RIGEL: PLATAFORMA EM NUVEM PARA APLICAÇÕES BIOLÓGICAS DE ALTO DESEMPENHO

JOÃO LUIZ DE ALMEIDA FILHO

UNIVERSIDADE ESTADUAL DO NORTE FLUMINENSE
DARCY RIBEIRO

CAMPOS DOS GOYTACAZES – RJ
JULHO – 2019

RIGEL: PLATAFORMA EM NUVEM PARA APLICAÇÕES BIOLÓGICAS DE ALTO DESEMPENHO

JOÃO LUIZ DE ALMEIDA FILHO

Tese apresentada ao Centro de Biociências e Biotecnologia da Universidade Estadual do Norte Fluminense, como parte das exigências para obtenção do título de Doutor em Biociências e Biotecnologia.

Orientador: Dr ª Jorge Hernandez Fernandez

CAMPOS DOS GOYTACAZES – RJ

JULHO – 2019

RIGEL: PLATAFORMA EM NUVEM PARA APLICAÇÕES BIOLÓGICAS DE ALTO DESEMPENHO

JOÃO LUIZ DE ALMEIDA FILHO

Tese apresentada ao Centro de Biociências e Biotecnologia da Universidade Estadual do Norte Fluminense, como parte das exigências para obtenção do título de Doutor em Biociências e Biotecnologia.

Aprovada em 29 de julho de 2019

Comissão Examinadora

Profa. Annabell Del Real Tamariz (D.Sc., LCMAT) – UENF

Prof. Evenilton Pessoa Costa (D.Sc.) – UFRJ Campus Macaé

Prof. Thiago Motta Venâncio (D.Sc., LQFPP) – UENF

Prof. Jorge Hernandez Fernandez (D.Sc., LQFPP) – UENF

(Orientador)

Dedico o presente trabalho ao meu “Tio” Jorge, (in memorian), por todo apoio naqueles primeiros anos.

AGRADECIMENTOS

Este manuscrito que você está lendo foi fruto de um trabalho intenso que durou cerca de quatro anos, durante este período muitas pessoas contribuíram direta ou indiretamente para a conclusão deste projeto. Aqui vai o meu agradecimento a alguma delas:

A meu orientador, Jorge Hernandez Fernandez, que abriu as portas do ambiente acadêmico para mim há 11 anos atrás quando eu era apenas um garoto da ciência da computação perdido em algoritmos e números. Especificamente neste projeto, agradeço sua participação em todas as decisões e ideias que proporcionaram o meu crescimento cientifico.

A minha família por terem sempre apoiado e incentivado todas as minhas decisões e por ficarem orgulhosos de mim mesmo não quando entendiam as explicações do que eu faço no laboratório.

A Mariana, minha namorada, que sempre me ajudou na escrita dos textos, no clareamento das minhas ideias, e apoio de meu crescimento acadêmico e profissional.

Aos meus professores, pois, sem o compartilhamento do conhecimento de vocês, eu jamais conseguiria chegar até aqui e realizar este sonho,

Ao pessoal do GRC da UENF, sempre foram solícitos em resolver os problemas de rede e computacionais deste projeto.

Ao CNPq, Capes e UENF pela concessão da bolsa de doutorado, apoio financeiro e infraestrutura para a realização deste projeto.

Á Deus por ter proporcionado todos os desafios e ferramentas para a minha formação. Por apresentar uma luz em cada momento escuro, o qual eu achava que estava tudo perdido.

E por último, e não menos importante, agradeço a você que está apreciando este manuscrito.

RESUMO

Experimentos de Biologia Computacional possuem uma alta demanda hoje em dia pela rapidez de análise de grandes quantidades de dados e baixo custo relativo. Muitos destes experimentos são realizados pela combinação de varias ferramentas computacionais o que aumenta a complexidade da usabilidade e o custo de processamento. Neste contexto, vários softwares para automatização destes *pipelines* foram criados para trabalhar na análise de dados da área da genômica devido às técnicas de sequenciamento de alto rendimento. Paralelamente a isto, há um hiato de programas especializados no desenvolvimento de *pipelines* de análises de dados da biologia computacional gratuitos, o que impacta negativamente a experimentação nas áreas de biotecnologia e farmacêutica. Assim, neste trabalho nós descrevemos uma nova plataforma de *pipelines* chamada Rigel: Uma ferramenta web que facilita a criação e utilização de *pipelines* de biologia computacional com a utilização de ferramentas disponíveis gratuitamente para o meio acadêmico. A versão atual de Rigel inclui *pipelines* de predição de estrutura de proteínas, *docking* molecular e *virtual screening*; chamados de AutoModel Web, MDR SurflexDock e HTP SurflexDock respectivamente. Estes dois últimos utilizam o protocolo de *ensemble docking* visando aumentar o espaço conformacional do receptor por meio da utilização de informação estrutural obtida da cristalografia e simulação molecular. Neste contexto, ilustramos a usabilidade de Rigel com a utilização de dois estudos de casos: O *Docking* de ligantes do tipo PxxP na proteína SH3, onde a utilização do protocolo de *ensemble docking* resultou em uma melhora no resultado do *docking* (Ki) de magnitude μM a nM, indicando uma melhor discrição das interações entre o receptor e os ligantes. O segundo exemplo foi o experimento de *virtual screening* utilizando a Enzima Conversora da Angiotensina humana I (hACE) com um conjunto de 1797 moléculas supostamente inativas e 46 inibidores conhecidos e caracterizados na literatura. A utilização do *ensemble docking* neste experimento permitiu um perfil de enriquecimento de ligantes muito mais abrangente quando comparado com o experimento realizado somente com a conformação obtida do cristal (condição controle). Assim, Rigel representa uma solução para a implementação de *pipelines* voltada para área de biologia computacional, diminuindo a complexidade de implementação e uso por parte dos usuários menos experientes. Rigel está disponível gratuitamente para uso acadêmico em: http://biocomp.uenf.br.

Palavras chaves: Biologia Computacional, Modelagem por homologia, *Docking molecular*, *Virtual Screening*, Gerenciamento de *pipelines*.

ABSTRACT

Experiments in Computational Biology have high demand nowadays for the rapidity of analysis of large amounts of data and low relative cost. Many of these experiments are performed by combining various computational tools which increases the complexity of usability and the cost of processing. In this context, several software for the automation of these pipelines were created to work on genomic data analysis due to high throughput sequencing techniques. Parallel to this, there is a hiatus of specialized programs in the development of pipelines of free data analysis of computational biology, which negatively impacts the experimentation in the biotechnology and pharmaceutical areas. Thus, in this work we describe a new pipeline platform called Rigel: A web tool that facilitates the creation and use of computational biology pipelines with the use of tools available for free to the academic world. The current version of Rigel includes predictive pipelines of protein structure, molecular docking and virtual screening; called AutoModel Web, MDR SurflexDock and HTP SurflexDock respectively. The latter two use the ensemble docking protocol to increase the conformational space of the receiver through the use of structural information obtained from crystallography and molecular simulation. In this context, we illustrate the usability of Rigel with the use of two case studies: The docking of PxxP-type ligands in SH3 protein, where the use of the ensemble docking protocol resulted in an improved docking result (Ki) of magnitude μM to nM, indicating a better discretion of the interactions between the receptor and the ligands. The second example was the virtual screening experiment using Human Angiotensin I Converting Enzyme (hACE) with a set of 1797 supposedly inactive molecules and 46 inhibitors known and characterized in the literature. The use of ensemble docking in this experiment allowed a much more comprehensive binder enrichment profile when compared to the experiment performed only with the conformation obtained from the crystal (control condition). Thus, Rigel represents a solution for the implementation of pipelines focused on the area of computational biology, reducing the complexity of implementation and usability by less experienced users. Rigel is available for free for academic use at: http://biocomp.uenf.br.

Keywords: Computational Biology, High Performance Computing, Homology Modeling, Molecular Docking, Virtual Screening.

SUMÁRIO

Índice de Figuras

Índice de tabelas

Capitulo 1: Introdução

A Bioinformática é um campo multidisciplinar que emprega técnicas de áreas de conhecimento como a Matemática, Estatística, Física, Biologia molecular, Genômica, Ecologia e ferramentas computacionais para o estudo de problemas e questões biológicas. Normalmente, os seus experimentos tem o objetivo de extrair informações úteis de dados brutos da identificação de genes, resoluções de estrutura tridimensional de proteínas, identificações de alvos moleculares terapêuticos, desenho racional de fármacos, montagens de árvores filogenéticas, além de outras aplicações (CHANG; TOMMASO, DI; NOTREDAME, 2014;DU; XIE; HUANG, 2014;GILISSEN et al., 2012;HWANG; VREVEN; WENG, 2014;MOULT et al., 2014). Neste contexto, alguns destes experimentos *in silico* são formados por um conjunto de tarefas ou algoritmos que necessitam ser executadas sequencialmente em um *pipeline* ou *workflow* (GUNARATHNE et al., 2011). Um *pipeline* de bioinformática estrutural bastante popular é a predição da estrutura tridimensional de proteínas a partir de sua sequência de aminoácidos utilizando o método de modelagem por homologia. Este método gera modelos com a utilização de uma proteína-molde e é formado por cinco etapas distintas (DORN et al., 2014). Estes *pipelines* demandam, em sua maioria, o desenvolvimento de *scripts* por meio da programação, como parte da metodologia, o que dificulta o uso cotidiano, e aumenta a curva de aprendizagem no uso destas ferramentas (SADEDIN; POPE; OSHLACK, 2012). Paralelamente a isto, alguns *pipelines* requerem o desenvolvimento de otimizações do processo de forma a possuir um melhor desempenho da aplicação, além do uso de técnicas de computação de alto desempenho (*High Performance Computing* – HPC) com o uso de servidores de alta *performance*, computação distribuída e programação em GPU (*Graphic Processing Unit*), entre outras abordagens (BUCK et al., 2014;D'ANGELO; RAMPONE, 2014;EVANS et al., 2016). O *structure-based virtual screening* (SBVS), por exemplo, é uma técnica computacional que utiliza a informação estrutural de uma proteína para encontrar possiveis inibidores de uma biblioteca de compostos que se ligam a proteína com maior afinidade (LI; SHAH, 2017). Normalmente, a classificação destas moléculas é dada por meio de experimentos de *docking molecular*, isto é, o cálculo do modo de ligação do ligante em relação a proteína receptora (FRADERA; BABAOGLU, 2018). Devido a sua característica exploratória, a execução de cada *docking* pode levar entre alguns minutos a algumas horas. Neste contexto, utilizar SBVS para avaliar milhares de moléculas sequencialmente nem sempre é viável, sendo obrigatório optar por

uso de técnica de paralelização de tarefas e o uso de técnicas de HPC (BOEHM, 2011). Além disso, a maioria das ferramentas de *docking* considera pouca ou nenhuma flexibilidade do receptor em seus cálculos o que impacta negativamente o SBVS (AMARO et al., 2018).

