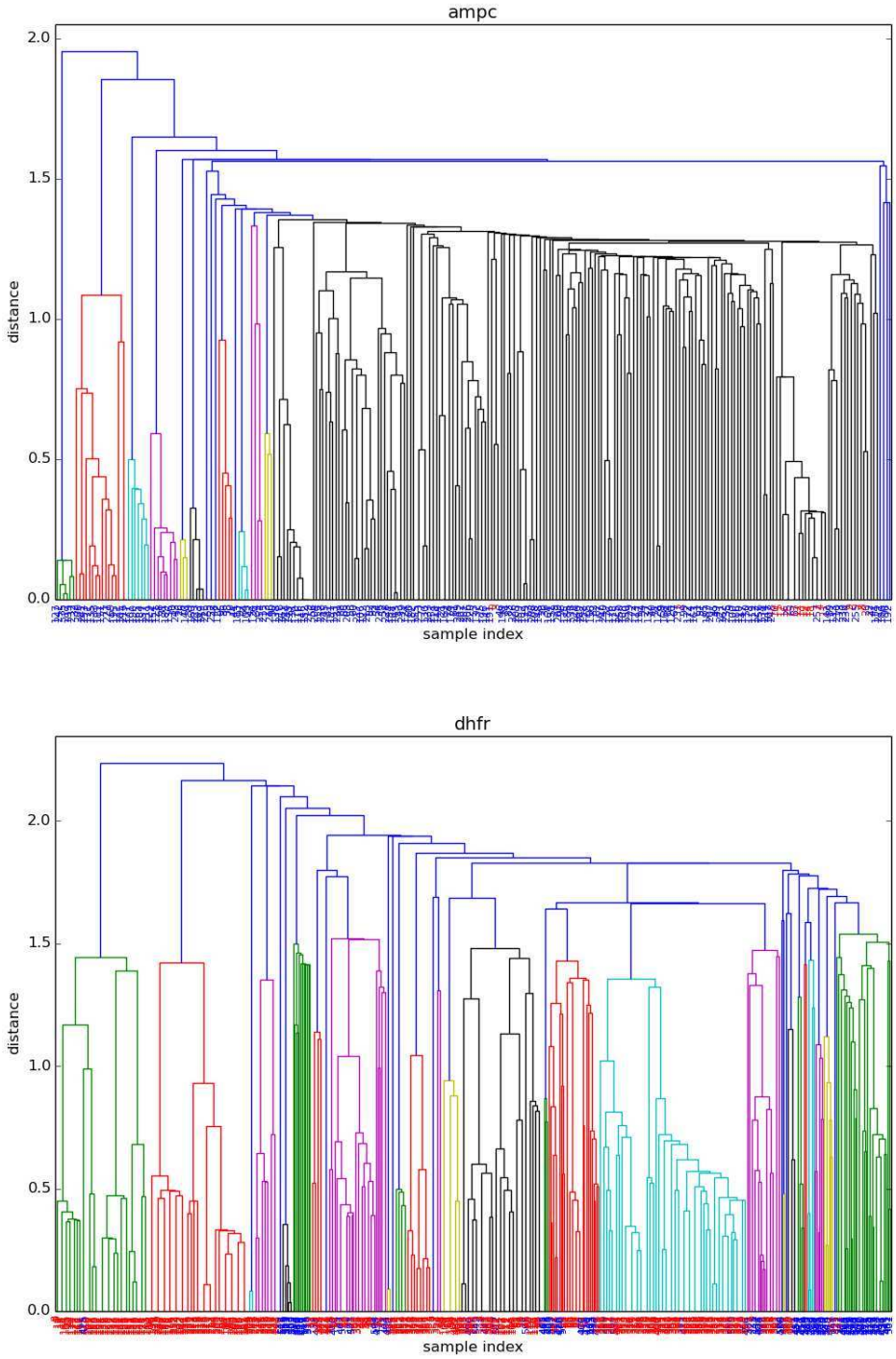


Figura 3.2.3.2: Dendrograma generado a partir de la matriz de distancias de Tanimoto de todos los compuestos semilla (azul) más todos los compuestos que son verdaderos ligandos aportados por DUD (rojo) para los targets AmpC y dhfr.

Por último, para analizar la performance del módulo de generación de estructuras, se analizaron para AmpC y dhfr la cantidad de elementos en cada etapa de la herramienta, completando de esta manera el pipeline utilizando los parámetros por defecto. Como puede observarse en la tabla 3.2.3.3, si bien es variable, la cantidad de geometrías generadas es cercana a un orden de magnitud superior a la cantidad de elementos del conjunto extendido, lo que es un punto a ser tenido en cuenta a la hora de ser más o menos permisivo con los parámetros de extensión del conjunto semilla.

Proteína	**Family**	**Ligandos en DUD**	**Compuestos Semilla**	**Conjunto extendido**	**Estructuras generadas**
AmpC beta lactamase (ampC)	PF00144	21	237	695	2913
Dihydrofolate reductase (dhfr)	PF00186	410	133	512	4147

Tabla 3.2.3.3: Cardinalidades de cada conjunto de salida del pipeline de ejecución de LigQ para los blancos proteicos ampC y dhfr con los parámetros estándar de sistema. Como puede observarse, la cantidad de elementos de cada conjunto aumenta bastante y modificar los parámetros para ser más permisivo, por ejemplo, en el índice de similitud, para obtener compuestos más novedosos como candidatos a ligandos, puede redundar en conjuntos muy grandes para docking molecular.

La segunda validación realizada para la herramienta fue la realización de experimentos de docking molecular para varios de las proteínas de DUD. En todos los casos se comparó los resultados obtenidos mejor rankeados para todo el conjunto de estructuras de compuestos obtenidas por la herramienta desarrollada, con los resultados del dockeo de los

compuestos etiquetados como ligandos por la base de datos originalmente. En todos los casos los resultados han sido comparables y, para los parámetros estándar de la herramienta, los compuestos no han sido demasiado novedosos químicamente.

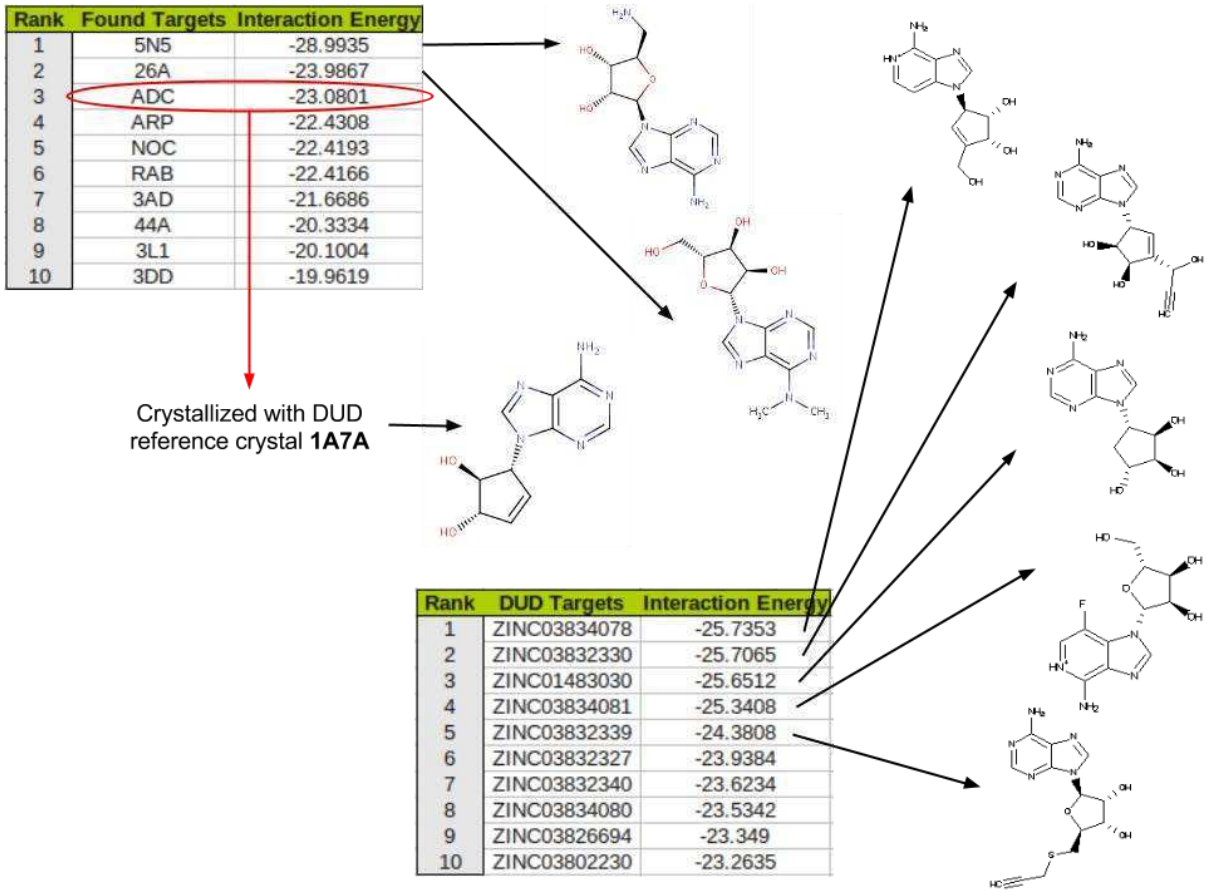


Figura 3.2.3.3: Comparación de los resultados de docking para los ligandos mejor rankeados encontrados por LigQ (arriba) versus los etiquetados como verdaderos lingandos por DUD (abajo). Se encuentran compuestos mejor rankeados en el docking que los ligandos propuestos en DUD y también que el compuesto co cristalizado con la proteína. Vemos que en el tope del ranking no surgen, para este caso, compuestos novedosos químicamente con respecto a ligandos conocidos.

Como en todos los capítulos de esta tesis, la herramienta desarrollada puede ser accedida y probada mediante un servidor web, el cual está disponible en http://ligq.qb.fcen.uba.ar/.

3.2.4 Discusión

En este capítulo hemos presentado la herramienta LigQ cuyo objetivo es proveer, dotado de una interfaz amigable, la posibilidad de generar distintos tipos de conjuntos para Virtual Screening: conjuntos que presenten novedad desde un punto de vista química u otros tendientes a optimizar las características fisicoquímicas que deben cumplir los ligandos encontrados, proveyendo además información acerca de cómo adquirir los compuestos de interés, o presentando ligandos candidatos para proteínas con poca o nula evidencia de ligandos con afinidad. La herramienta está organizada en cuatro módulos independientes, cada uno con una utilidad propia: determinar la cavidad receptora del ligando, conocer los ligandos para la proteńa de interés y las de su misma familia de proteínas, encontrar un conjunto de elementos con la propiedad de poder ser adquiridos enriquecido con binders potenciales y determinar todas las conformaciones tridimensionales probables para cada ligando.

La herramienta ha sido validada contra una base de datos de proteínas y sus ligandos la cual es de particular interés para testear algoritmos de docking molecular, para la misma se han encontrado una porción importante de ligandos "verdaderos" teniendo que analizar a posteriori una pequeña porción de la base de datos que contiene todos los ligandos posibles, generando de esta manera un enriquecimiento de los conjuntos para docking molecular. Para el conjunto de validación, dependiendo de la proteína uno podría ahorrarse entre un 50 y un 99% del tiempo de cómputo de las corridas de docking o incluso ahorrarse dicha etapa probando experimentalmente los compuestos de manera directa, dependiendo

del caso, lo cual de ser practicable maximiza las probabilidades de éxito y certeza en la búsqueda.

