WANDRÉ NUNES DE PINHO VELOSO

EASYVS: UMA FERRAMENTA PARA TRIAGEM VIRTUAL MISTA BASEADA EM ALVO E LIGANTE

Belo Horizonte
Julho de 2019

WANDRÉ NUNES DE PINHO VELOSO

EASYVS: UMA FERRAMENTA PARA TRIAGEM VIRTUAL MISTA BASEADA EM ALVO E LIGANTE

Tese apresentada ao Programa de Pós-Graduação em Bioinformática do Instituto de Ciências Biológicas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Doutor em Bioinformática.

Orientadores: Dr. Carlos Henrique da Silveira e Dr. Douglas Eduardo Valente Pires

Belo Horizonte

Julho de 2019

Nunes de Pinho Veloso, Wandré

EasyVS: Uma ferramenta para triagem virtual mista baseada em alvo e ligante / Wandré Nunes de Pinho Veloso. — Belo Horizonte, 2019
xxiii, 72 f. : il. ; 29cm

Tese (doutorado) — Universidade Federal de Minas Gerais
Orientadores: Dr. Carlos Henrique da Silveira e Dr. Douglas Eduardo Valente Pires

1. Biologia Computacional — Teses. 2. Bioinformática — Teses. I. Orientador. II. Título.

CDU

Universidade Federal de Minas Gerais
Instituto de Ciências Biológicas
Programa Interunidades de Pós-Graduação em Bioinformática da UFMG
Avenida Presidente Antônio Carlos, 6627 – Pampulha
31270-901 - Belo Horizonte – MG
Endereço eletrônico: bioinfo@icb.ufmg.br 55 31 3409-2554

"EASYVS: UMA FERRAMENTA PARA TRIAGEM VIRTUAL MISTA BASEADA EM ALVO E LIGANTE"

Wandré Nunes de Pinho Veloso

Tese aprovada pela banca examinadora constituída pelos Professores:

Prof. Carlos Henrique da Silveira - Orientador
UNIFEI

Prof. Lucas Bleicher
UFMG

Profa. Sabrina de Azevedo Silveira
UFV

Prof. Daniel Cristian Ferreira Soares
UNIFEI

Prof. Leonardo Henrique Franca de Lima
UFSJ

Lucianna Helene Silva dos Santos
UFMG

Belo Horizonte, 26 de julho de 2019.

Agradecimentos

Agradeço, primeiramente, a Deus, Fonte de Amor, Misericórdia e Consolo, pelo dom da vida, pelas dificuldades, conquistas e por nunca me abandonar.

À Virgem Maria, Mãe de Deus e minha, por me envolver em seu colo materno em todos os momentos.

À minha esposa Mariana, companheira durante todo esse período de estudo e trabalho, que me incentiva e me leva a Deus nas pequenas e grandes atitudes. Eu amo você.

À nossa pequena Alana que sabe, nas brincadeiras e sorrisos, por me permitir viver a paternidade todos os dias, me mostrar quão feliz pode ser a vida e o que é essencial.

Aos meus queridos pais, Waldir e Dária, meus irmãos, Duran (Rose e Tiago) e Giowana, grandes influenciadores do meu caráter, presentes em minha vida e sempre torcendo por mim. À minha sogra Socorro, Mônica, Anderson (Ana Júlia e Samuel), Éder e Pri (e Maria Gabriela) por também serem minha família em todos os sentidos.

Agradeço também a meus orientadores, Carlos e Douglas, pela confiança, amizade e ensinamentos, sempre obtidos a partir de calmas e detalhadas explicações. Aproveito para agradecer ao David Ascher, da Universidade de Melbourne, pela parceria no desenvolvimento do artigo e do sistema como um todo.

A Iara, José Cornélio e filhos, que me acolheram e acolhem como parte da família em inúmeros momentos. Deus os abençoe sempre!

Aos meus amigos e colegas de pesquisa que tanto me ajudaram durante esse processo. Meu companheiro/irmão de doutorado desde o primeiro dia, Biharck Muniz; à competente designer e ótima pessoa Pâmela Marinho; a João Linhares, pelos *feedbacks* durante o processo de desenvolvimento do EasyVS; a Joicy, querida amiga, madrinha de casamento e que ajudou esse trabalho em diversos pontos e momentos; a Vianei, pela amizade e os inúmeros suportes; a Francislon ("meu rei") pelo apoio no desenvolvimento do trabalho; e a Christian e Douglas, pelo apoio no desenvolvimento do novo código de agrupamento molecular. Aos amigos grandes Marielle, Neto (e João Pedro), Luciana,

Bruno (Lucas e Renata), Vivi e Rusbney, Carol e João, por estarem perto sempre.

Aos meus Professores, grandes incentivadores da minha profissão atual; e aos meus alunos, por me fazerem querer e ser melhor, mais humano e profissional a cada dia.

À UNIFEI, por me possibilitar um período de afastamento e também por hospedar o EasyVS em uma estrutura dedicada.

"Depois desse trabalho feito, depois da obra terminada, depois que toda essa vontade fica desacorrentada... Depois de ter o bom combate, depois de dar o xeque-mate, depois que a avidez se acaba e o cansaço nos abate. Restará depois da chuva forte os pedaços das palavras doces, semeados no fundo de cada coração. Restarão também canções perdidas nos olhares dessa despedida e os excessos de nós mesmos pelo chão"

(Maninho)

Resumo

O processo de descoberta de fármacos consiste em diversas etapas e a utilização de ferramentas computacionais pode ser parte essencial, em especial, durante os primeiros estágios da pesquisa. É possível, por exemplo, o estudo de grande número de moléculas a partir de propriedades fisico-químicas ou mesmo prevendo o possível modo e energia de ligação entre duas ou mais moléculas. O EasyVS é uma ferramenta WEB desenvolvida para simplificar o processo descrito anteriormente a partir da seleção de compostos e bibliotecas de compostos e triagem virtual de ligantes. Com uma interface intuitiva, a ferramenta permite aos usuários ir desde a seleção de uma proteína-alvo com estrutura conhecida, passando pela parametrização do *docking* até sua execução e visualização de resultados, em poucos *clicks*. O EasyVS também permite aos usuários a escolha dentre mais de 16 milhões de moléculas através de filtros baseados nas propriedades moleculares, assim como o agrupamento considerando índice de similaridade de Tanimoto. Essa ferramenta foi utilizada, em um estudo de caso, para a triagem virtual de ligantes para as GPCR (receptores acoplados a proteínas G) e apresentou resultados satisfatórios, indicando moléculas já conhecidas como ligantes, separando-as de *decoys*, assim como novos potenciais ligantes para serem estudados. Esse estudo mostrou-se relevante pois as GPCRs são consideradas a maior família de alvos para fármacos aprovados. Pretende-se aumentar a capacidade de processamento do servidor disponível para o EasyVS a fim de diminuir o tempo de computação das requisições, além de viabilizar novas funcionalidades para o EasyVS.

Abstract

The drug discovery process consists of several steps and the use of computational tools can be an essential part, especially during the early research stages. It is possible, for example, to study large numbers of molecules from various compound libraries from physico-chemical properties or even by predicting the mode and energy of binding between two or more molecules. EasyVS is a web tool developed to simplify the previously described process from the selection of compounds libraries and virtual screening of ligands. With an intuitive interface, this tool allows users to go from the selection of a protein target with known structure, through docking parameterization to its execution and results visualization, in a few clicks. EasyVS also allows users to choose from more than 16 million molecules through filters based on molecular properties, as well as the grouping through Tanimoto's similarity index. This tool was used, in a case study, for the virtual screening of ligands for GPCR (G protein coupled receptors) and presented satisfactory results, indicating molecules already known as ligands, separating them from *decoys*, as well as new potential ligands to be studied. This study proved to be relevant since GPCRs are considered the largest family of targets for approved drugs. We intend to increase the processing capacity of the server available for EasyVS in order to reduce the computation time of the requisitions, besides making viable new functionalities for EasyVS.

Lista de Figuras

Lista de Tabelas

Sumário

ADMET Absorção, Distribuição, Metabolismo, Excreção e Toxicidade

ANVISA Agência Nacional de Vigilância Sanitária

GPCR *G Protein-Coupled Receptors*

HTS *High Throughput Screening*

IFA Insumo Farmacêutico Ativo

InChI *IUPAC International Chemical Identifier* ou *International Chemical Identifier*

IUPAC *International Union of Pure and Applied Chemistry*

Kd *Dissociation Constant* ou Constante de dissociação

LBVS *Ligand-Based Virtual Screening*

NMR *Nuclear Magnetic Resonance*

PDB *Protein Data Bank*

RMSD *Root Mean Square deviation*

SDF *Structure-Data File*

SMILES *Simplified Molecular Input Line Entry Specification*

TBVS *Target-Based Virtual Screening*

Capítulo 1

Introdução

Para a Agência Nacional de Vigilância Sanitária (ANVISA), um fármaco, também denominado Insumo Farmacêutico Ativo (IFA), é uma substância química ativa ou matéria-prima que tenha "propriedades farmacológicas com finalidade medicamentosa, utilizada para diagnóstico, alívio ou tratamento, empregada para modificar ou explorar sistemas fisiológicos ou estados patológicos, em benefício da pessoa na qual se administra"[ANVISA, 2018]. Já farmacóforo, segundo *International Union of Pure and Applied Chemistry* (IUPAC) é uma molécula ou grupamento funcional com um conjunto de características eletrônicas e estéricas que são necessárias para assegurar as interações supramoleculares ótimas com um alvo biológico específico e para acionar (ou bloquear) sua resposta biológica (WERMUTH et al, 1998, *apud* Cortes-Cabrera et al. [2016]). Dentre essas características químicas, as que são mais comumente levadas em consideração são a quantidade de doadores e aceptores de hidrogênio, grupos carregados positiva ou negativamente, anéis aromáticos, grupos (hidrofóbicos) alifáticos/aromáticos (LEACH; GILLET, 2007, *apud* Cortes-Cabrera et al. [2016]).

Há muitas formas de se projetar um fármaco, seja por processos de descoberta de substâncias já existentes na natureza, como a prospecção de princípios ativos em produtos naturais ou minerais, seja por processos de invenção ou criação de novos fármacos, sintéticos, como seria o caso dos métodos de química combinatória e de desenho racional de fármacos [Keseru & Makara, 2006]. Na prática, descoberta e invenção confundem-se em um amplo e complexo processo conhecido mundialmente como "*drug discovery*".

Independentemente do método, todos podem didaticamente ser alicerçados em um tripé: ligante, alvo, função [Pitt et al., 2009]. Mais comumente, o ligante é uma pequena molécula orgânica (massa molecular < 500 Da[1]) capaz de interagir com al-

[1]Dalton é definido como 1/12 da massa de um carbono-12 em seu estado fundamental

vos biológicos, normalmente um receptor ou região invaginada na superfície de uma macromolécula (podendo, ou não, ser o *pocket* ou sítio ativo da biomolécula) ou em complexos de macromoléculas [Pitt et al., 2009]. De acordo com Nelson & Cox [2008] e Daintith [2008], o sítio ativo, catalítico ou alostérico (em inglês, *binding site*) é a região de uma enzima em que as moléculas influenciadas por ela se ligam ou se combinam e, geralmente, é complementar ao ligante em tamanho, forma e propriedades físico-químicas. Essa interação produz efeitos que se desdobram em alterações funcionais em diferentes escalas e condições: moleculares, celulares, fisiológicas, microbiológicas etc. Em termos práticos, tais mudanças funcionais, podem levar não somente ao diagnóstico e terapêutica de doenças (sejam humanas ou não), como também impactar contextos não relacionados diretamente a uma enfermidade, como, por exemplo, em nutrição, cosméticos, fertilizantes, processos bioquímicos industriais e saneamento.

A despeito de todas essas possibilidades de intervenção, o caminho que leva um ligante a se transformar em um fármaco não é simples e pode custar, de acordo com Avorn [2015], mais que 2,6 bilhões de dólares (sendo cerca de 1,2 bilhões são utilizados para capital e 1,4 bilhões para pesquisa de novos fármacos). Candidatos a fármacos, pré-selecionados nas complexas etapas de descoberta e invenção, têm que enfrentar todo um conjunto de etapas até chegar às prateleiras das farmácias. Normalmente, a jornada começa com a triagem de ligantes (do inglês *hits*) que demonstraram afinidade a alvos que possam ter envolvimento na bioatividade requerida. Nos casos em que os alvos não são de todo conhecidos, a triagem pode ter selecionado ligantes que tenham ou possam ter (no caso da triagem virtual ou *virtual screening*) correlação com o efeito funcional desejado. Seja como for, esses ligantes são apenas uma esperança como potenciais fármacos. Estudos experimentais, auxiliados por análises computacionais, precisam avaliar sua efetividade no contexto biológico, pela aferição de propriedades como: potência, seletividade (geralmente, por meio de estudos da Relação Estrutura-Atividade - REA), toxicidade e eficácia *in vivo* [Keseru & Makara, 2006]. Uma das vantagens do *virtual screening* é que podem ser estudados e testados, computacionalmente, componentes ainda não disponíveis fisicamente, além de possibilitar o uso de grandes bibliotecas de compostos.

Durante todo o processo, esses ligantes candidatos podem ter suas composições e estruturas reajustadas para otimizar tais propriedades, transformando-se em moléculas líderes (do inglês *leads*). Como tais, estão, em tese, preparadas para os primeiros testes "pré-clínicos", feitos em animais. Nessa fase, as propriedades citadas anteriormente são refinadas mais ainda, principalmente as características farmacocinéticas (Absorção, Distribuição, Metabolismo, Excreção e Toxicidade (ADMET)), o que pode exigir novos ciclos de otimizações na composição e estrutura das moléculas líderes [Pitt et al., 2009].

Após essas etapas, haverá alguns poucos candidatos a fármacos preparados para as fases clínicas em humanos. Na fase I, eles são administrados a um grupo de humanos sadios, para avaliações de toxicidade e segurança. Na fase II, são testados em poucos humanos doentes, para aferir a eficácia do candidato a fármaco em condições patológicas. Na fase III, amplia-se o número de humanos testados, bem como a coleta de dados sobre sua efetividade e segurança, para efeito de submissão às agências reguladoras. Se aprovado, poderá ser comercializado, e uma fase IV acompanhará os efeitos do seu uso generalizado por amostragens [Pitt et al., 2009]. Esse processo é representado na Figura 1.1.

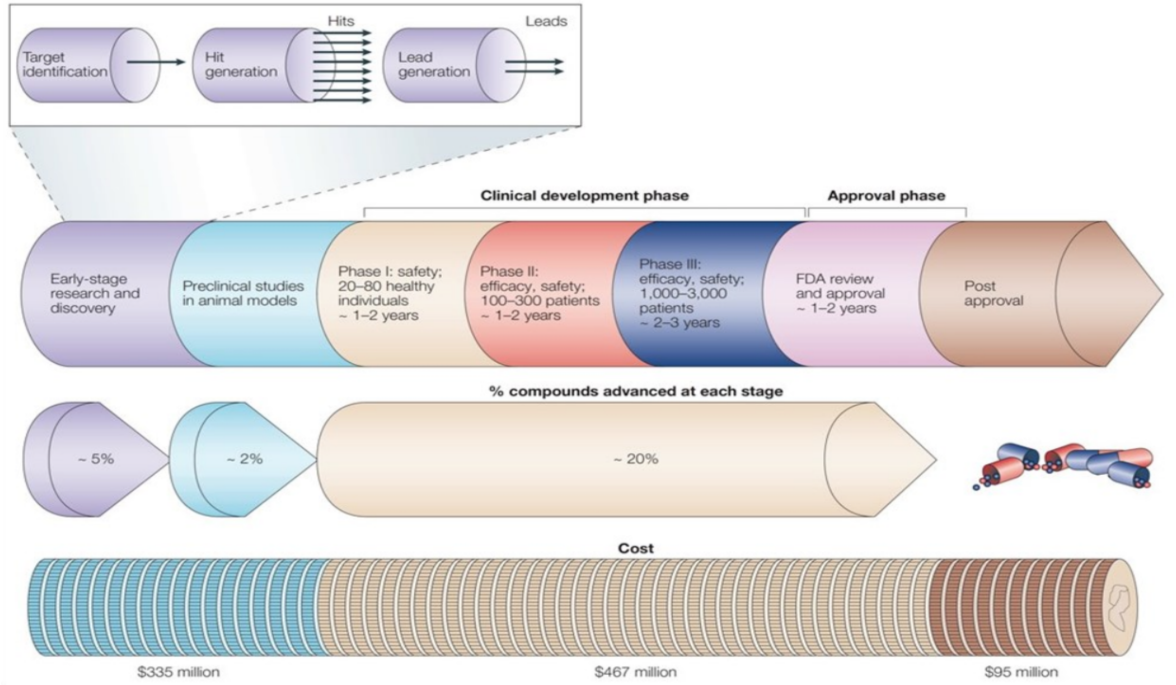

Figura 1.1. Fases no processo de *drug discovery*. Fonte: [Doganova, 2015, p.23].

Uma parte importante em todo esse processo é a fase inicial, que envolve a triagem dos ligantes e alvos. Se bem selecionados, minimizam as probabilidades de rejeição nas fases finais. O padrão da indústria farmacêutica tem sido o *High Throughput Screening* (HTS), no qual milhões de moléculas podem ser avaliadas em experimentos de larga escala ([Mayr & Fuerst, 2008] e [Bologa et al., 2019]). Mas, além da baixa taxa de sucesso, seu custo por candidato selecionado é muito elevado [Mayr & Fuerst, 2008]. Nesse sentido, a pesquisa em bioinformática estrutural e quimioinformática em uma triagem "virtual"(*in silico*) tem recebido atenção crescente como uma metodologia, de menor custo relativo, auxiliar a esse processo [Bielska et al., 2011].

É importante salientar que ferramentas *in silico* facilitam os estudos de um grande

número de moléculas, uma vez que, com o uso de *softwares* adequados é possível estimar a interação entre proteínas e possíveis ligantes, evitando a utilização de mais recursos em pesquisa *in vitro* ou mesmo *in vivo*. Diante dessa realidade, grandes esforços vêm sendo dedicados na elaboração de uma ferramenta que auxilie na triagem virtual de ligantes e que possa ser utilizada a partir de uma máquina com acesso a internet, sem necessidade de instalação de *softwares* específicos ou mesmo a configuração a partir de um conjunto de complexos parâmetros.

O EasyVS, que é descrito nesse documento, utiliza uma biblioteca de compostos atualizada e provê um conjunto de ferramentas que auxiliam a pesquisa *in silico* para diferentes tipos de usuários, incluindo os que pouco dominam *softwares* de *docking* ou mesmo agrupamento molecular.

1.1 Organização da Tese

Os demais Capítulos estão divididos da seguinte forma: o Capítulo 2 são detalhadas definições sobre Triagem Virtual de Ligantes/ *Virtual Screening* e abordagens baseadas em ligantes, alvo e mista; o Capítulo 3 trás os objetivos gerais e específicos; o Capítulo 4 apresenta uma explicação das formas de utilização da ferramenta assim como detalhes do desenvolvimento do EasyVS, como as tecnologias utilizadas, agrupamento molecular, *docking* e descrição das bibliotecas de moléculas utilizadas; já o Capítulo 5 contém a descrição e resultados do estudo de caso da ferramenta utilizando moléculas ativas e *decoys* gerados pelo DUD-E [Mysinger et al., 2012] para proteínas classificadas como GPCRs; por fim, o Capítulo 6 apresenta as conclusões assim como trabalhos futuros.

Capítulo 2

Triagem Virtual de ligantes

Duas abordagens de triagem virtuais são comumente empregadas: as baseadas em alvos (*Target-Based Virtual Screening* - TBVS) e as baseadas em ligantes (*Ligand-Based Virtual Screening* - LBVS) [Domingues & Lopes, 2012]. Na primeira, os algoritmos de seleção de ligantes, dentre um conjunto de moléculas, dependem de informações sobre os alvos para estimar as probabilidades de interação; na segunda, não há essa dependência. No LBVS, descritores envolvendo os próprios ligantes são usados como atributos discriminantes nos algoritmos de seleção e classificação, tendo como base de conhecimento um conjunto de ligantes ativos e não ativos para determinada funcionalidade. Se há informação estrutural e experimental sobre o alvo, técnicas TBVS tendem a ser mais usadas, mas não necessariamente com mais sucesso [Domingues & Lopes, 2012]. Ademais, como já dito, nem sempre o alvo é bem caracterizado ou conhecido. Nesses casos, técnicas LBVS podem ser mais promissoras, ou mesmo ser a única opção.

2.1 TBVS

Vale lembrar que a abordagem *Target-Based Virtual Screening* (TBVS) se baseia no conhecimento da estrutura da proteína-alvo ou receptor para a execução dos processos para descoberta de ligantes.

TBVS é interessante, especialmente, quando se tem acesso aos dados necessários para o estudo como o sítio de ligação, também denominado *binding site* ou *pocket*, para a construção de um ligante no próprio sítio ativo ou a realização de um estudo do *pocket* para estimar, computacionalmente, a afinidade com possíveis ligantes.

A abordagem que realiza a construção do ligante no *pocket* pode ser executada a partir de, por exemplo, complementaridade geométrica e físico-química.

Já outra possibilidade, complementar ou não a outras técnicas, de acordo com Cortes-Cabrera et al. [2016], tem como principal técnica o atracamento ou ancoramento molecular, também conhecido como *docking*. O *docking* fornecerá novos parâmetros para a afinidade dos possíveis ligantes com a molécula-alvo [Verli, 2014].

2.1.1 *Docking*

Dada uma estrutura tridimensional de uma proteína ou molécula-alvo e a constituição molecular de uma pequena molécula, qual é o modo de ligação (*binding mode*) da pequena molécula, isso é, sua posição, orientação e conformação quando ligada à proteína? Essa é a questão fundamental do "*docking problem*"[Sotriffer, 2016].

O conceito do *docking* baseia-se no modelo chave-fechadura, proposto por Emil Fischer, em 1984 [Fischer, 1894], ou seja, duas moléculas se encaixam de forma complementar. Porém, diferentemente de uma chave e uma fechadura reais, as moléculas envolvidas podem não ser rígidas, tornando esse problema complexo, ainda mais pelo fato de as moléculas alvos e ligantes serem rodeadas por outras moléculas, como água e íons.

A Figura 2.1 mostra a molécula-alvo, representada através da estrutura de PDB ID 1BR5, com as poses resultantes do *docking* da molécula NEO (Neopterina). Como a pequena molécula é, geralmente, mais flexível que a proteína-alvo (no caso do *docking* exemplificado, a proteína-alvo é rígida), as diversas configurações sugeridas, distinguíveis por suas cores, rodeiam o *pocket* da molécula-alvo. Pode-se perceber na Figura 2.1 que a pequena molécula Neopterina, ligante cristalográfico disponível junto à estrutura PDB 1BR5, através de um algoritmo voltado para estimar a probabilidade ou energia de interação entre moléculas, tenta ser alocada no sítio ativo da proteína, que, nesse caso, é bicavitário.

Os elementos constituintes do *docking* são: uma representação estrutural das moléculas (alvo e ligante) que irão interagir; definição do *pocket* alvo; um algoritmo de busca para gerar poses de ligação significantes do ligante no *pocket* alvo definido; e um método de pontuação (*score*) para avaliar as configurações geradas e identificar as que mais bem correspondam à realidade. Conforme Walters et al. [1998], esse (o método de pontuação) é um dos maiores problemas em todo o processo de *Virtual Screening*. Nesse sentido, é imprescindível que o modo de ligação seja experimentalmente verificável.