Os *pipelines* citados anteriormente executam programas que possuem interface de usuário em modo texto em sua grande maioria, o que favorece a sua automatização. Aproveitando-se disso, alguns laboratórios criam pacotes de software com interfaces gráficas amigáveis que facilitam a usabilidade desses *workflows*, sendo que alguns destes estão disponíveis pela internet. Além disso, existem frameworks especializados na criação de *pipelines* científicos chamadas de *Scientific Workflow Systems* (YU; BUYYA, 2005). Nestas ferramentas os *pipelines* podem ser adicionados pelo usuário por meio de uma linguagem de programação gráfica ou textual (ELHAI et al., 2009). No caso especifico de bioinformática, inúmeras soluções de gerenciamento de *workflows* por computador foram desenvolvidas, sendo a mais proeminente delas o *framework* Galaxy (AFGAN et al., 2011). No entanto, a maioria destes softwares contém componentes que tratam somente de protocolos relacionados à análise do sequenciamento de dados genômicos provenientes de técnicas de sequenciamento de alto rendimento, criando assim um hiato para programas gratuitos desta classe relacionados a outras áreas da biologia computacional, especialmente da simulação molecular e desenho de fármacos.

Parar suprir essa demanda, criamos uma plataforma web para o desenvolvimento de *pipelines*, chamada Rigel. A versão atual de Rigel inclui três *pipelines* experimentais: (I) AutoModel Web, responsável por predição de estrutura de proteínas utilizando o protocolo de modelagem por homologia; (II) o MDR SurflexDock (*Molecular Dynamics-based Receptor Surface Flexibility for Docking*) que permite avaliar a interação de pequenos ligantes no sítio ativo de receptores proteicos utilizando o protocolo de docking; (III) o HTP SurflexDock (*High Throughput Receptor Flexibility for Docking*), uma ferramenta que utiliza a técnica de *structure-based virtual screening* para encontrar candidatos a inibidores de uma proteína a partir de uma biblioteca de compostos. As ferramentas MDR SurflexDock e HTP SurflexDock, utilizam dados provenientes de cristalografia e simulação molecular para aumentar o espaço conformacional do receptor, em uma técnica chamada de *ensemble docking*. O objetivo do maior espaço conformacional do receptor é simular sua flexibilidade implicitamente, tornando, assim, os resultados do experimento *in silico* mais realísticos. Para avaliar a usabilidade e a confiabilidade destes *pipelines* foi realizado um estudo de

caso com cada ferramenta: No MDR SurflexDock foram utilizados ligantes do tipo PxxP e o receptor SH3 da vinculina humana para avaliar se a técnica de *ensemble docking* foi capaz de obter *dockings* com resultados melhores do que o experimento utilizando o docking rígido. Neste estudo de caso, os nossos resultados apontam que este *pipeline* de *docking* conseguiu reduzir a constante de inibição (Ki) da interação da magnitude de μM para nM. No caso do HTP SurflexDock foi avaliado se o *pipeline* implementado seria capaz de "identificar" os inibidores da Enzima Conversora da Angiotensina I a partir classificação de uma biblioteca contendo inibidores e outras moléculas supostamente sem atividade com a enzima. Neste experimento, o HTP SurflexDock apresentou um perfil de enriquecimento de ligantes muito mais amplo quando comparado com o experimento controle.

Neste contexto, diferente de outras ferramentas de *Scientific Workflow Systems*, Rigel é um *framework* que possui integrado um conjunto de ferramentas voltado para experimentos de simulações moleculares da biologia computacional. A partir de *pipelines* inclusos neste *framework*, o usuário poderá fazer experimentos de predição de estrutura de proteínas, simulação molecular e busca de drogas. Além disso, o usuário poderá instalar Rigel, em máquinas locais, como em ambientes de computação de alto desempenho ou em nuvem. Rigel está disponível gratuitamente para o uso acadêmico em http://biocomp.uenf.br.

1.1 Objetivos

Criar Rigel, um *framework* para experimentos em *pipelines* de Biologia Computacional utilizando ferramentas de uso gratuito de modelagem, simulação e *docking*. A ferramenta deverá poder ser executada com a utilização de computação em alto desempenho (HPC) ou em nuvem.

1.2 Objetivos Específicos:

- Implementar o AutoModel Web: *Pipeline* de modelagem molecular de proteínas por homologia.
- Implementar MDR SurflexDock: *Pipeline* de análise de interação proteína-ligante por meio de *ensemble docking.*
- Implementar HTP SurflexDock: *Pipeline* de análise de interação de inibidores (*virtual screening*) por meio de *ensemble docking.*

1.3 Organização do trabalho

Além desta introdução, o presente trabalho foi estruturado em mais 5 capítulos, a serem desenvolvido da seguinte forma:

- Capitulo 2: Introduz os conceitos teóricos de *pipelines* computacionais de modelagem simulação e *docking* utilizados para o desenvolvimento deste trabalho, como também apresenta os trabalhos relacionados.

- Capitulo 3: Apresenta a estrutura lógica da plataforma Rigel e dos *pipelines* de predição de estrutura de proteínas, *docking* e *virtual screening*

- Capitulo 4: Aprofunda os detalhes do *pipeline* de *ensemble docking* utilizado no MDR SurflexDock e apresenta um estudo de caso com o domínio SH3 da vinculina humana.

- Capitulo 5: Apresenta o HTP SurflexDock: o *pipeline* de *virtual screening* e ilustra sua utilização com um estudo de caso com a Enzima Conversora da Angiotensina I.

- Capitulo 6: Apresenta as conclusões deste trabalho.

Capitulo 2: Fundamentação teórica

2.1 *Pipeline* de modelagem de proteínas.

Técnicas experimentais como a cristalografia por raios-X e a ressonância magnética nuclear (NMR) conseguem determinar estruturas tridimensionais de proteínas em resoluções atômicas. No entanto, estes métodos são bastante onerosos e possuem uma grande quantidade de limitações quanto ao tamanho da proteína, flexibilidade, custo, entre outros (VERLI, 2014). Por outro lado, determinar a sequência de aminoácidos que compõe uma proteína é uma tarefa bem mais simples. Estes dois aspectos criam um enorme *gap* entre o número de sequencias e estruturas tridimensionais determinadas (CAVASOTTO e PHATAK, 2009). Como a estrutura terciária de uma proteína é determinada principalmente pela sua estrutura primária, métodos teóricos implementados em softwares de computador poderiam predizer a sua estrutura tridimensional a partir de sua estrutura primária (LEE et al., 2017). No entanto, estes métodos computacionais de predição também esbarram em uma série de desafios, como por exemplo, a enorme quantidade de estados conformacionais que pode ser adotado por uma proteína e que precisa ser explorado pelo software até que se encontre uma solução de menor energia (DE OLIVEIRA et al., 2015). Neste contexto, os métodos que modelam a estrutura de proteínas utilizando somente a sua sequência de aminoácidos são conhecidos modelagem de proteínas por primeiros princípios ou *ab initio* (LEE et al., 2017; YANG e ZHANG, 2015). Normalmente, a modelagem *ab initio* utiliza alguma heurística para otimizar a busca no espaço conformacional atuando com uma função de energia para escolha do melhor caminho até o enovelamento (KANDATHIL et al., 2018). Atualmente estas ferramentas são capazes de modelar com sucesso pequenos peptídeos com centenas de aminoácidos (RASHID et al., 2015). Paralelamente a esta técnica existem outros métodos de modelagem que além de usar a informação da sequência primária, utilizam informação estrutural de proteínas já determinadas. Um destes métodos chamado de modelagem por homologia, ou modelagem comparativa, baseia-se na evidência de que a estrutura tridimensional de proteínas é evolutivamente mais conservada do que sua sequência primária e que existe um número limitado de enovelamentos de proteínas (CAVASOTTO e PHATAK, 2009). Desta forma, é possível agrupar famílias de proteínas que possuem uma origem em comum, sendo estruturalmente semelhantes. Assim, conhecendo-se a qual família uma proteína pertence, pode-se utilizar esta informação como molde para a construção de

sua estrutura tridimensional (Figura 1) (WEBB e SALI, 2014a). No computador a modelagem por homologia ocorre em uma série de passos sequenciais ou iterativos que normalmente incluem:

(I) Busca e seleção de uma ou mais proteínas-moldes (*template*) que seja estruturalmente relacionada com a proteína de interesse. A busca por estruturas das proteínas-moldes são normalmente feitas por comparação da estrutura primária da proteína de interesse com sequencias de proteínas armazenadas no banco de dados do RSCB PDB (http://www.rcsb.org/) (BERMAN et al., 2000).

(II) Alinhamento de sequencias para a determinação de regiões estruturalmente conservadas,

(III) A construção do modelo tridimensional pode ser realizada por uma grande variedade de métodos, como por exemplo, importação de estruturas secundárias e *loops* dos *templates* ou através de restrições espaciais obtidas a partir do alinhamento,

(IV) Validação do modelo gerado através de análise de parâmetros estereoquímicos ou análise estatística (KELLEY et al., 2015; OSVALDO ANDRADE SANTOS FILHO, 2003; WEBB e SALI, 2014b).

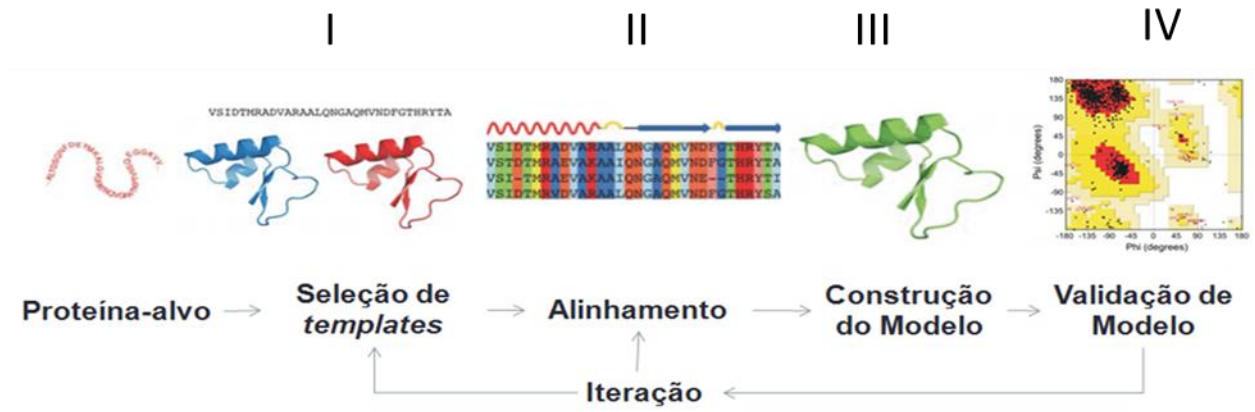


Figura 1- Etapas de uma modelagem molecular de proteínas por homologia.

Embora a modelagem por homologia possua uma técnica fácil de executar e de baixo custo computacional, esta técnica possui uma série de limitações. A principal delas está relacionada à necessidade de se encontrar uma proteína-molde que seja estruturalmente relacionada com a proteína de interesse. Além disso, quanto menor a similaridade entre as duas proteínas menor será a qualidade do modelo gerado, pois, existe uma maior probabilidade do aparecimento de gaps no alinhamento, isto é, espaços sem correspondência estrutural (ESWAR et al., 2006). O próprio alinhamento pode necessitar da intervenção do usuário, pois é comum encontrar regiões mal alinhadas.