Una futura extensión planeada para este desarrollo es la de poder realizar la ejecución del pipeline en el sentido opuesto: teniendo como entrada un compuesto, conocer cuáles son las proteínas que poseen cavidades con buenas probabilidades de acoplamiento del mismo. Esta extensión, que desde el punto de vista técnico tiene un costo bajo, sería útil en varias aplicaciones de la química orgánica y en la búsqueda de aplicaciones para productos naturales.

3.3. Análisis estructural del efecto de mutaciones no sinónimas de proteínas (VarQ)

3.3.1 Introducción

En en el campo de la medicina de precisión, o personalizada[110] uno de los problemas críticos a resolver es el entendimiento de cómo las variantes no sinónimas (aquellas que se traducen en un cambio de aminoácido) en proteínas afectan su actividad, su red de interacciones, su estabilidad, etc., y dan lugar finalmente a fenotipos patogénicos. Se han desarrollado muchos predictores de patogenicidad, los cuales clasifican a estas variantes en distintas categorías: desde "benignas" a "patogénicas", pasando por diferentes grises en el medio. Los mismos, en la mayoría de los casos, construyen una función de puntaje basada en información de secuencia, los cuales no tienen en cuenta la información estructural la cual ha demostrado ser vital para diagnosticar sus efectos efectos[111,112].

El desarrollo de la aplicación desarrollada para este capítulo, llamada VarQ, surge a partir de la necesidad preexistente de realizar un análisis de datos estructurales, de forma manual, con un bajo nivel de sistematización en el proceso de análisis de variantes. Por otro lado, sin una herramienta que permita automatizar el procesamiento y cálculo de propiedades estructurales relevantes, la aplicación de este tipo de análisis a gran escala se convierte en prohibitivo.

VarQ provee de una manera intuitiva información que permite a médicos, bioquímicos, genetistas y todos los profesionales involucrados en medicina personalizada, realizar una anotación comprehensiva de los efectos de mutaciones en relación con su patogenicidad a nivel estructural.

En nuestro desarrollo, cada una de las estructuras disponibles para una proteína es incorporada potencialmente al análisis, aunque solo son efectivamente analizadas aquellas

que no son redundantes con respecto a otras (ver métodos, 3.3.2), y de cada una de las posiciones se analizan las mutaciones reportadas en bases de datos públicas. También se ofrece la opción al usuario de especificar mutaciones no reportadas en bases de datos que sean de su interés. Para cada mutación se analiza cómo afecta el plegamiento[44], la actividad, su involucramiento en interfaces proteína-proteína[113], su pertenencia a sitio activo y cómo éste se ve afectada la potencial unión a compuestos tipo droga. La información obtenida se combina luego con el objetivo de clasificar estas variaciones de secuencia en función de su impacto en la función de la proteína: para cada propiedad calculada se han establecido umbrales simples para guiar el análisis del experto, pero no se pretende generar funciones de puntaje o valoración del efecto de las mutaciones en las proteínas, sino guiar al experto en su diagnóstico.

3.3.2 Materiales y Métodos

Como en todos los desarrollos de esta tesis, para resolver el problema planteado hemos desarrollado un pipeline de análisis que permite obtener la información y resultados necesarios de una manera automatizada y replicable.

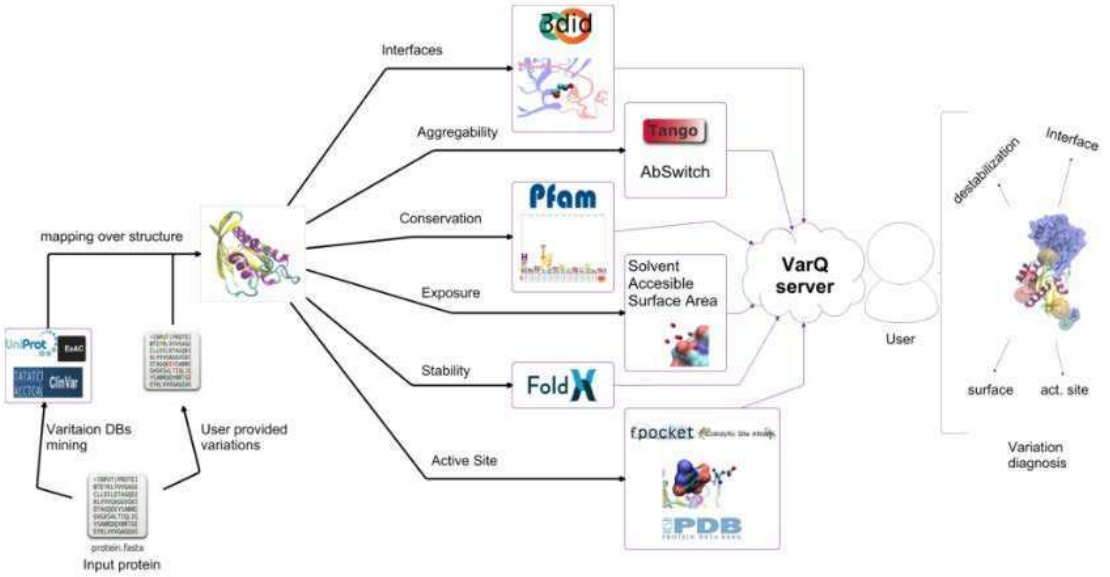


Figura 3.3.2.1: Pipeline de análisis de la herramienta VarQ. En la misma, teniendo como entrada el código UniProt de una proteína, se obtiene la anotación de todas las mutaciones definidas por el usuario y obtenidas de bases de datos, de manera asistir al experto en el diagnóstico de su efecto.

El primer paso, dado el código de la proteína de interés que sirve como entrada al pipeline de procesamiento, es el de obtener de todas las estructuras disponibles para la misma, y luego determinar aquellas que sean relevantes y no redundantes para seguir con ulteriores análisis. Para llevar a cabo este análisis seleccionamos todas aquellas estructuras disponibles de la proteína en el PDB que cubran diferentes segmentos de la secuencia de la proteína, prefiriendo aquellos con mayor cubrimiento de la misma en primer lugar y aquellos de mejor resolución experimental en segundo lugar. Solo seleccionamos cristales que tengan una longitud mayor a los 20 aminoácidos en secuencia.

Cuando dos cristales fueron resueltos cubriendo la misma porción de la secuencia de la proteína de interés, pero con distintos ligandos o cofactores, son considerados como dos configuraciones estructurales distintas. Lo mismo sucede cuando la proteína se encuentra cristalizada heterodiméricamente interactuando con distintas proteínas.

A modo de ejemplo, en la figura 3.3.2.2 se muestran un par de configuraciones estructurales obtenidas para la proteína HRAS humana, procesadas y presentadas de la misma manera en que lo hace la herramienta. En la misma puede observarse que existen dos configuraciones de esta proteína co-cristalizada interactuando con diferentes proteínas. Analizar las variaciones reportadas podría tener diferentes efectos dependiendo de las diferentes configuraciones estructurales. De cada proteína se informa el o los dominios cristalizados y cuáles son las moléculas (no solventes) presentes en cada cadena.

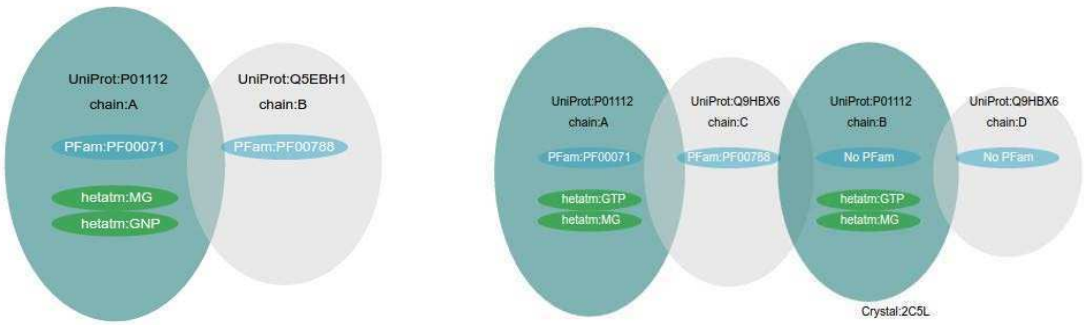


Figura 3.3.2.2: Según la página web de UniProt, a noviembre de 2016 la proteína de código P01112 (HRAS humana) tiene más de cien estructuras cristalográficas resueltas. Sin embargo, nuestro algoritmo de detección de configuraciones solo detecta nueve distintas, este análisis permite para estos casos reducir sensiblemente el tiempo de cómputo. Aquí mostramos dos configuraciones, la de los cristales 2C5L y 3DDC, como son presentados en la página web de la herramienta VarQ.

Para toda las secuencia de la proteína, se buscan entonces en las bases de datos de variantes disponibles de manera abierta (ver métodos generales de esta tesis) todas las mutaciones reportadas que pueden ser mapeadas sobre cada una de las estructuras

Para determinar cuáles posiciones definen el sitio activo de la proteína, se analiza en la estructura si esta está unida a un compuesto tipo droga, y de ser así se marca todos los

residuos en contacto (esta anotación se provee como información en el archivo .pdb del cristal) como de sitio activo. También, se busca en el Catalytic Site Atlas y se ejecuta el software de detección de cavidades Fpocket para encontrar cuál es el sitio candidato, siguiendo la misma estrategia nombrada en los desarrollos anteriores (ProteinQ y LigQ).

Para estimar el cambio en la estabilidad de la proteína, todas las mutaciones que son mapeadas dentro de la configuración estructural que se encuentra siendo analizada, son modeladas con el software FoldX, el cual posee muchas herramientas, una de las cuales es construir un modelo a partir de una estructura dada mutando residuos específicos. Además de construir el modelo, el software predice el impacto energético de la mutación en la estabilidad de la proteína o, en caso de tratarse de un complejo, en la estabilidad del mismo. Aquellas mutaciones que se calcule que generen una variación de energía mayor a 1 Kcal/Mol son etiquetadas como desestabilizantes. Para estructuras homoméricas, la variación provocada por una mutación es calculada en solo una de las cadenas del cristal.

El valor energético de una configuración estructural es calculado con el potencial empírico de FoldX el cual se expresa mediante la siguiente combinación lineal:

$$\Delta G = W_{vdw}\Delta G_{vdw} + W_{solvH}\Delta G_{solvH} + W_{solvP}\Delta G_{solvP} + W_{wb}\Delta G_{wb} +$$

$$W_{h\,bond}\Delta G_{h\,bond} + W_{el}\Delta G_{el} + W_{kon}\Delta G_{kon} + W_{mc}\Delta G_{mc} + W_{sc}\Delta G_{sc} + W_{clash}\Delta G_{clash}$$

En donde cada uno de los términos significan:

- vdw : contribución de Van der Waals.
- $solvH$: contribución de solventes hidrofóbicos.
- $solvP$: contribución de solventes polares.
- wb : contribución por puentes de hidrógeno de moléculas de agua con proteínas.
- $h\,bond$: contribución por puentes de hidrógeno intramoleculares.

- el : contribución por fuerzas electrostáticas.
- kon : contribución por fuerzas electrostáticas proveniente de complejos.
- mc : contribución por fuerzas generadas por la correción del backbone de la proteína en función de los ángulos phi-psi de sus aminoácidos.
- sc : contribución por las conformaciones de las cadenas laterales.
- $clash$: contribución generada por los clashes solapamientos estéricos.