Detalhando melhor esse processo, a preparação para o *docking* começa, geralmente, com a obtenção da estrutura 3D do alvo e/ou ligante por técnicas como *Nuclear Magnetic Resonance* (NMR), cristalografia por difração de raios-X ou de nêutrons, a partir de repositórios como o RCSB PDB - *Protein Data Bank* [Berman et al., 2000]

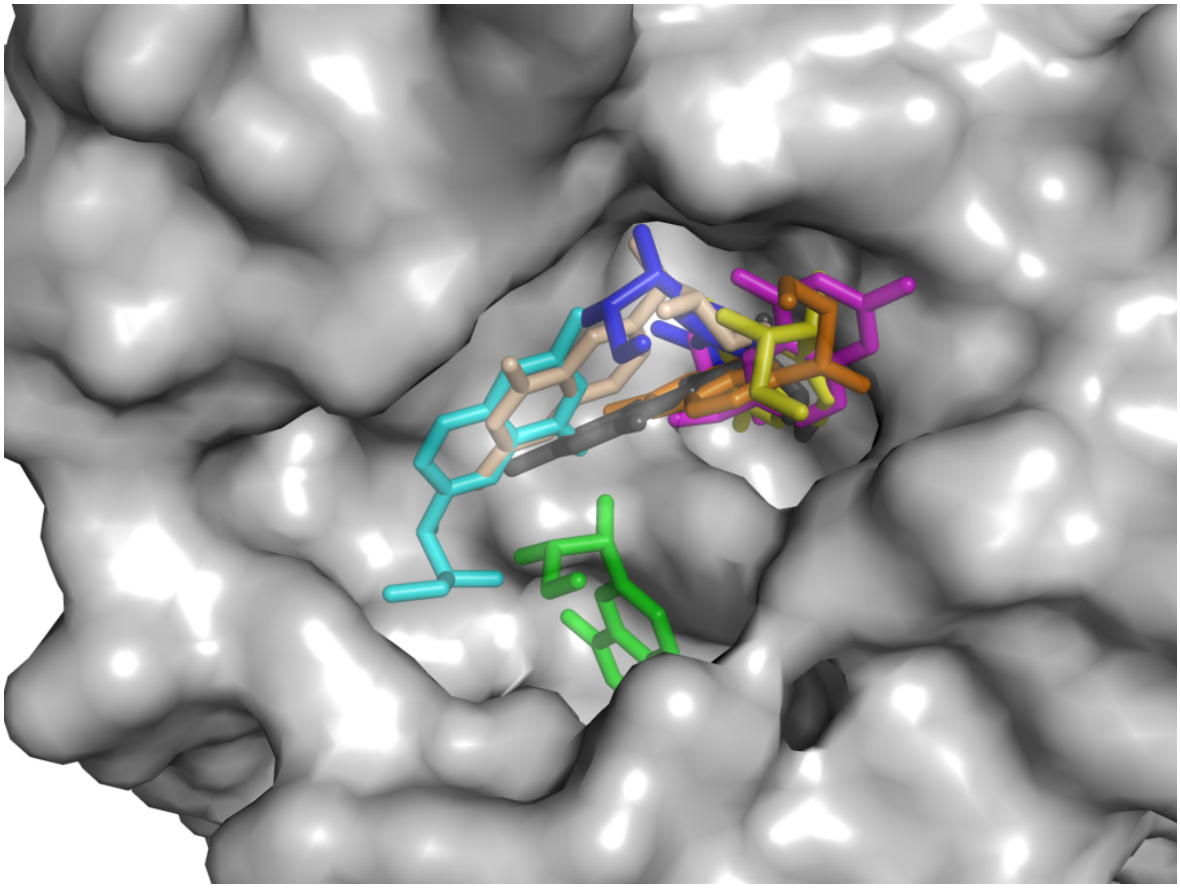

Figura 2.1. *Pocket* da cadeia A da Ricina (PDB ID 1BR5) com poses resultantes de *docking* com o ligante Neopterina.

ou mesmo a partir da criação/modelagem da estrutura em si. Sabe-se, porém, que é necessário atentar para a qualidade da estrutura alvo. No caso da cristalografia, é interessante checar a resolução e outros parâmetros que aferem a qualidade dos dados gerados, verificar resíduos ou átomos não presentes no mapa de densidade eletrônica, baixa ocupância e conformações alternativas. Também facilita o processo de *docking* se a área de busca do algoritmo for limitada a uma região específica (preferencialmente, no sítio ativo). Porém, pode-se realizar o *docking* de toda a superfície da proteína (denominado *blind docking*), embora seja muito custoso computacionalmente e não seja a opção ideal. Vale salientar que muitas estruturas PDBs apresentam moléculas de água na superfície que podem ou não obstruir o *pocket* e, logo, há a opção de se remover essas moléculas. Também, na cristalografia por difração de raios-X, em geral não é possível visualizar átomos de hidrogênio, podendo ser adicionados por programas específicos, que levam em conta também o pH e contexto químico-estrutural do alvo a ser protonado.

Com relação ao ligante, é comumente feita a análise atômica mais detalhada.

Como, geralmente, os possíveis ligantes são obtidos de uma base de dados somente com a representação atômica em duas dimensões, é necessária a conversão para o modelo tridimensional, em suas diversas conformações possíveis ou mais prováveis.

Em relação aos algoritmos, o problema de *docking* é, em essência, de otimização [Walters et al., 1998]. Uma vez dadas as estruturas individuais, o objetivo é encontrar uma combinação de complexos alvo-ligantes que tenham a melhor pontuação ou *score*.

Esse seria um simples problema de minimização se as funções de *score* fossem ideais. Os métodos de avaliação da energia livre de ligação objetivam otimizar a posição, orientação e conformação do ligante baseados em uma função objetiva, que leva em consideração as interações físico-químicas e adequações geométricas entre as moléculas.

As principais interações intermoleculares envolvidas nos complexos alvo-ligante são, conforme Verli [2014]: ligações de hidrogênio, interações de van der Waals, iônicas, empilhamentos aromáticos, cátion-Pi e coordenação com íons metálicos. No caso do programa utilizado para o *docking* nesta tese, o AutoDock Vina [Trott & Olson, 2010], o *score* é gerado levando em consideração campos de força, que podem ser calculados a partir de potenciais simplificados e aproximados de van der Waals e eletrostáticos, além de outros termos, assumindo sistemas massa-mola em energias de ligação, angulares, desvios do plano etc.

Sabe-se, porém, que nenhum programa de *docking* consegue estimar a energia de ligação satisfatoriamente (há uma aproximação desses valores), sendo necessária, dependendo do caso, a combinação de múltiplos programas para avaliar a predição da pose do ligante no processo de *docking*. Conforme Sotriffer [2016], Charifson et al, em 1999, e Clark et al, em 2002, elaboraram o conceito de "*consensus scoring*", que utiliza uma combinação de diversas funções de avaliação da predição de pose do ligante, já que a utilização de uma única função para *score* pode gerar diversos falsos positivos (quando um ligante é bem avaliado, mas, na realidade, não é bom) e falsos negativos (quando um bom ligante é mal avaliado). Outra característica que deve ser levada em consideração é que a utilização de uma única função de *score* não é o ideal, já que essas funções, geralmente, são elaboradas, testadas e parametrizadas a partir de um conjunto limitado de estruturas complexo-ligante.

Sendo assim, a utilização de mais de uma função de avaliação ou uma combinação de *softwares* para análise das conformações sugeridas pelo *docking* trará mais dados que possibilitam ao usuário melhor comparação entre as moléculas envolvidas e suas poses obtidas.

Warren et al (2006 *apud* Sotriffer [2016]) avaliaram 10 programas de *docking*, 37 funções de *score* para uma série de 8 proteínas de 7 tipos de moléculas-alvo para a predição do modo de ligação. Foram usadas, no total, 136 estruturas cristalográficas.

Concluiu-se que todos os programas de *docking* são capazes de gerar conformações de ligante similares à conformação cristalográfica do complexo proteína/ligante em ao menos uma das moléculas-alvo. Entretanto, as funções de pontuação foram menos bem sucedidas para distinguir a conformação cristalográfica do ligante do conjunto de poses de *docking* gerados (WARREN et al, 2006, *apud* [Sotriffer, 2016, p. 170]).

Percebe-se que os algoritmos de *docking* são capazes de explorar o espaço conformacional suficientemente bem para gerar poses do ligante corretamente. Entretanto, salienta-se, novamente, que as funções de pontuação/*score* necessitam de aprimoramento [Sotriffer, 2016] (ou mesmo não devem ser utilizadas isoladamente).

Uma questão que vale ser destacada é que o *docking*, idealmente, deve ser utilizado após uma prévia seleção das moléculas a serem envolvidas, uma vez que o *docking* em si é um procedimento que geralmente mais demorado que o procedimento de seleção ou filtro das moléculas envolvidas. Para tanto, se for utilizado em uma abordagem em que os ligantes da molécula-alvo já sejam conhecidos, podem ser utilizadas técnicas para seleção de moléculas com: características semelhantes a eles; ou características diferentes às moléculas conhecidas, quando se pretende descobrir moléculas novas. Sabe-se que moléculas similares podem ter efeitos bioquímicos similares (MAGGIORA; SHANMUGASUNDARAM, 2011, *apud* [Cortes-Cabrera et al., 2016]).

Então, se o objetivo do pesquisador for buscar moléculas comparando com os ligantes já conhecidos, a técnica TBVS deve incorporar processos inerentes às metodologias LBVS.

2.2 LBVS

O *Ligand-Based Virtual Screening* (LBVS) tem como principal abordagem a busca de novas moléculas candidatas a ligantes a partir de ligantes já conhecidos. A busca por novos ligantes pode ter como foco similaridades estruturais e nas atividades moleculares. Geralmente, conforme Cortes-Cabrera et al. [2016], o conjunto de moléculas é dividido em dois grupos, sendo que o primeiro contém moléculas com alta similaridade de atividade molecular, porém, baixa similaridade estrutural; e o segundo grupo, com baixa similaridade de atividade molecular e alta similaridade estrutural. Ambas as possibilidades, se utilizados critérios bem restritos quanto ao experimento, diminuem o espaço químico de busca, uma vez que somente moléculas semelhantes serão estudadas (mais detalhes na seção 2.2.1).

Para tanto, as duas etapas principais para a modelagem de um farmacóforo pela abordagem LBVS são a exploração do espaço conformacional dos ligantes e a determi-

nação das características químicas que são comuns aos ligantes conhecidos e que são responsáveis pela ligação com a molécula-alvo (*binding*). Para identificar quais são essas características, geralmente, os ligantes são alinhados e comparados por tamanho, forma (*shape methods*), distribuição de carga e estados conformacionais [Cortes-Cabrera et al., 2016].

2.2.1 Viés do análogo

Quando objetiva-se buscar moléculas com semelhante nível de atividade biológica dos ligantes conhecidos, encontrar moléculas que não sigam o "viés do análogo", ou *analogue bias* [Good & Oprea, 2008] é um desafio. O "viés do análogo"é caracterizado por uma busca de moléculas muito semelhantes entre si, restringindo a descoberta de plataformas de ligantes quimicamente diferentes, mas eficientes, consequentemente, reduzindo o espaço químico de análise e as possibilidades de inovação.

Já o "viés do enriquecimento artificial"[Verdonk et al., 2004] consiste em comparar um conjunto de moléculas ativas com outros muito dissimilares, podendo gerar diferenças enviesadas nos *scores*. Por exemplo, se um pesquisador descobre ou já conhece determinada molécula, de grande peso molecular, com boa energia de ligação com o alvo pesquisado, a comparação com outras moléculas de pesos moleculares muito mais baixos poderá dar a ilusão de que ele foi um excelente candidato, com *score* bem destacado dos demais, quando na verdade o que de fato ocorreu é que foram feitas comparações injustas.

Também não é interessante que o conjunto de moléculas seja enriquecido artificialmente pois a avaliação de um algoritmo de pontuação ou mesmo de mineração de dados poderá tendenciar a indicação de moléculas semelhantes às ativas, o que pode não ser desejado.

Há, então, de se buscar um equilíbrio entre esses dois vieses, constituindo um espaço de busca que não se limite a apenas análogos de um dado ligante de referência, nem se pulverize entre candidatos a ligantes muito diferentes entre si.

Para que a base de dados de moléculas não seja enviesada é importante que as estruturas, pertencentes a esse conjunto, sejam obtidas de diversas fontes e que não haja uma seleção ou filtro prévios, de forma que o espaço químico seja o mais completo e diversificado possível. Porém, sabe-se que se ter um espaço químico completo, de acordo com Kirkpatrick & Ellis [2004], é praticamente impossível, já que o total de pequenas moléculas orgânicas que populam o 'espaço químico' tem sido estimado entre 10^{60} [Kirkpatrick & Ellis, 2004, p. 823] e 10^{100} [Walters et al., 1998], números muito maiores do que a quantidade de moléculas que já foram feitas e ainda serão. A título

de comparação, de acordo com Walters et al. [1998], o espaço químico de moléculas que podem ser sintetizadas é de apenas 10^6 e a idade do universo, estimado em segundos, é na ordem de 10^{17}[Ade et al., 2016].

Sendo assim, o ideal é a utilização de um conjunto de moléculas não enviesado para que, de posse de informações sobre a molécula-alvo e/ou de ligantes conhecidos, a partir de ferramentas, seja possível identificar novas moléculas a serem estudadas como prováveis ligantes. Sabe-se, no entanto, que algumas características físico-químicas precisam ser obtidas de todo o conjunto, para que sejam utilizadas como parâmetro de comparação. Essas propriedades podem ser obtidas juntamente com o arquivo da molécula (quando obtida a partir de uma biblioteca de moléculas) ou podem ser calculadas com o auxílio de um *software* como o RDKit [Landrum, 2006], detalhado na seção 4.3.1.1. A comparação das moléculas entre si também pode ser feita a partir do *fingerprint* de cada uma delas, como descrito com detalhes na seção 4.4.1.

2.3 Abordagem mista

De acordo com Villoutreix et al. [2016], as diversas técnicas *in silico* para descoberta de novos fármacos podem ser combinadas e utilizadas em conjunto. O LBVS pode ser utilizado juntamente com o TBVS, quando se tem conhecimento da molécula-alvo e de alguns ligantes dessa molécula.

Pode ser feita a classificação de moléculas em uma base de dados de acordo com o cálculo da probabilidade da molécula ter, conforme verificação do pesquisador, boa energia de ligação à molécula-alvo. Nesse caso, utiliza-se de um conjunto de moléculas ativas (ligantes) e outro conjunto de moléculas inativas para treinar um algoritmo classificador que avalia, por meio de diversos métodos de aprendizado de máquina, se uma molécula, dentro de um conjunto desconhecido, seria ativa ou inativa. Cortes-Cabrera et al. [2016] diz que esse é um método ainda pouco utilizado e não necessita de informações sobre a molécula-alvo; porém, se forem utilizadas técnicas da abordagem TBVS, a precisão dos resultados pode melhorar consideravelmente.

Sendo assim, apresentamos o EasyVS, uma ferramenta web, *user-friendly* que pretende simplificar o processo de *Virtual Screening* através da seleção de uma proteína-alvo e de um conjunto de pequenas moléculas, além da parametrização do e execução *docking*, mostrando os resultados de forma visual para facilitar as análises dos usuários.

O EasyVS, por permitir ao usuário montar o seu próprio espaço químico, realizar agrupamento de moléculas a partir de propriedades físico-químicas (métodos LBVS) e possibilitar o *docking* de qualquer molécula de sua biblioteca ou de moléculas en-

viadas pelo usuário (método TBVS), pode ser classificada como uma ferramenta que utiliza uma abordagem mista TBVS e LBVS. Na seção 4.5.6 é descrita a biblioteca de compostos do EasyVS de forma que a ferramenta tenha um conjunto amplo de moléculas.

Capítulo 3

Objetivos

3.1 Objetivo Geral

Construir um sistema, acessível pela internet e com interface amigável, que integre um conjunto de ferramentas num fluxo de processos para triagem virtual de ligantes a partir de metodologias baseadas em estrutura dos alvos e ligantes.

3.2 Objetivos Específicos

- Coletar dados de moléculas a partir de diferentes bibliotecas de pequenas moléculas;
- Projetar o armazenamento de dados e resultados do *docking*;
- Implementar agrupamento molecular por similaridade, de modo a explorar melhor o espaço-químico de ligantes;
- Desenvolver um *front end web* como interface para facilitar o acesso dos usuários às ferramentas desenvolvidas;
- Realizar estudo de caso para validação da ferramenta, com a escolha de uma família de proteínas-alvo para a triagem virtual.

Capítulo 4

EasyVS

Diante da grande quantidade de dados, tanto de proteínas-alvo quanto de pequenas moléculas candidatas a fármaco/*hits*, foi proposto a criação de uma base de dados para armazenamento desses elementos, juntamente com um conjunto de ferramentas para triagem virtual combinadas em uma interface para interação a partir da internet.

A ideia central do EasyVS[1] é possibilitar ao usuário a utilização de uma ferramenta capaz de realizar *Virtual Screening* de um conjunto de moléculas sem a necessidade de instalação de nenhum programa na máquina do usuário além de um *browser* moderno. A ferramenta respeita as características que o usuário estabelecer, a partir de um, prévio e opcional, agrupamento dessas moléculas (o que torna viável a utilização do sistema com um vasto número de *hits*).

Antes do detalhamento do sistema em si, aqui serão expostos as ferramentas com funcionalidades semelhantes ao EasyVS, juntamente com suas características positivas e negativas. Ao final da comparação, antes da explanação dos passos do EasyVS, os principais atributos de cada uma das ferramentas, incluindo o EasyVS, serão sumarizados.

4.1 Ferramentas semelhantes

Procurou-se, durante todo o processo de desenvolvimento, o enriquecimento do trabalho como um todo a partir da comparação do EasyVS com as ferramentas disponíveis e com objetivos semelhantes, sendo elas gratuitas ou não. A Tabela 4.1 fornece um comparativo entre as ferramentas analisadas e a ferramenta proposta, o EasyVS, ordenados por data de publicação, ou de criação da ferramenta, mais antiga para mais recente.

[1] Acessível a partir do endereço: `https://easyvs.unifei.edu.br`

Alguns dos pontos que foram analisados dizem respeito a possibilidade de envio ou não de arquivos (tanto para proteína-alvo quanto para pequenas moléculas/*hits*), se existem bibliotecas de compostos disponíveis e, dessas, podem ser aplicados filtros para seleção de um conjunto de moléculas a partir de propriedades físico-químicas, entre outras características.

Uma das principais características verificadas, por essas serem ferramentas acessíveis pela internet, diz respeito ao *design* do sistema com foco na responsividade: se o sistema reage às necessidades dos usuários e seus dispositivos, alterando a disposição e/ou apresentação do conteúdo em diferentes formas e tamanhos de telas. Esse termo (*design* responsivo) foi primeiramente definido por Marcotte [2010].

Ressalta-se aqui que as ferramentas analisadas estabeleciam limitações de recursos ou de funcionalidades para seus utilizadores, podendo ser devido à grande utilização e objetivando prover o acesso a uma maior quantidade de pesquisadores. Até o presente momento, o EasyVS não limita os recursos a seus utilizadores, porém, sabe-se que pode ser necessário estabelecer limites para a quantidade de moléculas a ser trabalhada a cada requisição do usuário.

4.1.1 DockingServer

O DockingServer[2] [Bikadi & Hazai, 2009] é uma ferramenta que permite o *docking* entre proteínas e pequenas moléculas.

Com a possibilidade de mais recursos a usuários que pagam pela plataforma, a ferramenta estabelece diversas limitações para usuários visitantes e que realizaram o registro gratuito na plataforma. Dentre as limitações, estão a quantidade de *docking* diário, espaço para armazenamento do resultado e número de processadores dedicados, dentre outros.

O AutoDock 4 [Morris et al., 2009] é utilizado como ferramenta para o *docking* e a seleção de pequenas moléculas para *docking* pode ser feita somente pela busca manual (pelo nome) entre as moléculas disponíveis ou pelo *upload* de arquivos.

Aqui vale salientar que no processo de testes da referida ferramenta foram feitos envios de um arquivo para *target* e outro como ligante. Durante o processo de utilização da ferramenta, o ligante enviado não foi possível de ser encontrado, porém, a interface do sistema apresenta muitas moléculas disponíveis em uma lista, mostrando que o conjunto de moléculas enviadas por usuários não registrados são compartilhadas com os demais também não registrados. Sendo assim o teste não pode ocorrer como programado.

[2]Disponível no endereço `https://www.dockingserver.com/web/`. Acesso em: 8 de jul. 2018.

4.1.2 SwissDock

SwissDock [Grosdidier et al., 2011] é um serviço web que prediz interações moleculares que podem ocorrer entre uma proteína alvo e uma pequena molécula, podendo ser automaticamente preparadas para o *docking*. SwissDock[3] utiliza o EADock DSS *engine* e, após testes, foi possível identificar algumas limitações.

O SwissDock possui apenas 264 alvos (no artigo e no site não há descrição dos critérios utilizados para a seleção desses alvos) e além disso, não foi possível encontrar nenhuma GPCR (utilizada no estudo de caso desse documento). Foram pesquisados os termos: GPCR, *G protein-coupled receptor* e *protein-coupled receptor*, sem nenhum resultado.

É necessário que seja digitado o código da molécula a ser feito o *docking* pelo nome do ZINC, pela categoria de molécula ou *upload* de arquivo (até 5MB). Em relação aos filtros, só é possível fazê-lo pelo nome ou categoria da molécula.

O sistema também não detecta os *pockets* automaticamente, não mostra as moléculas (*targets* ou ligantes), não exige e-mail para recuperação dos resultados e a página web não é responsiva.

4.1.3 idock

O idock[4] [Li et al., 2012] é bem semelhante ao EasyVS, porém, faz *docking* de todas as moléculas que pertencem ao filtro selecionado (sem realizar agrupamento ou remoção de moléculas similares) e os *jobs* submetidos gastam, ao menos, 15 dias para terem resultado.

O idock utiliza um *software* próprio para *docking*, que, segundo os autores [Li et al., 2012], é de 8,69 a 37,51 vezes mais rápido que o Autodock Vina. Além disso, utiliza o RF-Score [Ballester & Mitchell, 2010] e o ZINC nas versões 2012-04-26, 2013-01-10 e 2013-12-18 com 23.129.083 de hits. O idock gasta, conforme publicação, cerca de 1 segundo por ligante, para realizar o *docking*.

A página web do idock é responsiva, porém a visualização da proteína não está responsiva, além disso permite o download de somente 1.000 moléculas por job. Em relação aos *pockets*, ele detecta apenas um *pocket* automaticamente e não mostra mais opções. A execução dos jobs não leva em consideração filas de prioridades, ou seja, as seleções muito grandes podem atrasar jobs pequenos que poderiam ser rapidamente processados. Por fim, notou-se também uma falta de privacidade, uma vez que não há

[3]Disponível no endereço `http://www.swissdock.ch/`. Acesso em: 8 de jul. 2018.
[4]Disponível no endereço `http://istar.cse.cuhk.edu.hk/idock/`. Acesso em: 8 de jul. 2018.

separação por sessão ou usuário, todos podem ver tudo que está sendo processado e acessar os resultados.

4.1.4 DockThor

DockThor[5] [de Magalhães et al., 2014] também é uma ferramenta para *docking* entre proteína e ligante, desenvolvida no Brasil. O sistema faz *docking* com algoritmo desenvolvido pelo próprio grupo e utiliza o JSMol para visualização dos resultados.

Algumas das limitações estão em eles não detectarem *pockets*, não possuírem uma base de dados de proteínas e hits e exigirem registro (sendo necessário o envio e aprovação de um projeto) para *dockings* com mais de 1.000 moléculas.

Objetivando a melhor visualização dos principais pontos levados em consideração nessa comparação, alguns desses estão sumarizados na Tabela 4.1.