A série de passos pode ser executada por vários programas ou por um único software como no Modeller (WEBB e SALI, 2017). Este é um programa bastante utilizado pela comunidade acadêmica que realiza a modelagem através do cumprimento de restrições espaciais. A interface de uso do Modeller é através de scripts na linguagem python, o que facilita o desenvolvimento de *pipelines* utilizando este programa. Embora esta interface dificulte o uso do usuário iniciante, facilita o desenvolvimento de softwares baseados nessa ferramenta. Desta forma, existem vários programas que executam o Modeller de forma totalmente autônoma ou manual através de scripts (WEBB e SALI, 2014). Um destes programas de modelagem é o AutoModel (DE A FILHO et al., 2018) que é um software semiautomático que realiza a modelagem em uma arquitetura cliente-servidor. Através de uma interface simples, o usuário possui o controle de pontos chaves da modelagem como, por exemplo, escolha da proteína-molde, heteroátomos, edição manual do alinhamento. As tarefas de processamento intensivo são executadas no servidor. O AutoModel foi desenvolvido em Python e de forma modular para facilitar a manutenção do software como também o seu código pode ser reutilizado para a criação de outros softwares. Além do AutoModel, existe uma série de serviços on-line de modelagem molecular. A maioria destes serviços oferece a modelagem por meio de uma página Web e são totalmente automatizados como o ESyPred3D (LAMBERT et al., 2002), Swiss-Model (BIASINI et al., 2014) e Robetta (KIM et al., 2004). Além disso, alguns destes programas utiliza o Modeller em alguma etapa da modelagem como ESyPred3D.

Um grande desafio ao criar *pipeline*s para a modelagem por homologia é a compatibilidade de formatos gerados pelos diferentes softwares que compõe o *pipeline*. Por exemplo, embora o Modeller reconheça uma série de formatos de arquivos de sequência, em algumas etapas utiliza o formato pir que é próprio do programa. Nem todos os programas tem a capacidade de trabalhar com

este formato de arquivo o que torna a conversão necessária e podendo ser bastante complicada. Além disso, o Modeller como motor de modelagem não aproveita os recursos de multiprocessadores ou multicore o que degrada o desempenho em análises de alto rendimento.

2.2 Análises de sitio ativo por meio do *pipeline* de *docking molecular*

As interações entre proteínas-ligantes são os principais atuantes em todos os processos fisiológicos e patológicos e as proteínas envolvidas nesses processos representam os maiores alvos para o desenvolvimento de fármacos. Neste sentido, um dos protocolos de bioinformática mais utilizados é o *docking molecular*, este método é capaz de predizer a interação não covalente de complexos de proteína-ligantes de forma tridimensional (ANTUNES et al., 2015). Para fazer isto, o software *docking* utiliza a informação estrutural de uma proteína receptora para tentar "encaixar" a molécula ligante em nível atômico, assim, permitindo predizer a localização do sitio de ligação na superfície do receptor e a sua afinidade com ligante (RENTZSCH e RENARD, 2015). Neste sentido, a partir dessas informações, é possível compreender de que forma ocorrem as interações entre proteínas-ligantes, assim como prever qual é a energia livre de ligação destas interações.

Os primeiros softwares de *docking* surgiram no início da década de 80 (EWING et al., 2001), e evoluíram de forma a serem capazes de produzir resultados com bastante precisão e um relativo baixo custo computacional (DOS SANTOS MUNIZ e NASCIMENTO, 2015). A entrada do software normalmente é feita carregando-se as estruturas tridimensionais do receptor e do ligante. Normalmente, o programa de *docking* calcula vários modelos tridimensionais da interação e os classifica de acordo com a energia livre de ligação, assim, a qualidade do modelo de interação gerada é afetada principalmente pela resolução atômica da estrutura do receptor. Para um bom experimento sugere-se o uso de estruturas oriundas de espectrometria por difração de raios-X, NMR, entre outras técnicas. Bons resultados podem ser alcançados por meio de modelos obtidos através de modelagem molecular, caso faltem estruturas de alta resolução (SEDDON et al., 2012).

Normalmente, os experimentos de *docking* são utilizados para dois experimentos distintos: (I) caracterização do modo de ligação do ligante na formação de um complexo com o receptor, de forma a identificar a conformação tridimensional adotada pelo ligante em relação a uma região de interesse do receptor. Outro parâmetro importante é o cálculo da afinidade do ligante com o receptor. Este tipo de estudo pode ser utilizado para caracterizar o sítio ativo do receptor. O outro experimento (II) se constitui em experimentos de *virtual screening*. Nesse estudo, tenta-se

identificar as moléculas de uma biblioteca de compostos que apresentam atividade com o receptor. Tais moléculas poderão ser utilizadas como estruturas líderes para o desenho racional de fármacos (PAGADALA et al., 2017).

O programa de *docking molecular* tenta acomodar o ligante em uma determinada região do receptor e é gerado um modo de ligação a cada acomodação calculada. O modo de ligação, também conhecido como pose, contém a orientação, posição e conformação do ligante em relação à superfície da proteína. Além destes, é calculada a energia livre de ligação através de abstrações das contribuições eletrostáticas envolvidas na interação entre as duas moléculas (DOS SANTOS MUNIZ e NASCIMENTO, 2015).

Entre os algoritmos de busca utilizados para a realização do *docking* podem-se mencionar as redes neurais (WASSERMAN, 1989), os algoritmos evolutivos (Layton, 2008) e os algoritmos genéticos (KOZA et al., 1999). Este último, utilizado no software AutoDock (RENTZSCH e RENARD, 2015), um dos mais utilizados da área. O AutoDock é uma ferramenta de *docking* que usa um campo de força semi-impirico para calcular as conformações do complexo durante o protocolo de *docking*. Este programa calcula o *docking* por meio de um *pipeline* contendo os seguintes passos:

(I) Inicialmente os arquivos de coordenadas do receptor e do ligante são convertidos para o formato PDBQT, que possui as cargas parciais, os pontos de torção e tipos para cada átomo;

(II) São calculados mapas de afinidades para diferentes tipos de átomos do ligante e do receptor dentro de um espaço pré-determinado;

(III) Para aumentar a eficiência da algoritmo é utilizado um algoritmo de busca, sendo o mais usado o algoritmo lamarckiano (LGA);

(IV) A análise dos resultados pode ser feita por meio da visualização das conformações, agrupamentos de conformações similares, análise de energia, entre outros métodos.

O AutoDock é uma ferramenta de modo texto, todavia, o desenvolvedor disponibiliza um ambiente gráfico e *scripts python* para preparação de arquivos por meio do AutoDockTools (MORRIS et al., 2009).

2.3 Análise de interação de inibidores por meio de *virtual screening*

O desenho racional de fármacos é um processo de busca e desenvolvimento drogas baseado no conhecimento corrente de um determinado alvo biológico (MANDAL et al., 2009). Atualmente, a busca por novas drogas pode ser realizada por métodos automatizados de última geração chamados de *High Throughput Screening* (HTS) (DÖRR et al., 2016). Estes métodos de HTS normalmente utilizam equipamentos especiais para realizar milhões de testes químicos, farmacêuticos e genéticos por dia, seguido pela seleção dos melhores compostos (compostos ativos) através do processamento dos dados gerados (SZYMAŃSKI PAWEŁAND MARKOWICZ e MIKICIUK-OLASIK, 2012; XU e HAGLER, 2002). Uma desvantagem das técnicas de HTS é que é um processo dispendioso, pois, a cada teste é necessário um conjunto de compostos que são organizados em bibliotecas (MACALINO et al., 2015). Além disto, a geração destas bibliotecas por si só pode ser um desafio pois, a quantidade de moléculas no espaço químico conhecido atualmente é superior a grandeza de 10^{60} (REYMOND et al., 2010). Desta forma, selecionar compostos promissores através de filtros especiais ajuda a reduzir custos e o tempo de desenvolvimento do fármaco (BOEHM, 2011; PYZER-KNAPP et al., 2015). Por outro lado, utilizando ferramentas de desenho de fármacos assistido por computador (*Computer Aided Drug Design*), como o *Virtual High Throughput Screening* (vHTS) é possível encontrar um conjunto de compostos teoricamente ativos entre milhares de moléculas de uma biblioteca virtual de compostos (HASSAN BAIG et al., 2016). Esta metodologia, conhecida também como *Virtual Screening* (VS), avalia a interação entre compostos e alvos biomoleculares através de cálculos teóricos e podem ser complementares as técnicas de HTS ao criar filtros que ajudam a reduzir o número de moléculas a serem avaliadas (BOEHM, 2011; MACALINO et al., 2015; PYZER-KNAPP et al., 2015).

Os métodos de *Virtual Screening* são classificados em dois tipos: o primeiro, chamado de *Ligand Based Virtual Screening* (LBVS) busca por novos candidatos a ligantes que possuam propriedades químicas semelhantes a um ligante conhecido do alvo biomolecular. O segundo tipo é conhecido como *Structure Based Virtual Screening* (SBVS) e utiliza a informação estrutural do alvo biomolecular, normalmente uma proteína, para docar uma grande quantidade de moléculas em seu sitio ativo. Ao final, utiliza um critério de avaliação para filtrar um subconjunto das melhores moléculas que serão avaliadas posteriormente em ensaios biológicos (LI e SHAH, 2017; MACALINO et al., 2015). Neste contexto, o amadurecimento dos programas de *docking* aliado ao aumento do poder computacional durante a última década permite a avaliação da interação de

milhares de moléculas em um receptor proteico utilizando um computador pessoal em questão de horas. Assim, o SBVS é um método acessível, pois, necessita apenas de uma biblioteca virtual de moléculas como o Zinc (STERLING e IRWIN, 2015), estrutura tridimensional do alvo biomolecular, o software de SBVS e um computador.

2.4 O desafio da representação da flexibilidade do receptor no *docking*

A popularização da modelagem molecular e *docking molecular* permitiram que estas ferramentas fossem amplamente utilizadas para acelerar a descoberta de novos candidatos a drogas. No entanto, devido a simplificações com o objetivo de reduzir o custo computacional ambas ferramentas desconsideram os efeitos da flexibilidade das proteínas. As técnicas de predição da estrutura de proteínas fornecem uma imagem estática tridimensional da proteína e o *docking molecular*, por sua vez, calcula a interação de ligantes flexíveis, possivelmente drogas, em receptores estáticos mimetizando a teoria da "chave e fechadura". A qual diz que um receptor estático foi evolutivamente selecionado para acomodar outra molécula sem sofrer alterações em sua conformação (CAVASOTTO et al., 2005). Contudo, dados cristalográficos e de NMR apontam que as proteínas apresentam uma variedade conformacional que é importante para a interação com o substrato. Assim, o modelo "chave e fechadura" logo foi abandonado em favor do modelo do encaixe induzido (*induced fit*) onde a interação do substrato com o receptor induz alterações conformacionais na proteína (CHAUDHURY e GRAY, 2008). Neste contexto, algumas proteínas como a enzima conversora da angiotensina I, consegue reconhecer vários substratos com propriedades químicas diferentes em seu sitio ativo (NATESH et al., 2004). Diferentes tipos de ligantes podem induzir diferentes conformações na proteína receptora (CAVASOTTO et al., 2005). Por outro lado, a maioria das ferramentas de SBVS utiliza o receptor rígido em seus experimentos devido ao alto custo computacional (ANTUNES et al., 2015; SINKO et al., 2013). Assim, quando a proteína é modelada ou cristalizada com algum ligante, ela tende possuir uma conformação do sitio de ligação quimicamente complementar ao ligante devido ao efeito do encaixe induzido, este "efeito memória" da superfície de contato do receptor acaba gerando falsos negativos de ligantes do receptor que possuem propriedades químicas distintas do ligante cristalizado (JAIN, 2008) em experimentos de *docking*.