Los valores de cada término W han sido determinados empíricamente en función de valores observados en estructuras cristalográficas resueltas. Las formas funcionales de cada una de las contribuciones de ΔG se encuentran explicadas en la bibliografía[CITA] y no es nuestro objetivo discutirlas en detalle en esta tesis.

En el caso de estar analizando mutaciones, el valor que nos interesa para determinar la contribución energética es la variación provocada en la energía de plegamiento, por lo que estaremos interesados en el $\Delta\Delta G$ entre el cristal original y el modelo con la mutación modelada: para ello se genera una estructura modelada en función de la original pero en el cual se produce la variante en la secuencia de la proteína. Para las dos estructuras ahora disponibles se calcula con el potencial descrito más arriba la energía libre y se las compara.

Otra de las propiedades calculadas es la superficie expuesta al solvente de la cadena lateral de cada aminoácido. Para su cálculo se divide el área total de la cadena lateral versus aquella que se se encuentra expuesta al solvente. La estrategia de cálculo es la de crear una superficie para cada uno de los átomos de la cadena lateral del residuo que está siendo calculado, usando los radios de Van Der Waals y luego intentar ubicar átomos del solvente (por defecto de un radio igual al del agua) sin que entren en colisión con otros átomos de la proteína. Aquellos átomos con más de un 50% de su cadena lateral expuesta al solvente son etiquetados como "de superficie".

Para evaluar si un residuo pertenece o no a una interfaz proteína-proteína se han implementado dos estrategias: por un lado, todo residuo que tenga alguno de sus átomos a una distancia menor a 5A de un residuo de otra cadena es etiquetado como de interfaz; por otro lado, se ha montado de manera local la base de datos 3did, en la cual, para cada posición expresada en el modelo de una familia PFam que está mapeada en un cristal, se encuentra guardado con qué familia interactúa (y cuántas veces). En la figura 3.3.2.3 puede observarse el grafo de interacciones de la familia "Ras" con otras familias de proteínas. Determinar si una posición mutada afecta a la interfaz proteica es fundamental para decidir si afecta su función en cuanto a su red de interacciones.

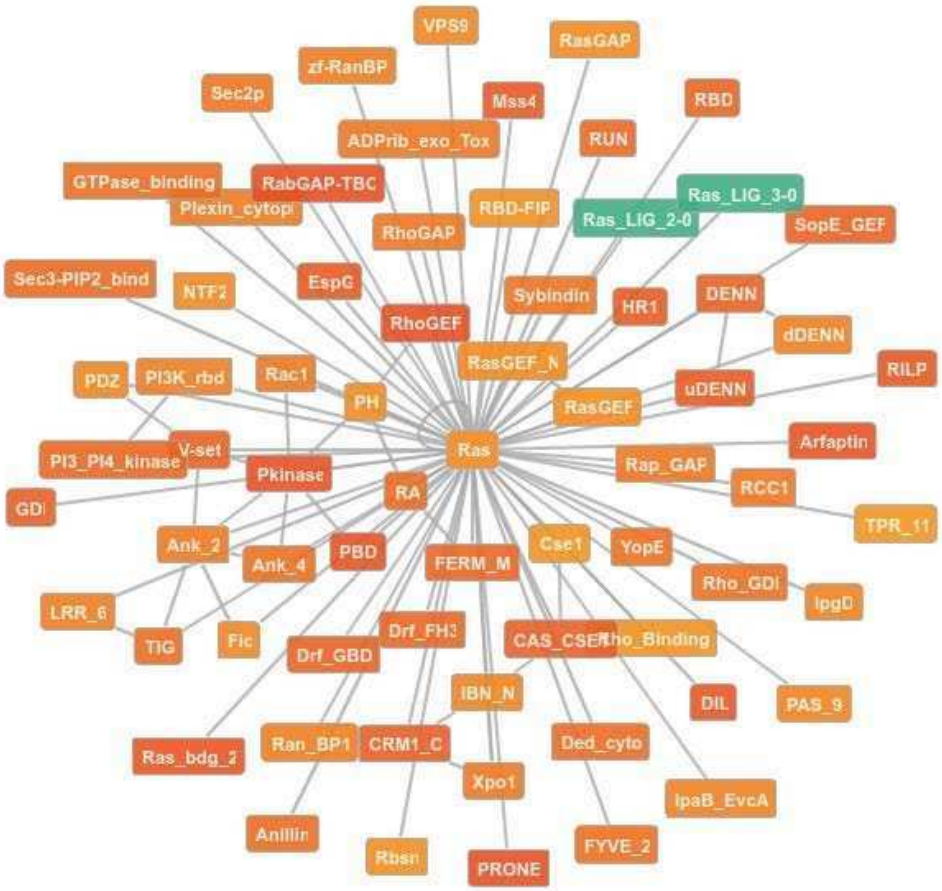


Figura 3.3.2.3: Visión en forma de gráfos de la red de interacciones de cualquiera de los residuos de las proteínas cristalizadas que tienen asignado el dominio PFam Ras. En escala de rojos se grafica la cantidad de evidencias cristalográficas depositadas en el PDB.

Otras propiedades calculadas para los aminoácidos de cada una de las configuraciones estructurales son:

- La conservación en bits del aminoácido, siempre que este pueda ser mapeado a una posición en una familia PFam asignada. Residuos con un alto valor de conservación tendrán, potencialmente, un impacto mayor sobre la función de la proteína ya que afectarán aminoácidos constitutivos de la familia.
- El factor de agregabilidad del aminoácido: el software Tango[114] evalúa cuán propenso es un aminoácido a generar agregación en una proteína desde un punto

de vista estructural, lo cual se sabe que está relacionado con distintas enfermedades.

- El factor de "switching" del aminoácido: el software ABSwitch[115] evalúa cuán propenso es un motivo secuencial de cinco residuos (centrado en el residuo que se está evaluando) a generar un cambio de hélice alfa a hoja beta en la estructura, lo que tiene implicaciones fuertes en el plegamiento y por consiguiente en la función de la proteína.
- El BFactor de cada residuo y su valor relativo a los demás aminoácidos de la estructura, lo que ayuda al usuario a dilucidar el el mismo se encuentra en una región móvil o rígida del cristal que está siendo analizado.

3.3.3 Resultados

Para validar la herramienta desarrollada hemos tomado un set de mutaciones de literatura, para el cual se ha realizado previamente el tipo de análisis que VarQ efectúa de manera sistematizada pero de forma manual. El mismo comprende un conjunto de proteínas involucradas en cáncer y trastornos del desarrollo denominados RASopatías, las cuales se encuentran reportadas en la literatura y analizadas mediante curación manual en detalle orientada a encontrar el mismo tipo de información que intenta obtenerse mediante nuestra herramienta.

	Kiel & Serrano 2014[111]	VarQ
Mutaciones totales	956	1109
Mapeadas en estructura	427	566

Tabla 3.3.3.1: Totalización de mutaciones recabadas y mapeadas en estructura de la herramienta VarQ comparado con el set de validación generado de manera manual en literatura.

Al tratarse de un trabajo del año 2014, nuestra herramienta pudo recabar de manera automatizada de bases de datos de variantes de público acceso un mayor número de mutaciones que las analizadas previamente. De las 427 mutaciones que en literatura pudieron ser mapeadas sobre la estructura, pudimos mapear exitosamente 414, las 152 restantes (para completar las 566 reportadas en la tabla) son nuevos casos no estudiados en el trabajo anterior. En análisis más detallados, encontramos que las mutaciones restantes se encuentran reportadas en la base de datos COSMIC[116] la cual no hemos tenido en cuenta, principalmente por dos razones: por un lado, el acceso programático a esta base de datos, y el proceso de mapeo de posiciones en cromosomas a posiciones efectivas en la secuencia uniprot de una proteína es un método de difícil automatización, y porque, en segunda instancia, la información contenida en esta base es exclusiva de cáncer, mientras que nuestra herramienta pretende ser general. De todas maneras, hemos recuperado de manera automática ~97% de las mutaciones reportadas previamente, lo que es una cifra muy significativa.

	Kiel & Serrano 2014	VarQ
Interdominio	79(18%)	74(17%)
Sitio Activo	147(34%)	203(49%)
Inhibición	53(12%)	19(4%)
Plegamiento	145(33%)	118(29%)
Localización	3(1%)	-
Total	**427**	**414**

Tabla 3.3.3.2: Clasificación en base a los umbrales establecidos para las propiedades generadas por VarQ sobre el conjunto de validación comparados con la clasificación generada de manera manual en literatura del conjunto análogo.

En la tabla 3.3.3.2, se ha realizado con VarQ un análisis cuantitativo de las mutaciones obtenidas en literatura previa con el fin de poder comparar la clasificación realizada por expertos de manera manual con el aporte de nuestra herramienta. Las posiciones inter-dominio fueron derivados de la base de datos 3did. Todas las posiciones con altos valores de switchabilidad o agregabilidad que pertenecían a interfaces heterodiméricas fueron marcados de manera separada como inhibitorias. Las posiciones de sitio activo son aquellas que se encuentran marcados como unidos a ligando en los archivos PDB o que pertenecen al mismo pocket que se encuentre conteniendo estos residuos nombrados o aquellos que pertenezcan al Catalytic Site Atlas. Los residuos que afectan el plegamiento fueron identificados como aquellos que no afectan la interfaz entre dominios ni el sitio activo. Hubo trece mutaciones que no pudieron ser mapeados automáticamente con

ninguno de estos conjuntos. Se observa que, cuantitativamente, la clasificación realizada por la herramienta arroja clasificaciones comparables en cuanto a proporciones y, aunque no puede visualizarse en la tabla, también en cuanto a los casos, los cuales poseen una alta intersección.

Para las 556 mutaciones encontradas que mapeamos sobre estructura, tomando la etiqueta provista por las bases de datos para considerar la mutación como *patogénica* (por ejemplo, en ClinVar, aquellos con etiqueta "probable-pathogenic" o "pathogenic") o *no patogénica*, hemos computado el histograma de energía $\Delta\Delta G$ provista por el software FoldX causado por cada mutación (Figura 3.3.3.1). Una variante es clasificada como patogénica por las bases de datos de casos clínicos, cuando la misma se encuentra reportada en pacientes que han sido secuenciados y poseen una enfermedad identificada en una cantidad significativa estadísticamente.

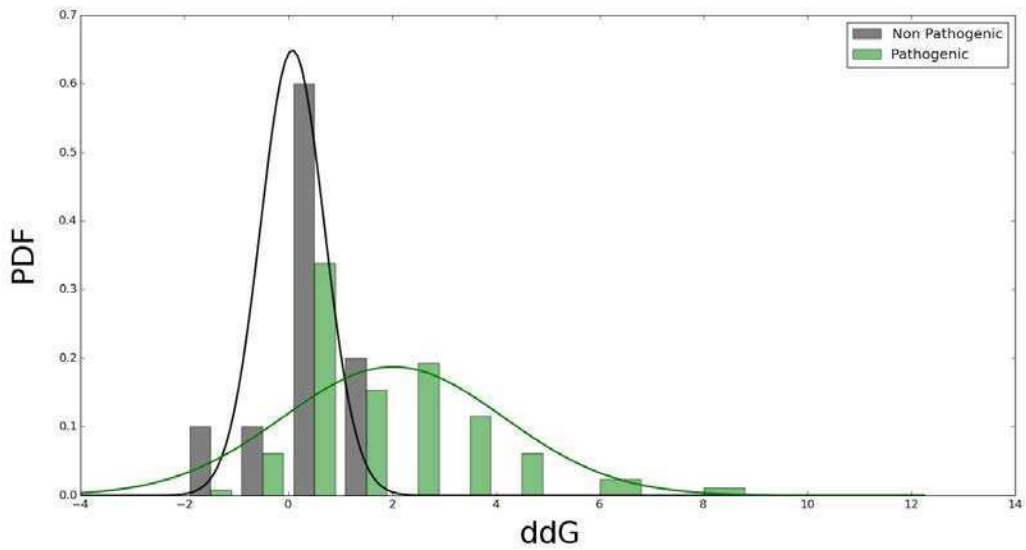


Figura 3.3.3.1: Histograma del cambio en la energía de plegamiento generada por mutaciones patogénicas(verde) y no patogénicas(negro) calculada por el software FoldX. Para cambios de energía altos (> 2 Kcal/Mol) las mutaciones resultan siempre ser causantes de enfermedades, dentro del conjunto de estudio.