Tabela 4.1. Resumo da comparação do EasyVS com ferramentas semelhantes

Nome	**Fonte targets; hits**	**Visualização de moléculas**	**Limitações**
DockingServer	*Upload* e RCSB PDB; PubChem	JSMol	2 *docking* diários; Registro no sistema para utilização
SwissDock	RCSB PDB; Zinc	JSMol	264 proteínas disponíveis; Não detecta *pockets*
idock	*Upload*; Zinc	GLmol	Sem filas de prioridade; Download de 1000 moléculas por job; Detecta 1 *pocket*
DockThor	*Upload*; *Upload*	JSMol	Não detecta *pockets*; Aprovação de projeto para mais de 1000 moléculas
EasyVS	*Upload* e RCSB PDB; *Upload*, diversas bibliotecas[6]	NGLView	Envio de pequenas moléculas somente no formato SDF; Não é possível agrupar moléculas além dos grupos pré-processados

4.2 Fluxo de utilização do EasyVS

O fluxo principal de utilização da ferramenta é apresentado nessa seção. Aqui são listadas, com detalhes, as funcionalidades presentes nos quatro passos em que o

[5]Disponível no endereço `https://www.dockthor.lncc.br`. Acesso em: 8 de jul. 2018.
[6]ChEMBL, Chembridge, DrugBank, HMDB, Maybridge, Super Natural II e Zinc

usuário, a partir de uma interface gráfica simples, poderá chegar a alguns resultados de *Virtual Screening*.

Para facilitar o entendimento do funcionamento do EasyVS, foi elaborada a Figura 4.1, onde são destacadas as páginas correspondentes aos quatro principais passos que compõem o fluxo de utilização do sistema e seus processos internos.

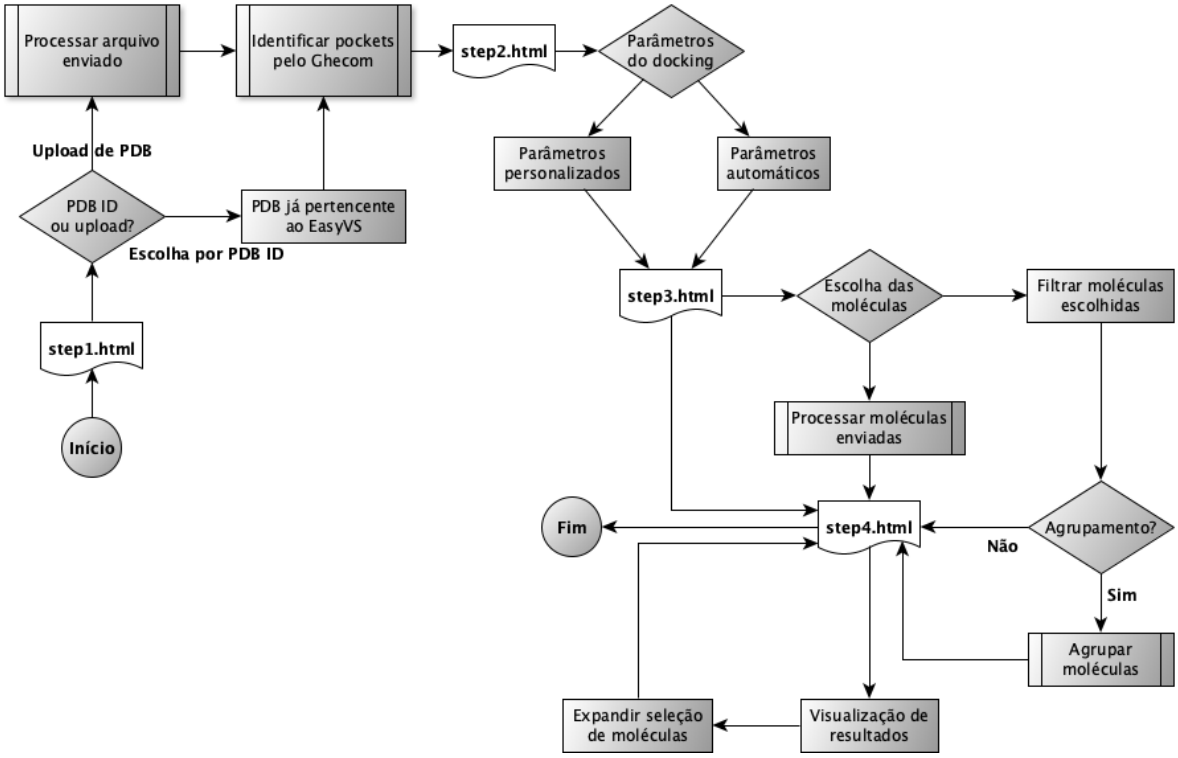

Figura 4.1. EasyVS - Diagrama geral: Em destaque estão as páginas principais do sistema e são mostradas as decisões que podem ser tomadas pelo usuário no seu fluxo de utilização.

4.2.1 Passo 1 - Escolha do alvo/proteína

O primeiro passo do usuário no sistema é a definição de uma proteína-alvo (ou *target*) para realizar o estudo. Nesse momento, o usuário poderá escolher dentre as proteínas depositadas no RCSB PDB[7], identificadas pelo PDB ID (*accession code* composto de quatro caracteres), como apresentado na Figura 4.2 (A) ou enviar o seu próprio arquivo, no formato PDB, para o EasyVS, como pode ser visualizado na Figura 4.2 (B). No caso de *upload*, o usuário poderá ser notificado por e-mail quando o processamento for concluído e receberá o *link* para acesso direto aos dados. Salienta-se que arquivos

[7]Acessível em `https://www.rcsb.org/`. Acesso em: 8 de jul. 2018.

enviados por usuários não estarão disponíveis por outra forma além do *link* enviado ao usuário.

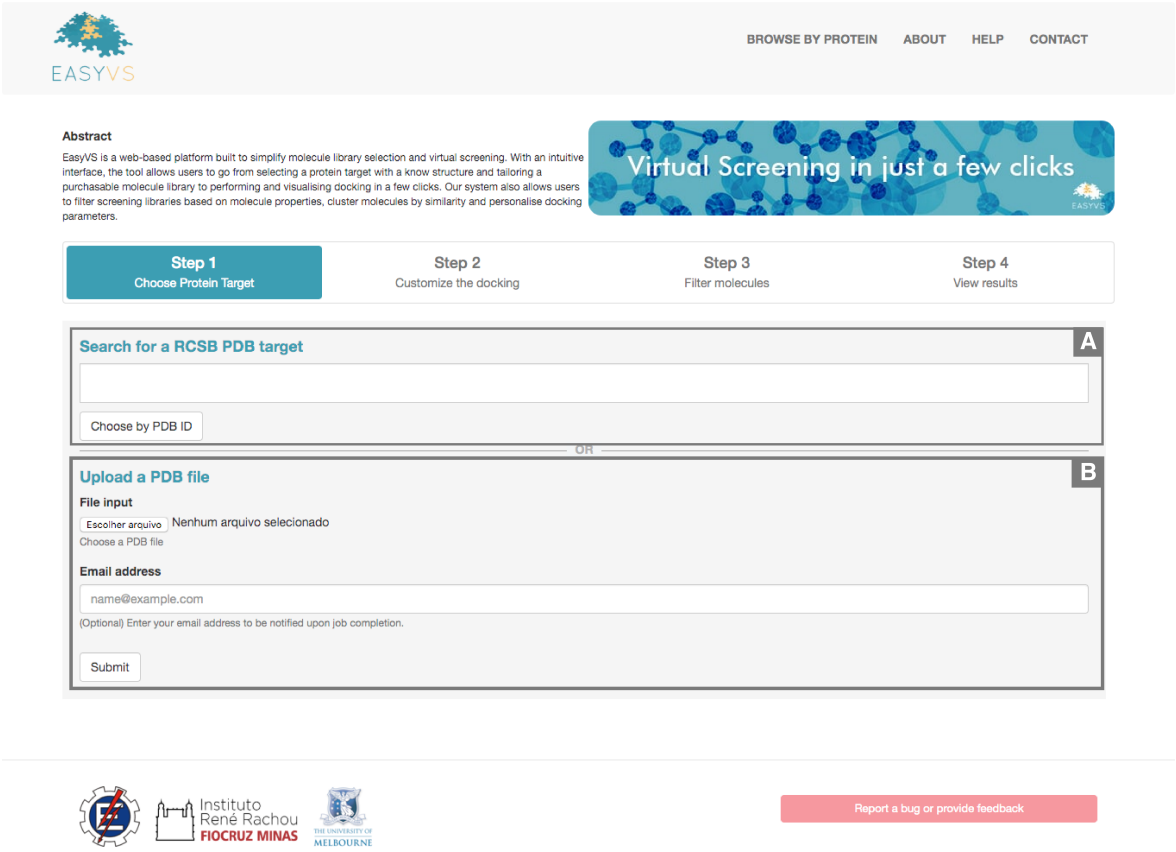

Figura 4.2. EasyVS - Página inicial: Seleção da proteína-alvo para estudo, a partir do PDB ID (A), considerando as proteínas existentes no EasyVS, enviando arquivo PDB (B), preenchendo um e-mail para receber um *link* para acesso ao arquivo enviado e processado no EasyVS. Fonte: `https://easyvs.unifei.edu.br/index`. Acesso em: 8 de jul. 2018.

Quando a opção de escolha pelo PDB ID é selecionada, o sistema busca diretamente do *site* do RCSB PDB pelos arquivos necessários, realiza o processamento e encaminha o usuário para o segundo passo. No caso do usuário digitar uma entrada inválida será emitido um aviso e o processamento não ocorrerá.

Caso o usuário queira fazer envio do arquivo PDB (*upload*) e, se esse arquivo possuir mais de um *model* (que pode ser definido como um conjunto de coordenadas em um arquivo PDB) e/ou com a presença de heteroátomos, somente o primeiro *model* será considerado, e os heteroátomos também serão desconsiderados no processo do *docking*.

Na atual versão do EasyVS, todos os heteroátomos são removidos durante a preparação da proteína-alvo para o *docking*. Na seção 6.1.6 é descrito que, em versões

futuras, pretende-se possibilitar ao usuário a escolha das moléculas a serem removidas e mantidas.

4.2.2 Passo 2 - Configurar parâmetros para *docking*

Uma vez que a proteína-alvo for definida, no segundo passo é solicitado que o usuário, a partir de uma visualização tridimensional da proteína-alvo, forneça os parâmetros desejados para o *docking* molecular, seja através dos dados sugeridos automaticamente pela plataforma ou personalizando-os.

Inicialmente, o usuário verá a proteína à esquerda, na Figura 4.3 (A), escolhendo a visualização a partir de *cartoon* e/ou *surface*, podendo ou não mostrar os ligantes cristalográficos (presentes no arquivo PDB selecionado no Step 1) e a caixa/*box*, onde o *docking* será realizado.

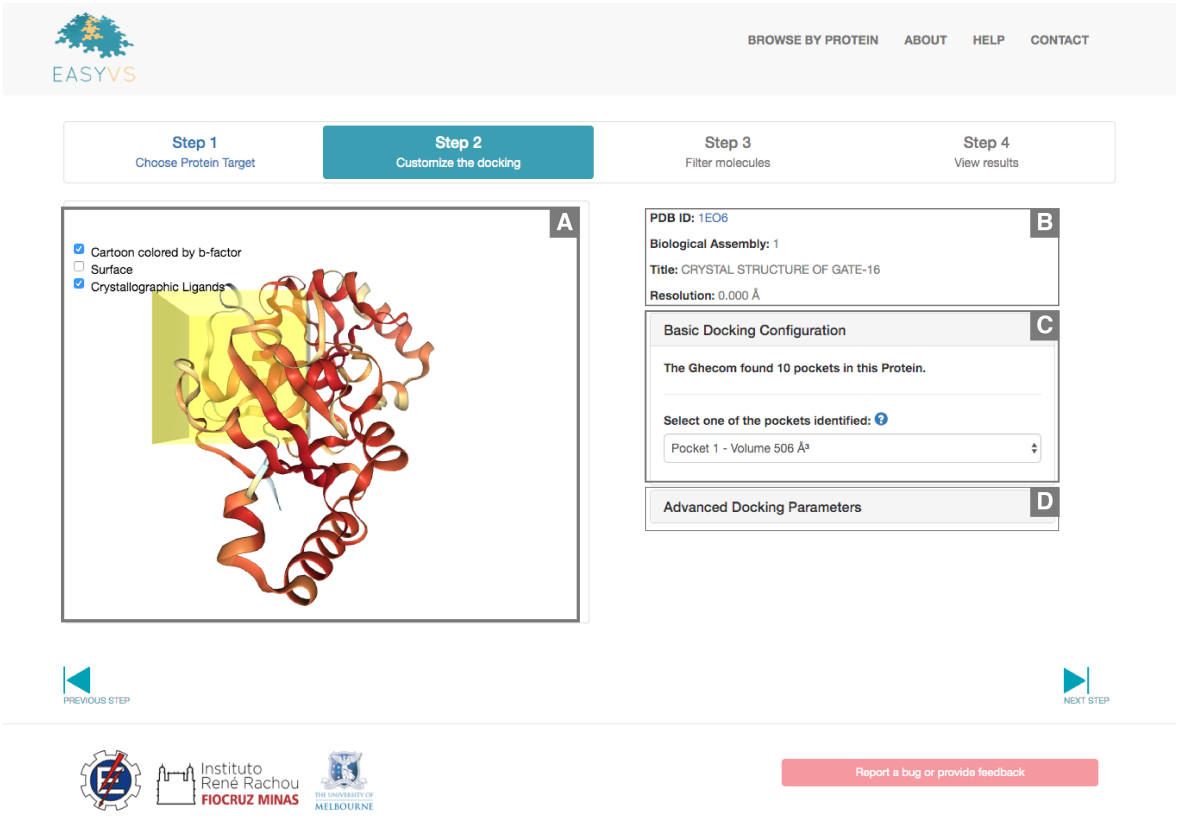

Figura 4.3. EasyVS - Configurações do *docking*: A partir da visualização da proteína-alvo (A) e confirmação de dados do alvo (B), o usuário escolhe a posição do *box* (C) a partir das sugestões da ferramenta ou opta pela personalização a partir de parâmetros avançados incluindo posição detalhada do *box*, tamanho, *exhaustiveness*, *energy range* e número de poses (D). Fonte: `https://easyvs.unifei.edu.br/step2/141572`. Acesso em: 8 de jul. 2018.

Como configuração disponível, caso o usuário não queira personalizar muitos detalhes do *docking* (apresentado na interface como *Basic Docking Configuration*), está o posicionamento do centro do *box*, conforme identificado pelo Ghecom [Kawabata, 2010], com a área do *pocket* mostrada em ångströms[8] cúbicos (Å^3). O Ghecom [Kawabata, 2010] é um *software* utilizado para identificar cavidades/*pockets* na estrutura, e, internamente ao EasyVS, as coordenadas que compõem essas cavidades são utilizadas para cálculo dos centros geométricos individuais como sugestões, ao usuário, para o centro do *box*.

Vale salientar que as coordenadas retornadas pelo Ghecom não se referem a cavidades que tenham ligantes próximos ou internos. Cabe ao usuário, com o auxílio da imagem tridimensional da estrutura, apresentada nessa mesma página, escolher as coordenadas dentre as sugeridas (ordenadas conforme o volume identificado, em ordem decrescente) ou inserir manualmente.

No caso do usuário desejar personalizar as configurações do *docking* (denominado *Advanced Docking Parameters*), é possível alterar os parâmetros a serem inseridos no AutoDock Vina [Trott & Olson, 2010], que são o tamanho da caixa, em ångströms (a mesma dimensão nos eixos X, Y e Z), *exhaustiveness*, número máximo de poses geradas, *energy range*, além do posicionamento detalhado do *box*. O valor da *exhautiveness*, conforme o manual *online* do AutoDock Vina[9], é um valor que altera a quantidade de tempo que o algoritmo utilizará para achar o ponto de menor energia. Ainda conforme o manual, geralmente, não é necessário aumentar muito esse valor. Porém, caso o usuário queira alterar a *exhautiveness*, ao aumentar seu valor, o tempo de busca aumenta linearmente e diminui exponencialmente a probabilidade de não encontrar o mínimo. O próprio EasyVS já sugere o valor 8 como padrão (mesmo valor sugerido pelo manual do AutoDock Vina).

Os dois últimos parâmetros personalizados que podem ser informados são correlatos, uma vez que o número máximo de poses que serão geradas deverá obedecer à diferença de energia estimada pelo parâmetro *energy range*. Em termos simples, a primeira e a última pose geradas deverão ter diferença de energia estimada pelo *energy range*, valor calculado pelo AutoDock Vina (energia essa estimada em kcal/mol). Para exemplificar, se um usuário escolher um *energy range* de valor 3 kcal/mol e um máximo de 20 poses, então terá como resultado até 20 poses, desde que a diferença de energia entre "melhor"e a "pior"pose seja estimada em 3 kcal/mol.

Para o usuário que opte por utilizar os parâmetros sugeridos pelo sistema, o valor

[8] 1 ångström equivale a 10^{-10} metro

[9] Disponível em `http://vina.scripps.edu/manual.html#exhaustiveness`. Acesso em: 8 de jul. 2018.

de *exhaustiveness* será 8, número máximo de poses geradas será 20 e *energy range* será 3 (valores sugeridos no manual do AutoDock Vina).

4.2.3 Passo 3 - Filtro para pequenas moléculas

Com a proteína-alvo escolhida e parâmetros para *docking* configurados, o usuário deverá definir o conjunto de pequenas moléculas que deseja realizar o estudo, como apresentado na Figura 4.4. O usuário pode optar dentre as bibliotecas de moléculas disponíveis (Figura 4.4 (A)) ou enviar o seu próprio conjunto de moléculas. No caso da utilização das bibliotecas internas ao EasyVS, são disponibilizados filtros para esses compostos a partir de número de átomos, peso molecular, quantidade de doadores e aceptores de hidrogênio, número de anéis, ligações rotacionáveis e valor de LogP[10], como apresentado na Figura 4.4 (C).

Para facilitar a interação do usuário com o sistema, já é disponibilizado na interface um conjunto de botões para a seleção das moléculas que obedecem Lipinski RO5 e RO3, como destacado na Figura 4.4 (C).

Lipinski et al. [2001] citam que, após estudos experimentais e computacionais, foi estabelecido um conjunto de critérios voltados para otimizar propriedades farmacológicas relativas a absorção, distribuição, metabolismo e excreção (ADME). O *Rule Of Five* (RO5) original de Lipinski et al. [2001] cita que a molécula deve seguir as seguintes características:

- não ter mais que cinco doadores de hidrogênio;
- ter até dez aceptores de hidrogênio;
- massa molar menor que 500 daltons (Da);
- LogP menor ou igual a cinco.

Já a adaptação dessa regra, tornando-a mais restritiva, é o *Rule Of Three* (RO3), que objetiva selecionar as moléculas caracterizadas como *lead-like*. A RO3 consiste em selecionar moléculas com até três doadores de hidrogênio, três aceptores de hidrogênio, três ligações rotacionáveis, LogP até três e massa molar menor que 300 daltons.

Um grande conjunto de moléculas é disponibilizado ao usuário, sendo obtidas de diversas fontes como ChEMBL [Gaulton et al., 2017], Chembridge Core e Chembridge Express [Desai et al., 2004], Drugbank [Wishart et al., 2006], HMDB [Wishart et al.,

[10]LogP (ou coeficiente de partição) representa a relação das concentrações da substância em um solvente orgânico e em água. O LogP será maior quanto menor for a polaridade da substância.

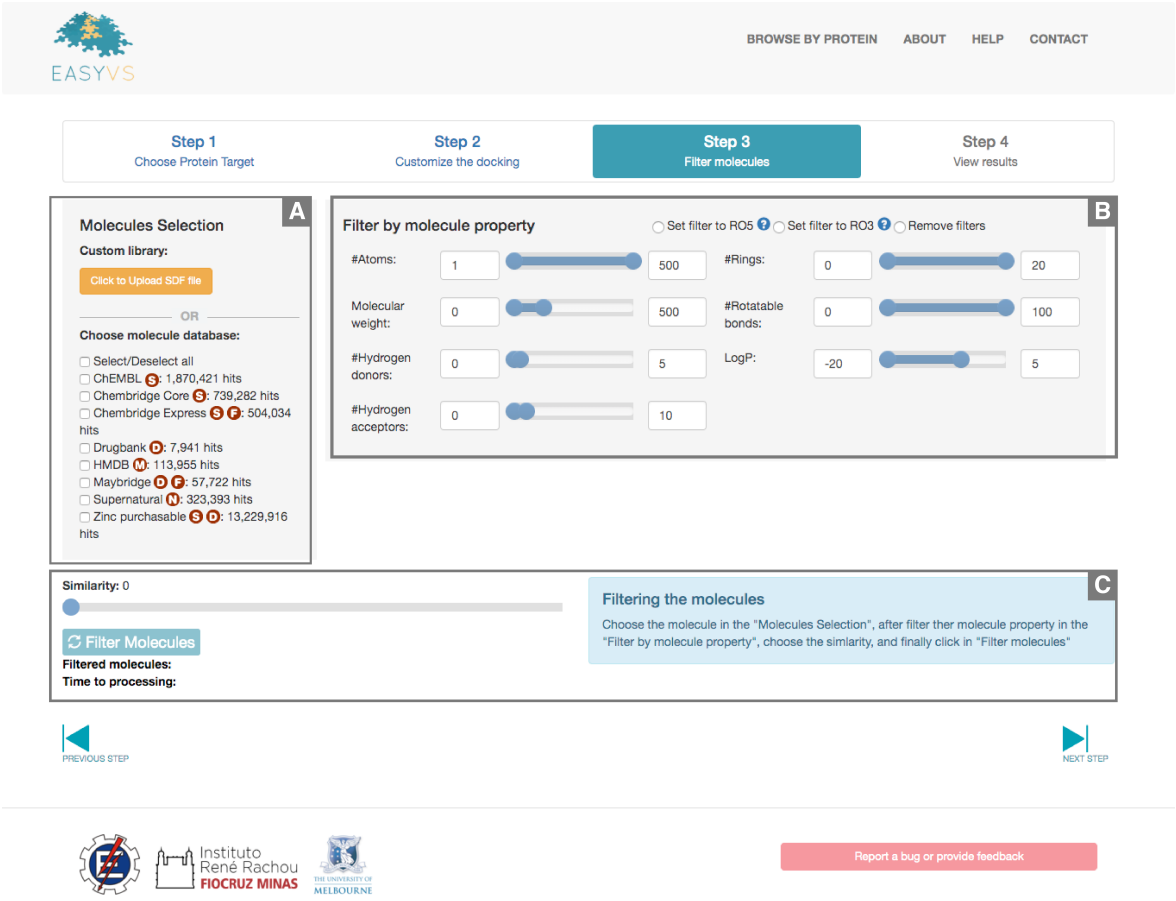

Figura 4.4. EasyVS - Filtro de pequenas moléculas: Página que apresenta as bibliotecas de pequenas moléculas (A) possibilita a definição manual de filtros a partir de propriedades físico-químicas (B) ou utilizando filtros pré-determinados, baseados em Lipinski et al. [2001] (C), além de permitir a utilização de agrupamento molecular, quando selecionado valor de similaridade diferente de zero (D) e poder visualizar a quantidade de moléculas filtradas e uma estimativa de tempo para o término no processamento do *docking* para o conjunto selecionado de moléculas (E). Fonte: `https://easyvs.unifei.edu.br/step3/141572&3.340&19.500&-58.510&20&8`. Acesso em: 8 de jul. 2018.

2018b], Maybridge [Fisher Scientific, 2018]), Super Natural II [Banerjee et al., 2015] e Zinc 15 *purchasable in stock* [Sterling & Irwin, 2015], somando mais que 16 milhões de moléculas.