Para evitar o problema de "efeito memória" pode ser utilizada a técnica de *ensemble docking* que utiliza um conjunto de conformações relevantes do receptor para simular a flexibilidade

deste implicitamente. Nesta técnica, é feito *docking* dos ligantes com cada uma das conformações pertencentes ao *ensemble* (ANTUNES et al., 2015). As conformações podem ser obtidas através de experimentos como a cristalografia por raios-X e NMR ou por meio de simulações moleculares (AMARO et al., 2018). A simulação da dinâmica molecular utiliza a física newtoniana para simular a movimentação de átomos com baixo custo computacional. A tecnologia atual permite simular o comportamento dinâmico de proteínas em ambiente solvatado, o que permite o estudo das mudanças conformacionais destas biomoléculas (ABRAHAM et al., 2015). Utilizando a dinâmica molecular é possível obter um conjunto de conformações relevantes através de técnicas de *clustering* (agrupamento) e estas conformações podem ser utilizadas para compor o *ensemble docking*.

2.5 Obtenção de conformações por meio da Simulação Molecular

A simulação computacional é uma ferramenta utilizada na ciência e na indústria que tem como objetivo simular um fenômeno ou um processo do mundo real, o qual demandaria perigo, altos custos e inviabilidade de sua reprodução. Dentre os tipos de simulações que podem ser realizadas no computador está a dinâmica molecular de biomoléculas - por vezes classificada como química computacional e utilizada para o estudo de interações microscópicas entre moléculas. Neste sentido, a simulação molecular de biomoléculas pode ser utilizada como um pequeno laboratório, muitas vezes estendendo experimentos do mundo real por permitir a visualização e análise de fenômenos que não seriam possíveis ou muito difíceis por técnicas convencionais, pelo fato de se constituírem em eventos microscópicos, rápidos ou demasiadamente lentos.

Um dos métodos de simulação molecular mais utilizados é a Dinâmica Molecular que se vale dos princípios das mecânica newtoniana e hamiltoniana para fornecer informações sobre o comportamento dinâmico de todos os átomos que compõem o sistema em estudo. Isto é, a simulação molecular fornece uma trajetória onde especifica posições e velocidades de todas as partículas do sistema ao longo do tempo (DE VIVO et al., 2016). Por basear-se somente em princípios da mecânica clássica, a dinâmica molecular está restrita a simular as interações entre os átomos das moléculas que não haja mudanças nas ligações covalentes entre os átomos ou processos que levem a alterações eletrônicas dos átomos (GANESAN et al., 2017).

A dinâmica molecular consiste em solucionar equações do movimento para cada átomo que compõe o sistema em estudo. Além disso, as forças de interação sobre cada átomo são calculadas a cada instante por meio da derivada da função de energia potencial (LEIMKUHLER e MATTHEWS,

2016). O conjunto de todos os potenciais de interação utilizados pelo software de simulação molecular são definidos por funções de campos de força. A maioria dessas incluem termos para descrever ligações intramoleculares (força de interação dos átomos ligados covalentemente) e forças de interação para átomos não ligados. As ligações intramoleculares são tratadas como forças elásticas e harmônicas, em que os potenciais harmônicos podem ser descritos de forma simplificada pela lei de Hooke (KARPLUS e PETSKO, 1990)**.** Por outro lado, as forças de interação entre átomos não-ligados covalentemente são descritos pelos potenciais de Lennard-Jones e Coulomb (LEIMKUHLER e MATTHEWS, 2016). Importante destacar que os campos de força podem possuir parâmetros específicos para o tratamento de ligações de hidrogênios, diedros impróprios, penalizações associadas pelo desvio de valores de referência, entre outros parâmetros. A escolha do campo de força depende do sistema em estudo, mas grande parte dos softwares de simulação molecular suporta mais de um campo de força, sendo os mais comuns o AMBER (SONG et al., 2019), CHARMM (BROOKS et al., 2009), OPLS (JORGENSEN et al., 1996), GROMOS (NESTER et al., 2019), entre outros.

A resolução das equações diferenciais do sistema é feita por meio do método das diferenças finitas, sendo que os mais utilizados são os algoritmos de "*Verlet"* e "*Leap-frog*". Ambos são algebricamente equivalente mas oferecem melhor precisão em relação ao método de Euler, para as equações de newton (HAIRER et al., 2003; LEIMKUHLER e MATTHEWS, 2016; MAZUR, 1997). Alcança-se o conjunto de posições e velocidades de cada átomo, denominado trajetória, por meio da resolução de equações de forma sucessiva para cada instante (NAMBA et al., 2008). Além de especificar o campo de força, outras configurações iniciais da simulação são importantes para a precisão do experimento *in silico.* Dentre esses parâmetros estão o tipo de solvatação utilizado para a molécula e as posições iniciais dos átomos que constituem o sistema.

Para realizar a simulação pode ser interessante a inclusão da molécula estudada em ambiente solvatado, onde em ferramentas de simulação molecular é possível realizar a solvatação de forma explícita ou implícita. No modelo de solvatação explícito, moléculas do solvente, (água) são adicionadas ao sistema. Combinados ao solvente a fim de neutralizar as cargas do sistema, átomos de Cl^- e Na^+, por exemplos, são também adicionados (NAMBA et al., 2008). Vários modelos para descrever as moléculas de água foram desenvolvidos, dentre estes, os mais utilizado são da família SPC (*Simple Point Charge*) (GOPAL et al., 2015) e da família TIP (*Transferable Intermolecular*

Potentials) (ABASCAL e VEGA, 2005; JORGENSEN, 1981). Os membros destas famílias possuem diferenças estruturais e termodinâmicas importantes que devem ser avaliadas em cada estudo. Em simulações de sistemas grandes pode ser interessante o uso do modelo de solvatação implícita, pois, nesse sistema o solvente é representado como um continuo que apresenta um uso computacional por ter menor grau de liberdade do sistema. No entanto, a solvatação implícita não permite a simulação de águas estruturais (SCHERAGA et al., 2007).

A quantidade de moléculas de solventes inclusa no sistema afeta de forma diretamente proporcional o uso computacional. Consequentemente, os programas de simulação adicionam o mínimo de moléculas de solvente possível em uma caixa de simulação de tamanho finito. São utilizadas condições periódicas de contorno (*Periodic Boundary Conditions*) para simular que o sistema está dentro de uma caixa de dimensões infinitas e para isso a caixa é espelhada em todas as suas faces de modo que quando uma partícula sai por uma face da caixa de simulação outra partícula entra com a mesma velocidade da face oposta, mantendo assim o número de partículas e a energia do sistema (Figura 2) (ALLEN e OTHERS, 2004).

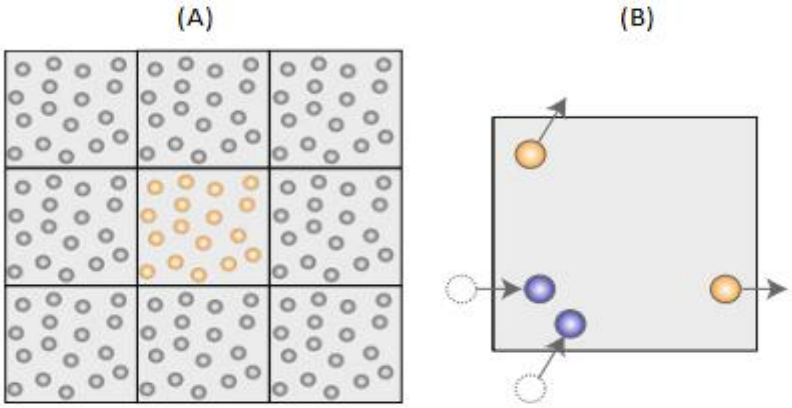


Figura 2 - Caixa de simulação com condições periódicas de contorno ilustrada em duas dimensões. As caixas vizinhas contém as mesmas partículas da caixa principal (A), assim, quando uma partícula sai da caixa principal, ele entra novamente pela face oposta, mantendo, assim, o mesmo número de partículas (B). Fonte: (LEIMKUHLER e MATTHEWS, 2016).

Após adicionar a molécula na caixa de simulação é interessante fazer uma otimização na geometria do tamanho e ângulos das ligações covalentes da molécula, como também melhorar o acesso do solvente na molécula por meio de algoritmos de minimização de energia. Entre os vários

métodos de minimização de energia, o *steepest descent* (FLETCHER, 2013) é o mais utilizado, pois apresenta baixo custo computacional. Com o sistema minimizado pode-se iniciar a criação da trajetória por meio da simulação temporal do sistema. O início dessa simulação coincide com a primeira etapa que se chama período de equilíbrio, cujas propriedades do sistema não se mantêm constantes para evitar a "explosão" do mesmo.

As propriedades que se mantêm constantes durante o período de equilíbrio são chamadas *ensemble* estatístico (RAPAPORT, 2004). Normalmente são utilizados um dos seguintes *ensembles:* o NVT (NOSÉ, 1984), o qual se mantém o número de partículas, volume e temperatura constante, variando a pressão; o NPT (ANDERSEN, 1980), o qual se mantém o número de partículas, pressão e temperatura constante, variando o volume e o NVE (HAILE, 1992), o qual se mantém o número de partículas, volume, estabilizando a energia. Após a fase de equilíbrio, inicia-se a produção da trajetória, e podem-se avaliar as diferentes propriedades do sistema de interesse.

Determinada temperatura e pressão do sistema podem ser alcançadas durante a produção da trajetória por meio dos algoritmos de termostato e barostato respectiva. Os primeiros algoritmos para este fim, foi desenvolvido por Berendsen (1984) e nele as velocidades dos átomos são reajustadas de acordo com um fator de correção, até que se alcance as temperatura e pressão desejadas. No algoritmo de termostato simula-se que o sistema está acoplado fracamente a um banho externo com a temperatura desejada. No entanto, estes algoritmos possuem uma baixa precisão e o reajuste continuo da velocidade dos átomos gera um amortecimento exagerado da velocidade das partículas, o que consequentemente causa artefatos na simulação, não respeitando nenhum dos *ensembles* estatísticos (BRAUN et al., 2018). Softwares de simulações atuais, utilizam outros algoritmos mais precisos como o termostato Nosé-Hoover (HOOVER, 1985; NOSÉ, 1984) e barostato Parrinello-Rahman (PARRINELLO, Michele e RAHMAN, 1981). Estes algoritmos fixam uma temperatura e a pressão na simulação, mas permitem flutuações de temperatura de modo que se respeite o *ensemble* estatístico utilizado.