En el análisis de esta herramienta, hemos dilucidado que la misma sirve para detectar mutaciones que serán patogénicas, que es cuando el cambio de energía sea alto (si es mayor a 2 kcal/mol observamos que esa variante siempre está reportada como patogénica). Las proteínas en las que se ha producido una variante que genera una alta desestabilización, al menos en los casos reportados, afectan la función de la proteína de manera tal que este cambio repercute en el desarrollo de enfermedades relacionadas con el correcto funcionamiento de estas proteínas (RASopatías).

Para valores bajos de desestabilización, las causas de patogenicidad deberán ser determinadas mediante otras propiedades, generando un "árbol de decisión" como el que hemos desarrollado de manera manual para el análisis de la tabla 3.3.3.2, que en última instancia siempre requerirá una evaluación exhaustiva del caso realizado por un experto

El servidor web desarrollado para consultar los casos de validación o correr cálculos en nuevas proteínas, está disponible en http://varq.qb.fcen.uba.ar/. En el mismo, un nuevo trabajo se define mediante el código UniProt de la proteína de interés.

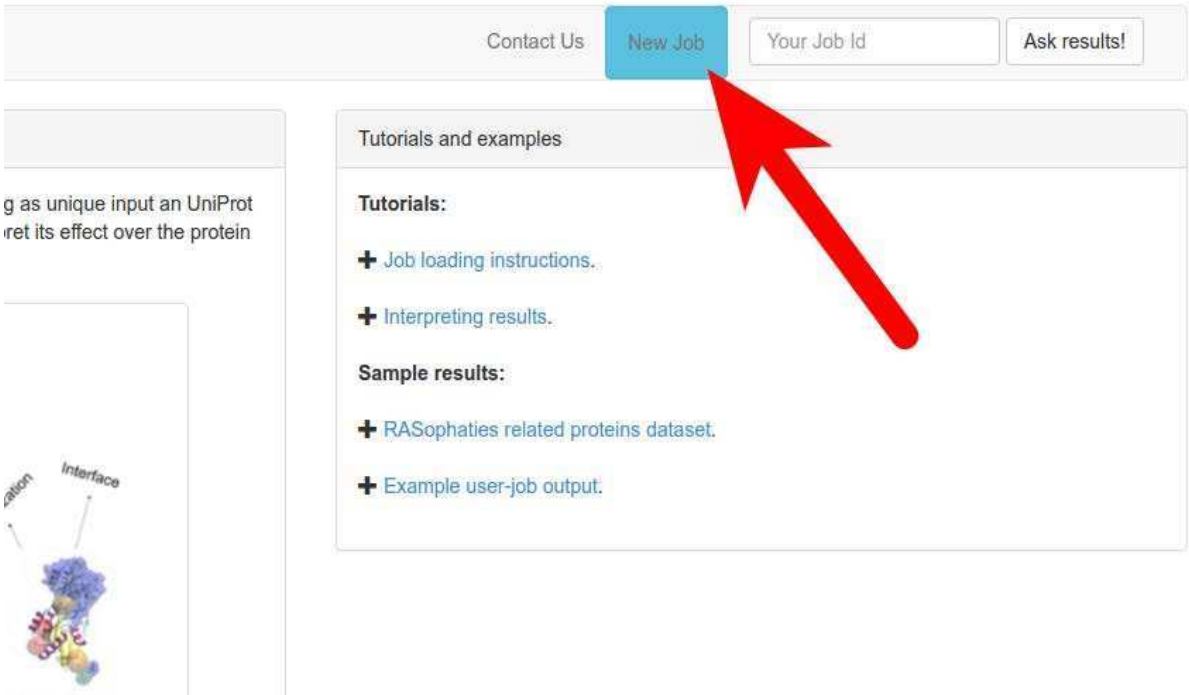


Figura 3.3.3.2: Visualización de la pantalla principal de la herramienta VarQ, señalando el vínculo que lleva a la pantalla para la carga de nuevos cálculos sobre una proteína de interés.

Se pueden ingresar, además de buscar mutaciones en las bases de datos de acceso público ya explicadas, una lista de mutaciones de interés para ejecutar los cálculos del pipeline sobre las mismas y luego analizar el resultado de manera interactiva.

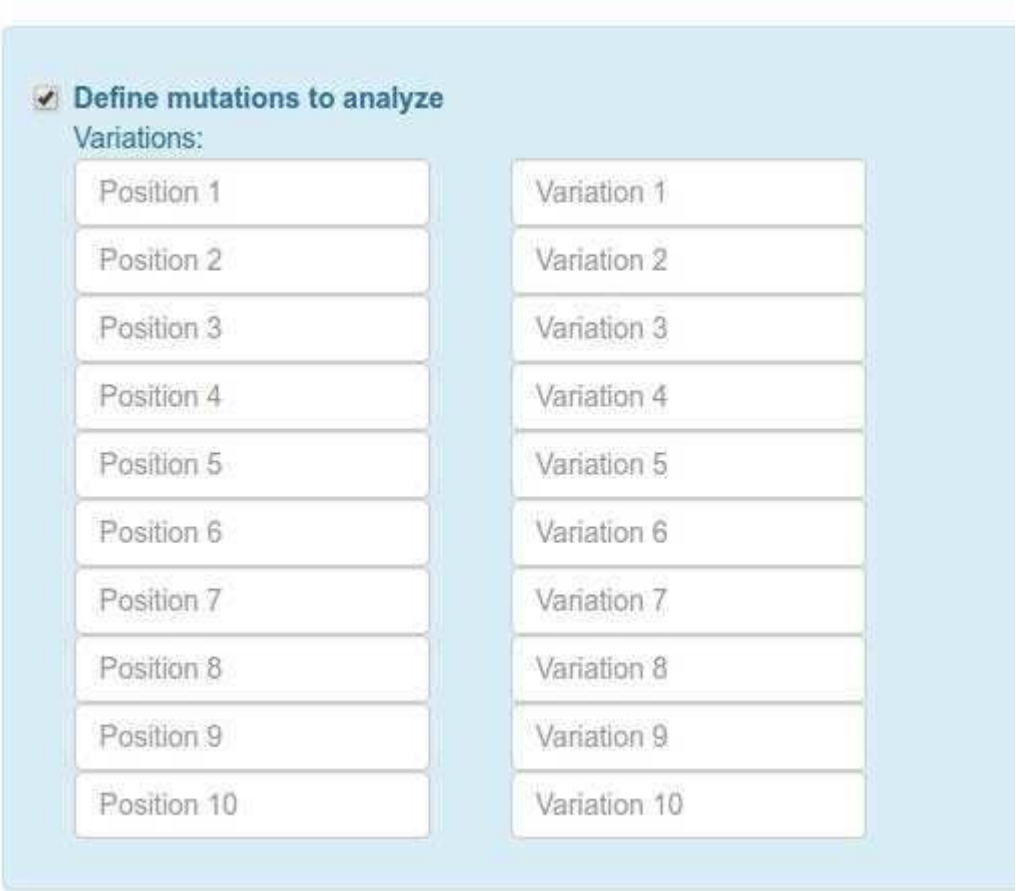


Figura 3.3.3.3: En la pantalla de carga de nuevos trabajos, una de las opciones ofrecidas por la herramienta es la de cargar para posiciones de la secuencia, mutaciones que sean de interés para que sean analizadas aunque no se encuentren reportadas en bases de datos de mutaciones.

Cuando el trabajo se encuentra calculado, se envía un email al usuario para informarlo de que puede consultar la información, si es que lo ingresó en el momento de generar el nuevo trabajo. Para la proteína ingresada, se mostrarán en una primer pantalla todas las configuraciones estructurales disponibles, y para cada una de ellas se explicitará el grafo de interacciones y cómo las mutaciones y las familias PFam de la proteína se mapean sobre la estructura.

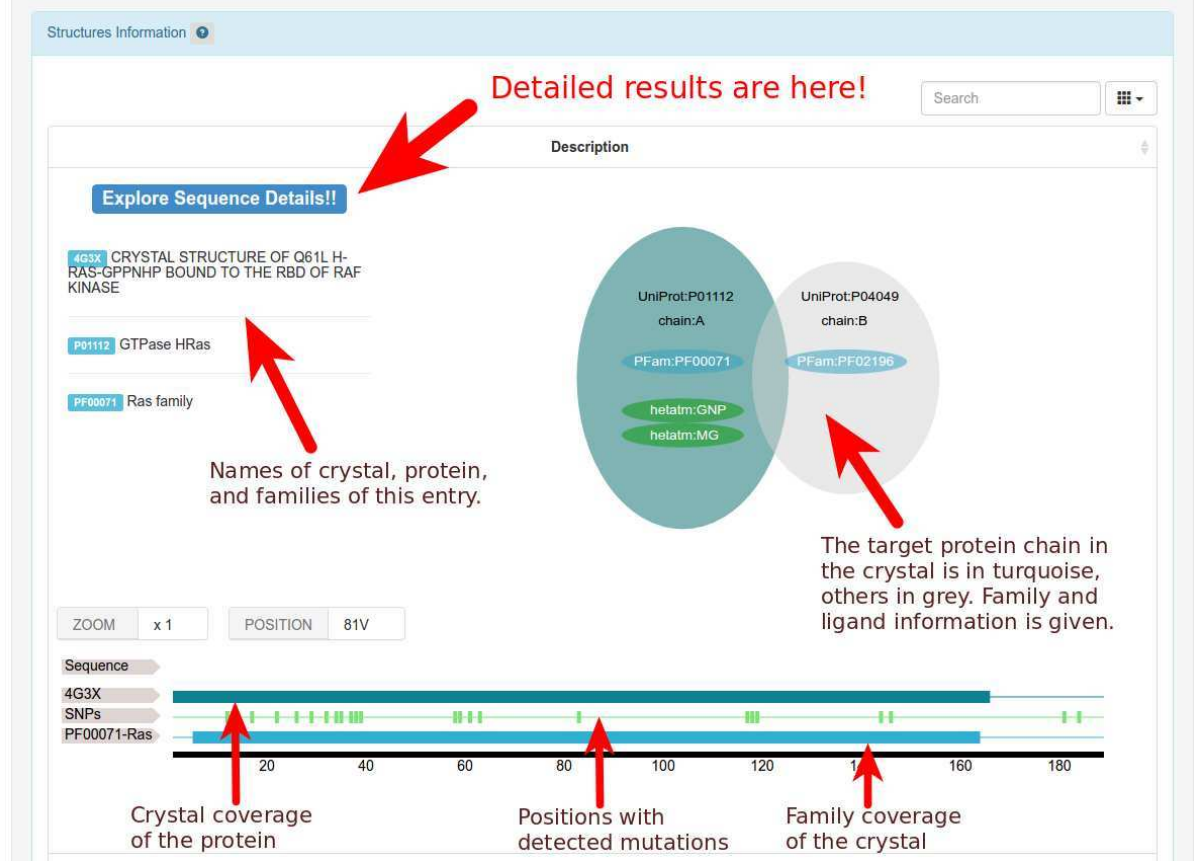


Figura 3.3.3.4: Visualización de la pantalla principal de un trabajo que ha terminado de ejecutar exitosamente. En la misma, podemos ver que para cada una de las configuraciones estructurales se presenta información sumarizada acerca del mapeo de las mutaciones y cómo está definida la configuración en base a sus interacciones. Vemos que la proteína P01112 (HRAS) tiene un cristal (4G3X) en la cual interactúa con la proteína P04049, en la cual se han mapeado mutaciones las cuales pueden ser exploradas en pantallas posteriores.