O EasyVS possibilita também que o usuário envie seu próprio conjunto de moléculas em um arquivo, no formato SDF. Esse arquivo poderá ser utilizado como alternativa às moléculas já disponíveis no EasyVS e nenhum filtro será aplicado nos *hits* do arquivo enviado (todas as moléculas presentes no arquivo SDF serão consideradas para o estudo realizado no Step 4, ou seja, não será feito agrupamento das mesmas).

O próximo passo, ainda nessa página, consiste no agrupamento, opcional, das

moléculas selecionadas (Figura 4.4 (D)). O agrupamento ocorre conforme algoritmo de Butina [1999], detalhado na seção 4.4.2. Caso o usuário escolha valor zero para similaridade, não será feito agrupamento e, no próximo passo, será feito o *docking* de todas as moléculas que corresponderem aos filtros selecionados. Qualquer outro valor escolhido será utilizado como valor limiar para o agrupamento (também chamado de *cutoff*).

Há a sugestão, pelo sistema, que o usuário opte por realizar o agrupamento, uma vez que o EasyVS apresenta uma estimativa de tempo (Figura 4.4 (E)) para o término do *docking* considerando o conjunto de moléculas selecionadas e seu agrupamento ou não. Como, no caso do agrupamento molecular, há a escolha de representantes de cada grupo, o tempo de processamento será menor que o *docking* de todas as moléculas filtradas.

4.2.4 Passo 4 - Visualização de resultados

Essa é a última etapa do fluxo principal do EasyVS. Conforme seleções das etapas anteriores, o sistema irá iniciar o preparo do grupo de moléculas a ser estudado, havendo ou não agrupamento molecular, e, logo após, iniciar o *docking* molecular. A Figura 4.5 mostra a página com a visualização dos resultados obtidos.

Nessa página há uma tabela, em destaque na Figura 4.5 (A), com todos os resultados de *docking* até o momento que o usuário carregou a página. Nessa tabela são apresentados, em cada linha, uma molécula que pertença aos filtros do passo 3, juntamente com propriedades físico-químicas e valores de afinidade em kcal/mol (estimado pelo AutoDock Vina, Trott & Olson [2010]) e nMolar (estimado pelo NNScore 2.01, Durrant & McCammon [2011]) da mesma e um botão que, quando clicado, mostra as poses preditas pelo *docking* com detalhes na Figura 4.5 (C), (D) e (E). Salienta-se que a quantidade de moléculas da tabela dependerá da opção ou não pelo agrupamento molecular, já que o EasyVS faz o *docking* de 1 molécula por grupo, no caso de escolha de um valor de *cutoff*/limiar diferente de zero (como já exposto) ou o *docking* será executado para todas as moléculas filtradas (com *cutoff* sendo zero). Quanto mais próximo de 1 o valor do *cutoff*, maior a quantidade de grupos, pois somente moléculas mais semelhantes estarão no mesmo grupo. Os detalhes do algoritmo de agrupamento estão presentes na seção 4.4 deste documento.

Toda a tabela referida anteriormente poderá ser baixada como um arquivo CSV (sem a representação molecular) a partir de um *link* destacado na Figura 4.5 (B).

Na segunda parte dessa página, onde é possível visualizar as poses do *docking* com detalhes, o usuário poderá ver os nomes identificadores do alvo e da molécula,

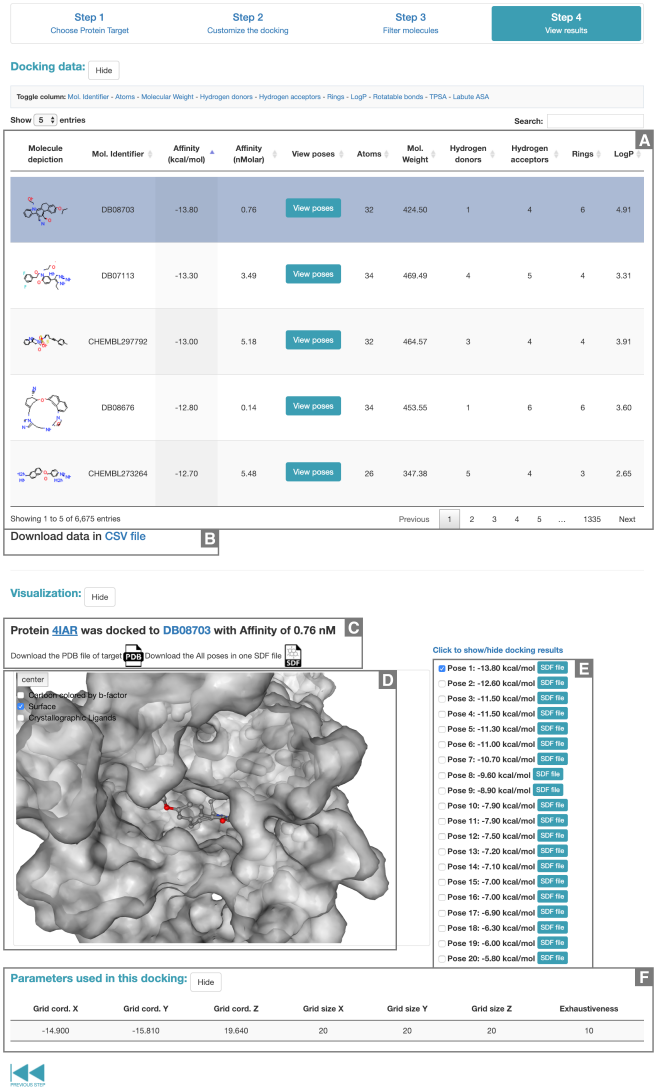

Figura 4.5. EasyVS - Visualização de resultados: A página apresenta todos os resultados do *docking* até o momento (A), disponibiliza o *download* de todos os dados da tabela em um arquivo CSV (B), além de mostrar, quando selecionada uma molécula específica, detalhes do alvo e da molécula, juntamente com *links* para acesso individual (C), representação tridimensional (D) das poses selecionadas (E) e os parâmetros utilizados para o *docking* (F). Fonte: `https://easyvs.unifei.edu.br/step4/141572&1`. Acesso em: 8 de jul. 2018.

juntamente com um *link* para acesso às páginas com detalhes de cada um, juntamente com o valor de afinidade, predito em nMolar pelo NNScore 2.01 [Durrant & McCammon, 2011], na Figura 4.5 (C). Logo abaixo é disponibilizado ao usuário o *download* dos arquivos PDB da proteína-alvo e SDF de todas as poses preditas pelo sistema através do AutoDock Vina [Trott & Olson, 2010].

Para cada molécula, o EasyVS apresenta as poses preditas pelo algoritmo de

docking, utilizando as configurações descritas nos passos anteriores. Além de ver, como representado na Figura 4.5 (D), a pose tridimensional da molécula junto ao alvo, o usuário poderá alternar entre a visualização das poses ou mesmo baixar os arquivos SDF individuais de cada uma dessas, como destacado na Figura 4.5 (E).

No fim da página é mostrado ao usuário quais os parâmetros que foram utilizados para a realização do *docking* (Figura 4.5 (F)), juntamente com um botão para voltar ao passo anterior (botão padrão em todas as telas do sistema). Vale salientar que diversos desses elementos aqui apresentados podem ser ocultados pelo usuário para que tenha uma interface personalizada, visualizando somente o que lhe for essencial.

Uma vez que o usuário conseguirá visualizar todos os resultados de *docking* da ferramenta, é natural que ele encontre um conjunto de moléculas que sejam mais bem caracterizadas como possíveis ligantes da proteína-alvo e ele mesmo reinicie o processo de estudo com o EasyVS alterando parâmetros como posição da caixa do *docking* ou um filtro mais detalhado das moléculas compondo um conjunto de moléculas voltado ao seu objetivo específico.

Para facilitar o processo de interação do usuário com o EasyVS, é possibilitado ao usuário a seleção uma molécula, dentre os resultados apresentados na página do passo 4, para que seja gerado um novo conjunto de compostos, selecionados automaticamente, baseados na similaridade entre moléculas e no agrupamento. Esse fluxo poderá acontecer quantas vezes o usuário quiser, realizando o processo de refinamento personalizado dentre as moléculas selecionadas por ele e disponíveis no sistema.

4.3 Detalhes técnicos da ferramenta

Nesta seção serão detalhadas características técnicas do EasyVS como as linguagens e ferramentas utilizadas, assim como detalhes relativos ao agrupamento e *docking*.

4.3.1 Python

O sistema do EasyVS é totalmente desenvolvido na linguagem de programação Python na versão 3.6.6, distribuído pelo Anaconda [Anaconda, 2016]. O Python é uma linguagem de programação interpretada, interativa e orientada a objetos que pode ser executada em diversos sistemas operacionais como Unix, Linux, Mac e Windows [Python, 2009]. Desde a versão 2.1 a linguagem é mantida pela Python *software* Foundation e é uma linguagem muito utilizada na área de Bioinformática, especialmente a partir das bibliotecas RDKit [Landrum, 2006] e BioPython [Cock et al., 2009]. No EasyVS, o BioPython só é utilizado como um facilitador para obter alguns dados textu-

ais dos arquivos no formato PDB das proteínas-alvo. Já o RDKit é bastante utilizado, o que será descrito na seção 4.3.1.1.

O Python foi escolhido como principal linguagem do EasyVS por ser de fácil utilização, permitir a utilização de muitas bibliotecas voltadas para a Bioinformática e, em especial, o RDKit. Para a leitura, conversão e cálculo de propriedades físico-químicas das moléculas é utilizado o RDKit, que é uma biblioteca do Python. Como principal ferramenta e para a interface *web* é utilizado o Flask [Ronacher, 2018], que também é uma biblioteca do Python.

Em acréscimo, para gerenciar o paralelismo no Python, assim como a possibilidade de execução de funções de maneira assíncrona é utilizado o RQ [Driessen, 2018].

O RQ é uma biblioteca do Python, *open-source*, utilizada para enfileirar funções e executá-las a partir de um recurso denominado *Worker*. Ele utiliza do servidor Redis [Sanfilippo & Noordhuis, 2018] para gerenciar as filas de códigos a serem executados (o RQ é um acrônimo de *Redis Queue*).

A maioria das funcionalidades relacionadas ao agrupamento de moléculas, rotinas para preparação e execução do *docking* e inserção de moléculas no sistema são gerenciadas pelo RQ.

Cabe ressaltar que a versão do Python utilizada no EasyVS é distribuída pelo Anaconda [2016], em especial, por permitir a geração do InChI pelo RDKit.

4.3.1.1 RDKit

O RDKit [Landrum, 2006] é um conjunto de ferramentas, *open-source*, escrito nas linguagens de programação C++ e Python, voltadas para a quimioinformática e aprendizado de máquina. Ele possibilita a manipulação de moléculas, importadas de diversos formatos, assim como o cálculo de diversas propriedades como peso molar, quantidade de anéis, átomos, busca por similaridade entre muitas outras funcionalidades.

Dentro do EasyVS, ele é utilizado como uma extensão da linguagem Python a partir da leitura e escrita de arquivos SDF, PDB, SMILES[11], geração de InChI[12], conversão entre formatos de moléculas com coordenadas tridimensionais, bidimensionais ou textuais (*string*), além da criação de *fingerprints* (detalhados na seção 4.4.1), entre outros recursos. O RDKit é muito utilizado por aceitar uma grande gama de arquivos e representações de moléculas como entrada e saída, sendo os principais citados anteriormente. Destaca-se que SMILES e InChI podem ser descritos como formatos

[11] Acrônimo de *Simplified Molecular Input Line Entry Specification*, é uma forma de representação de uma molécula somente utilizando caracteres convencionais, também chamados de ASCII

[12] Acrônimo de *IUPAC International Chemical Identifier*, *International Chemical Identifier* ou, em português, Identificador Químico Internacional

de representação textual de moléculas (*string*), porém, geralmente o InChI é utilizado quando se pretende identificar unicamente uma molécula, implicando na pequena possibilidade de moléculas diferentes gerarem o mesmo descritor.

Vale salientar que todas as moléculas inseridas no EasyVS passam pelo RDKit antes de estarem disponíveis ao usuário e, em diversos casos, a ferramenta identificou erros nos arquivos originais das moléculas. Os possíveis erros encontrados foram na conformação ou, principalmente, na quantidade de elétrons na camada de valência ou número de ligações atômicas além do permitido (a maior quantidade de moléculas com indicação de erro teve a descrição deste ressaltando quantidade de elétrons na camada de valência além do permitido). Nos casos que o RDKit detectou algum erro que impediu a correta conversão ou leitura da molécula, o *software* OpenBabel [O'Boyle et al., 2011] foi utilizado para a conversão da molécula, geração do InChI e SMILES, porém, essas moléculas foram marcadas como não lidas pelo RDKit e, na atual versão do EasyVS, não são consideradas nos filtros do passo 3 e, consequentemente, *docking*.

Durante o desenvolvimento da ferramenta, buscou-se contato com o desenvolvedor e criador do RDKit para a identificação e correção dos erros (sem modificação dos arquivos originais) a fim de incorporar a maior quantidade de moléculas das bases de dados no EasyVS. Após constantes mudanças e adequações, o sistema tem conseguido importar todas as moléculas que não tiveram erros identificados pelo RDKit.

4.3.1.2 Flask

O Flask [Ronacher, 2018] é um *framework* para desenvolvimento *web* bem pequeno, lançado inicialmente em 2010. Pequeno o bastante para ser chamado de "*microframework*", afirma Grinberg [2014].

De acordo com a documentação do Flask [Ronacher, 2018], o fato de ele ser micro não significa que ele tenha alguma deficiência, mas que ele mantém seu núcleo simples e extensível.

No geral, *frameworks* tendem a guiar a forma de desenvolvimento do projeto. Já com este *microframework* não é bem isso que ocorre. De acordo com Dwyer et al. [2017], utilizando o Flask é possível escolher o que fazer, já que ele deixa funcionalidades mais avançadas como conexões a Banco de Dados para extensões especializadas.

Um dos pilares do Flask, conforme Grinberg [2014], é o Jinja2, um *plugin* para *templates*. De acordo com Dwyer et al. [2017], o Jinja permite a definição de blocos dinâmicos em HTML, que poderão ser populados futuramente. O Flask foi construído em cima do Jinja, o que faz com que eles tenham métodos específicos para trabalhar em conjunto. Uma vez criado o *template*, os códigos do Flask são utilizados para

renderizar a página, analisando o código Jinja, inserindo qualquer dado dinâmico e criando o HTML a ser retornado para o usuário.

4.3.2 Desenvolvimento WEB da ferramenta

Diante do exposto anteriormente, a linguagem Python foi utilizada como principal recurso no *back-end* (processamento interno do servidor) e no *front-end* (em essência, parte do sistema que interage com o usuário) foram utilizadas as linguagens Python (com Flask e Jinja2), JavaScript (com jQuery) e Bootstrap (conjunto de ferramentas para facilitar o desenvolvimento de interfaces HTML com CSS e JavaScript).

Como será detalhado em uma seção específica, todos os dados do EasyVS estão armazenados em um Banco de Dados PostgreSQL (exceto os arquivos originais em formato PDB das proteínas-alvo).

4.4 Agrupamento molecular

Como já citado, uma das características do EasyVS é a possibilidade do usuário ter acesso a um vasto número de moléculas para estudo e *docking* individual a partir do AutoDock Vina [Trott & Olson, 2010].

Como detalhado na seção 2.1.1, o processo de *docking* é demorado e custoso computacionalmente, o que torna impeditivo o *docking* de todas as moléculas armazenadas no EasyVS. Por isso, as moléculas filtradas pelo usuário formam um espaço químico personalizado, podendo ser agrupadas conforme algumas características individuais, descritas a partir de um *fingerprint*, como detalhado a seguir.

Devido à grande quantidade de moléculas disponíveis no EasyVS, antes do agrupamento, as moléculas foram divididas em 8 grupos conforme peso molecular e LogP (ambas propriedades calculadas pelo RDKit). No caso, todas as moléculas foram divididas em quartis de acordo com peso molecular e, cada um desses grupos, divididos em 2 outros grupos conforme LogP. Esses número foram calculados após diversos testes com agrupamento, pois, após essa divisão os grupos gerados foram de cerca de 2.1 milhões de moléculas, tornando viável, em tempo e utilização de memória pelos servidores, o agrupamento dessa quantidade de moléculas.

Optou-se por essa divisão também para realizar o pré-processamento de todos os agrupamentos possíveis para permitir ao usuário menor tempo de resposta na utilização do EasyVS. Sendo assim, na atual versão do sistema, as moléculas filtradas pelo usuário são agrupadas conforme dados já disponíveis na ferramenta, não sendo necessário a

espera pela execução do algoritmo, que poderia ser de muitas horas ou dias, dependendo do número de moléculas.

4.4.1 *Fingerprint*

Conforme Cortes-Cabrera et al. [2016], o conceito de *fingerprint* molecular é referenciado na década de 1980, quando foram desenvolvidos algoritmos para identificar propriedades estruturais moleculares, tais como presença de determinados átomos, tipos de ligação e fragmentos para, então, comparar as moléculas e encontrar similaridade entre elas. A Figura 4.6 ilustra um exemplo de um *fingerprint* baseado na presença (1) ou não (0) de determinada subestrutura na molécula em questão. Por serem armazenados somente 0s e 1s, os valores puderam ser guardados em estruturas binárias (comumente denominadas de *bitstrings*), o que torna a comparação entre *fingerprints* muito rápida.

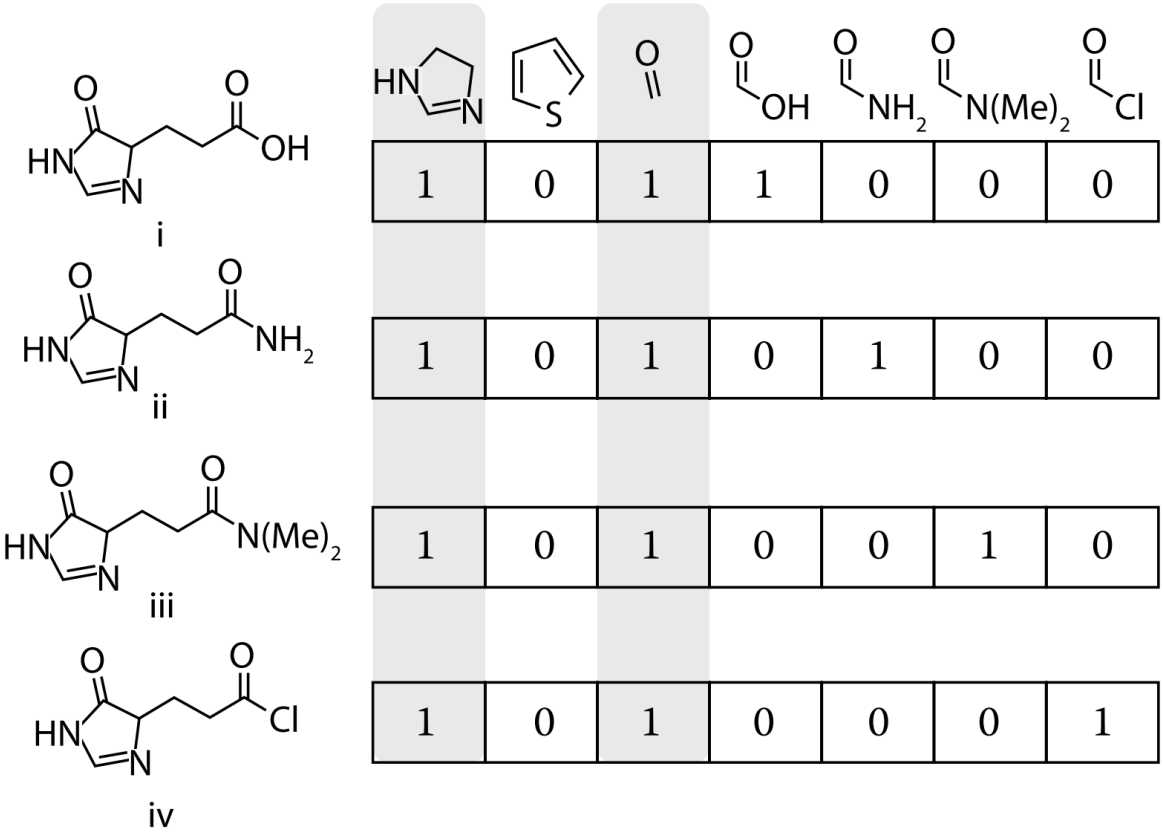

Figura 4.6. Exemplo de *fingerprint* para encontrar similaridade entre moléculas. Fonte: [Cortes-Cabrera et al., 2016, p.111].

Conforme Cereto-Massagué et al. [2015], um dos passos mais importantes para medir a similaridade entre moléculas, etapa essencial no agrupamento, é a escolha da

representação molecular e sua complexidade. Quanto maior a abstração na representação molecular, mais fácil e rápida será a comparação. Porém, representações mais detalhadas representam a molécula com maior fidelidade e individualidade. O que aumenta, por consequência e de forma considerável, o custo para comparação (tanto de memória quanto de tempo).

A maior parte dos *fingerprints* moleculares utilizam métodos que transformam a molécula em uma sequência de *bits* que indicam se determinada subestrutura está, ou não, presente na molécula analisada. Os principais tipos de *fingerprints* e alguns dos seus representantes são descritos nas seções a seguir.

4.4.1.1 *Fingerprints* baseados em estrutura-chave

Esses tipos de fingerprints identificam, com um *bit* (1 ou 0) a presença ou não de determinada subestrutura, respectivamente.

Representantes desse tipo de fingerprint são o MACCS 166 ou o 960 [Durant et al., 2002]. O que diferencia entre eles é a quantidade de *bits* a ser utilizada (166 no primeiro caso e 960 no segundo). Destaca-se: quanto maior a quantidade de descritores de um *fingerprint* (também denominado *features*), mais detalhada será a representação, porém, mais custoso serão seu processamento e armazenamento. Esse *fingerprint* utiliza de SMARTS (linguagem específica para descrever padrões moleculares) para o estabelecimento dos descritores. Uma característica relevante desse *fingerprint* é que cada um dos seus *bits* indica uma subestrutura específica[13], permitindo ao utilizador saber quais estruturas estão presentes ou não (essa característica não é presente em todos os *fingerprints*, como será citado em outros exemplos).

Outro *fingerprint* que pode ser citado é o PubChem [Bolton et al., 2008], utilizado na base de dados de moléculas homônima[14]. Esse *fingerprint* contém 881 *bits* e é utilizado nos mais de noventa milhões de compostos cadastrados para a pesquisa por similaridade dentro do site.

Já o TGD e o TGT *fingerprints* [Sheridan et al., 1996], contendo 735 e 13.824 *bits*, respectivamente, utilizam pares de átomos como descritores usando sete *features* e distância de até quinze ligações entre eles.

[13] A definição dos *bits* utilizada no *fingerprint* MACCS166, gerado pelo RDKit, está acessível em `http://www.scbdd.com/pybel_desc/fps-maccs/`. Acesso em: 8 de jul. 2018.

[14] Acessível em `https://pubchem.ncbi.nlm.nih.gov/`. Acesso em: 8 de jul. 2018.

4.4.1.2 *Fingerprints* topológicos ou baseados em caminho

Esse tipo de *fingerprint* percorre a molécula, linearmente, a partir de um átomo; e vai dividindo cada parte percorrida da molécula como uma estrutura.

Um exemplo da representação molecular a partir desse tipo de técnica pode ser visto na Figura 4.7. Nela é possível ver que, a técnica consiste em percorrer os caminhos a partir de um átomo de origem e assinalar cada caminho como um *bit* do *fingerprint*. Esse tipo de técnica pode ser utilizada para busca de subestruturas em moléculas.