Após a produção da trajetória, a obtenção de conformações mais representativas podem ser obtidas através do agrupamento de conformações análogas (*clustering*). Um dos algoritmos de *clustering* mais utilizados é o GROMOS (DAURA et al., 1999) que agrupa estruturas da simulação que são similares. A similaridade é medida por meio do cálculo do RMSD (*Root Mean Square Diferences*) entre as estruturas. Após essa classificação, o algoritmo seleciona a estrutura central de

cada agrupamento. O perfil do agrupamento pode ser visualizado por meio do gráfico de desvio médio quadrático (RMSD) em função do tempo de simulação, como demostrado na figura 3. Estas conformações obtidas por *clustering* podem ser utilizadas pra a construção do conjunto utilizado em *pipelines* de *ensemble docking.*

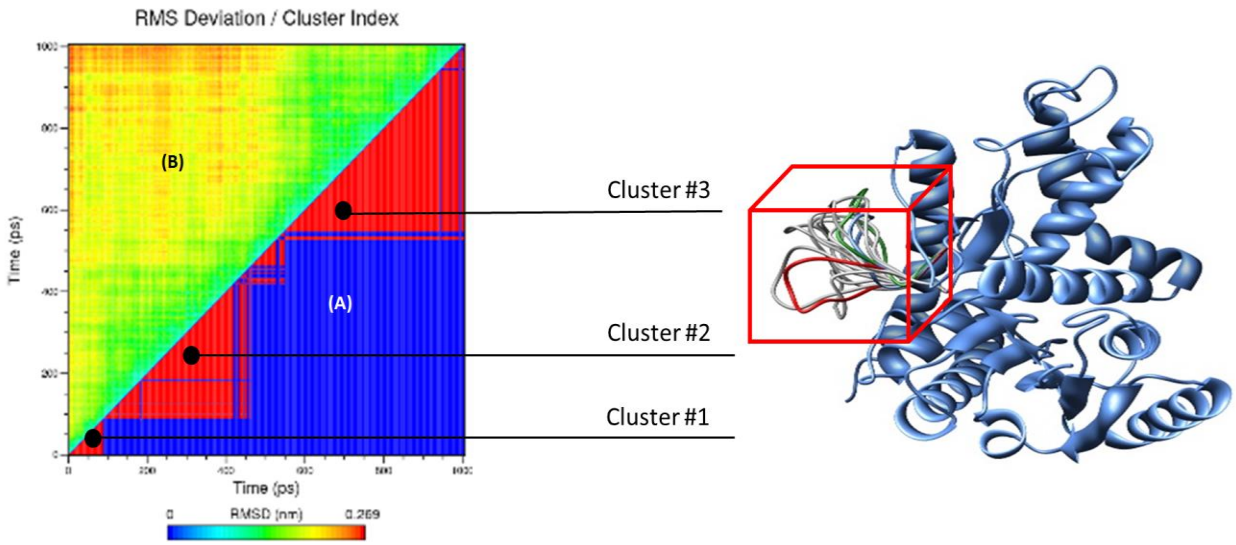


Figura 3 - Gráfico de desvio médio quadrático (RMSD) em função do tempo de simulação. Este gráfico bi-dimensional indica a diferença de RMSD entre as estruturas em função do tempo. (A) Os padrões bem definidos em vermelho indicam a formação de clusters de estruturas semelhantes. (B) O código de cores indica a diferença entre 2 conformações bidimensionalmente.

2.6 Trabalhos relacionados

Scientific Workflow Systems (SWS) são ferramentas bastante utilizadas no meio cientifico para gerência de *pipelines* que executam programas ou manipulações de dados sequencialmente. Alguns destes SWS têm como objetivo otimizar os recursos computacionais para garantir o melhor desempenho da aplicação, com a utilização de computação em nuvem ou em grade (YU; BUYYA, 2005;YUAN et al., 2010). No caso especifico da biologia computacional foram desenvolvidos algumas ferramentas como listadas abaixo. Sendo que a maioria delas é voltada especialmente para projetos relacionados a área da genômica.

O Anduril (CERVERA et al., 2019) é um sistema SWS escrito em Java e desenvolvido pela Universidade de Helsinki. O objetivo do Anduril é fornecer analises eficientes para dados obtidos de sequenciamento de ultima geração. Sua interface foi desenvolvida para ser flexível na construção

dos *pipelines* e com suporte ao uso de ferramentas externas (OVASKA et al., 2010). A especificação dos *pipelines* é feito por meio de uma linguagem em script própria chamada de AndurilScript Language ou por meio da linguagem Scala (ODERSKY et al., 2004). Novos módulos chamados de componentes podem ser adicionados aos mais de 400 componentes já inclusas. Estes novos componentes podem acessar ferramentas externas por meio da especificação dos programas externos e seus parâmetros em arquivos do tipo XML. Anduril está disponível gratuitamente em http://anduril.org.

O BioBike (ELHAI et al., 2009) é um sistema em nuvem baseado no paradigma de Plataforma como um Serviço (PaaS), isto é, BioBike fornece uma API para o desenvolvimento de *pipelines* para ser utilizada em ambientes de computação em nuvem. O objetivo principal do BioBike é fonecer um ambiente de implementação de *pipelines* para biólogos com baixo conhecimento em linguagens programação. Desta forma, a especificação dos *pipeline* é feita por meio de uma linguagem de programação gráfica chamada de BioLingua. Assim como o Anduril, o BioBike fornece módulos para o tratamento de resultados de experimentos da área da genômica. BioBike possui suporte a programas externos que podem ser adicionadas aos módulos pré-existentes através de sua descrição em arquivos do tipo XML. BioBike está disponível em: https://github.com/jeffshrager/biobike.

Discovery Net (ĆURČIN et al., 2002) é uma ferramenta de desenvolvimento de *workflows* científicos que permite o uso de computação distribuída. Originalmente este programa não possuía ferramentas para a construção de *pipelines* de bioinformática. No entanto, softwares para anotações de genes e proteínas foram adicionados por ROWE e colaboradores (2003). A especificação dos *pipelines* no Discovery Net é realizada por meio de uma linguagem de programação própria chamada de *Discovery Process Markup Language* (DPML) que é baseada em XML como também pode ser realizado graficamente por meio de grafos orientados.

Galaxy Project (AFGAN et al., 2011) é a ferramenta mais conhecida para o uso e gerenciamento de *workflows* em bioinformática. Esta plataforma foi desenvolvida inicialmente para fornecer *pipelines* para pesquisas relacionada à área genômica de forma colaborativa podendo ser implementado servidores de alto desempenho ou na nuvem. No entanto, atualmente é usado com uma ferramenta para gerenciamento de *pipelines* para outra áreas da biologia computacional e interface web para uma série de programas relacionados com de predição de estrutura proteínas,

refinamento de loops, detecção de domínios e *docking* por meio do servidor do GalaxyWeb e GalaxyDock (KO et al., 2012). Diferente do Galaxy Project original, o GalaxyWeb não permite alterar os *pipelines* já existentes e seus experimentos são realizados de forma interativa com o usuário. Novos *pipelines* podem ser adicionado ao Galaxy Project por meio de arquivos no formato XML e pelo software BioBlend (SLOGGETT; GOONASEKERA; AFGAN, 2013) que fornece uma API de alto nível para a especificação dos *workflows*. O Galaxy Project está disponível em https://usegalaxy.org/.

Schrödinger (https://www.schrodinger.com/) é uma companhia que desenvolve plataformas de experimentos *in silico* com alta precisão para as áreas farmacêutica, biotecnologia, química, engenharia de materiais e eletrônica. Seus produtos são utilizados por grande parte das empresas farmacêuticas para acelerar a busca e desenvolvimento de medicamentos (KALYAANAMOORTHY; CHEN, 2011). Suas suítes de programas incluem softwares de modelagem molecular, simulação molecular em nível atômico, *docking molecular*, *virtual screening* e etc (DAGAN-WIENER et al., 2017; ROOS et al., 2019). Estas ferramentas podem ser utilizadas em protocolos para experimentos que necessitam maior precisão, como, por exemplo, o protocolo *Schrödinger's Induced Fit* (IFD) permite realizar o *docking* levando em consideração os efeitos do encaixe induzido. Este protocolo utiliza o Glide (BHACHOO; BEUMING, 2017), o software da Schröndiger responsável pelo *docking*, em conjunto com o software de modelagem Prime (SINDHIKARA et al., 2017). O protocolo IFD realiza *docking* com o receptor flexível implicitamente através da diminuição dos raios de van der Waals e Coulomb (CLARK et al., 2016). Para cada pose calculada é feita a acomodação do ligante por meio da reorientação das cadeias laterais do sítio ativo minimização de energia pelo software Prime.

Tome-3 (PONS; LABESSE, 2009) não é um software para gerenciamento de *workflows*, mas é um servidor web que oferece *pipelines* para a predição da estrutura de proteínas por modelagem por homologia e *docking molecular* rígido. O *pipeline* de modelagem é realizado através de vários programas como o ClustalW (LARKIN et al., 2007), T-Coffee (TOMMASO, DI et al., 2011) e Muscle (EDGAR, 2004) para alinhamento. A etapa de modelagem ainda conta com o reconhecimento de tipos de enovelamento feito software FUGUE (SHI; BLUNDELL; MIZUGUCHI, 2001) e SP3 (ZHOU; ZHOU, 2005) e a validação é feita por meio do Virify3d (EISENBERG; LÜTHY; BOWIE, 1997). Após a obtenção do modelo é feito o *docking* rígido dos

ligantes presentes nas proteínas-moldes utilizados para a predição da estrutura tridimensional da proteína. O Tome-3 foi escrito em Perl e seu *pipeline* de modelagem é semi-interativo, necessitando que o usuário tome algumas decisões no *pipeline*. O Tome-3 é de uso gratuito para usuários acadêmicos e está disponível em http://abcis.cbs.cnrs.fr/AT2/.

Capitulo 3: Plataforma para HPC: Rigel.

Para alcançar os objetivos deste projeto, desenvolvemos a plataforma web Rigel que possui inicialmente 3 *pipelines* de experimentos de Biologia Computacional. Neste sentido, o software desenvolvido inclui os *pipelines* para modelagem de proteínas por homologia, análises de sítio ativo por meio de *ensemble docking* e análises de interação com inibidores através *virtual screening*. Em Rigel, cada *pipeline* é constituído por uma série de algoritmos implementados em diferentes programas e módulos de bioinformática já consagrados sob uma arquitetura em camadas sobrepostas conforme Figura 4: Subsistema *WebService*, camada de módulos de bioinformática e camada de gerenciamento de multitarefa. O desenvolvimento em multicamadas permite a reutilização do código como também a implementação em diferentes arquiteturas computacionais. Todas as camadas de Rigel foram desenvolvidas utilizando testes automatizados, um modelo de testes estruturados que compara os resultados encontrados com os resultados esperados previamente especificados de cada módulo do programa, a fim de encontrar inconsistências (GEORGE e WILLIAMS, 2004).

3.1 Subsistema WebService

O subsistema Web Service é a camada na qual o usuário interage com o sistema (Figura **4**:A), para isso ela utiliza o *framework* Django (“Django”, 2019) em um modelo em camadas MVC (*Model-View-Controller).* Django é um framework web de código aberto (*open*-source) que é escrito em Python. Um software gerado pelo Django consiste em um projeto contendo uma coleção de componentes para gerar páginas *Web* responsivas de forma fácil e segura. Um projeto Django, contem aplicativos que são responsáveis pelos sites disponíveis ao usuário e efetivam tarefas no sistema, como por exemplo, gravar dados em um banco de dados. Assim, o Rigel consiste em um projeto Django e cada aplicativo corresponde a um *pipeline* experimental que contém uma ou mais páginas para controle do experimento. Por sua vez, estes aplicativos utilizam módulos que se comunicam com o subsistema tarefas e multiprocessamento, responsável por executar as ferramentas de biologia computacional no servidor onde o Rigel está instalado.