De cada una de las configuraciones estructurales que se hayan seleccionado para la proteína, se puede entonces acceder al detalle de lo calculado para cada posición de su secuencia, en donde la información sumarizada y las etiquetas aplicadas en función de los umbrales definidos son mostradas en una lista de posiciones (Figura 3.3.3.5).

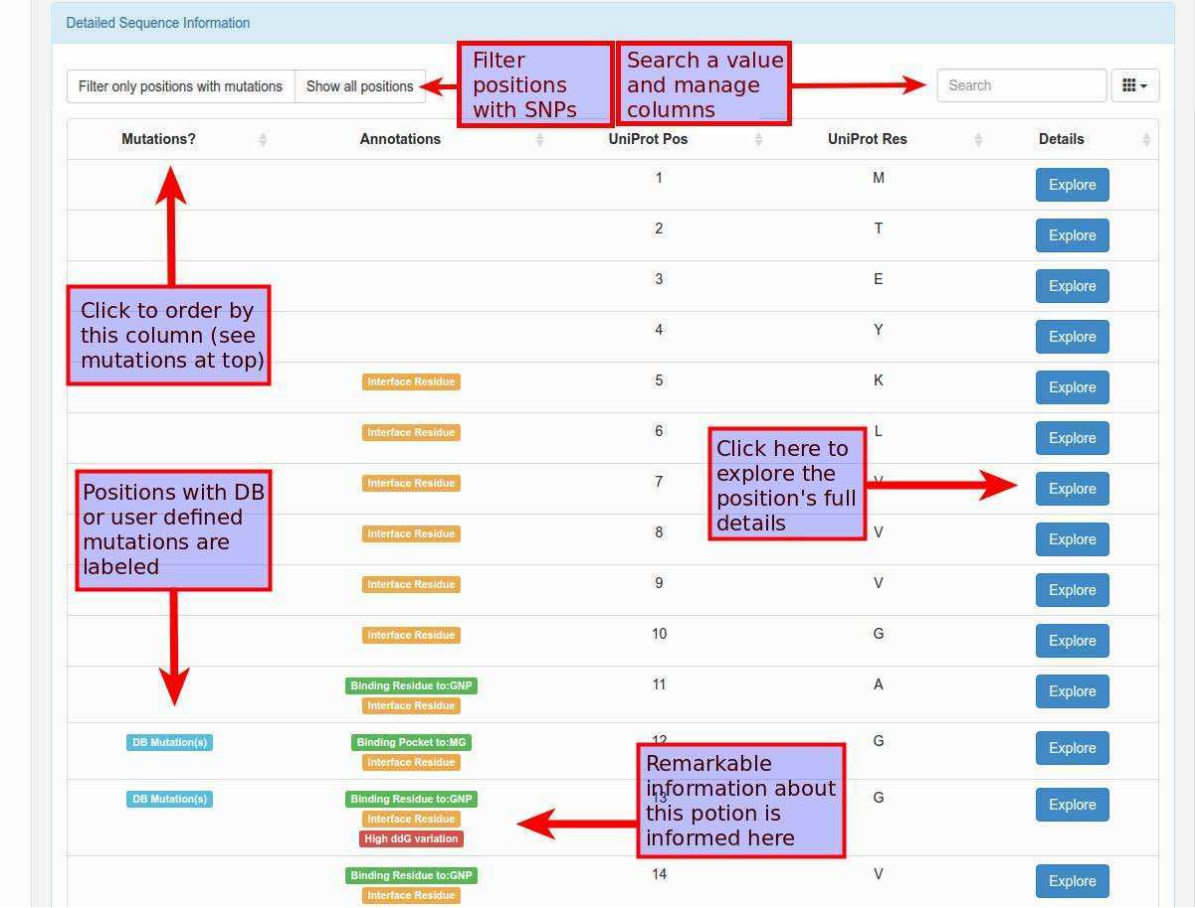


Figura 3.3.3.5: Visualización del detalle de una configuración estructural. Cada una de las posiciones está etiquetada con la descripción que les corresponde en función de los valores de sus propiedades. Cada posición tiene indicada en la primera columna si existen mutaciones reportadas o ingresadas por el usuario que han sido calculadas.

Para acceder a la información específica de la salida de cada uno de los algoritmos utilizados sobre la posición de interés, una pantalla desplegará los detalles de la misma.

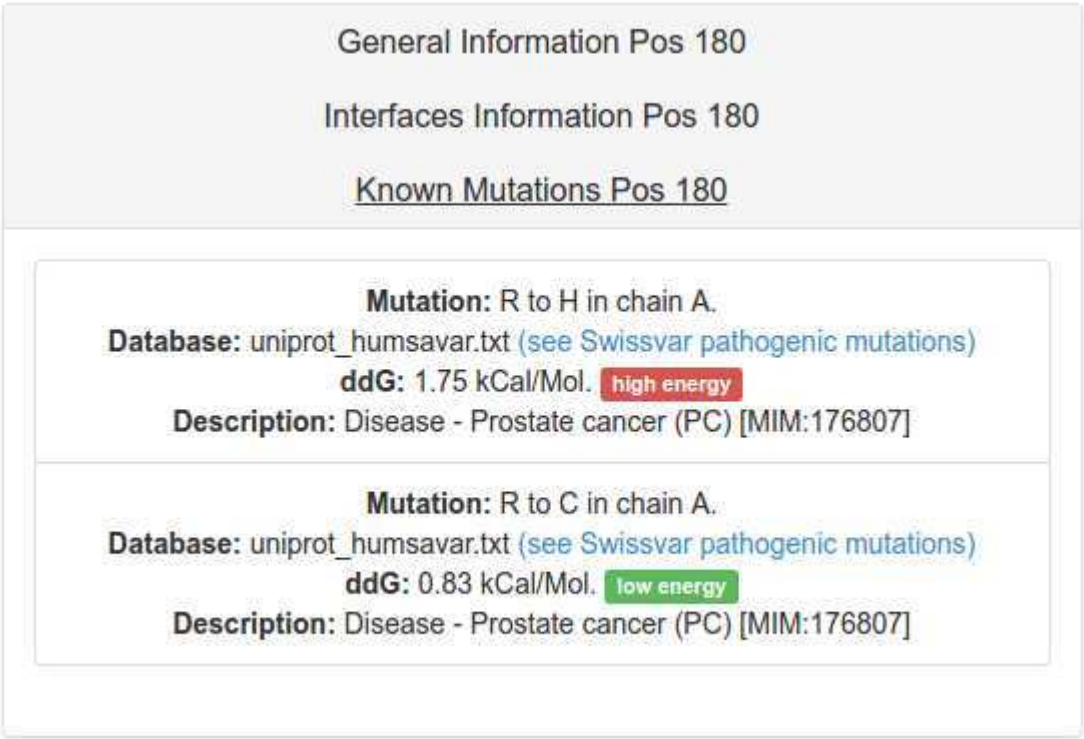


Figura 3.3.3.5: Visualización del detalle de las propiedades de una posición. La misma está organizada en secciones, en la presente figura se encuentra desplegada la información que concierne a las mutaciones reportadas para una posición puntual.

En la misma pantalla, además, podrá visualizarse la posición de la mutación dentro de la estructura que está siendo analizada, mediante el plugin de visualización JMol. El mismo ofrece diferentes modos de visualización y análisis (medida de distancias, análisis de puentes de hidrógeno, etc.) que pueden contribuir al diagnóstico del efecto de las mutaciones que se estén analizado (Figura 3.3.3.6).

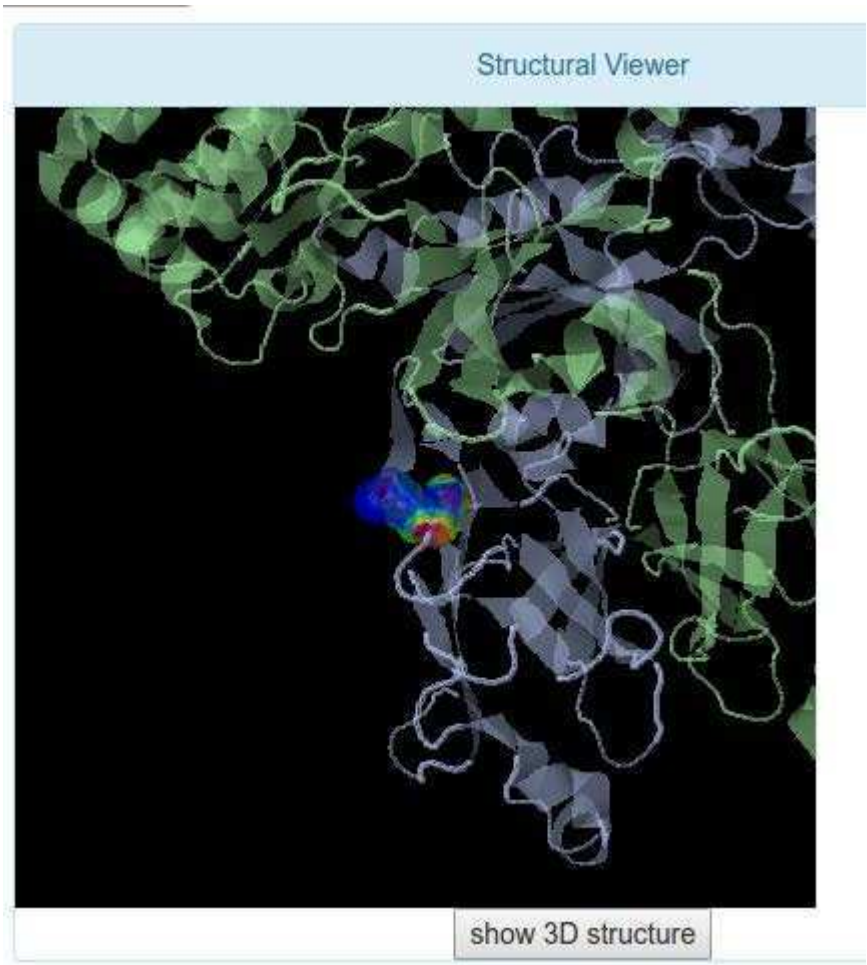


Figura 3.3.3.6: Visualización de la estructura de la proteína en el contexto de la exploración de detalles de una posición dentro de una configuración estructural, donde se encuentra remarcada la posición que está siendo explorada.

3.3.4 Discusión

En este capítulo hemos presentado la herramienta VarQ, la cual tiene el objetivo de asistir al especialista en el diagnóstico de posibles efectos de mutaciones puntuales en las secuencias de proteínas sobre su función y los efectos subyacentes de las mismas.

Además, para la herramienta se ha desarrollado un servidor web que permite tanto ejecutar nuevos cálculos como analizar de manera visual los resultados de los mismos, ayudando también de esta manera al diagnóstico.