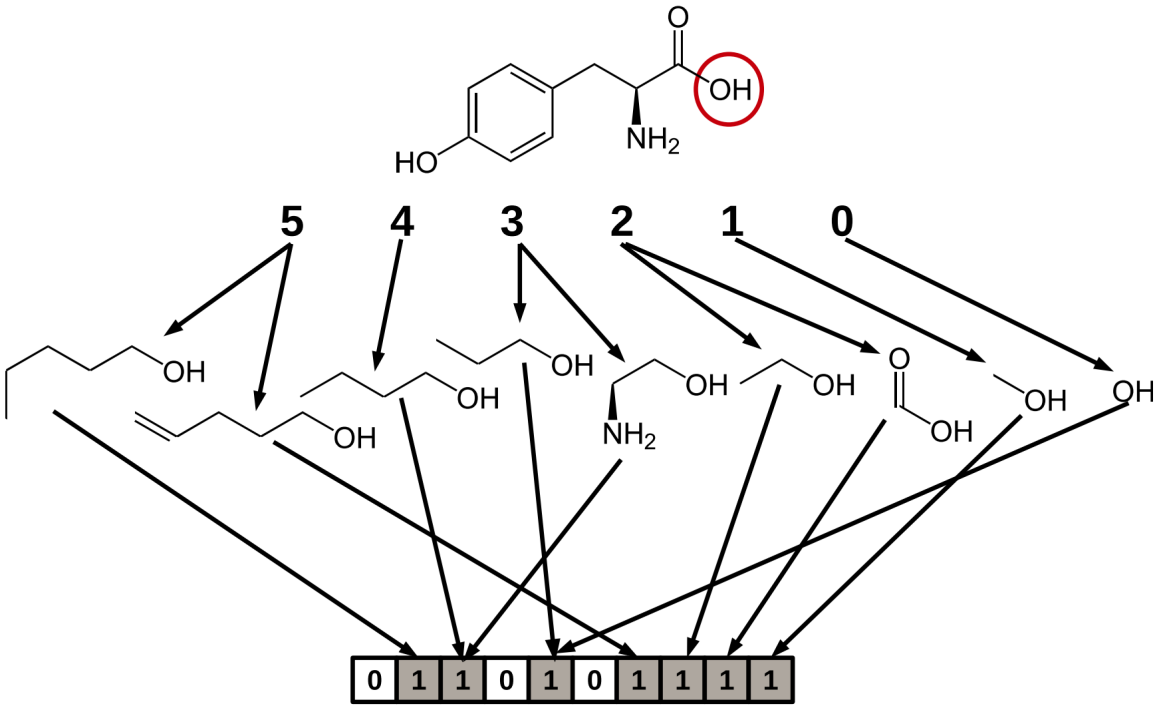

Figura 4.7. Exemplo da construção de um *fingerprint* topológico com distância máxima de cinco ligações. Fonte: [Cereto-Massagué et al., 2015, p.59].

O Daylight *fingerprint* [Daylight, 2008] é um representante desse tipo, com 2.048 *bits* de precisão.

4.4.1.3 *Fingerprints* circulares

Os *fingerprints* circulares, diferentemente dos topológicos, não percorrem a molécula linearmente, mas a partir de um raio e procuram, a cada átomo, se tal átomo faz parte de uma estrutura (como representado na Figura 4.8).

Nessa classificação, de acordo com Cereto-Massagué et al. [2015], os tipos mais utilizados são o ECFP e o FCFP. O ECFP (*Extended-Connectivity Fingerprint*) é baseado no algoritmo de Morgan [1965] e pode registrar tanto a presença de uma

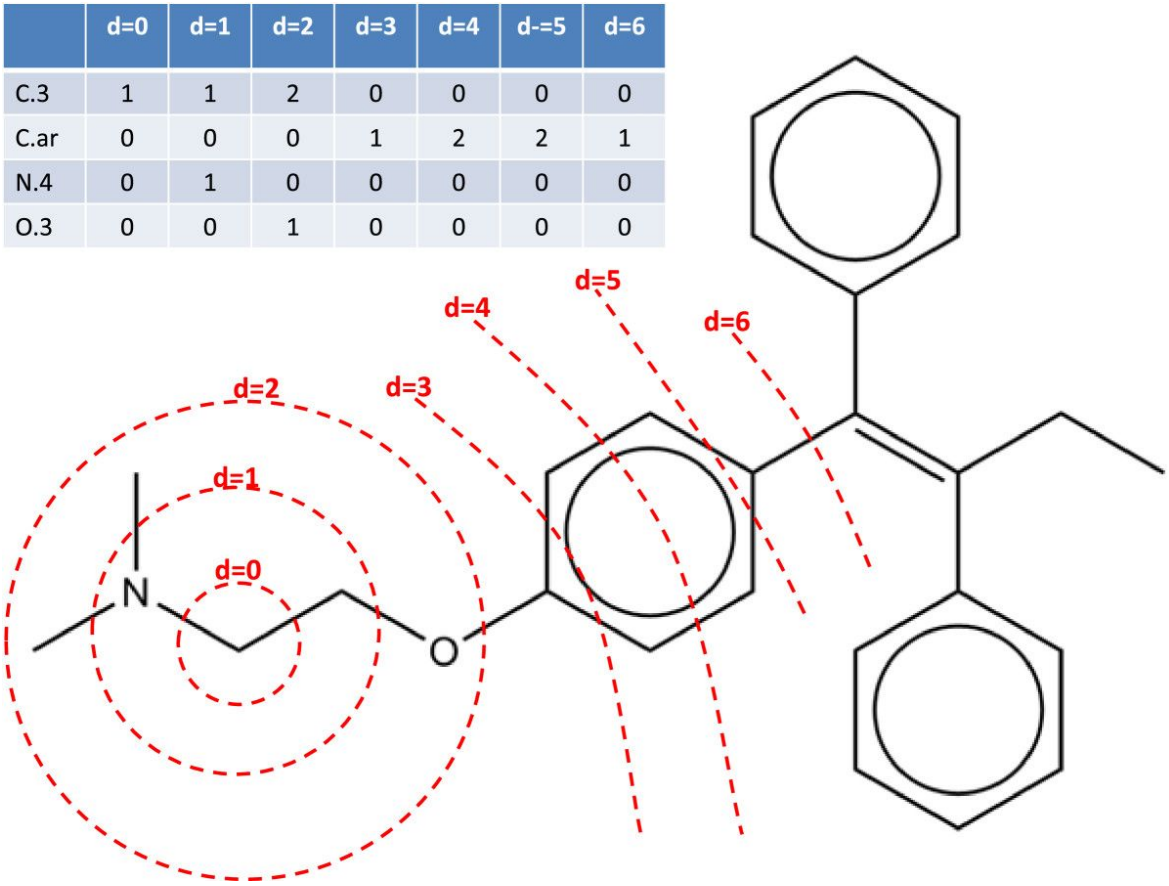

Figura 4.8. Exemplo da construção de um *fingerprint* circular com distância máxima de seis ligações. Fonte: [Tyzack et al., 2014, p.4]

estrutura a partir de um *bit* (ainda denominado ECFP, porém significando *Extended-Connectivity Bit String*) ou a frequência que essa estrutura aparece na molécula (nesse caso, a denominação passa a ser ECFC - *Extended-Connectivity Count Vector*). O RDKit Landrum [2006] implementa essa técnica, referindo-o como Morgan *fingerprint*.

Já o FCFP (*Functional-Class Fingerprint*) é uma variação do ECFP que, em vez de separar cada átomo em um *key*/*feature* diferente, agrupa átomos que possuem funções similares em uma mesma *key* (cada implementação dessa técnica estabelece seu conjunto de átomos similares). Seria como se fosse ter uma só *key* para estruturas similares (quando os átomos que se diferem forem substituídos e a função da estrutura não for alterada ou os átomos forem da mesma "classe"ou "grupo").

4.4.1.4 *Fingerprints* baseados em farmacóforos

Um farmacóforo representa características relevantes e interações necessárias para que uma molécula seja ativa, dado determinado alvo [Cereto-Massagué et al., 2015].

Os algoritmos desse *fingerprint* levam em consideração a distância entre os farmacóforos, podendo ser, inclusive, distância tridimensional entre eles (desde que a molécula esteja representada nesse tipo de coordenada).

4.4.2 Algoritmo de agrupamento Butina

O algoritmo de agrupamento, ou *clustering* de Butina [1999], é muito utilizado na área de bioinformática e quimioinformática para a classificação não-supervisionada de moléculas. A sua implementação pode ser dividida em três etapas:

1. gerar os *fingerprints* para cada indivíduo a ser agrupado;
2. identificar todos os prováveis indivíduos que podem ser o centroide de um grupo ou *cluster* e seus prováveis vizinhos;
3. excluir todos indivíduos repetidos entre os *clusters*, começando pelos que têm maior quantidade de itens.

Primeiramente, é calculado o índice de dissimilaridade entre as moléculas a partir do *fingerprint*. Essa dissimilaridade representa, de zero a 100%, o quanto uma molécula difere da outra. Esse valor é obtido a partir do índice de Tanimoto, descrito por Butina [1999] e apresentado nas equações 4.1 e 4.2.

$$\text{Índice de Tanimoto} = \frac{BC}{B1 + B2 - BC} \tag{4.1}$$

$$\text{Índice de Dissimilaridade} = 1 - \text{Índice de Tanimoto} \tag{4.2}$$

Nessas equações referidas, BC representa número de *bits* com o valor 1 que são iguais entre as duas *fingerprints*, B1 é o número de *bits* com o valor 1 no primeiro *fingerprint* e B2 o número de *bits* com o valor 1 no segundo *fingerprint*.

Logo após é feito uma lista com todos os possíveis vizinhos de cada molécula, considerando o índice de dissimilaridade e um valor de limiar/*cutoff*, definido previamente. Ao comparar uma molécula com outra, caso o valor de dissimilaridade esteja dentro do limiar, a molécula que está sendo comparada deverá ser inserida na de referência.

A próxima etapa consiste na ordenação da lista de vizinhos (possíveis *clusters*) de cada molécula, indo do maior conjunto de vizinhos ao menor. Caso alguma molécula de uma lista com maior quantidade de vizinhos esteja em uma lista menor, essa será retirada da lista com menor quantidade. Caso seja necessário, é feito a reordenação das listas afetadas.

Esse processo é denominado esfera de exclusão e, conforme Butina [1999], faz com que os elementos disputados por dois ou mais possíveis *clusters* integrem o *cluster* com maior número de elementos.

É salientado no artigo de referência desse algoritmo que essa técnica pode gerar *singletons* (grupos com somente 1 elemento), pois, *clusters* com mais elementos tem prioridade. A Figura 4.9 representa uma situação em que três moléculas (enumeradas como 1, 2 e 3) estão em grupos distintos. A característica da esfera de exclusão faz com que as moléculas 1 e 2, mesmo sendo mais distantes de alguns representantes de seu grupo que a molécula 3 (em especial, das moléculas mais próximas do limiar do grupo), por pertencerem a *clusters* maiores, acabam fazendo com que essas moléculas estejam nos grupos maiores, criando *singleton* (molécula 3 sendo único representante de seu grupo).

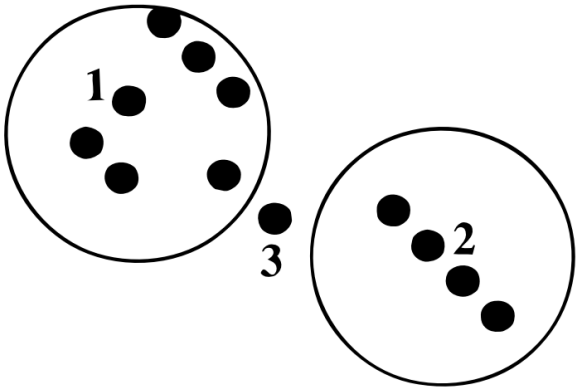

Figura 4.9. Representação de três grupos, sendo a molécula 3, *singleton*. Fonte: [Butina, 1999, p.748].

É importante destacar que o algoritmo Butina [1999] não necessita de um número de *clusters* para o agrupamento, e sim um índice de dissimilaridade, o que reflete diretamente na homogeneidade dos grupos criados (todos eles deverão estar conforme o *cutoff* estabelecido).

4.4.3 Melhorias no algoritmo de agrupamento Butina do RDKit

O RDKit [Landrum, 2006] realiza o agrupamento de grande quantidade de moléculas utilizando o algoritmo baseado no trabalho do Butina [1999]. Na atual forma dessa implementação, armazenamento dos *fingerprints*, similaridades e agrupamentos conforme *cutoff*, o código só consegue, de acordo com testes, processar 50 mil moléculas por vez em uma máquina de 64GB de memória RAM.

Foi necessário realizar a mudança de um algoritmo original para agrupamento do RDKit, para possibilitar o agrupamento de maior quantidade de moléculas de uma só vez (almeja-se a execução de um único agrupamento para todas as moléculas do EasyVS). A separação das moléculas em conjuntos não é ideal pois moléculas que poderiam ser classificadas em um mesmo grupo, caso estejam em conjuntos distintos (compostos de 50 mil, separados inicialmente), estarão em grupos distintos.

Após a nova implementação, com código paralelizado e utilizando a linguagem de programação C++, mantendo o mesmo resultado do algoritmo do RDKit, o código foi testado e ficou 28 vezes mais rápido e consumiu aproximadamente 500 vezes menos memória. Na descrição dos trabalhos futuros (seção 6.1.1) são citados detalhes da implementação utilizando CUDA C, que realiza o processamento dos cálculos utilizando placas de vídeo da NVidia, cuja otimização está sendo testada e ultrapassa 700 vezes a velocidade de processamento e 20 mil vezes menos memória utilizada.

4.5 *Docking* molecular

Como exposto na seção 2.1.1, o *docking* ou atracamento molecular consiste na utilização de um *software* para estimar uma pose e energia de ligação entre uma proteína-alvo e uma molécula (geralmente pequena), aqui denominada de ligante.

No caso do EasyVS, esse processo é feito pelo AutoDock Vina [Trott & Olson, 2010]. A escolha e detalhamento dessa ferramenta é descrita a seguir, assim como os passos para a preparação das moléculas envolvidas.

4.5.1 AutoDock Vina

O AutoDock Vina, conforme Trott & Olson [2010], é uma nova geração do AutoDock 4 [Morris et al., 2009], também desenvolvido pelos mesmos autores, com melhorias na rapidez dos resultados (aproximadamente duas ordens de grandeza mais rápido) e também uma significante melhoria da precisão do modo de ligação entre as moléculas envolvidas.

O código do AutoDock Vina é paralelizado, de forma que ele consegue dividir seus cálculos em múltiplos núcleos do processador (*cores*) e os executa repetidas vezes, conforme números aleatórios que são gerados (a quantidade de repetições e número de cálculos simultâneos é diretamente relacionada ao valor de *exhaustiveness*). Sabe-se que seu algoritmo realiza a busca da melhor pose, que para esse *software* implica em menor energia livre. Para cada pose é gerada uma pontuação/*score*, levando em consideração campos de força, que podem ser calculados a partir de potenciais simplificados

e aproximados envolvendo ligações de Van der Waals e eletrostáticas, além de outros termos.

A parte da superfície em que o AutoDock Vina realiza a busca é especificada a partir das coordenadas de uma caixa (denominada *docking box* ou mesmo *box*). Essas coordenadas determinam o espaço de busca do algoritmo.

Uma vez que a busca é iniciada a partir de um valor numérico aleatório (podendo também ser informado pelo usuário), denominado semente/*seed*, a reexecução de um experimento contendo um par proteína-ligante pode não gerar o mesmo resultado (já, no caso da inserção do valor da semente, o experimento poderá ser reproduzido).

O AutoDock Vina tem como entrada o receptor e ligante como arquivos no formato PDBQT (como será detalhado nas próximas seções), além de necessitar das coordenadas do *box* (tamanho e posição no espaço tridimensional), número máximo de poses a serem geradas, variação de energia (*energy range* é o valor máximo de energia entre a melhor pose predita e a pior, dentre o número máximo de poses), e *exhaustiveness* (já detalhado anteriormente). Ao fim da execução, também é gerado um arquivo, com diversos *models*, ou conjunto de coordenadas, no formato PDBQT contendo todas as poses preditas, juntamente com a estimativa de energia.

No caso do EasyVS, ao fim do processo de *docking*, os arquivos resultantes, incluindo os originados do NNScore 2.01 [Durrant & McCammon, 2011], usado para nova estimativa de energia de ligação e valor de afinidade/Kd, são salvos em uma pasta conforme um código identificador.

Salienta-se que o AutoDock 4 [Morris et al., 2009], diferentemente da sua versão derivada, o AutoDock Vina [Trott & Olson, 2010], provê dados para uma estimativa de Kd, sem a necessidade de uma ferramenta externa como o NNScore 2.01 para esse cálculo.

4.5.2 Processo de preparação do *target*

No caso do EasyVS, a preparação da proteína (*target* ou receptor) começa removendo todos os heteroátomos do arquivo PDB original (pretende-se, em futuras versões do sistema, permitir ao usuário a escolha dos heteroátomos que não serão removidos). Como se pretende fazer o *docking* de novas moléculas, a superfície da proteína precisa estar livre de outras moléculas (em especial, dos ligantes cristalográficos) para que o *docking* aconteça. Após essa etapa, o arquivo PDB é convertido para PDBQT a partir do MGLTools (utilizando o *script* prepare_receptor4.py[15], que, pela configuração

[15]Encontrado em `https://github.com/sahrendt0/Scripts/blob/master/docking_scripts/prepare_receptor4.py`. Acesso em: 8 de jul. 2018.

padrão, remove moléculas de água e adiciona átomos de hidrogênio.

O PDBQT permite que sejam armazenadas, além das coordenadas atômicas da molécula nesse formato, cargas parciais (o formato PDBQT permite armazenar as cargas parciais, porém, o AutoDock Vina não as utiliza) e tipo atômico do AutoDock, para cada um dos átomos do ligante (marcados como HETATM) e do receptor (marcados como ATM). Esse tipo atômico do AutoDock permite, por exemplo, diferenciar um nitrogênio que aceita ligação de hidrogênio de um que não aceita esse tipo de ligação. Arquivos no formato PDBQT são essenciais para o *docking* com o AutoDock e AutoDock Vina.

Com o arquivo PDBQT da proteína pronto para o *docking*, basta realizar a preparação dos arquivos dos *hits* e do arquivo de configuração (contendo os parâmetros do *docking* já citados).

4.5.3 Processo de preparação das moléculas candidatas

Cada uma das moléculas, provenientes de diversas bibliotecas, são lidas pelo RDKit que, a partir dos códigos a seguir, adiciona átomos de hidrogênio (polares), busca obter as coordenadas iniciais da molécula, realiza otimização da conformação gerada e salva a nova molécula, com coordenadas tridimensionais, em um arquivo no formato SDF.

```
1 mol = Chem.MolFromInchi(inchi_of_hit)
2 newmol = Chem.AddHs(mol)
3 mol_conforms = AllChem.EmbedMolecule(newmol)
4 if mol_conforms < 0:
5     mol_conforms = AllChem.EmbedMolecule(newmol, useRandomCoords=True)
6 AllChem.UFFOptimizeMolecule(newmol, 200, 10.0, -1)
7 writer = Chem.SDWriter(folder + 'hit' + hitid +'3d.sdf')
8 writer.write(newmol)
9 writer.close()
```

Com a conformação tridimensional gerada, a molécula é convertida para o formato MOL2 pelo OpenBabel [O'Boyle et al., 2011] e, logo após, para PDBQT pelo *script* do MGLTools prepare_ligand4.py[16] e está pronta para o *docking*.

4.5.4 Processo de *docking*

Tendo os arquivos do *target* e dos *hits* envolvidos, para execução do *docking* basta executar o AutoDock Vina Trott & Olson [2010] passando os parâmetros selecionados

[16]Disponibilizado em `https://github.com/sahrendt0/Scripts/blob/master/docking_scripts/prepare_ligand4.py`. Acesso em: 8 de jul. 2018.

como configuração. Geralmente todas as configurações são salvas em 1 único arquivo, que contém a lista de parâmetros desejados e seus valores.

A lista de parâmetros para o *docking* foi detalhada na seção 4.2.2. Em adição a esses, há a definição dos nomes e diretórios para localização dos arquivos do *target* e de cada um dos *hits* (1 para cada execução do *docking* com o AutoDock Vina). Ao término da execução é gerado um arquivo no formato PDBQT para cada resultado da combinação alvo-ligante. Nesse arquivo estão todos os átomos, compondo as poses preditas para cada *hit*, energia estimada da referida pose (em kcal/mol) e o valor de RMSD de cada uma das poses em comparação ao melhor resultado (com menor energia predita).

4.5.5 Processo de *rescoring*

Após a finalização do processo do *docking* pelo AutoDock Vina [Trott & Olson, 2010], os arquivos PDBQT das poses preditas são passados para o NNScore 2.01 [Durrant & McCammon, 2011].

O NNScore 2 utiliza vinte redes neurais treinadas para os cálculos relativos à avaliação das conformações resultantes do *docking*. O NNScore 2.01, versão mais recente e utilizada no EasyVS, gera, ao fim dos cálculos, três valores: *score*, *standard deviation* (desvio padrão) e valor estimado de Kd (constante de dissociação que indica a concentração de ligante para ocupar 50% dos receptores). Os valores de *score* e Kd são calculados para cada uma das vinte funções de pontuação que o NNScore 2 utiliza e, por fim, é apresentado o melhor valor de *score* e Kd e os valores médios dentre as pontuações, assim como o desvio padrão dentre esses valores.

Ao término de sua execução essa ferramenta gera um relatório detalhado dos cálculos que foram realizados e seus resultados. Esse próprio relatório sugere que o valor médio, fornecido por ele, seja utilizado como métrica de avaliação das poses (é esse o valor apresentado ao usuário no momento de visualização dos resultados). O Kd médio é calculado a partir dos valores estimados entre as vinte redes neurais e todas as poses disponíveis no arquivo PDBQT utilizado como entrada no processo de *rescoring*.

O que é apresentado ao usuário do EasyVS é somente o valor estimado de Kd, convertido para nMolar[17]. Os demais valores gerados pelo NNScore 2.01 permanecem armazenados no sistema, assim como o arquivo resultante do *rescoring*.

[17] 1 Nanomolar = $1*10^{-9}$ Molar

4.5.6 Fonte dos dados do EasyVS

O EasyVS possui, atualmente, moléculas de diversas bases e, nessa seção todas elas são listadas, juntamente com a quantidade de registros disponíveis nas bibliotecas de compostos até o mês de maio de 2019.

Para facilitar o entendimento, no caso das bibliotecas de pequenas moléculas, os principais dados estão sumarizados na Tabela 4.2.

Tabela 4.2. Resumo das bibliotecas de pequenas moléculas disponíveis no EasyVS

Nome	Data da versão	Moléculas
ChEMBL	10/12/2018	1.8 milhões
Chembridge Express	28/05/2019	500 mil
Chembridge Core	28/05/2019	700 mil
DrugBank	02/04/2019	10 mil
HMDB	09/07/2018	114 mil
Maybridge	15/05/2019	57 mil
Super Natural II	04/2018	325 mil
Zinc	01/06/2019	13 milhões

4.5.6.1 RCSB PDB

O RCSB PDB [Berman et al., 2000], denominado *Protein Data Bank* é a principal origem das proteínas cadastradas no EasyVS. Exceto as proteínas enviadas pelo usuário a partir da tela inicial do sistema, todas são originadas do RCSB PDB, que possui mais de 142 mil proteínas depositadas (no momento da escrita desta tese).

4.5.6.2 ChEMBL

O ChEMBL [Gaulton et al., 2017] provê um conjunto de 1.879.206 moléculas (disponíveis em 1.870.461 arquivos) na versão 25, disponibilizada em 10 de dezembro de 2018.

A característica do ChEMBL é sustentar um conjunto de moléculas bioativas, pequenas e *drug-like*, além de algumas moléculas terem, inseridas no arquivo SDF, propriedades calculadas como LogP, peso molecular dentre outras. E o RDKit [Landrum, 2006] é utilizado para o cálculos dessas propriedades.