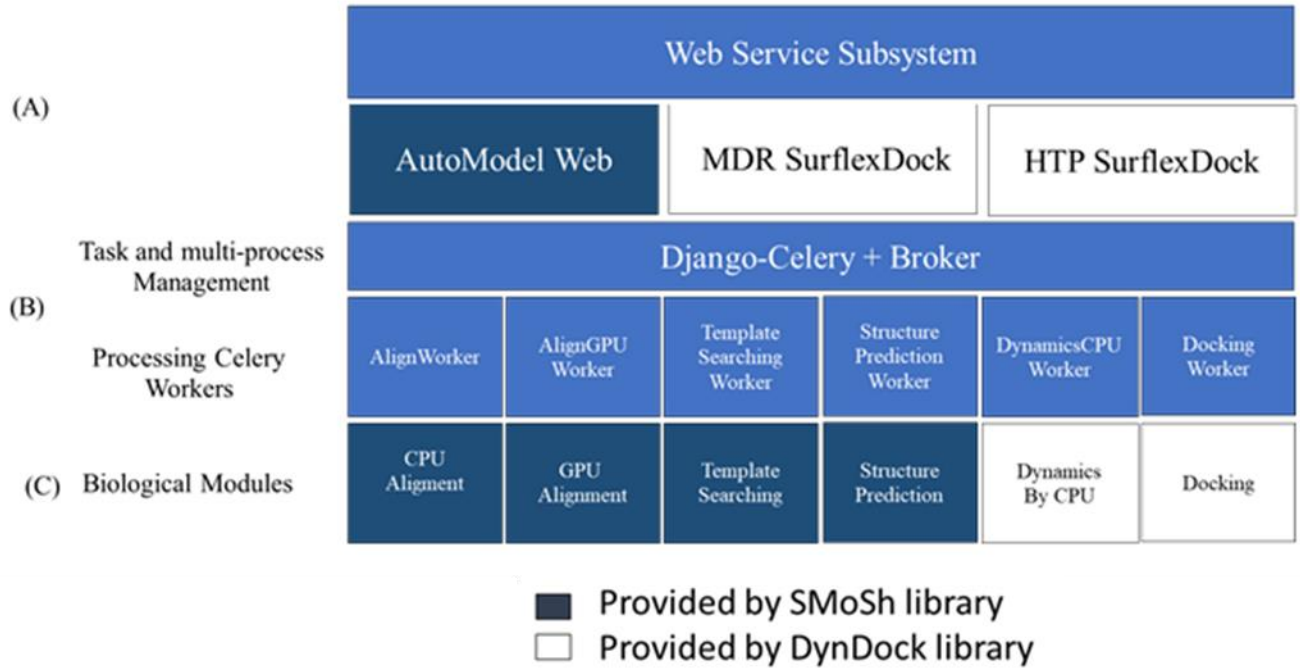


Figura 4- Arquitetura em camadas do sistema Rigel. (A) Os diferentes pipelines de experimentação biológica implementados em Rigel e cuja interface de usuário é gerada pela camada webservices. (B) A camada de escalonamento de multitarefa gerencia a fila de tarefas a ser executada e os Workers disponíveis utilizando a biblioteca Celery. (C) Camada de módulos de bioinformática utilizados pelo os pipelines de experimentação e fornecido pelas bibliotecas SMoSh e DynDock, desenvolvidas para esta plataforma.

Os *pipelines* da plataforma Rigel possuem pequenos módulos de biologia computacional (Figura **4**:C). Onde, cada módulo controla um software ou biblioteca externo, juntamente com os seus dados de entrada e saída. Na criação de cada módulo foi especificados tipos de arquivos a serem utilizados para permitir a comunicação entre as ferramentas. As conversões de arquivos foram feitas por meio da biblioteca BioPython. Para facilitar o desenvolvimento, estes módulos foram agrupados em bibliotecas temáticas: SMoSh que é utilizado pelo *pipeline* de modelagem por homologia e DynDock que é utilizado pelo por *pipelines* que utilizam softwares de dinâmica molecular e *docking*.

3.2.1 Biblioteca SMoSh

A biblioteca SMoSh (*Simple Modeling Shell*) foi desenvolvida utilizando os módulos de modelagem do AutoModel em conjunto com a API utilizada em Rigel. O AutoModel é um software cliente-servidor semiautomático de modelagem molecular por homologia (ALMEIDA FILHO et al., 2018). O objetivo desta biblioteca é prover um ambiente de desenvolvimento comum entre o AutoModel e o *pipeline* de modelagem molecular de Rigel, facilitando atualizações de ambas as

ferramentas. Desta forma as funcionalidades adicionadas em SMoSh estarão disponíveis para o AutoModel e para o *pipeline* de modelagem molecular. Como a maioria das chamadas da API SMoSh são para o software Modeller, a biblioteca foi estruturada para trabalhar com este software. No Modeller, em cada etapa é necessário criar um *script* python personalizado contendo os parâmetros necessários para a execução do programa. Esta interface de usuário embora dificulte o uso diário, facilita a criação de softwares, como o AutoModel, ou bibliotecas, como o SMoSh, baseadas nesta ferramenta, como também o seu uso em larga escala.

No SMoSh, cada etapa da modelagem é executada por meio de dois comandos (métodos do paradigma de Orientação a Objetos): o primeiro recebe configurações para a etapa a ser executada [__make_script__] e gera o *script* personalizado (Figura 5:B) e o segundo, posteriormente, executa o Modeller (Figura 5:A). A cada etapa da modelagem é criada uma pasta temporária no sistema de arquivos do sistema, onde serão armazenados os arquivos gerados pela modelagem. Estes arquivos estarão disponíveis para as outras camadas de Rigel ou o AutoModel através do comando 'get_files'.

Como o SMoSh trabalha com ferramentas externas, foi desenvolvido um sistema de captura de erros de execução do Modeller, este dispara uma exceção quando uma etapa termina de forma anormal para, por exemplo, exibir uma mensagem de erro amigável na tela do usuário (Figura 5:C). Outra característica presente na biblioteca é que o armazenamento do banco de dados do RCSB PDB (BERMAN et al., 2000) é desnecessário, desta forma, o download de proteínas-moldes é feito sob demanda utilizando o serviço *PDB service restful* presente no site do RCSB PDB. Até a versão 0.5 do AutoModel era necessário manter localmente no servidor, o banco de dados do RCSB PDB completo. Além disto, SMoSh permite executar tarefas de busca de candidatos a proteínas-molde utilizando o software Modeller, alinhamento de sequências em *CPU* e *GPU* utilizando vários algoritmos diferentes, modelagem de proteínas por homologia, download de *template*s no banco de dados do RSCB pdb, avaliação da qualidade do melhor modelo através do gráfico de score DOPE e, por final, refinamento de regiões de *loops*. Cada uma destas tarefas pode ser executada independentemente.

Especificamente na tarefa de alinhamento, SMoSh pode utilizar: (1) O software G-PAS 2.0 (FROHMBERG et al., 2012) que faz alinhamentos utilizando a *GPU* do computador e sua saída compatível com o Modeller e outros softwares utilizados no projeto. Através da linha de comando

G-PAS 2.0 permite que o usuário faça alinhamento utilizando os algoritmos Needleman-Wunsch (GOTOH, 1982) e Smith-Waterman (SMITH e WATERMAN, 1981). Para o SMoSh foi escolhido o Needleman-Wunsch por ser baseado em alinhamento global. (2) A própria ferramenta de alinhamento do Modeller que é baseada no algoritmo do software Comparer (SALI e BLUNDELL, 1990), e realiza alinhamento baseado em programação dinâmica. (3) software MUSCLE (EDGAR, 2004) que utiliza um algoritmo próprio baseado em processos estocásticos (EDGAR, 2004).

Na etapa de modelagem, SMoSh gera dez modelos, sendo o melhor modelo baseado em sua energia interna é devolvida pela API. Na tarefa de refinamento de regiões de *loops*, SMoSh depende que seja especificado o número dos resíduos que representam o início e o final da região a ser refinada.

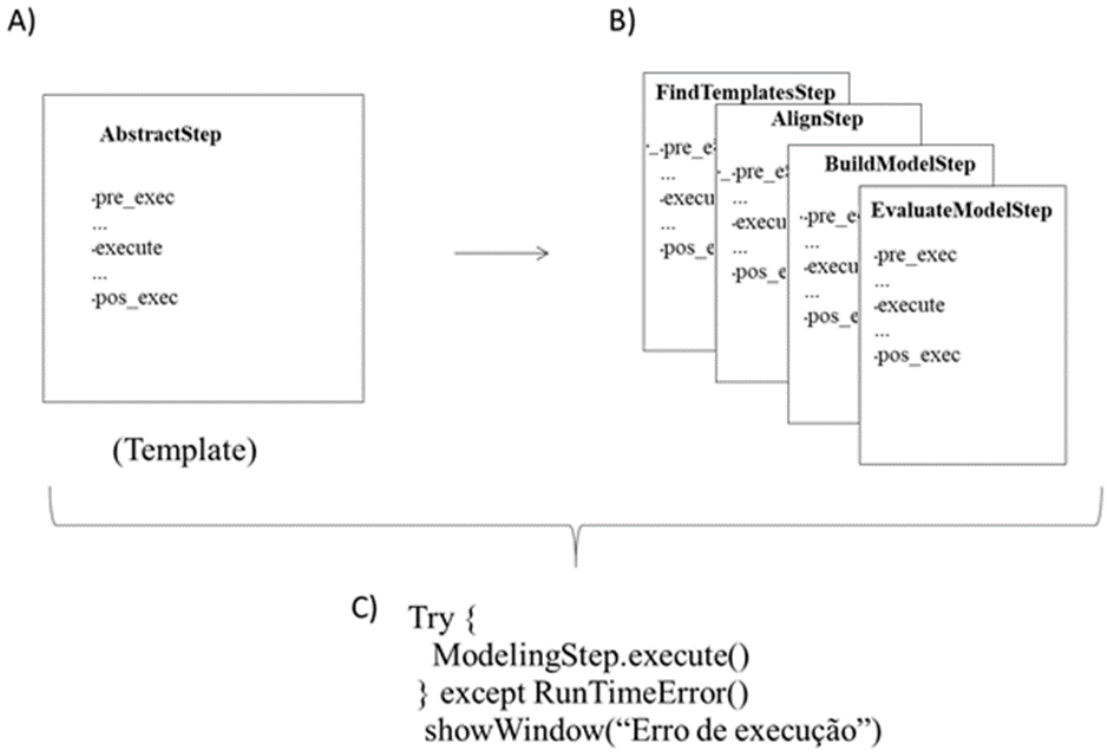


Figura 5 - Esquema da biblioteca SMoSh. (A) Classe abstrata de programação (Template), a qual possui toda a lógica de SMoSh. (B) Copias concretas de template a qual possui apenas dois comandos executáveis. (C) Exemplo de captura exceção, a qual é exibida uma janela com a mensagem "Erro de execução".

3.2.2 Biblioteca DynDock Engine

A biblioteca DynDock fornece os *pipelines* de dinâmica molecular através do software Gromacs 4.6 (ABRAHAM et al., 2015) e *Docking molecular* por meio do AutoDock 4.2 (NORGAN et al., 2011).

Nesta biblioteca, a classe DynDock.Dynamics permite que o software hospedeiro execute e gerencie as etapas da simulação molecular (Figura 1). Como todos os métodos realizam chamadas para executáveis do software externo Gromacs, um tratamento especial de erros foi elaborado que consiste em verificar o código de saída do executável (um número), caso o código de saída seja diferente de zero, é levantada a exceção RunException.

No Gromacs, antes das etapas de uso computacional intensivo é necessário preparar os arquivos a serem processados através do pré-processador grompp que também utiliza parâmetros estimados pelo usuário em um arquivo com extensão mdp. Após a execução do grompp o programa mdrun executa a etapa da dinâmica. Desta forma, na classe DynDock.Dynamics cada etapa é executada em três métodos. O primeiro comando é a preparação do arquivo. mdp através do método ‘write_XXX_mdp, onde XXX é o nome da etapa, seguido da execução do grompp através do método run_grompp, a etapa é finalizada pelo método run_mdrun.