La herramienta ha sido validado contra un conjunto de mutaciones con un alto nivel de curación manual pertenecientes a un conjunto de proteínas relacionados con enfermedades del desarrollo y cáncer y para las mismas han podido reproducirse, esta vez de manera automática, resultados comparables desde el punto de vista cuantitativo y de clasificación. El análisis cualitativo de cada caso, nos parece importante remarcar, tiene que ser realizado por el experto en cada ocasión, ya que es él quien entiende del contexto donde la mutación se produce, las implicancias en redes de interacción, etc.

El desarrollo realizado constituye un avance en el diagnóstico del efecto de mutaciones no sinónimas a gran escala teniendo en cuenta información estructural, lo cual tiene un impacto directo en todas las disciplinas relacionadas con la medicina personalizada.

En el futuro, la extensión de este desarrollo permitirá tomar como punto de entrada la secuenciación de un paciente mapeando automáticamente, cuando fuere posible, cada una de sus mutaciones, con las estructuras disponibles de proteínas, realizando el actual análisis. Una segunda extensión será la incorporación de algoritmos de inteligencia artificial para la predicción del efecto de las variantes reportadas.

4. Conclusiones y perspectivas

El primer análisis que debe hacerse al momento de plantear las conclusiones de este trabajo es en relación a los objetivos propuestos. Desde un punto de vista general se planteó la meta de desarrollar y utilizar métodos bioinformáticos volcados a resolver problemas biológicos relacionados con la salud humana. En ese sentido, todo los desarrollos que hemos presentado han estado orientados y dedicados a atacar estas problemáticas. Por supuesto que para cada uno de los desarrollos existen perspectivas para ampliar o mejorar las capacidades de los mismos, pero en su estadío actual, los mismos representan avances concretos para el cumplimiento de este objetivo genera y están disponibles para su uso por parte de la comunidad.

En cuanto al primer objetivo específico, que planteaba desarrollar una herramienta que asista en la elección de blancos proteicos a escala genómica en base a su drogabilidad estructuraVarQ

Análisis de mutaciones en pacientes para diagnóstico.

Extensión a paneles de proteínas relacionadas con otras enfermedades.

Procesamiento automático de VCFsl, hemos desarrollado TuberQ. El desarrollo ha logrado constituirse como una herramienta que facilita y sustenta la búsqueda de blancos interesantes. Aunque hemos presentado el primer desarrollo centrado en Mtb, la metodología ha demostrado ser útil y escalable y ya se ha replicado en otros organismos bacterianos. Toda la información de la base de datos generada es de público acceso, permitiendo a cada usuario hacer su propio análisis. Este enfoque es sustancialmente diferente de otros desarrollos en los que los resultados son presentados en forma cerrada

en una tabla que indica cuales son los mejores blancos para atacar con miras a tratar el patógeno que sea de interés en el trabajo.

El segundo objetivo específico planteado hablaba de desarrollar una herramienta que permita encontrar compuestos que sean candidatos de modular o inhibir la actividad de un blanco proteico determinado. Este objetivo se enlaza de manera directa con el anterior, ya que es la continuación natural en cuanto al desarrollo de herramientas *in-silico* en la búsqueda de nuevos tratamientos para enfermedades bacterianas (además de otros usos potenciales). Para este objetivo hemos desarrollado la herramienta LigQ que ha demostrado enriquecer los conjuntos de compuestos que en etapas posteriores formarán parte de experimentos de Virtual Screening y/o conjuntos de compuestos que serán probados de manera experimental. La herramienta, cuyo testeo formal es complicado, porque cualquier compuesto novedoso que sea postulado tiene que ser confirmado experimentalmente, objetivo que no forma parte de este trabajo, ha sido validada contra un conjunto de blancos proteicos que son un estándar en el campo del Docking Molecular, el Directory of Useful Decoys. Para estos targets, la herramienta ha demostrado producir un notable enriquecimiento en los conjuntos propuestos.

Finalmente, el tercer objetivo específico planteaba desarrollar una herramienta que permitiese analizar los efectos de mutaciones puntuales en proteínas, para el cual hemos desarrollado la herramienta VarQ. La misma obtiene de manera automática las mutaciones reportadas de manera pública para estas proteínas y también permite al usuario ejecutar el análisis sobre mutaciones que sean de su interés. Este objetivo no tiene una secuencialidad directa con los dos anteriores, pero sí puntos de contacto en cuanto al uso de la bioinformática en el desarrollo de herramientas que permitan el análisis automático centrado en la estructura de proteínas con aplicaciones en salud. El desarrollo ha reproducido con éxito, y de manera automática, análisis que en el pasado han insumido una cantidad de tiempo considerable con resultados comparables.

Todos los desarrollos realizados cumplen además una premisa que nos planteamos en el comienzo del trabajo, que es el de ser reproducibles, de uso libre y ser extensibles. Ya planteamos cómo se ha extendido el desarrollo de TuberQ a otros organismos, en cuanto a los otros dos desarrollos, los mismos se encuentran *online* y son de público acceso, para que cualquier persona los utilice, valide y extienda más allá de los casos con los que nosotros nos hemos valido tanto para verificar con funcionamiento como para aplicar a casos de interés.

Como en todo trabajo científico han quedado temas pendientes de desarrollo, debido tanto al tiempo finito que se posee para realizarlos, como a que estos temas están enmarcados en líneas de investigación que tienen un comienzo anterior y un final posterior al alcance de la presente tesis.

La herramienta TuberQ está siendo extendida a otros organismos y se está desarrollando una herramienta que automatiza la generación de la base de datos y análisis, de manera que el usuario pueda trabajar interactivamente con el proteoma que sea de su interés. La herramienta aún no se encuentra publicada pero está disponible en versión de prueba en http://www.biargentina.com.ar/xomeq/. A su vez, otros miembros del laboratorio se han dedicado a las pruebas experimentales de inhibidores de blancos de interés seleccionados en Mtb.

En cuanto a LigQ, dos de los pasos posteriores planteados para su extensión son los de a) utilizar algoritmos de inteligencia artificial para vincular las propiedades que puedan calcularse sobre el sitio activo de la proteína con las que debieran cumplir los compuestos candidatos finales con miras a mejorar los conjuntos presentados por la herramienta; y b) desarrollar una herramienta análoga pero "inversa", en el sentido de que el punto de partida es el compuesto o producto natural que el investigador tiene en su poder, para determinar

cuáles son los potenciales blancos proteicos que tienen alguna probabilidad de unirse a dicho compuesto y a través de esa unión modular su actividad.

Para VarQ el paso próximo en el desarrollo es el de generar una clasificación más sofisticada de las mutaciones que la actual, la cual se basa en umbrales muy simples producto del análisis de las distribuciones de las propiedades. Uno de los primeros productos de este análisis es el de generar un "puntaje de patogenicidad" basado en información estructural, siendo que los actuales predictores incorporan en su mayoría información secuencial únicamente. Pero, más allá de un puntaje, el objetivo de este desarrollo es el de ayudar al diagnóstico del efecto de mutaciones presentando información estructural combinada de una manera inteligente en la que se sugieran causas y se ayude verdaderamente a diagnosticar los mecanismos afectados por el cambio aminoacídico. Otro de los desarrollos planteados a futuro es la incorporación de información metabólica para determinar la relevancia a nivel red de interacciones de las mutaciones analizadas en la proteína de interés.

En resumen, en este trabajo se han desarrollado herramientas que pueden enmarcarse dentro del campo de la bioinformática estructural, con diferentes campos de aplicación que en nuestro análisis se han vinculado al tratamiento de patologías. Las herramientas desarrolladas, si bien nos hemos concentrado en aplicaciones particulares para validación y aplicación, tienen un potencial uso general y extensivo que las convierten en desarrollos bioinformáticos con peso propio.

Publicaciones realizadas a partir de esta tesis

→ **TuberQ: a Mycobacterium tuberculosis Protein's Druggability Database**

Leandro Radusky, Lucas A Defelipe, Esteban Lanzarotti, Javier Luque, Xavier Barril, Marcelo A Marti, Adrián G Turjanski - Oxford Database Journal 2014: bau035 doi:10.1093/database/bau035

→ **An integrated structural proteomics approach along the druggable genome of Corynebacterium pseudotuberculosis species for putative druggable targets**

Leandro G Radusky, Syed S Hassan, Esteban Lanzarotti, Sandeep Tiwari, Syed B Jamal, Javed Ali, Amjad Ali, Rafaela S Ferreira, Debmalya Barh, Artur Silva, Adrián G Turjanski, Vasco AC Azevedo - BMC Genomics, 2014, doi: 10.1186/1471-2164-16-S5-S9

→ **A whole genome bioinformatic approach to determine potential latent phase specific targets in Mycobacterium tuberculosis**

Lucas A Defelipe, Leandro Radusky, Esteban Lanzarotti, Adrián G Turjanski, Marcelo A Marti, Dario Fernández Do Porto, Ezequiel Sosa, Pablo Ivan Pereira Ramos, Marisa Fabiana Nicolás - Tuberculosis, 2015

→ **VarQ: A tool for the strutural analysis of protein variants**

Leandro Radusky, Javier Delgado, Sebastian Vishnopolska, Juan P. Bustamante, Christina Kiel, Marcelo A. Martí, Luis Serrano, Adrián G. Turjanski *(en preparación)*

→ **LigQ: A tool for the enrichment of Virtual Screening compound sets**

Leandro Radusky, Sergio Ruiz-Carmona, Xavier Barril, Adrián G Turjanski, Marcelo A. Martí *(enviado a Journal of Chemical Information and Modelling)*

Otras publicaciones del candidato

→**Protein frustratometer: a tool to localize energetic frustration in protein molecules**

Michael Jenik, R. Gonzalo Parra, Leandro G. Radusky; Adrian Turjanski, Peter G. Wolynes; Diego U. Ferreiro - Nucleic Acids Research 2012, doi: 10.1093/nar/gks447

→**Using crystallographic water properties for the analysis and prediction of lectin–carbohydrate complex structures**

Carlos Modenutti, Diego Gauto, Leandro Radusky, Juan Blanco, Adrian Turjanski and Marcelo A. Marti - Glycobiology, 2015, doi: 10.1093/glycob/cwu102

→**Evolutionary and Functional Relationships in the Truncated Hemoglobin Family**

Juan Pablo Bustamante, Leonardo Boechi, Leandro Radusky, Darío Estrin, Arjen ten Have, Marcelo Adrián Martí - PLoS Computational Biology, 2016 12(1): e1004701. doi:10.1371/journal.pcbi.1004701

→**Protein Frustratometer 2: a tool to localize energetic frustration in protein molecules, now with electrostatics**

Parra, Gonzalo; Schafer, Nicholas; Radusky, Leandro; Tsai, Min-Yeh; Guzovsky, A. Brenda; Wolynes, Peter; Ferreiro, Diego -Nucl. Acids Res. (08 July 2016) 44 (W1): W356-W360. doi: 10.1093/nar/gkw304