4.5.6.3 Chembridge

O Chembridge [Desai et al., 2004], diferentemente de outras bases de moléculas citadas nesse documento, possui duas bases distintas de moléculas, voltadas para *high-throughput screening* e *fragment-based screening*. Essas moléculas estão disponíveis aos usuários cadastrados no sistema deles e, pelo seu identificador, é possível acessar uma página com *link* para compra da maioria dos compostos.

A versão do Chembridge utilizada no EasyVS, para as duas biliotecas de compostos, contém as moléculas disponibilizadas no dia 28 de maio de 2019.

O primeiro conjunto de moléculas, denominado de *EXPRESS-Pick Collection Stock* (no EasyVS é referenciado somente como *Chembridge Express*), possui mais de quinhentos mil compostos feitos à mão, incluindo fragmentos. Segundo os mantenedores, esse conjunto foi elaborado a partir de análise de propriedades de fármacos, levando em consideração a diversidade, inovação e quase vinte anos de experiência do grupo.

Já o Chembridge *CORE Library Stock* (referenciado como *Chembridge Core*) possui mais de setecentos mil compostos voltados para gerar um espaço químico não coberto pela *EXPRESS-Pick Collection Stock* ou qualquer outra biblioteca comercial de moléculas existente.

Esses conjuntos de moléculas têm a característica de ser voltados para o *Virtual Screening* e também ter um *link* para acesso e compra da maior parte das moléculas, individualmente, direcionando para a loja virtual deles[18].

4.5.6.4 DrugBank

Wishart et al. [2018a] apresenta versão do DrugBank 5.1.3, datada de 2 de abril de 2019, disponível do EasyVS. Essa é uma biblioteca de compostos, iniciada em 2006, contendo 10.256 fármacos, mecanismos de atuação, interações e alvos. Vale salientar que mais da metade (5.766) é composta de fármacos marcados como experimentais, sendo que 205 são ilícitos e 3.732 são aprovadas.

Todas as moléculas do DrugBank, inseridas no EasyVS, são identificadas pelo seu número identificador (iniciado com os caracteres "DB"). E, assim como as moléculas do ChEMBL, HMDB, Zinc e Chembridge, quando o usuário visualiza a molécula no sistema, ele conseguirá acessar a molécula diretamente no *site* de onde ela foi obtida, podendo ter mais informações ou mesmo comprar o composto.

[18] Acessível em `https://www.hit2lead.com`. Acesso em: 8 de jul. 2018.

4.5.6.5 HMDB

O HMDB, acrônimo de *The Human Metabolome Database*, é uma biblioteca de 114.100 metabólitos humanos [Wishart et al., 2018b], disponíveis em 09 de julho de 2018. O próprio artigo do HMDB 4.0 define o metaboloma humano como a coleção completa de "pequenas moléculas encontradas no corpo humano incluindo peptídios, lipídios, aminoácidos, ácidos nucleicos, carboidratos, ácidos orgânicos, aminas biogênicas, vitaminas, minerais, aditivos alimentares, drogas, cosméticos, contaminantes, poluentes e qualquer outro produto químico que os humanos ingerem, metabolizam, catalisam ou entram em contato"[Wishart et al., 2018b, p. 608].

Todas as moléculas do HMDB estão disponíveis gratuitamente sem a necessidade de cadastro prévio e em formato SDF ou XML.

4.5.6.6 Maybridge

O Maybridge[19] contém diversos conjuntos de moléculas e, desses, os selecionados para comporem o EasyVS são os denominados *Screening* e *Fragment Libraries* e, somados, são mais de 57 mil compostos (disponíveis em 15 de maio de 2019). O primeiro conjunto contém compostos orgânicos produzidos pelo Maybridge, e projetados individualmente. Já o segundo conjunto é composto basicamente de fragmentos, voltados para grande diversidade, obedecendo aos filtros de RO3 (*Rule Of Three* ou *lead-like*). A RO3, como já explicado, é uma adequação do Lipinski RO5 [Lipinski et al., 2001] que, no filtro com *Rule of Three*, considera moléculas com LogP menor ou igual a três, massa molecular menor que 300 daltons e até três doadores de hidrogênio, aceptores de hidrogênio e ligações rotacionáveis.

4.5.6.7 Super Natural

O conjunto de moléculas denominado Super Natural [Dunkel et al., 2006], na sua versão Super Natural II [Banerjee et al., 2015], contém 325.508 moléculas naturais (mais recente atualização ocorrida em abril de 2018). São compostos sintetizados por organismos vivos. Essa biblioteca de moléculas, iniciada em 2006, provê dados sobre toxicidade além da conformação espacial 2D e dados físico-químicos como LogP, número de anéis, ligações rotacionáveis entre outros.

[19] Acessível a partir do endereço `https://www.maybridge.com`. Acesso em: 8 de jul. 2018.

4.5.6.8 Zinc

O Zinc [Sterling & Irwin, 2015] é a maior biblioteca de compostos disponível no EasyVS. A versão usada é a Zinc 15 que disponibiliza 13.229.915 compostos compráveis (que possuem, no site do Zinc, um endereço para acesso direto a um distribuidor do composto com possibilidade de compra) e disponíveis (que, em consulta a algum dos distribuidores do composto, este encontra-se em estoque), com coordenadas tridimensionais e preparadas para o *docking*.

Ao navegar no site do Zinc é perceptível um vasto número de moléculas disponíveis e de seus conjuntos. Optou-se pela utilização de compostos compráveis e disponíveis em estoque ao invés de somente compráveis (quantidade maior que 600 milhões).

Capítulo 5

Estudo de caso com GPCR

Receptores acoplados à proteína G (GPCRs - *G Protein-Coupled Receptors*) são proteínas caracterizadas por terem 7 domínios transmembrana, conforme apresentado na Figura 5.1. Eles são encontrados somente em eucariotos. Os ligantes desses receptores incluem compostos sensíveis a luz, odores, feromônios, hormônios e neurotransmissores, que variam em tamanho entre pequenas moléculas, peptídios, até grandes proteínas [Joost & Methner, 2002].

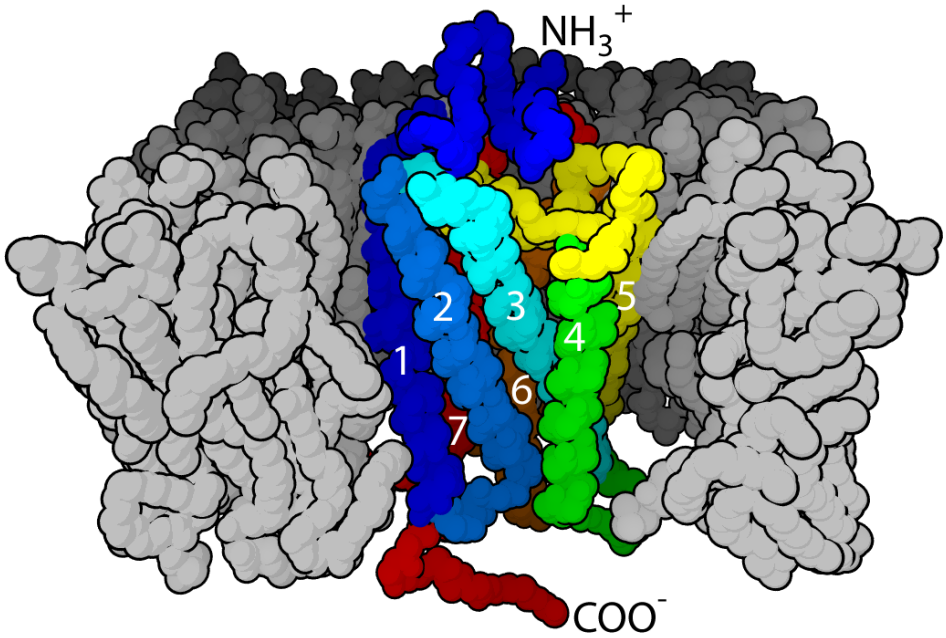

Figura 5.1. Exemplo dos sete domínios transmembrana de uma GPCR. Fonte: `https://en.wikipedia.org/wiki/G_protein-coupled_receptor`. Acesso em: 8 de jul. 2018.

Esses receptores estão envolvidos na transdução de sinal de muitos processos fi-

siológicos vitais, dos quais podemos citar: controle celular de divisão e proliferação, modulação de sinapse neuronal, regulação de transporte de íons, homeostases e modificação de morfologia celular [Salon et al., 2011]. Esses receptores compõem uma superfamília de proteínas contendo, de acordo com análise genômica, 800 genes em humanos [Niimura, 2009].

A superfamília foi dividida inicialmente em três principais classes: A, B e C. Nos últimos anos, além dessas 3 classes, foram adicionadas mais 2 (Taste2 e F) como demonstrado na Figura 5.2. Na Figura 5.2 também é possível visualizar as várias subdivisões dentro das classes. A maior família é a classe A, contendo aproximadamente 85% das GPCRs. Metade dos receptores da classe A são receptores olfativos, o restante são ativados por moléculas endógenas ou são classificados como órfãos. Os receptores órfãos não possuem ligantes conhecidos [Trzaskowski et al., 2012]. A família das GPCRs estão também envolvidas em patologias como Parkinson, Alzheimer, doenças cardiovasculares, asma, diabetes e em muitas outras [Salon et al., 2011].

5.1 Início dos estudos da GPCR

O modo como as células podiam sentir o ambiente ficou obscuro por muito tempo. Cientistas há muito tempo já sabiam que moléculas têm efeitos de grande impacto nos organismos, como por exemplo, o hormônio adrenalina, capaz de aumentar a pressão sanguínea e causar aceleração de batimento cardíaco. Suspeitava-se de pequenas proteínas de superfície celular capazes de interagir com essas moléculas.

Em 1968 Lefkowitz e seus colaboradores incorporaram o isótopo Iodo-125 [Lefkowitz et al., 1970] à vários hormônios, e graças a isso descobriram vários receptores. Em 1980, nessa mesma equipe, Kobilka conseguiu isolar o gene de um receptor beta-adrenérgico e após extensivas análises, constatou que era similar ao que capturava luz nos olhos. Essa equipe de pesquisadores percebeu a existência de uma família de receptores que se parecem e funcionam da mesma maneira. Referiram-se a essa família como receptores acoplados à proteína G. Em 2012, Lefkowitz e Kobilka ganharam o prêmio Nobel de química por esses trabalhos [Roth & Marshall, 2012].

5.2 Mecanismo de ação envolvendo a GPCR

A estrutura terciária desses receptores apresenta grande diversidade. Em termos estruturais, são caracterizadas por uma uma porção N-terminal extracelular, seguida de sete α-hélices transmembrana conectadas por três *loops* intracelulares e três *loops*

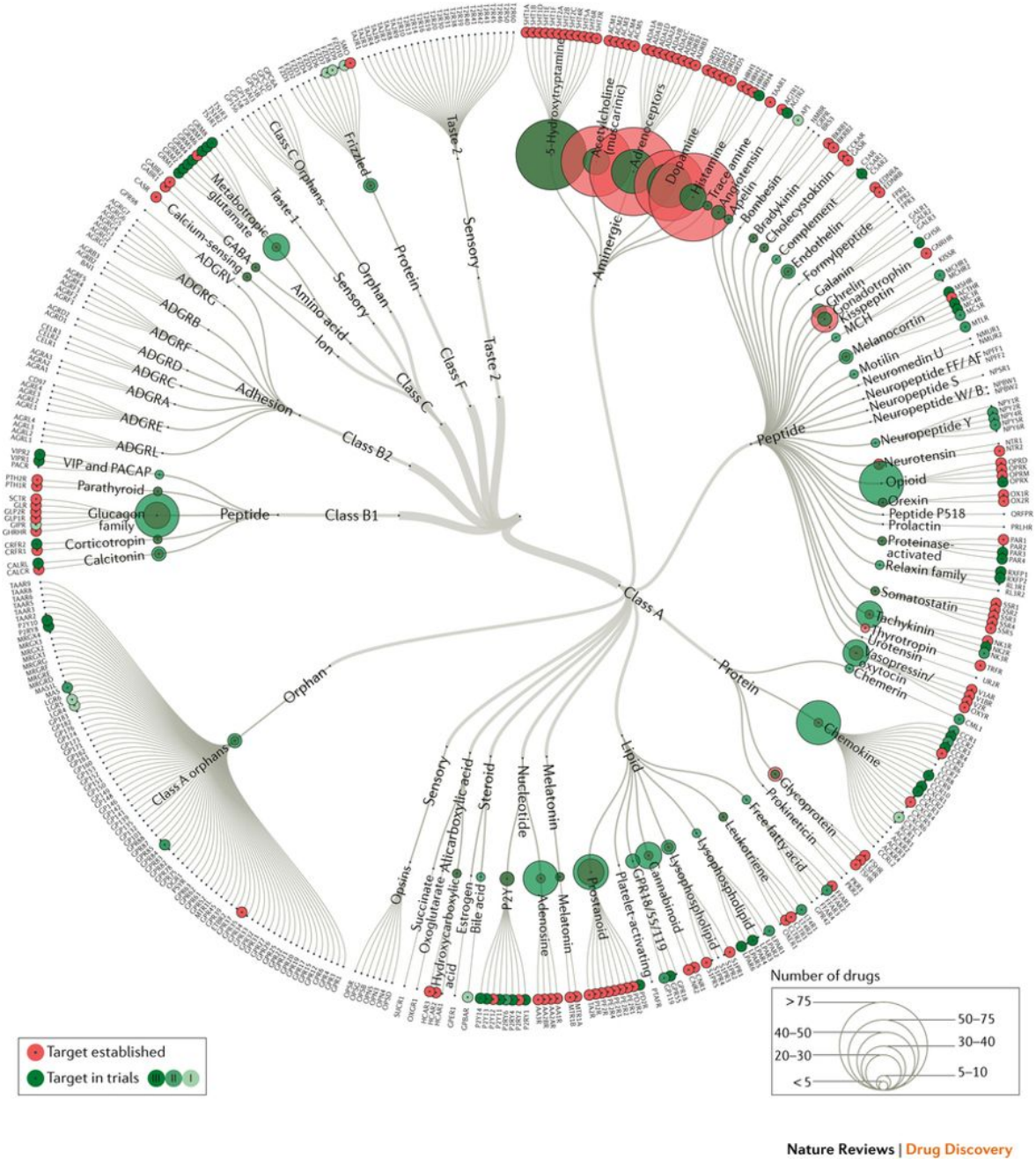

Figura 5.2. Divisão das classes da superfamília das GPCRs. Fonte: [Hauser et al., 2017]

extracelulares e, por último, uma porção C-terminal intracelular. A organização da estrutura terciária desses receptores se assemelha ao formato de um barril, de modo que os sete domínios transmembranas formam uma cavidade na membrana celular, local onde ocorre a interação com os ligantes [Trzaskowski et al., 2012].

A GPCR em seu estado inativo está ligada a um complexo heterotrimérico cha-

mado proteína G. A interação do ligante com a GPCR causa uma mudança conformacional no receptor que é transmitida a subunidade Gα da proteína G. A subunidade Gα ativada tem sua molécula GDP trocado por uma molecula de GTP. Essa mudança leva a dissociação da subunidade Gα do complexo, restando no complexo a subunidade dimérica Gβγ. A subunidade Gα dissociada e a subunidade Gβγ interagem com outras proteínas intracelulares gerando cascatas de sinalização, através de duas rotas bioquímicas possíveis: cAMP e fosfatidilinositol [Digby et al., 2006].

5.3 Resultados da GPCR pelo EasyVS

Nessa seção são detalhados alguns resultados obtidos, a partir do EasyVS, visando a validação da ferramenta. Para tanto foi feito o *redocking* dos ligantes cristalográficos de algumas estruturas e, nessas mesmas, foi feito um estudo a partir de *decoys* gerados pelo DUD-E [Mysinger et al., 2012], detalhado mais adiante.

5.3.1 Resultados do *redocking* com GPCR

Conforme Verli [2014], o *redocking* consiste na realização de estudos objetivando reproduzir determinados estudos experimentais do complexo receptor-ligante. Nesses casos, geralmente, são analisados valores de *Root Mean Square deviation* (RMSD) entre as poses preditas e as experimentais além de estimar a afinidade entre a proteína e o ligante.

Para melhor detalhamento dos pontos supracitados, a descrição dos resultados da afinidade estimada das poses obtidas pelo EasyVS será feita na seção 5.3.2, juntamente com a apresentação dos resultados dos ligantes em comparação com os *decoys*.

As estruturas proteicas-alvo selecionadas são GPCRs que possuem um ligante cristalográfico disponível na estrutura. Como já explicado na seção 2.1.1, Sotriffer [2016] apresenta um estudo em que verificou-se que os programas de *docking* analisados são capazes de, a partir de uma análise proteína-ligante, gerar conformações similares às cristalográficas. O estudo aqui apresentado corrobora com esse trabalho e também valida os parâmetros *defaults* disponíveis no EasyVS, uma vez que nenhum deles foi alterado, exceto a escolha do *pocket* adequado, conforme já exposto.

Aqui serão graficamente apresentados os resultados do *redocking*. Em todos os casos o ligante cristalográfico será colorido de azul e a pose predita de vermelho e as distâncias destacadas são entre os mesmos átomos nas moléculas referidas. Primeiramente a molécula GBK ligada à proteína 6HLL teve *Dissociation Constant* ou Constante de dissociação (Kd) estimado em 28,79nM e RMSD 0,81Å(valores calculados levando em

consideração a melhor pose conforme o *docking*). A melhor pose pode ser visualizada na Figura 5.3.

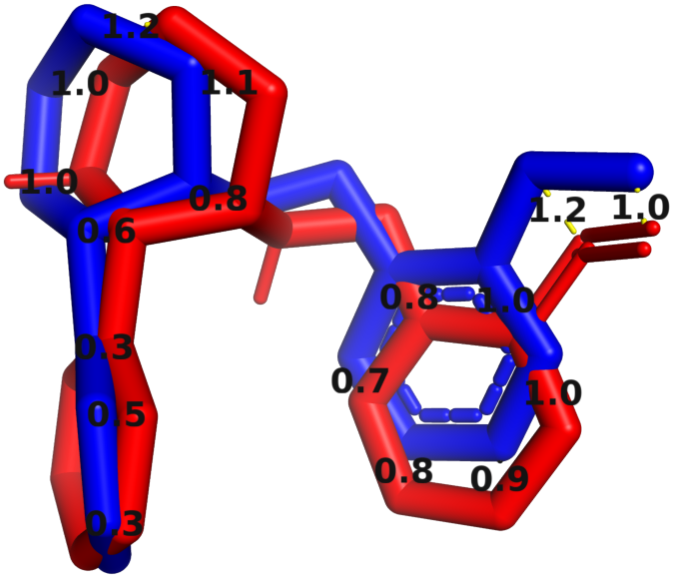

Figura 5.3. *Redocking* do ligante GBK à estrutura 6HLL, com Kd 28,79nM e RMSD 0,81Å.

A estrutura 6M9T, de ligante J9P, foi predita com Kd de 1,12nM e RMSD 1,41Å. Esses dados são representados na Figura 5.4.

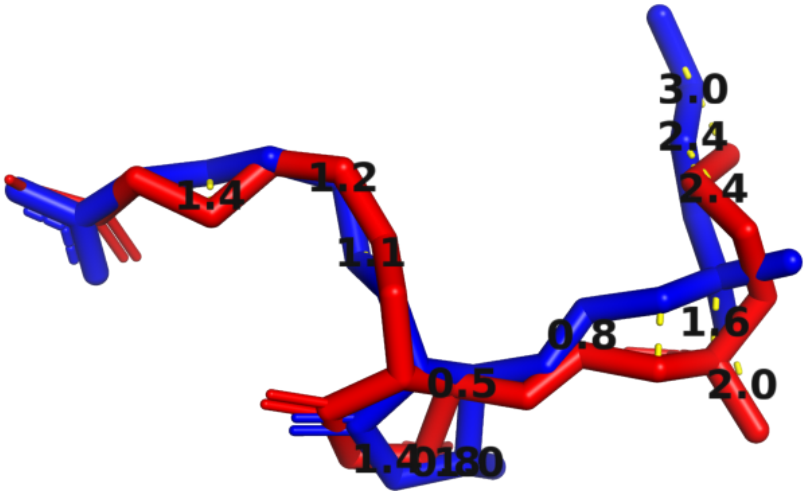

Figura 5.4. *Redocking* do ligante J9P à estrutura 6M9T, com Kd 1,12nM e RMSD 1,41Å.

As estruturas 6GPS e 6GPX possuem o mesmo ligante, F7N. Quando ligado à 6GPS teve Kd 0,09nM e RMSD 1,61Å (Figura 5.5) e, ao interagir com a 6GPX obteve Kd 0,10nM e RMSD 1,64Å (Figura 5.6).

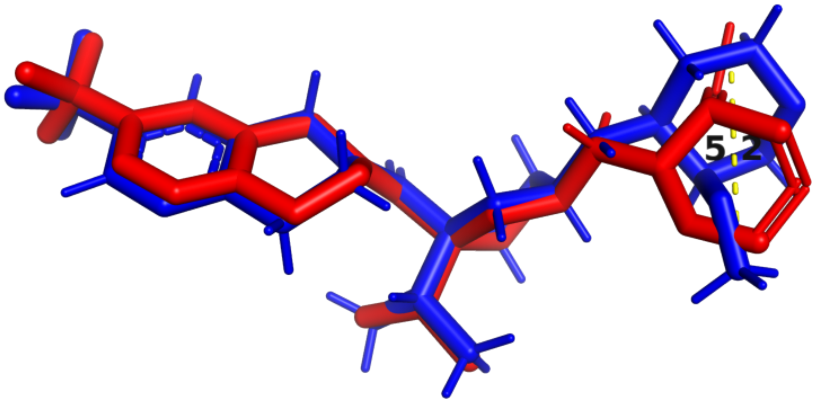

Figura 5.5. *Redocking* do ligante F7N à estrutura 6GPS, com Kd 0,09nM e RMSD 1,61Å.

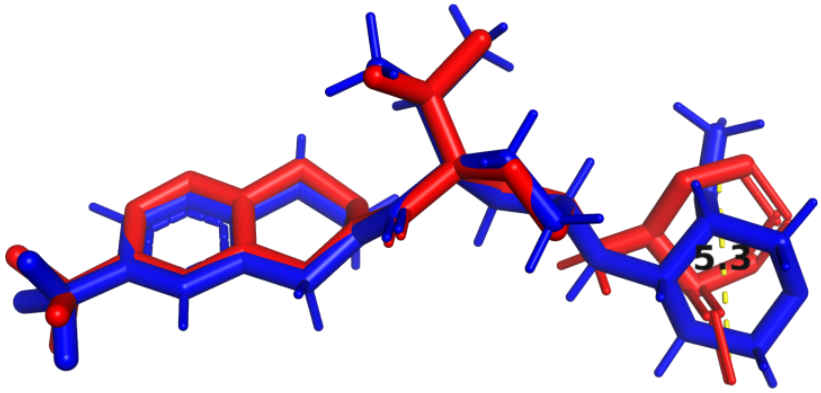

Figura 5.6. *Redocking* do ligante F7N à estrutura 6GPX, com Kd 0,10nM e RMSD 1,64Å.

É interessante notar que o ligante F7N, após o resultado do *redocking*, tanto com o alvo 6GPS (Figura 5.5) quanto com 6GPX (Figura 5.6) o posicionamento do anel, porção mais profunda do ligante no *pocket* da proteína, está diferente da original, porém, com bons valores de Kd.

Já a interação entre a proteína de PDB ID 6HLP e seu ligante cristalográfico GAW foi predita, conforme apresentado na Figura 5.7, com Kd 12,26nM e RMSD 0,26Å.