Após a classe Dynamics ser instanciada, o experimento de simulação inicia-se com a conversão do arquivo de coordenadas da proteína para o formato Gromacs (extensão .gro) e a criação do arquivo de topologia deste mesmo arquivo através do método “generate_topology”. Neste método utilizamos o software pdb2gmx e os parâmetros do campo de força gromos53a6 (OOSTENBRINK et al., 2005). Além disso, foi utilizado no pdb2gmx o parâmetro para a obtenção da topologia mesmo faltando alguns átomos na estrutura da proteína.

A caixa de solvatação com água do tipo *Single Point Charge* (SPC) modelo 216 (TELEMAN et al., 1987) e íons Na+ e Cl- adicionados aleatoriamente em uma concentração 0.01 mol/L é preparadas através dos métodos generate_box e write_ions_mdp e são posteriormente processadas pelos métodos run_grompp e run_mdrun. Em DynDock utilizamos a caixa de solvatação do tipo triclínica para minimizar o custo computacional.

Na caixa de solvatação é realizada a minimização de energia utilizando os write_em_mdp, run_grompp e run_mdrun. Em Dynamics a minimização de energia utiliza, por padrão, o algoritmo *steep-descent* em 50000 passos ou quando força máxima chegar a 1000 KJ/mol/nm.

Com o sistema minimizado, DynDock permite fazer termalizações graduais em temperaturas especificadas como parâmetro pelo método generate_term_mdp. Esta termalizações são feitas utilizando o "*NVT ensemble"*, isto é, mantendo o número de partículas e volume constante mas sendo aplicada a temperatura no sistema através do termostato do tipo *V-rescale* (BUSSI et al., 2007). A termalização termina quando o sistema todo atingir o patamar de temperatura especificado. A próxima etapa é a estabilização da densidade do sistema, através do método generate_npt_mdp, em regime de número de partículas, volume e temperatura constante à aplicação de uma pressão constante (*ensemble NPT)* até o sistema alcançar a pressão de um atm., utilizando um barostato do tipo Parrinello-Rahman (PARRINELLO, Michele e RAHMAN, 1981).

A produção da trajetória é pré-configurada pelo método write_md_mdp, que é executada pelos métodos run_grompp e run_mdp em 2500000 passos de 2 femtosegundos totalizando 5 nanosegundos e utilizando 6 a 10 núcleos de processamento.

Após a etapa de produção da trajetória a biblioteca DynDock permite fazer o agrupamento de conformações representativas de alguma região da proteína por meio do método "clustering". Essa região compreende todos os resíduos que estão a uma distância especificada, em nanômetros, de um resíduo central também especificado como parâmetro neste método (Figura **6**). O agrupamento é feito através do algoritmo GROMOS (DAURA et al., 1999) utilizando um corte de 0.1 nm. A partir de informações do agrupamento é feito um gráfico em formato *postscript* e as estruturas da proteína dos 3 grupos (*clusters)* mais representativos pode ser obtida em formato .pdb pelo método get_pdbs_from_clusters.

Além de dinâmica, a biblioteca DynDock fornece também a possibilidade de fazer *docking molecular*. Neste caso, este é realizado pela classe Dyndock.Docking e possui como construtores (parâmetros de inicialização), o caminho da estrutura da proteína receptora (arquivo de coordenadas), caminho do ligante, o número de poses a serem geradas e a localização do espaço de busca na proteína receptora. Todas as etapas são realizadas utilizando o pacote AutoDockTools e o software AutoDock 4.2. Assim como na classe Dynamic, a classe Docking faz uma série de

chamadas a executáveis e *scripts* externos e caso alguma dessas chamadas termine com o código diferente de 0 é levantada a exceção RunException. Além disto, o *docking* é realizado através de vários métodos executados sequencialmente.

O *docking* é iniciado pelo método prepare_ligand, que verifica a estrutura ligante através do script 'prepare_ligand4.py' e este retorna o arquivo "ligand.pdbqt". Após a preparação do ligante, o receptor é preparado pelos métodos remove_heteroatoms que remove todos os heteroátomos do receptor e os armazena em um arquivo temporário e o método prepare_receptor, responsável por converter o arquivo pdb do receptor em pdbqt com hidrogênios adicionados. Com o receptor em formato pdbqt são adicionados os heteroátomos, removidos anteriormente, pelo método add_hetatms_pdbqt. Estes heteroátomos, quando são íons (Zn, Cl, Na e Mg), são convertidos em formato pdbqt com cargas +1.2, -0.8, +0.8 e +1.2 respectivamente. O passo seguinte é a preparação do arquivo GPF (*Grid Parameter File*) pelo método prepare_GPF. O arquivo GPF contém vários parâmetros utilizados pelo executável autogrid4, entre eles está a localização da grade tridimensional, onde serão calculados os *dockings* no receptor. No AutoDock, o centro da grade tridimensional é definido por um ponto do espaço cartesiano e um número de pontos de grid para cada dimensão xyz. Este número de pontos quando multiplicado pelo parâmetro *spacing*, cujo valor padrão 0.375Å, que corresponde ao diâmetro da grade tridimensional nas dimensões X, Y e Z em angstrons. No entanto, para manter a compatibilidade com a classe Dynamic, no cálculo da grade tridimensional, o método prepare_GPF utiliza como parâmetro um resíduo, onde seu carbono alfa será o centro da grade, e a distância do carbono alfa até a borda da grade tridimensional, definido em nanômetro (Figura **6**).

O autogrid é executado, método run_autogrid, para a criação dos mapas de afinidade para cada tipo de átomo. A próxima etapa é a preparação do arquivo DPF (*Docking Parameter File*) pelo método prepare_dpf. O docking é preparado para utilizar o algoritmo genético Lamarckiano com 2500000 avaliações de energia para cada pose. O único parâmetro que o método prepare_dpf recebe é o número de poses, sendo que este valor deve ser um valor entre 10 e 300. As repetições são geradas com a execução do software autodock4 pelo método run_autodock.

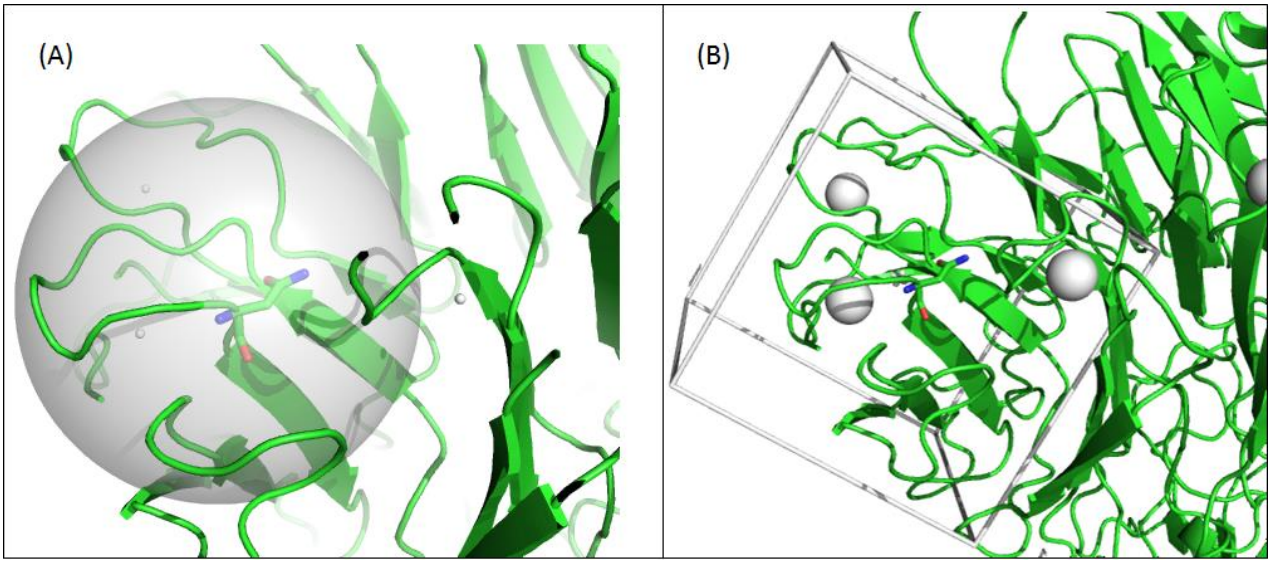


Figura 6 - Definição do espaço cartesiano para o agrupamento de conformações mais representativas e da grade tridimensional do docking. (A) Espaço utilizado para o algoritmo de clustering gerado pelo método Dynamic.clustering. (B) Grade tridimensional gerado pelo método Docking.prepare_GPF. Ambos os métodos possuem como parâmetro um resíduo, onde o carbono alfa será utilizado como centro e uma distância do centro até o limite do espaço de busca. Desta forma, é possível sobrepor o espaço de clustering com o cubo que define o espaço de cálculo do docking. Esta abordagem é utilizada no MDR SurFlexDock e no HTP SurFlexDock para manter a continuidade do espaço cartesiano.

3.3 Camada de escalonamento de multitarefa.

O subsistema de multiprocessamento foi desenvolvido utilizando a biblioteca Django-Celery (*Queue, Celery: Distributed Task*) e o software Redis[1] (Figura 4:B), estes fornecem um gerenciador de filas de processamento distribuído. Django-Celery permite, também, a execução de tarefas bloqueantes concorrentemente sem congelar o sistema. Além disso, permite que estas sejam executada localmente e em ambientes de rede multiprocessados.

A cada novo experimento, o subsistema de multiprocessamento recebe requisições dos aplicativos de Rigel, isto é, dos módulos de experimentação e delega a tarefa a um *Worker* (Figura 7) se este estiver disponível ou coloca em uma fila de espera (Figura 7:A). Estes *Workers* são pequenos serviços que aguardam e processam estas requisições (Figura 7:B-C), após o processamento é realizado o processo inverso e, assim, o módulo recebe o resultado que pode ser enviado a outro módulo ou para a página do usuário.

A cada novo módulo adicionado a um *pipeline* do sistema é necessário especificar uma tarefa no arquivo ‘tasks.py’ existente dentro do diretório do aplicativo e desenvolver a página de forma a

[1] https://redis.io/

chamar esta tarefa, todo o funcionamento do gerenciador de filas de processamento é transparente para o usuário e o desenvolvedor sendo que o tamanho da fila e o número de *workers* podem ser configurados através do arquivo de configurações do projeto.

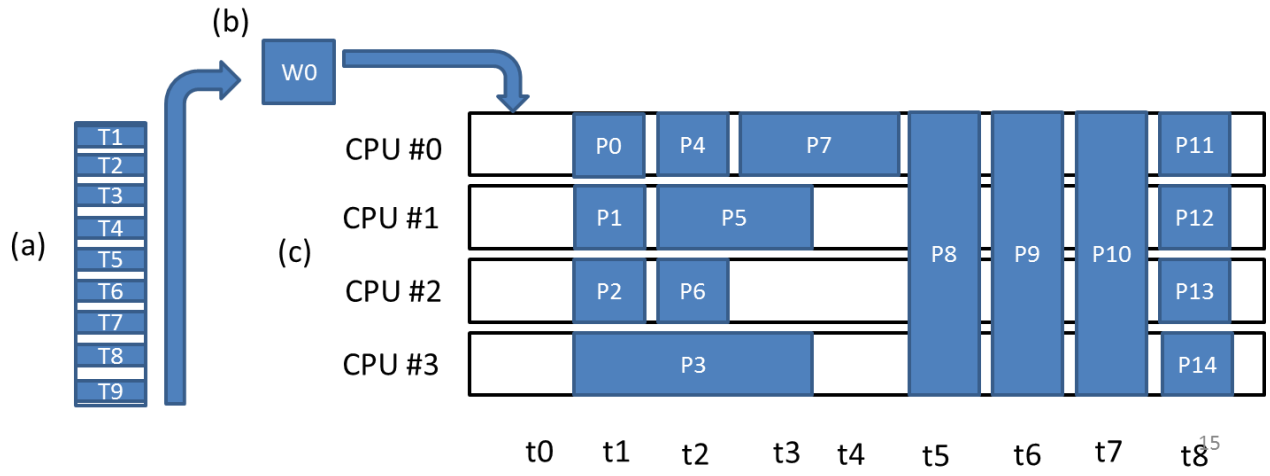


Figura 7 - Esquema do funcionamento da camada de escalonamento de multitarefa. Rigel, utiliza uma fila de tarefas (A) que será processada pelos workers presentes no sistema (b). Os Workers por sua vez atribuem as tarefas nas CPUS disponíveis como processos.