Referencias

1. Luscombe, N. M., Greenbaum, D., & Gerstein, M. (2001). What is bioinformatics? A proposed definition and overview of the field. Methods of information in medicine, 40(4), 346-358.
2. Doolittle, R. F. (1981). Similar amino acid sequences: chance or common ancestry. *Science*, *214*(4517), 149-159.
3. Howe, D., Costanzo, M., Fey, P., Gojobori, T., Hannick, L., Hide, W., ... & Twigger, S. (2008). Big data: The future of biocuration. *Nature*, *455*(7209), 47-50.
4. Crick, F., Barnett, L., Brenner, S., & Watts-Tobin, R. J. (1961). *General nature of the genetic code for proteins*. Macmillan Journals Limited.
5. Sanger, F., Nicklen, S., & Coulson, A. R. (1977). DNA sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences*,*74*(12), 5463-5467.
6. Sjölander, K. (2004). Phylogenomic inference of protein molecular function: advances and challenges. *Bioinformatics*, *20*(2), 170-179.
7. Pellegrini, M., Marcotte, E. M., Thompson, M. J., Eisenberg, D., & Yeates, T. O. (1999). Assigning protein functions by comparative genome analysis: protein phylogenetic profiles. *Proceedings of the National Academy of Sciences*, *96*(8), 4285-4288.
8. Sander, C., & Schneider, R. (1991). Database of homology-derived protein structures and the structural meaning of sequence alignment. *Proteins: Structure, Function, and Bioinformatics*, *9*(1), 56-68.
9. Levin, J. M., Robson, B., & Garnier, J. (1986). An algorithm for secondary structure determination in proteins based on sequence similarity. *FEBS letters*,*205*(2), 303-308.
10. Edgar, R. C. (2004). MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic acids research*, *32*(5), 1792-1797.
11. Benson, D. A., Cavanaugh, M., Clark, K., Karsch-Mizrachi, I., Lipman, D. J., Ostell, J., & Sayers, E. W. (2013). GenBank. *Nucleic acids research*, *41*(D1), D36-D42.
12. McGuffin, L. J., Bryson, K., & Jones, D. T. (2000). The PSIPRED protein structure prediction server. *Bioinformatics*, *16*(4), 404-405.
13. Pruitt, K. D., Tatusova, T., & Maglott, D. R. (2007). NCBI reference sequences (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. *Nucleic acids research*, *35*(suppl 1), D61-D65.
14. Bernstein, F. C., Koetzle, T. F., Williams, G. J., Meyer, E. F., Brice, M. D., Rodgers, J. R., ... & Tasumi, M. (1977). The protein data bank. *European Journal of Biochemistry*, *80*(2), 319-324.
15. Xenarios, I., Salwinski, L., Duan, X. J., Higney, P., Kim, S. M., & Eisenberg, D. (2002). DIP, the Database of Interacting Proteins: a research tool for studying cellular networks of protein interactions. *Nucleic acids research*, *30*(1), 303-305.
16. Schuster, S. C. (2007). Next-generation sequencing transforms today's biology.*Nature*, *200*(8), 16-18.

17. Gomez-Cabrero, D., Abugessaisa, I., Maier, D., Teschendorff, A., Merkenschlager, M., Gisel, A., ... & Tegnér, J. (2014). Data integration in the era of omics: current and future challenges. *BMC systems biology*, *8*(2), 1.
18. Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., & Byers, A. H. (2011). Big data: The next frontier for innovation, competition, and productivity.
19. Higgins, D. G., & Sharp, P. M. (1988). CLUSTAL: a package for performing multiple sequence alignment on a microcomputer. *Gene*, *73*(1), 237-244.
20. Eddy, S. R. (1996). Hidden markov models. *Current opinion in structural biology*, *6*(3), 361-365.
21. Feng, D. F., & Doolittle, R. F. (1987). Progressive sequence alignment as a prerequisitetto correct phylogenetic trees. *Journal of molecular evolution*, *25*(4), 351-360.
22. Rose, P. W., Bi, C., Bluhm, W. F., Christie, C. H., Dimitropoulos, D., Dutta, S., ... & Quinn, G. B. (2013). The RCSB Protein Data Bank: new resources for research and education. *Nucleic acids research*, *41*(D1), D475-D482.
23. Simons KT, Bonneau R, Ruczinski I, Baker D (1999) Ab initio protein structure prediction of CASP III targets using ROSETTA. Proteins Suppl 3:171-6
24. Webb, B., & Sali, A. (2014). Comparative protein structure modeling using Modeller. *Current protocols in bioinformatics*, 5-6.
25. Barril, X. (2013). Druggability predictions: methods, limitations, and applications. *Wiley Interdisciplinary Reviews: Computational Molecular Science*,*3*(4), 327-338.
26. Schmidtke, P., & Barril, X. (2010). Understanding and predicting druggability. A high-throughput method for detection of drug binding sites. *Journal of medicinal chemistry*, *53*(15), 5858-5867.
27. Porter, C. T., Bartlett, G. J., & Thornton, J. M. (2004). The Catalytic Site Atlas: a resource of catalytic sites and residues identified in enzymes using structural data. *Nucleic acids research*, *32*(suppl 1), D129-D133.
28. UniProt Consortium. (2008). The universal protein resource (UniProt). *Nucleic acids research*, *36*(suppl 1), D190-D195.
29. Bateman, A., Coin, L., Durbin, R., Finn, R. D., Hollich, V., Griffiths-Jones, S., ... & Studholme, D. J. (2004). The Pfam protein families database. *Nucleic acids research*, *32*(suppl 1), D138-D141.
30. Pearl, F., Todd, A., Sillitoe, I., Dibley, M., Redfern, O., Lewis, T., ... & Akpor, A. (2005). The CATH Domain Structure Database and related resources Gene3D and DHS provide comprehensive domain family information for genome analysis. *Nucleic acids research*, *33*(suppl 1), D247-D251.
31. Velankar, S., Dana, J. M., Jacobsen, J., van Ginkel, G., Gane, P. J., Luo, J., ... & Kleywegt, G. J. (2012). SIFTS: structure integration with function, taxonomy and sequences resource. *Nucleic acids research*, gks1258.
32. Gaulton, A., Bellis, L. J., Bento, A. P., Chambers, J., Davies, M., Hersey, A., ... & Overington, J. P. (2012). ChEMBL: a large-scale bioactivity database for drug discovery. *Nucleic acids research*, *40*(D1), D1100-D1107.
33. Irwin, J. J., & Shoichet, B. K. (2005). ZINC-a free database of commercially available compounds for virtual screening. *Journal of chemical information and modeling*, *45*(1), 177-182.

34. Demšar, J., Curk, T., Erjavec, A., Gorup, Č., Hočevar, T., Milutinovič, M., ... & Štajdohar, M. (2013). Orange: data mining toolbox in Python. *Journal of Machine Learning Research*, *14*(1), 2349-2353.
35. Goodstadt, L. (2010). Ruffus: a lightweight Python library for computational pipelines. *Bioinformatics*, *26*(21), 2778-2779.
36. Rumbaugh, J., Jacobson, I., & Booch, G. (2004). *Unified Modeling Language Reference Manual, The*. Pearson Higher Education.
37. MySQL, A. B. (2001). MySQL reference manual.
38. Van Rossum, G. (2007, June). Python Programming Language. In *USENIX Annual Technical Conference* (Vol. 41).
39. Eclipse, I. D. E. (2009). for JAVA Developers. *URL http://www. eclipse. org/.–2008*.
40. Bray, T., Paoli, J., Sperberg-McQueen, C. M., Maler, E., & Yergeau, F. (1998). Extensible markup language (XML). *World Wide Web Consortium Recommendation REC-xml-19980210. http://www. w3. org/TR/1998/REC-xml-19980210*, *16*, 16.
41. Le Guilloux, V., Schmidtke, P., & Tuffery, P. (2009). Fpocket: an open source platform for ligand pocket detection. *BMC bioinformatics*, *10*(1), 168.
42. Li, W., & Godzik, A. (2006). Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, *22*(13), 1658-1659.
43. Eddy, S. (2010). HMMER3: a new generation of sequence homology search software. *URL: http://hmmer. janelia. Org*.
44. Schymkowitz, J., Borg, J., Stricher, F., Nys, R., Rousseau, F., & Serrano, L. (2005). The FoldX web server: an online force field. *Nucleic acids research*,*33*(suppl 2), W382-W388.
45. Owens, M., & Allen, G. (2010). *SQLite*. Apress LP.
46. www.mongodb.com
47. Cock, P. J., Antao, T., Chang, J. T., Chapman, B. A., Cox, C. J., Dalke, A., ... & de Hoon, M. J. (2009). Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, *25*(11), 1422-1423.
48. Holland, R. C., Down, T. A., Pocock, M., Prlić, A., Huen, D., James, K., ... & Schreiber, M. J. (2008). BioJava: an open-source framework for bioinformatics.*Bioinformatics*, *24*(18), 2096-2097.
49. Stajich, J. E., Block, D., Boulez, K., Brenner, S. E., Chervitz, S. A., Dagdigian, C., ... & Lehväslaiho, H. (2002). The Bioperl toolkit: Perl modules for the life sciences. *Genome research*, *12*(10), 1611-1618.
50. Johnson, R., Hoeller, J., Arendsen, A., Sampaleanu, C., Davison, D., Kopylenko, D., ... & Harro, R. (2004). Spring-Java/J2EE application framework.*Reference Documentation, Version*, *1*(7), 265-278.
51. Hellkamp, M. (2012). Bottle: Python web framework.
52. Raggett, D., Le Hors, A., & Jacobs, I. (1999). HTML 4.01 Specification. *W3C recommendation*, *24*.
53. Bodin, M., Charguéraud, A., Filaretti, D., Gardner, P., Maffeis, S., Naudziuniene, D., ... & Smith, G. (2014). A trusted mechanised JavaScript specification. *ACM SIGPLAN Notices*, *49*(1), 87-100.
54. Bos, B., Çelik, T., Hickson, I., & Lie, H. W. (2005). Cascading style sheets level 2 revision 1 (css 2.1) specification. *W3C working draft, W3C, June*.
55. Lerner, R. M. (2012). At the forge: twitter bootstrap. *Linux Journal*, *2012*(218), 6.

56. Henikoff, S., & Henikoff, J. G. (1992). Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences*, *89*(22), 10915-10919.
57. Altschul, S. F., Gish, W., Miller, W., Myers, E. W., & Lipman, D. J. (1990). Basic local alignment search tool. *Journal of molecular biology*, *215*(3), 403-410.
58. Smith, T. F., & Waterman, M. S. (1981). Comparison of biosequences.*Advances in applied mathematics*, *2*(4), 482-489.
59. Thomsen, M. C. F., & Nielsen, M. (2012). Seq2Logo: a method for construction and visualization of amino acid binding motifs and sequence profiles including sequence weighting, pseudo counts and two-sided representation of amino acid enrichment and depletion. *Nucleic acids research*, *40*(W1), W281-W287.
60. Bairoch, A., & Apweiler, R. (2000). The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. *Nucleic acids research*, *28*(1), 45-48.
61. Brooks, B. R., Bruccoleri, R. E., Olafson, B. D., States, D. J., Swaminathan, S., & Karplus, M. (1983). CHARMM: a program for macromolecular energy, minimization, and dynamics calculations. *Journal of computational chemistry*,*4*(2), 187-217.
62. Shen, M. Y., & Sali, A. (2006). Statistical potential for assessment and prediction of protein structures. *Protein science*, *15*(11), 2507-2524.
63. Weininger, D. (1988). SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, *28*(1), 31-36.
64. O'Boyle, N. M., Banck, M., James, C. A., Morley, C., Vandermeersch, T., & Hutchison, G. R. (2011). Open Babel: An open chemical toolbox. *Journal of cheminformatics*, *3*(1), 1.
65. Bajusz, D., Rácz, A., & Héberger, K. (2015). Why is Tanimoto index an appropriate choice for fingerprint-based similarity calculations?. *Journal of cheminformatics*, *7*(1), 1.
66. McNaught, A. (2006). The iupac international chemical identifier. *Chemistry international*, 12-14.
67. Pletnev, I., Erin, A., McNaught, A., Blinov, K., Tchekhovskoi, D., & Heller, S. (2012). InChIKey collision resistance: an experimental testing. *Journal of cheminformatics*, *4*(1), 1.
68. Ruiz-Carmona, S., Alvarez-Garcia, D., Foloppe, N., Garmendia-Doval, A. B., Juhos, S., Schmidtke, P., ... & Morley, S. D. (2014). rDock: a fast, versatile and open source program for docking ligands to proteins and nucleic acids. *PLoS Comput Biol*, *10*(4), e1003571.
69. Csizmadia, F. (2000). JChem: Java applets and modules supporting chemical database handling from web browsers. *Journal of Chemical Information and Computer Sciences*, *40*(2), 323-324.
70. Mayo, S. L., Olafson, B. D., & Goddard, W. A. (1990). DREIDING: a generic force field for molecular simulations. *Journal of Physical chemistry*, *94*(26), 8897-8909.
71. Holm, L., & Sander, C. (1999). Protein folds and families: sequence and structure alignments. *Nucleic acids research*, *27*(1), 244-247.
72. Liu, T., Lin, Y., Wen, X., Jorissen, R. N., & Gilson, M. K. (2007). BindingDB: a web-accessible database of experimentally determined protein–ligand binding affinities. *Nucleic acids research*, *35*(suppl 1), D198-D201.