A molécula GBQ é o ligante cristalográfico das estruturas 6J20 e 6J21. Na primeira obteve Kd de 0,52nM e RMSD 0,93Å (visível na Figura 5.8), enquanto na segunda o Kd foi 1,13nM e RMSD 0,93Å (Figura 5.9).

Por fim, a última estrutura analisada é a de PDB ID 6IIU, com ligante AX8, o valor de Kd obtido foi 1,92nM e RMSD 0,23Å e a Figura 5.10 representa a melhor pose para esse complexo proteína-ligante.

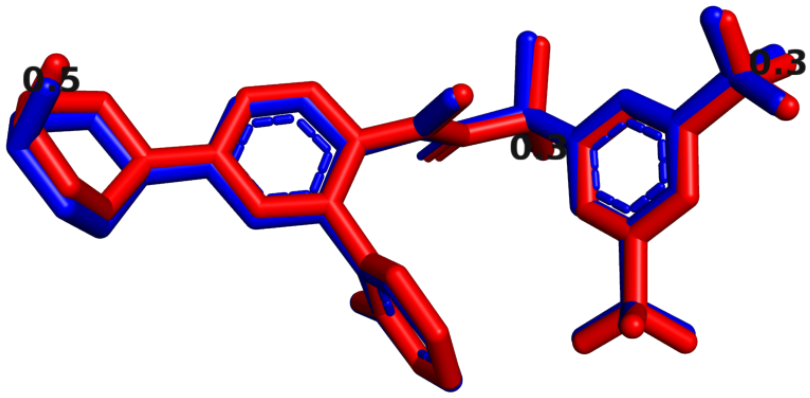

Figura 5.7. *Redocking* do ligante GAW à estrutura 6HLP, com Kd 12,26nM e RMSD 0,26Å.

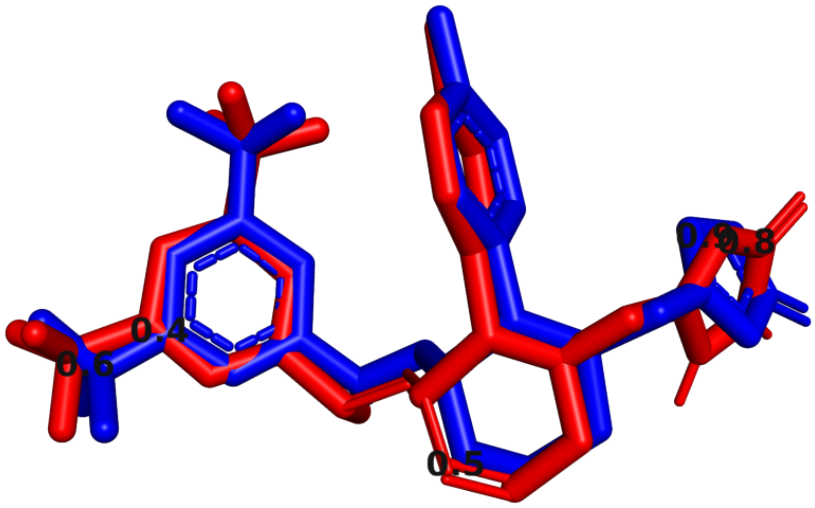

Figura 5.8. *Redocking* do ligante GBQ à estrutura 6J20, com Kd 0,52nM e RMSD 0,93Å.

O resultado de *redocking* da molécula AX8 com o alvo 6IIU obteve o menor RMSD dos aqui listados. A Figura 5.10 ressalta as maiores distâncias entre os átomos da pose original e da predita, sendo que o maior valor foi de 0.4Å e a grande parte dos átomos teve essa distância menor que 0.1Å.

5.3.2 Resultados do DUD-E com GPCR

O termo *decoys* consiste em um conjunto de compostos que possuem propriedades fisico-químicas semelhantes a compostos ativos, porém, são topologicamente distintos. Para este estudo, os compostos ativos serão os ligantes cristalográficos das GPCRs analisadas e, para cada um desses, foi gerado um conjunto de *decoys* a partir do DUD-

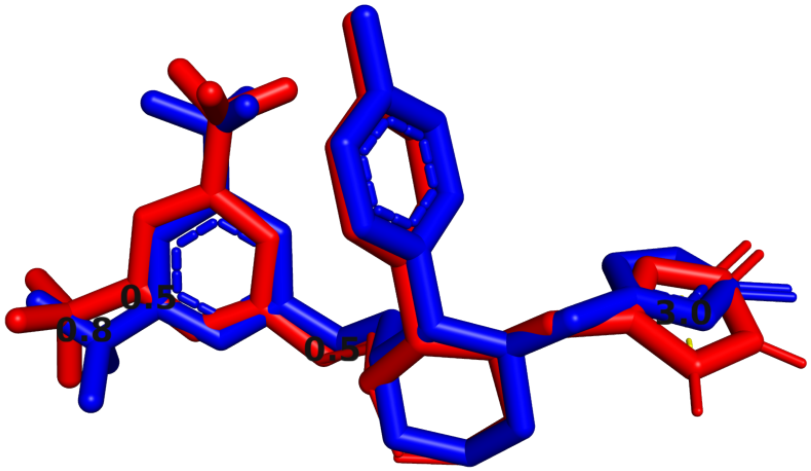

Figura 5.9. *Redocking* do ligante GBQ à estrutura 6J21, com Kd 1,13nM e RMSD 0,93Å.

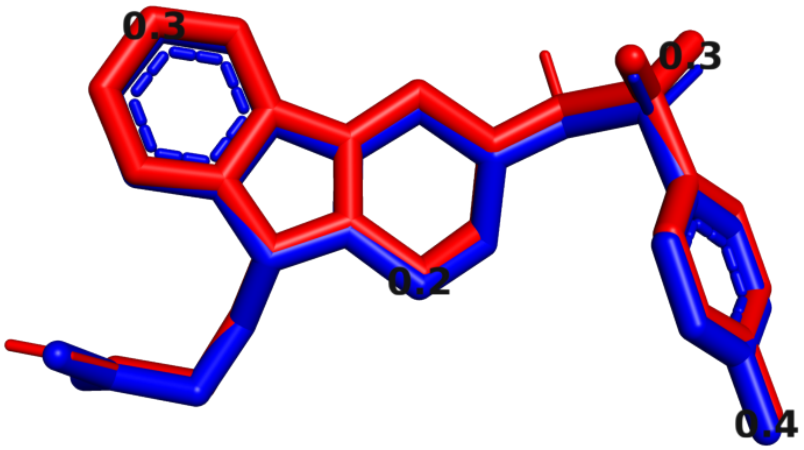

Figura 5.10. *Redocking* do ligante AX8 à estrutura 6IIU, com Kd 1,92nM e RMSD 0,23Å.

E [Mysinger et al., 2012].

A versão anterior ao DUD-E, denominada DUD [Huang et al., 2006], tinha seu acrônimo descrito como *Directory of Useful Decoys* (DUD), depois *Directory of Useful Decoys, Enhanced* (DUD-E) e atualmente, no site do DUD-E[1], é nominado como *A Database of Useful Decoys: Enhanced*. Essa ferramenta surgiu, em sua primeira versão, como uma biblioteca de compostos, contendo 40 proteínas e um conjunto de ligantes e *decoys* para cada uma dessas. Na versão mais recente esse número aumentou para 102 proteínas e mais de 22 mil compostos ativos para esses alvos proteicos.

Para o EasyVS, embora o DUD-E já tenha em sua biblioteca diversos compostos

[1]Acessível a partir do endereço `http://dude.docking.org/`. Acesso em: 8 de jul. 2018.

ativos e *decoys* para GPCRs (há 5 dessas disponíveis na ferramenta), optou-se por utilizar uma funcionalidade específica do DUD-E, que é a geração de *decoys*, sob demanda, a partir de uma lista de compostos ativos. Nesse caso, o DUD-E gera uma lista de 50 *decoys* para cada composto ativo, possuindo uma topologia dissimilar, porém, similar peso molecular, LogP, número de ligações rotacionáveis, doadores e aceptores de hidrogênio e carga total da molécula (essa última propriedade não estava presente na versão anterior do DUD-E).

Sendo assim, as mesmas proteínas citadas na seção 5.3.1 foram utilizadas no DUD-E. A Tabela 5.1 mostra o PBD ID, o código de cada um dos ligantes dessas estruturas, valores estimados de afinidade (Kd), p-value (detalhado a seguir) e RMSD comparando a pose original do ligante e a predita pelo EasyVS.

Tabela 5.1. Resumo dos resultados do DUD-E

PDB ID	**Ligante**	**Kd (nM)**	***p-value***	**RMSD (Å)**
6HLL	GBK	28,79	1,44E-11	0,81
6M9T	J9P	1,12	1,02E-10	1,41
6GPS	F7N	0,09	3,05E-10	1,61
6GPX	F7N	0,10	1,07E-08	1,64
6HLP	GAW	12,26	8,58E-06	0,26
6J20	GBQ	0,52	2,20E-05	0,93
6J21	GBQ	1,13	2,20E-05	0,93
6IIU	AX8	1,92	0,01	0,23

Para o experimento com os dados apresentados nessa seção foram utilizados os parâmetros padrão do EasyVS para *exhaustiveness* (valor 8), tamanho do *box* (20Å) e *energy range* (valor 3). Para o posicionamento do *box* foi utilizada a metodologia proposta pela ferramenta no Step 2, que utiliza o Ghecom [Kawabata, 2010] para identificar *pockets*. Vale salientar que é possível realizar a personalização de qualquer um desses parâmetros, porém, para validação da ferramenta optou-se pela utilização dos parâmetros sugeridos pelo sistema. A única exceção é a escolha do posicionamento do *box*, já que o EasyVS sugere uma lista de *pockets*, ordenada pelo volume (de maneira decrescente), porém, a metodologia que foi utilizada para obtenção dos resultados aqui presentes foi a escolha de um dos posicionamentos do *box* que fosse o mais próximo possível do ligante cristalográfico (no caso, usando o valor de 20Å para o tamanho, percebeu-se que praticamente todos os ligantes estavam localizados internamente em cada *box*). Um exemplo dessa escolha é exemplificado na Figura 5.11.

O *p-value*, apresentado na Tabela 5.1, foi calculado do valor médio de Kd dos

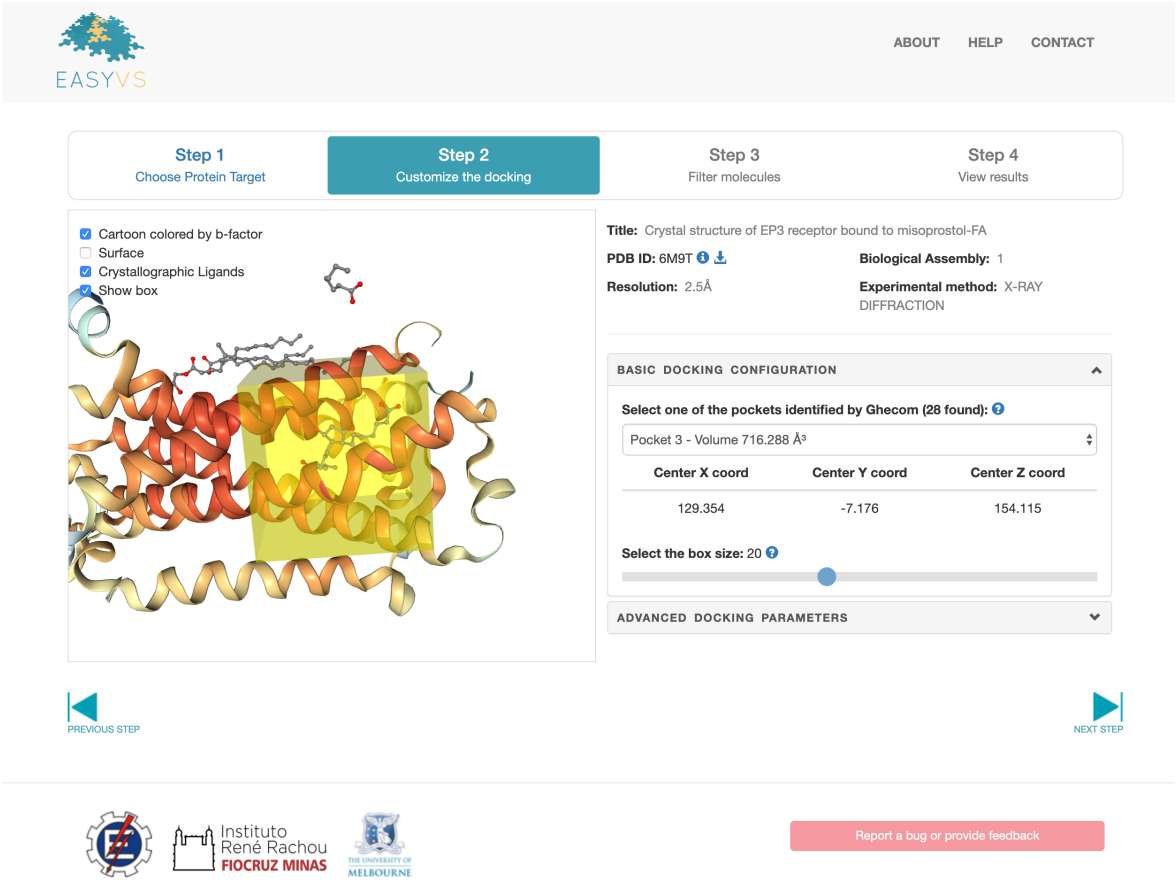

Figura 5.11. Exemplo de escolha do *pocket*, com o ligante localizado na área interna ao *box*, sendo escolhido o terceiro *pocket* com maior volume.

decoys, desvio padrão e valor do Kd do ligante após o *redocking*. Todos esses valores foram obtidos a partir do NNScore 2.01 [Durrant & McCammon, 2011]. Utilizando como exemplo o *p-value* obtido para o ligante GBK após *redocking* com o alvo proteico de PDB ID 6HLL, o número $1,44^{-11}$ evidencia que o ligante não faz parte do grupo dos *decoys*. Não há dúvidas que a interpretação desses valores indica que todos casos apresentados evidenciam esse fato.

Capítulo 6

Conclusões

O EasyVS foi proposto como sistema acessível por meio de um navegador web, que disponibilizaria ferramentas para triagem virtual de ligantes a partir de uma abordagem mista, baseada em alvo e ligantes. Esse objetivo foi alcançado uma vez que a plataforma está acessível, podendo ser utilizada inclusive a partir de computadores pessoais ou *smartphones* e possui configurações simples, automáticas e personalizáveis, para seleção de conjunto de pequenas moléculas, preparação e execução do *docking* molecular.

Para validação do sistema foram escolhidas 6 GPCRs que possuíam ligantes cristalográficos, a fim de verificar se as configurações utilizadas para o *docking* foram capazes de estimar a forma de ligação entre elas. O resultado obtido foi satisfatório, uma vez que os valores estimados de Kd ficaram, em sua maioria, menores que 2 nM (exceto 6HLL ligada à GBK e 6HLP ligada à GAW, que tiveram 28,79 nM e 12,26 nM, respectivamente). Também os valores de RMSD, na comparação da pose cristalográfica com a melhor pose predita, tiveram o maior valor de 1,64Å. Por fim, para cada uma dessas proteínas, foram gerados *decoys* de seus respectivos ligantes. O *p-value* indicou que, com base nos valores individuais de Kd entre os ligantes e *decoys*, há uma clara separação entre esses conjuntos de moléculas. As que realmente são ligantes das proteínas-alvo foram percebidas como diferentes das moléculas que foram geradas com base nessas, com propriedades físico-químicas semelhantes, porém, não possuem topologia similar e, por isso, não deveriam se ligar.

Com a possibilidade de utilização das proteínas contidas no RCSB PDB, assim como o *upload* de novos alvos drogáveis, o usuário da ferramenta pode iniciar sua pesquisa sem limitação de alvo (bastando o envio de um arquivo PDB). Essa característica, somada ao conjunto de mais de 16 milhões de pequenas moléculas, provenientes de variadas bibliotecas de compostos, e à possibilidade do envio de um arquivo SDF com moléculas, indica que o EasyVS se apresenta como uma ferramenta versátil e que pode

ser utilizada por qualquer pesquisador da área de triagem virtual de ligantes, mesmo que esse não seja um profundo conhecedor das configurações necessárias para o *docking* ou mesmo que não tenha uma grande quantidade de compostos acessíveis.

6.1 Trabalhos futuros

Nesta seção serão apresentadas as perspectivas deste trabalho, bem como suas limitações e o que se pretende fazer para superá-las. Cada tarefa é discutida e apresentada nas subseções seguintes.

6.1.1 Agrupamento molecular utilizando CUDA

Foi apresentado na Seção 4.4.3 que a implementação em Python do algoritmo de agrupamento baseado no trabalho do Butina [1999], disponível no RDKit [Landrum, 2006], limitava o processamento do agrupamento em cinquenta mil moléculas. Esse fato motivou o estudo visando melhoria de utilização de memória e tempo de processamento do referido código.

O código que está funcionando no EasyVS utiliza a linguagem de programação C++ e, para o processamento, realiza os cálculos em CPU. Já está em fase de testes a implementação desse mesmo algoritmo utilizando CUDA C que, em vez de realizar os cálculos em CPU, esses são feitos em GPU (placas gráficas NVidia).

A utilização da GPU nos testes já realizados poderá trazer muitos benefícios em termos de tempo de resposta e uso de memória quando for necessário o agrupamento molecular (em especial para grandes conjuntos). Isto porque a melhoria, em tempo de resposta, tem sido de mais de setecentas vezes. E o uso de memória (antes, fator limitador ao agrupamento de até cinquenta mil moléculas por vez), os testes apresentaram melhoria de mais de vinte mil vezes.

Após a finalização dessa nova abordagem, espera-se a utilização da máquina disponível e que tem uma placa GPU Tesla K40M, dentre outras que também possuem essa mesma placa, porém, não são partes componentes da estrutura física do EasyVS atualmente.

6.1.2 Utilização de mais lâminas do *cluster*

O EasyVS está alocado em duas máquinas dedicadas, entretanto, essas são algumas das máquinas/lâminas que compõe o *cluster* Beowulf, localizado na Universidade Federal de Itajubá (UNIFEI).

Espera-se, em um futuro próximo, que o EasyVS possa utilizar do poder de processamento do *cluster*, que possui 192 núcleos de processamento, 40TB de HD e cinco GPU Tesla K40M (cada uma com 2.880 CUDA *cores*), dispostos em lâminas conectadas por uma rede InfiniBand.

Os ganhos com relação ao acesso a essa estrutura poderão ser perceptíveis não somente no agrupamento molecular (que poderá executado nas placas gráficas e para cada conjunto de moléculas escolhida pelo usuário), mas também no *docking* (que utiliza muito processamento de CPU) e no tempo de resposta do servidor.

6.1.3 Testes com outros *fingerprints*

O EasyVS utiliza, no presente momento, somente o MACCS166 [Durant et al., 2002] como *fingerprint* para o agrupamento das moléculas. Após a conclusão da nova abordagem para agrupamento, utilizando GPU, serão testados outros *fingerprints* com maior quantidade e variedade de *features*, possibilitando maior precisão e, consequentemente, consumindo mais tempo de processamento e memória.

6.1.4 Classificação de moléculas como íons ou artefatos de cristalografia

Os trabalhos de Strömbergsson & Kleywegt [2009] e Carolan & Lamzin [2014] listam diversas moléculas que poderiam ser separadas das demais, uma vez que esses trabalhos as colocam em um conjunto de moléculas que não se caracterizariam como ligantes (sendo essas artefatos de cristalografia ou carboidratos).

Essa categorização ainda não foi incorporada ao EasyVS pois necessita de maiores estudos para que resultados válidos não sejam desconsiderados.

6.1.5 Sincronização com o RCSB

Todas as moléculas depositadas no RCSB PDB [Berman et al., 2000] estão disponíveis para uso no EasyVS a partir do momento em que o usuário solicita a sua utilização. No primeiro acesso a molécula é obtida do RCSB PDB e salva nos servidores locais. Pretende-se sincronizar, em tempo real, as proteínas do EasyVS com o RCSB PDB de forma que o usuário não espere o *download* e processamento das mesmas após sua requisição.

Também se espera armazenar todos os ligantes de proteínas do RCSB PDB, além de todas as classificações das proteínas.

6.1.6 Opção por remover ou manter heteroátomos no *docking*

No vigente fluxo de utilização da ferramenta e seus algoritmos, todos os heteroátomos são removidos da proteína-alvo (do arquivo PDB original) durante o processo de preparação do receptor para o *docking*, como descrito na seção 4.5.

Uma funcionalidade que será desenvolvida consiste em possibilitar ao usuário a escolha de quais heteroátomos deverão permanecer ou ser removidos durante o método de preparação da proteína para o *docking*. Em especial, moléculas de água ou íons presentes no sítio ativo da proteína ou localizados.

6.1.7 Aumento do número de execuções do *docking*

Sabe-se que o AutoDock Vina [Trott & Olson, 2010] utiliza um parâmetro para início da pesquisa por poses, denominado *seed* (semente). Esse é um valor aleatório que é gerado antes da execução do *docking* (podendo ser definido manualmente também).

A execução do *docking*, para as mesmas estruturas proteína-ligante pode ser realizado mais de uma vez (em duplicata, triplicata etc), sendo necessário a geração de um novo valor para a *seed*, já que o mesmo valor, mantendo os demais parâmetros, implica no mesmo resultado da execução.

Pretende-se, em futuras versões, possibilitar ao usuário a escolha de quantas vezes ele deseja que cada *docking* seja executado, possibilitando maior variedade na pesquisa de resultados para cada combinação entre proteína e ligante.

6.1.8 Filtro de moléculas por carga total

O Passo 3 do EasyVS possibilita ao usuário o estabelecimento do espaço químico a partir da escolha das bibliotecas de compostos em conjunto com a utilização de diversos filtros contendo diversas propriedades fisico-químicas.

Pretende-se aumentar a quantidade de parâmetros a serem utilizados pelo usuário para melhor serem definidas as moléculas filtradas pelo mesmo. Inicialmente, além dos sete filtros já existentes, será adicionado o valor de carga total e outros parâmetros podem ser acrescentados conforme forem necessidade.

6.1.9 Busca por proteínas a partir do FASTA

Já existem alguns algoritmos internos do EasyVS que estão prontos para a realização de busca e comparação de proteínas utilizando da sequência de aminoácidos

(arquivo FASTA), a partir do *software* BLASTp (aplicação do BLAST para comparação entre proteínas), conforme o trabalho de Camacho et al. [2009].

Pretende-se disponibilizar ao usuário essa funcionalidade para pesquisa de proteínas não somente a partir do PDB ID.

6.1.10 Acréscimo de outras bibliotecas de moléculas

Espera-se incorporar outras bibliotecas de compostos como o BindingDB [Gilson et al., 2016] ao EasyVS. Ela contém um conjunto de dados sobre ligação de mais de sete mil proteínas e mais de 650 mil pequenas moléculas. Juntamente com o PDBbind [Wang et al., 2004], trará mais detalhes sobre os ligantes conhecidos das proteínas do RCSB PDB.

Além da inserção dos demais compostos compráveis, disponíveis no Zinc 15 [Irwin et al., 2012], não somente os que estão em estoque, espera-se utilização de outras bibliotecas de compostos.

Esse recurso, juntamente com o refinamento dos resultados a partir da similaridade de um determinado composto (funcionalidade disponível no último passo do fluxo principal de utilização do sistema), possibilitará ao usuário o *docking* de moléculas semelhantes aos ligantes já conhecidos.

6.1.11 Integração ao pkCSM

O pkCSM [Pires et al., 2015] é uma ferramenta que utiliza uma assinatura baseada em grafos para predizer propriedades farmacocinéticas e toxicidade - ADMET (absorção, distribuição, metabolismo, excreção e toxicidade).

Pretende-se integrar o EasyVS ao pkCSM de forma que o usuário tenha disponível mais informações sobre cada molécula além das provenientes das bases de moléculas originais e do RDKit.

6.1.12 Integração ao CSM-lig

De maneira semelhante ao pkCSM, estuda-se a integração do EasyVS com o CSM-lig [Pires & Ascher, 2016], mas, nesse caso, provendo dados sobre a afinidade da proteína com o possível ligante sugerido pelo EasyVS.

Referências Bibliográficas

Ade, P. A. R.; Aghanim, N.; Arnaud, M.; Ashdown, M.; Aumont, J.; Baccigalupi, C.; Banday, A. J.; Barreiro, R. B.; Bartlett, J. G.; Bartolo, N.; Battaner, E.; Battye, R.; Benabed, K.; Benoît, A.; Benoit-Lévy, A.; Bernard, J.-P.; Bersanelli, M.; Bielewicz, P.; Bock, J. J.; Bonaldi, A.; Bonavera, L.; Bond, J. R.; Borrill, J.; Bouchet, F. R.; Boulanger, F.; Bucher, M.; Burigana, C.; Butler, R. C.; Calabrese, E.; Cardoso, J.-F.; Catalano, A.; Challinor, A.; Chamballu, A.; Chary, R.-R.; Chiang, H. C.; Chluba, J.; Christensen, P. R.; Church, S.; Clements, D. L.; Colombi, S.; Colombo, L. P. L.; Combet, C.; Coulais, A.; Crill, B. P.; Curto, A.; Cuttaia, F.; Danese, L.; Davies, R. D.; Davis, R. J.; de Bernardis, P.; de Rosa, A.; de Zotti, G.; Delabrouille, J.; Désert, F.-X.; Di Valentino, E.; Dickinson, C.; Diego, J. M.; Dolag, K.; Dole, H.; Donzelli, S.; Doré, O.; Douspis, M.; Ducout, A.; Dunkley, J.; Dupac, X.; Efstathiou, G.; Elsner, F.; Enßlin, T. A.; Eriksen, H. K.; Farhang, M.; Fergusson, J.; Finelli, F.; Forni, O.; Frailis, M.; Fraisse, A. A.; Franceschi, E.; Frejsel, A.; Galeotta, S.; Galli, S.; Ganga, K.; Gauthier, C.; Gerbino, M.; Ghosh, T.; Giard, M.; Giraud-Héraud, Y.; Giusarma, E.; Gjerløw, E.; González-Nuevo, J.; Górski, K. M.; Gratton, S.; Gregorio, A.; Gruppuso, A.; Gudmundsson, J. E.; Hamann, J.; Hansen, F. K.; Hanson, D.; Harrison, D. L.; Helou, G.; Henrot-Versillé, S.; Hernández-Monteagudo, C.; Herranz, D.; Hildebrandt, S. R.; Hivon, E.; Hobson, M.; Holmes, W. A.; Hornstrup, A.; Hovest, W.; Huang, Z.; Huffenberger, K. M.; Hurier, G.; Jaffe, A. H.; Jaffe, T. R.; Jones, W. C.; Juvela, M.; Keihänen, E.; Keskitalo, R.; Kisner, T. S.; Kneissl, R.; Knoche, J.; Knox, L.; Kunz, M.; Kurki-Suonio, H.; Lagache, G.; Lähteenmäki, A.; Lamarre, J.-M.; Lasenby, A.; Lattanzi, M.; Lawrence, C. R.; Leahy, J. P.; Leonardi, R.; Lesgourgues, J.; Levrier, F.; Lewis, A.; Liguori, M.; Lilje, P. B.; Linden-Vørnle, M.; López-Caniego, M.; Lubin, P. M.; Macías-Pérez, J. F.; Maggio, G.; Maino, D.; Mandolesi, N.; Mangilli, A.; Marchini, A.; Maris, M.; Martin, P. G.; Martinelli, M.; Martínez-González, E.; Masi, S.; Matarrese, S.; McGehee, P.; Meinhold, P. R.; Melchiorri, A.; Melin, J.-B.; Mendes, L.; Mennella, A.; Migliaccio, M.; Millea, M.; Mitra, S.; Miville-Deschênes, M.-A.; Moneti, A.; Montier, L.; Morgante, G.; Mortlock, D.;

Moss, A.; Munshi, D.; Murphy, J. A.; Naselsky, P.; Nati, F.; Natoli, P.; Netterfield, C. B.; Nørgaard-Nielsen, H. U.; Noviello, F.; Novikov, D.; Novikov, I.; Oxborrow, C. A.; Paci, F.; Pagano, L.; Pajot, F.; Paladini, R.; Paoletti, D.; Partridge, B.; Pasian, F.; Patanchon, G.; Pearson, T. J.; Perdereau, O.; Perotto, L.; Perrotta, F.; Pettorino, V.; Piacentini, F.; Piat, M.; Pierpaoli, E.; Pietrobon, D.; Plaszczynski, S.; Pointecouteau, E.; Polenta, G.; Popa, L.; Pratt, G. W.; Prézeau, G.; Prunet, S.; Puget, J.-L.; Rachen, J. P.; Reach, W. T.; Rebolo, R.; Reinecke, M.; Remazeilles, M.; Renault, C.; Renzi, A.; Ristorcelli, I.; Rocha, G.; Rosset, C.; Rossetti, M.; Roudier, G.; Rouillé d'Orfeuil, B.; Rowan-Robinson, M.; Rubiño-Martín, J. A.; Rusholme, B.; Said, N.; Salvatelli, V.; Salvati, L.; Sandri, M.; Santos, D.; Savelainen, M.; Savini, G.; Scott, D.; Seiffert, M. D.; Serra, P.; Shellard, E. P. S.; Spencer, L. D.; Spinelli, M.; Stolyarov, V.; Stompor, R.; Sudiwala, R.; Sunyaev, R.; Sutton, D.; Suur-Uski, A.-S.; Sygnet, J.-F.; Tauber, J. A.; Terenzi, L.; Toffolatti, L.; Tomasi, M.; Tristram, M.; Trombetti, T.; Tucci, M.; Tuovinen, J.; Türler, M.; Umana, G.; Valenziano, L.; Valiviita, J.; Van Tent, F.; Vielva, P.; Villa, F.; Wade, L. A.; Wandelt, B. D.; Wehus, I. K.; White, M.; White, S. D. M.; Wilkinson, A.; Yvon, D.; Zacchei, A. & Zonca, A. (2016). Planck 2015 results. *Astronomy & Astrophysics*, 594:A13.

Anaconda (2016). Anaconda Software Distribution.

ANVISA, A. N. d. V. S. (2018). Definições da ANVISA.

Avorn, J. (2015). The $2.6 Billion Pill — Methodologic and Policy Considerations. *New England Journal of Medicine*, 372(20):1877--1879. ISSN 0028-4793.

Ballester, P. J. & Mitchell, J. B. (2010). A machine learning approach to predicting protein-ligand binding affinity with applications to molecular docking. *Bioinformatics*. ISSN 13674803.

Banerjee, P.; Erehman, J.; Gohlke, B.-O.; Wilhelm, T.; Preissner, R. & Dunkel, M. (2015). Super Natural II—a database of natural products. *Nucleic Acids Research*, 43(D1):D935–D939. ISSN 1362-4962.

Berman, H. M.; Westbrook, J. D.; Feng, Z.; Gilliland, G.; Bhat, T. N.; Weissig, H.; Shindyalov, I. N. & Bourne, P. E. (2000). The Protein Data Bank. *Nucleic Acids Research*, 28(1):235--242. ISSN 0305-1048.

Bielska, E.; Lucas, X.; Czerwoniec, A.; Kasprzak, J. M.; Kaminska, K. H. & Bujnicki, J. M. (2011). Virtual screening strategies in drug design – methods and applications. *Biotechnologia*, 92(3):249--264.

Bikadi, Z. & Hazai, E. (2009). Application of the PM6 semi-empirical method to modeling proteins enhances docking accuracy of AutoDock. *Journal of Cheminformatics*, 1(1):15. ISSN 1758-2946.

Bologa, C. G.; Ursu, O. & Oprea, T. I. (2019). How to Prepare a Compound Collection Prior to Virtual Screening. Em *Bioinformatics and Drug Discovery*, pp. 119--138. Humana Press, New York, NY.

Bolton, E. E.; Wang, Y.; Thiessen, P. A. & Bryant, S. H. (2008). PubChem: Integrated Platform of Small Molecules and Biological Activities. *Annual Reports in Computational Chemistry*, 4:217--241. ISSN 1574-1400.

Butina, D. (1999). Unsupervised data base clustering based on daylight's fingerprint and Tanimoto similarity: A fast and automated way to cluster small and large data sets. *Journal of Chemical Information and Computer Sciences*, 39(4):747--750. ISSN 00952338.

Camacho, C.; Coulouris, G.; Avagyan, V.; Ma, N.; Papadopoulos, J.; Bealer, K. & Madden, T. L. (2009). BLAST+: architecture and applications. *BMC Bioinformatics*, 10(1):421. ISSN 1471-2105.

Carolan, C. G. & Lamzin, V. S. (2014). Automated identification of crystallographic ligands using sparse-density representations. *Acta Crystallographica Section D Biological Crystallography*, 70(7):1844--1853. ISSN 1399-0047.

Cereto-Massagué, A.; Ojeda, M. J.; Valls, C.; Mulero, M.; Garcia-Vallvé, S. & Pujadas, G. (2015). Molecular fingerprint similarity search in virtual screening. *Methods*, 71(C):58--63. ISSN 10462023.

Cock, P. J.; Antao, T.; Chang, J. T.; Chapman, B. A.; Cox, C. J.; Dalke, A.; Friedberg, I.; Hamelryck, T.; Kauff, F.; Wilczynski, B. & De Hoon, M. J. (2009). Biopython: Freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422--1423. ISSN 13674803.

Cortes-Cabrera, A.; Murcia, P. A. S.; Morreale, A. & Gago, F. (2016). Ligand-Based Drug Discovery and Design. Em Cavasotto, C. N., editor, *In Silico Drug Discovery and Design: Theory, Methods, Challenges, and Applications*, capítulo 4, pp. 99--116. CRC Press, Londres.

Daintith, J. (2008). *Dictionary of Chemistry*. Oxford University Press, Nova Iorque, 6 edição. ISBN 978-0-19-920463-2.

Daylight (2008). Daylight chemical information systems.

de Magalhães, C. S.; Almeida, D. M.; Barbosa, H. J. C. & Dardenne, L. E. (2014). A dynamic niching genetic algorithm strategy for docking highly flexible ligands. *Information Sciences*, 289:206--224. ISSN 0020-0255.

Desai, P. V.; Patnym, A.; Sabnis, Y.; Tekwani, B.; Gut, J.; Rosenthal, P.; Srivastava, A. & Avery, M. (2004). Identification of Novel Parasitic Cysteine Protease Inhibitors Using Virtual Screening. 1. The ChemBridge Database. *Journal of Medicinal Chemistry*, 47(26):6609--6615.

Digby, G. J.; Lober, R. M.; Sethi, P. R. & Lambert, N. A. (2006). Some G protein heterotrimers physically dissociate in living cells. *Proceedings of the National Academy of Sciences*. ISSN 0027-8424.

Doganova, L. (2015). Que vaut une molécule ? Formulation de la valeur dans les projets de développement de nouveaux médicaments. *Revue d'anthropologie des connaissances*, 9(1):17. ISSN 1760-5393.

Domingues, B. F. & Lopes, J. C. D. (2012). *3D-Pharma: Uma Ferramenta para Triagem Virtual Baseada em Fingerprints de Farmacoforos*. Tese de doutorado, Universidade Federal de Minas Gerais.

Driessen, V. (2018). RQ: Simple job queues for Python.

Dunkel, M.; Fullbeck, M.; Neumann, S. & Preissner, R. (2006). SuperNatural: a searchable database of available natural compounds. *Nucleic acids research*, 34(Database issue):678--683. ISSN 1362-4962.

Durant, J. L.; Leland, B. A.; Henry, D. R. & Nourse, J. G. (2002). Reoptimization of MDL keys for use in drug discovery. *Journal of Chemical Information and Computer Sciences*, 42(6):1273--1280. ISSN 00952338.

Durrant, J. D. & McCammon, J. A. (2011). NNScore 2.0: A neural-network receptor-ligand scoring function. *Journal of Chemical Information and Modeling*, 51(11):2897--2903. ISSN 15499596.

Dwyer, G.; Aggarwal, S. & Stouffer, J. (2017). *Flask: Building Python Web Services*. Pakt, Birmingham, Reino Unido. ISBN 9781787288225.

Fischer, E. (1894). Einfluss der Configuration auf die Wirkung der Enzyme. *Berichte der deutschen chemischen Gesellschaft*, 27(3):2985--2993. ISSN 03659496.

Fisher Scientific (2018). Maybridge.com.

Gaulton, A.; Hersey, A.; Nowotka, M.; Bento, A. P.; Chambers, J.; Mendez, D.; Mutowo, P.; Atkinson, F.; Bellis, L. J.; Cibrián-Uhalte, E.; Davies, M.; Dedman, N.; Karlsson, A.; Magariños, M. P.; Overington, J. P.; Papadatos, G.; Smit, I. & Leach, A. R. (2017). The ChEMBL database in 2017. *Nucleic Acids Research*, 45(D1):D945–D954. ISSN 0305-1048.

Gilson, M. K.; Liu, T.; Baitaluk, M.; Nicola, G.; Hwang, L. & Chong, J. (2016). BindingDB in 2015: A public database for medicinal chemistry, computational chemistry and systems pharmacology. *Nucleic acids research*, 44(D1):1045--1053. ISSN 1362-4962.

Good, A. C. & Oprea, T. I. (2008). Optimization of CAMD techniques 3. Virtual screening enrichment studies: A help or hindrance in tool selection? *Journal of Computer-Aided Molecular Design*, 22(3-4):169--178. ISSN 0920654X.

Grinberg, M. (2014). *Flask Web Development*. O'Reilly, 1 edição. ISBN 9781449372620.

Grosdidier, A.; Zoete, V. & Michielin, O. (2011). SwissDock, a protein-small molecule docking web service based on EADock DSS. *Nucleic acids research*, 39(Web Server issue):270--7. ISSN 1362-4962.

Hauser, A. S.; Attwood, M. M.; Rask-Andersen, M.; Schiöth, H. B. & Gloriam, D. E. (2017). Trends in GPCR drug discovery: New agents, targets and indications. *Nature Reviews Drug Discovery*. ISSN 14741784.

Huang, N.; Shoichet, B. K. & Irwin, J. J. (2006). Benchmarking Sets for Molecular Docking. *Journal of Medicinal Chemistry*, 49(23):6789--6801. ISSN 0022-2623.

Irwin, J. J.; Sterling, T.; Mysinger, M. M.; Bolstad, E. S. & Coleman, R. G. (2012). ZINC: A free tool to discover chemistry for biology. *Journal of Chemical Information and Modeling*, 52(7):1757--1768. ISSN 15499596.

Joost, P. & Methner, A. (2002). Phylogenetic analysis of 277 human G-protein-coupled receptors as a tool for the prediction of orphan receptor ligands. *Genome biology*. ISSN 1474-760X.

Kawabata, T. (2010). Detection of multiscale pockets on protein surfaces using mathematical morphology. *Proteins: Structure, Function, and Bioinformatics*, 78(5):1195--1211. ISSN 08873585.

Keseru, G. M. & Makara, G. M. (2006). Hit discovery and hit-to-lead approaches. *Drug discovery today*, 11(15-16):741--748. ISSN 1359-6446.

Kirkpatrick, P. & Ellis, C. (2004). Chemical space. *Nature*, 432:823--823. ISSN 0028-0836.

Landrum, G. (2006). RDKit Documentation.

Lefkowitz, R. J.; Roth, J. & Pastan, I. (1970). Radioreceptor assay of adrenocorticotropic hormone: new approach to assay of polypeptide hormones in plasma. *Science (New York, N.Y.)*, 170(3958):633--5. ISSN 0036-8075.

Li, H.; Leung, K.-S. & Wong, M.-H. (2012). idock: A multithreaded virtual screening tool for flexible ligand docking. Em *2012 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pp. 77--84. IEEE.

Lipinski, C. A.; Lombardo, F.; Dominy, B. W. & Feeney, P. J. (2001). Experimental and computational approaches to estimate solubility and permeability in drug discovery and development setting. *Advanced Drug Delivery Reviews*, 46:3--26.

Marcotte, E. (2010). Responsive Web Design.

Mayr, L. M. & Fuerst, P. (2008). The future of high-throughput screening. *Journal of biomolecular screening*, 13(6):443--448. ISSN 1087-0571.

Morgan, H. L. (1965). The Generation of a Unique Machine Description for Chemical Structures-A Technique Developed at Chemical Abstracts Service. *Journal of Chemical Documentation*, 5(2):107--113. ISSN 0021-9576.

Morris, G. M.; Ruth, H.; Lindstrom, W.; Sanner, M. F.; Belew, R. K.; Goodsell, D. S. & Olson, A. J. (2009). Software news and updates AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility. *Journal of Computational Chemistry*, 30(16):2785--2791. ISSN 01928651.

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. ISSN 0022-2623.

Nelson, D. L. & Cox, M. M. (2008). *Lehninger Principles of Biochemistry*, volume 1. W. H. Freeman And Company, Nova Iorque, 5 edição. ISBN 13: 978-0-7167-7108-1.

Niimura, Y. (2009). Evolutionary dynamics of olfactory receptor genes in chordates: interaction between environments and genomic contents.

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):33. ISSN 1758-2946.

Pires, D. E. V. & Ascher, D. B. (2016). CSM-lig: a web server for assessing and comparing protein–small molecule affinities. *Nucleic Acids Research*, 44(W1):W557–W561. ISSN 0305-1048.

Pires, D. E. V.; Blundell, T. L. & Ascher, D. B. (2015). pkCSM: Predicting Small-Molecule Pharmacokinetic and Toxicity Properties Using Graph-Based Signatures. *Journal of Medicinal Chemistry*, 58(9):4066--4072. ISSN 0022-2623.

Pitt, W. R.; Higueruelo, A. P. & Groom, C. R. (2009). Structural bioinformatics in drug discovery. Em Gu, J. & Bourne, P. E., editores, *Struct. Bioinf. (2nd Ed.)*, pp. 809--845, Nova Jersey. ISSN 0076-6941.

Python (2009). The Python tutorial.

Ronacher, A. (2018). Flask (a python microframework).

Roth, B. L. & Marshall, F. H. (2012). Studies of a ubiquitous receptor family. *Nature*, 492(7427):57--57. ISSN 0028-0836.

Salon, J. A.; Lodowski, D. T. & Palczewski, K. (2011). The Significance of G Protein-Coupled Receptor Crystallography for Drug Discovery. *Pharmacological Reviews*. ISSN 0031-6997.

Sanfilippo, S. & Noordhuis, P. (2018). Redis.

Sheridan, R. P.; Miller, M. D.; Underwood, D. J. & Kearsley, S. K. (1996). Chemical Similarity Using Geometric Atom Pair Descriptors. *Journal of Chemical Information and Computer Sciences*, 36(1):128--136. ISSN 0095-2338.

Sotriffer, C. A. (2016). Protein-Ligand Docking: From Basic Principles to Advanced Applications. Em Cavasotto, C. N., editor, *In Silico Drug Discovery and Design: Theory, Methods, Challenges, and Applications*, capítulo 6, pp. 155--188. CRC Press.

Sterling, T. & Irwin, J. J. (2015). ZINC 15 - Ligand Discovery for Everyone. *Journal of Chemical Information and Modeling*, 55(11):2324--2337. ISSN 1549-9596.

Strömbergsson, H. & Kleywegt, G. J. (2009). A chemogenomics view on protein-ligand spaces. *BMC Bioinformatics*, 10(Suppl 6):S13. ISSN 1471-2105.

Trott, O. & Olson, A. J. (2010). AutoDock Vina: Improving the Speed and Accuracy of Docking with a New Scoring Function, Efficient Optimization, and Multithreading. *Journal of Computational Chemistry*, 31(2):455--61. ISSN 1096-987X.

Trzaskowski, B.; Latek, D.; Yuan, S.; Ghoshdastider, U.; Debinski, A. & Filipek, S. (2012). Action of molecular switches in GPCRs–theoretical and experimental studies. *Current medicinal chemistry*, 19(8):1090--109. ISSN 1875-533X.

Tyzack, J. D.; Mussa, H. Y.; Williamson, M. J.; Kirchmair, J. & Glen, R. C. (2014). Cytochrome P450 site of metabolism prediction from 2D topological fingerprints using GPU accelerated probabilistic classifiers. *Journal of Cheminformatics*, 6(1):29. ISSN 1758-2946.

Verdonk, M. L.; Berdini, V.; Hartshorn, M. J.; Mooij, W. T. M.; Murray, C. W.; Taylor, R. D. & Watson, P. (2004). Virtual screening using protein-ligand docking: Avoiding artificial enrichment. *Journal of Chemical Information and Computer Sciences*, 44(3):793--806. ISSN 00952338.

Verli, H. (2014). *Bioinformática: da Biologia à Flexibilidade Molecular*. Sociedade Brasileira de Bioquímica e Biologia Molecular - SBBq, São Paulo. ISBN 9788569288008.

Villoutreix, B. O.; Kuenemann, M. A.; Lagorce, D.; Sperandio, O. & Miteva, M. A. (2016). In silico approaches assisting the rational design of low molecular weight protein–protein interaction modulators. Em Cavasotto, C. N., editor, *In Silico Drug Discovery and Design: Theory, Methods, Challenges, and Applications*, capítulo 17, pp. 441--482. CRC Press, Londres.

Walters, W.; Stahl, M. T. & Murcko, M. A. (1998). Virtual screening—an overview. *Drug Discovery Today*, 3(4):160--178. ISSN 13596446.

Wang, R.; Fang, X.; Lu, Y. & Wang, S. (2004). The PDBbind Database: Collection of Binding Affinities for Protein-Ligand Complexes with Known Three-Dimensional Structures. *Journal of Medicinal Chemistry*, 47(12):2977--2980. ISSN 0022-2623.

Wishart, D. S.; Feunang, Y. D.; Guo, A. C.; Lo, E. J.; Marcu, A.; Grant, J. R.; Sajed, T.; Johnson, D.; Li, C.; Sayeeda, Z.; Assempour, N.; Iynkkaran, I.; Liu, Y.; Maciejewski, A.; Gale, N.; Wilson, A.; Chin, L.; Cummings, R.; Le, D.; Pon, A.; Knox, C. & Wilson, M. (2018a). DrugBank 5.0: a major update to the DrugBank database for 2018. *Nucleic acids research*, 46(D1):D1074–D1082. ISSN 1362-4962.

Wishart, D. S.; Feunang, Y. D.; Marcu, A.; Guo, A. C.; Liang, K.; Vázquez-Fresno, R.; Sajed, T.; Johnson, D.; Li, C.; Karu, N.; Sayeeda, Z.; Lo, E.; Assempour, N.; Berjanskii, M.; Singhal, S.; Arndt, D.; Liang, Y.; Badran, H.; Grant, J.; Serra-Cayuela, A.; Liu, Y.; Mandal, R.; Neveu, V.; Pon, A.; Knox, C.; Wilson, M.; Manach, C. & Scalbert, A. (2018b). HMDB 4.0: the human metabolome database for 2018. *Nucleic Acids Research*, 46(D1):D608–D617. ISSN 0305-1048.

Wishart, D. S.; Knox, C.; Guo, A. C.; Shrivastava, S.; Hassanali, M.; Stothard, P.; Chang, Z. & Woolsey, J. (2006). DrugBank: a comprehensive resource for in silico drug discovery and exploration. *Nucleic acids research*, 34(Database issue):668--72. ISSN 1362-4962.