73. Flanagan, S. E., Patch, A. M., & Ellard, S. (2010). Using SIFT and PolyPhen to predict loss-of-function and gain-of-function mutations. *Genetic testing and molecular biomarkers*, *14*(4), 533-537.
74. Scott, S. A., Edelmann, L., Kornreich, R., & Desnick, R. J. (2008). Warfarin pharmacogenetics: CYP2C9 and VKORC1 genotypes predict different sensitivity and resistance frequencies in the Ashkenazi and Sephardi Jewish populations. *The American Journal of Human Genetics*, *82*(2), 495-500.
75. Roses, A. D. (2000). Pharmacogenetics and the practice of medicine. *Nature*,*405*(6788), 857-865.
76. Aitman, T. J., Cooper, L. D., Norsworthy, P. J., Wahid, F. N., Gray, J. K., Curtis, B. R., ... & Hill, A. V. (2000). Population genetics: Malaria susceptibility and CD36 mutation. *Nature*, *405*(6790), 1015-1016.
77. Huang, N., Agrawal, V., Giacomini, K. M., & Miller, W. L. (2008). Genetics of P450 oxidoreductase: sequence variation in 842 individuals of four ethnicities and activities of 15 missense mutations. *Proceedings of the National Academy of Sciences*, *105*(5), 1733-1738.
78. Sherry, S. T., Ward, M. H., Kholodov, M., Baker, J., Phan, L., Smigielski, E. M., & Sirotkin, K. (2001). dbSNP: the NCBI database of genetic variation. *Nucleic acids research*, *29*(1), 308-311.
79. Landrum, M. J., Lee, J. M., Riley, G. R., Jang, W., Rubinstein, W. S., Church, D. M., & Maglott, D. R. (2014). ClinVar: public archive of relationships among sequence variation and human phenotype. *Nucleic acids research*, *42*(D1), D980-D985.
80. Lappalainen, I., Lopez, J., Skipper, L., Hefferon, T., Spalding, J. D., Garner, J., ... & Paschall, J. (2013). DbVar and DGVa: public archives for genomic structural variation. *Nucleic acids research*, *41*(D1), D936-D941.
81. Mottaz, A., David, F. P., Veuthey, A. L., & Yip, Y. L. (2010). Easy retrieval of single amino-acid polymorphisms and phenotype information using SwissVar.*Bioinformatics*, *26*(6), 851-852.
82. Szmant, H. H. (1989). *Organic building blocks of the chemical industry*. John Wiley & Sons.
83. Lucas, X., Grüning, B. A., Bleher, S., & Günther, S. (2015). The purchasable chemical space: a detailed picture. *Journal of chemical information and modeling*, *55*(5), 915-924.
84. Caminero,J.A., Sotgiu,G., Zumla,A. et al. (2010) Best drug treatment for multidrug-resistant and extensively drug-resistant tuberculosis. Lancet Infec. Dis., 10, 621–629.
85. Reddy,T., Riley,R., Wymore,F. et al. (2009) TB database: an ntegrated platform for tuberculosis research. Nucleic Acids Res., 37, D499–D508.
86. Schilling, C.H., Schuster, S., Palsson, B.O. et al. (1999) Metabolic pathway analysis: Basic concepts and scientific applications in the post-genomic era. Biotechnol. Prog., 15, 296–303.
87. Aguero,F., Al-Lazikani,B., Aslett,M. et al. (2008) Genomic-scale prioritization of drug targets: the TDR Targets database. Nat. Rev. Drug Discov., 7, 900–907.
88. Jamshidi,N. and Palsson,B.Ø. (2007) Investigating the metabolic capabilities of Mycobacterium tuberculosis H37Rv using the insilico strain iNJ661 and proposing alternative drug targets. BMC Syst. Biol., 1, 26.

89. Hasan,S., Daugelat,S., Rao,P.S.S. et al. (2006) Prioritizing genomic drug targets in pathogens: application to Mycobacterium tuberculosis. PLoS Comput. Biol., 2, 0539–0550.
90. Sassetti,C.M. and Rubin,E.J. (2003) Genetic requirements for mycobacterial survival during infection. Proc. Natl Acad. Sci. USA, 100, 12989–12994.
91. Rengarajan,J., Bloom,B.R. and Rubin,E.J. (2005) Genome-wide requirements for Mycobacterium tuberculosis adaptation and survival in macrophages. Proc. Natl Acad. Sci. USA, 102, 8327–8332.
92. Radusky, L., Defelipe, L. A., Lanzarotti, E., Luque, J., Barril, X., Marti, M. A., & Turjanski, A. G. (2014). TuberQ: a Mycobacterium tuberculosis protein druggability database. *Database*, *2014*, bau035.
93. Halgren,T.A. (2009) Identifying and characterizing binding sites and assessing druggability. J. Chem. Inf. Model., 49, 377–389.
94. Sassetti,C.M., Boyd,D.H. and Rubin,E.J. (2003) Genes required for mycobacterial growth defined by high density mutagenesis. Mol. Microbiol., 48, 77–84.
95. Nakane, T. (2014). GLmol-Molecular Viewer on WebGL/Javascript, Version 0.47.
96. DeLano, W. L. (2002). The PyMOL molecular graphics system.
97. Humphrey, W., Dalke, A., & Schulten, K. (1996). VMD: visual molecular dynamics. *Journal of molecular graphics*, *14*(1), 33-38.
98. Ouellet, H., Johnston, J. B., & de Montellano, P. R. O. (2010). The Mycobacterium tuberculosis cytochrome P450 system. *Archives of biochemistry and biophysics*, *493*(1), 82-95.
99. Lyle, T. A., Chen, Z., Appleby, S. D., Freidinger, R. M., Gardell, S. J., Lewis, S. D., ... & Ng, A. S. (1997). Synthesis, evaluation, and crystallographic analysis of L-371,912: A potent and selective active-site thrombin inhibitor. *Bioorganic & Medicinal Chemistry Letters*, *7*(1), 67-72.
100. Jaeger, T., Budde, H., Flohé, L., Menge, U., Singh, M., Trujillo, M., & Radi, R. (2004). Multiple thioredoxin-mediated routes to detoxify hydroperoxides in Mycobacterium tuberculosis. *Archives of biochemistry and biophysics*, *423*(1), 182-191.
101. Defelipe, L. A., Do Porto, D. F., Ramos, P. I. P., Nicolás, M. F., Sosa, E., Radusky, L., ... & Marti, M. A. (2016). A whole genome bioinformatic approach to determine potential latent phase specific targets in Mycobacterium tuberculosis. *Tuberculosis*, *97*, 181-192.
102. Kendall, S. L., Movahedzadeh, F., Rison, S. C. G., Wernisch, L., Parish, T., Duncan, K., ... & Stoker, N. G. (2004). The Mycobacterium tuberculosis dosRS two-component system is induced by multiple stresses. *Tuberculosis*, *84*(3), 247-255.
103. Rawat, R., Whitty, A., & Tonge, P. J. (2003). The isoniazid-NAD adduct is a slow, tight-binding inhibitor of InhA, the Mycobacterium tuberculosis enoyl reductase: adduct affinity and drug resistance. *Proceedings of the National Academy of Sciences*, *100*(24), 13881-13886.
104. Radusky, L. G., Hassan, S. S., Lanzarotti, E., Tiwari, S., Jamal, S. B., Ali, J., ... & Turjanski, A. G. (2015). An integrated structural proteomics approach along the druggable genome of Corynebacterium pseudotuberculosis species for putative druggable targets. *BMC genomics*, *16*(5), 1.

105. Lipinski, C. A. (2004). Lead-and drug-like compounds: the rule-of-five revolution. *Drug Discovery Today: Technologies*, *1*(4), 337-341.
106. Lyne, P. D. (2002). Structure-based virtual screening: an overview. *Drug discovery today*, *7*(20), 1047-1055.
107. Morley, S. D., & Afshar, M. (2004). Validation of an empirical RNA-ligand scoring function for fast flexible docking using RiboDock®. *Journal of computer-aided molecular design*, *18*(3), 189-208.
108. Nelder, J. A., & Mead, R. (1965). A simplex method for function minimization.*The computer journal*, *7*(4), 308-313.
109. Mysinger, M. M., Carchia, M., Irwin, J. J., & Shoichet, B. K. (2012). Directory of useful decoys, enhanced (DUD-E): better ligands and decoys for better benchmarking. *Journal of medicinal chemistry*, *55*(14), 6582-6594.
110. Hamburg, M. A., & Collins, F. S. (2010). The path to personalized medicine.*New England Journal of Medicine*, *363*(4), 301-304.
111. Kiel, C., & Serrano, L. (2014). Structure-energy-based predictions and network modelling of RASopathy and cancer missense mutations. *Molecular systems biology*, *10*(5), 727.
112. Kiel, C., Vogt, A., Campagna, A., Chatr-aryamontri, A., Swiatek-de Lange, M., Beer, M., ... & Serrano, L. (2011). Structural and functional protein network analyses predict novel signaling functions for rhodopsin. *Molecular systems biology*, *7*(1), 551.
113. Stein, A., Russell, R. B., & Aloy, P. (2005). 3did: interacting protein domains of known three-dimensional structure. *Nucleic acids research*, *33*(suppl 1), D413-D417.
114. Fernandez-Escamilla, A. M., Rousseau, F., Schymkowitz, J., & Serrano, L. (2004). Prediction of sequence-dependent and mutational effects on the aggregation of peptides and proteins. *Nature biotechnology*, *22*(10), 1302-1306.
115. Diaz, C., Corentin, H., Thierry, V., Chantal, A., Tanguy, B., David, S., ... & Edgardo, F. (2014). Virtual screening on an α-helix to β-strand switchable region of the FGFR2 extracellular domain revealed positive and negative modulators. *Proteins: Structure, Function, and Bioinformatics*, *82*(11), 2982-2997.
116. Bamford, S., Dawson, E., Forbes, S., Clements, J., Pettett, R., Dogan, A., ... & Wooster, R. (2004). The COSMIC (Catalogue of Somatic Mutations in Cancer) database and website. *British journal of cancer*, *91*(2), 355-358.