3.4 Hierarquia de arquivos

Como Rigel foi desenvolvido utilizando o *framework* Django, ele possui a estrutura de arquivos e pastas semelhante a projetos deste *framework*. Desta forma, as configurações da ferramenta são feitas através do arquivo Rigel/settings.py. Neste arquivo os parâmetros mais importantes são o ALLOWED_HOSTS que contém o IP da máquina que irá oferecer os serviços de Rigel; Parâmetros EMAIL, utilizado para enviar e-mails relacionados a cada experimento, como por exemplo, quando um experimento termina é enviado um e-mail para o usuário, e o parâmetro Q_CLUSTER, que permite configurar a camada de escalonamento de multitarefa, permitindo selecionar quantos *workers* estarão disponíveis para os experimento e o tamanho máximo da fila de experimentação. No script rigel/hooks.py contém os comandos que serão processados pelos nos *workers*. Estes comandos utilizam as bibliotecas desenvolvidas para este projeto para executar cada *pipeline*. O script rigel/model.py contém a assinatura do banco de dados do projeto. Neste, script a classe *process* é no armazenamento de informações sobre cada experimento. Os scripts rigel/views.py e rigel/urls.py armazenam os detalhes de acesso de cada página presente no sistema.

Como todo projeto Django, o arquivo manage.py permite iniciar o servidor web e banco de dados. No caso de Rigel, este comando também é utilizado para iniciar a Camada de escalonamento de multitarefa através do parâmetro qcluster. Este comando deve ser executado toda vez que iniciar o servidor Rigel manualmente: <python manage.py qcluster>

3.5. *Pipelines* implementados em Rigel.

3.5.1. *Pipeline* para modelagem molecular – AutoModel Web.

A modelagem por homologia é uma metodologia que consiste em resolver a estrutura tridimensional de proteínas computacionalmente através de sua sequência de aminoácidos, utilizando, como molde, a estrutura tridimensional de uma proteína homologa previamente determinada (KRIEGER et al., 2003).

O processo de modelagem ocorre em uma série de passos sequenciais (Figura 1) que normalmente incluem (i) seleção de uma ou mais proteínas-molde (*template*), (ii) alinhamento de sequencias para a determinação das regiões estruturalmente conservadas e restrições espaciais, (iii) construção do modelo tridimensional por meio da satisfação de restrições tridimensional, (iv) validação do modelo gerado através de análise de parâmetros estereoquímicos ou análise estatística (SANTOS FILHO, 2003). Neste contexto, desenvolvemos um *pipeline* de modelagem por homologia por meio de um aplicativo em Rigel chamando de AutoModel Web, este *pipeline* utiliza módulos de uma biblioteca chamada SMoSh para criar um sistema de interação média com o usuário (semiautomático) (Figura 8).

No *pipeline* desenvolvido, devido a sua característica semi-automática, cada etapa da modelagem utiliza uma ou mais páginas *web* para a interação do usuário: (i) o usuário carrega a sequência de resíduos da proteína que se deseja modelar em formato FASTA, seleciona uma proteína-molde de uma lista gerada pelo *pipeline* (Figura 8), da proteína-molde o usuário escolhe a cadeia e heteroátomos que melhor representam a proteína a ser modelada, posteriormente o alinhamento entre as sequência da proteína molde e da proteína a ser é realizada seguido da modelagem pelo Modeller. A validação é feita utilizando a função de pontuação DOPE (*Discrete Optimized Protein Energy*) (SHEN, 2006) gerado pelo Modeller e também através de gráficos do tipo Ramachandran.

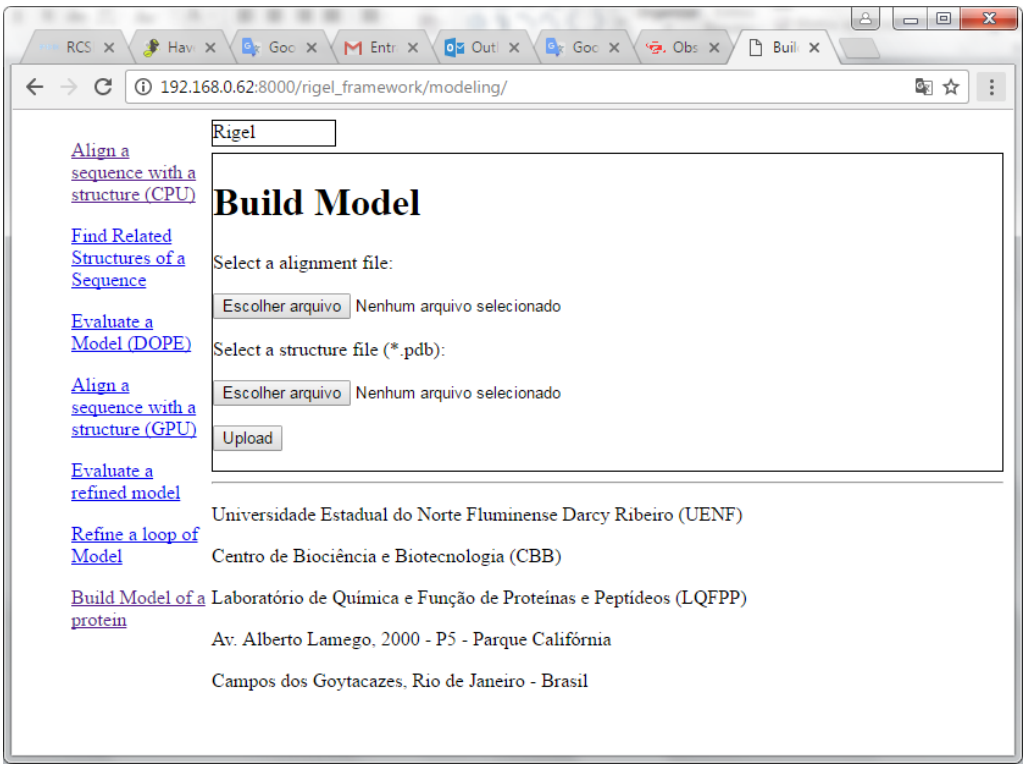


Figura 8 - Interface Web gerado por Rigel para a modelagem de proteínas. A esquerda da página está o acesso para as ferramentas no pipeline individualmente.

3.5.2 *Pipelines* de análise de sitio ativo (MDR SurflexDock) e análise de interação de inibidores (HTP SurflexDock)

Os *pipelines* análise de sitio ativo e análise de interação de inibidores foram desenvolvidos utilizando a técnica de *ensemble docking*. Neste sentindo, o *docking molecular* é uma metodologia que tenta encontrar orientações de um ligante em relação a um receptor proteico de forma a propiciar uma boa energia de ligação ao formar um complexo estável (MIH\UA\CSAN, 2012). No entanto, esta é uma técnica que pode possuir um alto custo computacional, e desta forma são feitas abstrações de fenômenos físicos que consequentemente impactam negativamente no modelo do complexo gerado (ANTUNES et al., 2015). Como por exemplo, a maioria das ferramentas de docking desconsidera a flexibilidade do receptor, no máximo permitindo movimentações de cadeias laterais localizadas no sitio catalítico (SINKO et al., 2013). Neste contexto, o *ensemble docking* permite a simulação da flexibilidade do receptor através do uso de um pacote de conformações mais representativas provenientes de cristalografia ou simulação molecular (AMARO et al., 2018). Assim, é feito *dockings* dos ligantes com cada uma das conformações que compõe o ensemble. A grande vantagem do *ensemble docking* é que não é necessária a modificação dos algoritmos de

dockings já existentes. Além disso, entre as técnicas de docking com flexibilidade do receptor, o *ensemble docking* é uma das que possui o custo computacional mais baixo (ANTUNES et al., 2015).

Em Rigel, a implementação do *ensemble docking* em ambos *pipelines* foi possível através da biblioteca DynDock, responsável por controlar as ferramentas de *docking* e dinâmica utilizadas pelo o sistema. Resumidamente os *pipelines* obtém o *ensemble* composto de três conformações mais representativas de uma dinâmica molecular simulando o ambiente fisiológico. Com estas conformações são realizados os *docking* com os ligantes.

No entanto, embora, ambos *pipelines* utilizem o *ensemble docking*, eles realizam diferentes tipos de experimentos utilizando o *docking molecular*: O *pipeline* de análise do sítio ativo foi implementado com o nome de MDR SurFlexDock tem como objetivo, tão somente, oferecer poses (experimentos de *docking*) de poucos ligantes em diferentes conformações do receptor. Além disso, o MDR SurFlexDock permite que o usuário faça uma análise estatística das melhores poses de cada conformação do *ensemble*. Por outro lado, o *pipeline* análise de interações de inibidores foi implementado com o nome de HTP SurFlexDock e permite que o usuário faça experimentos de *virtual screening* utilizando o *ensemble docking*. *Virtual screening* é uma técnica de *High Throughput Screening virtual* que permite selecionar os melhores ligantes, normalmente inibidores, de um determinado alvo molecular. No HTP SurFlexDock utilizamos o *ensemble docking* para evitar falsos negativos decorrentes da utilização do *docking* rígido em receptores otimizados para certos tipos de ligantes. Ambas as ferramentas foram desenvolvidas de forma a funcionar automaticamente, não requerendo a participação ativa do usuário após o início do experimento. Os detalhes sobre o funcionamento do MDR SurFlexDock e HTP SurFlexDock nos capítulos 4 e 5.

Capitulo 4: MDR SurFlexDock: a semi-automatic webserver for discrete receptor-ensemble docking

Com a presente plataforma, foi desenvolvido o MDR SurflexDock, um servidor semi-automático para análises de interação proteína-ligante. O documento científico foi submetido à "*Computational and Structural Biotechnology Journal*".

MDR SurFlexDock: a semi-automatic webserver for discrete receptor-ensemble docking

João Luiz de Almeida Filho[1]*, Jorge Hernandez Fernandez[1]

[1]BioComp, LQFPP, Centro de Biociência e Biotecnologia (CBB), Universidade Estadual do Norte Fluminense (UENF), Campos dos Goytacazes, Rio de Janeiro, Brazil

*To whom correspondence should be addressed.

Abstract

Background: In current computational biology, docking is one of the most popular tools used to find the best fit of one ligand relative to its molecular receptor in forming a complex. However, a good representation of the fit between the receptor and ligand is not always possible, since most of the tools do not take into account the flexibility of the receptor due to computational cost. As a result, the conformational changes caused by the induced fit are ignored in exploratory docking experiments. In this context and to improve the predictive capacity of docking, a good strategy is to simulate the flexibility of the receptor with the use of key conformations that can be obtained by mixing crystallography and computer simulations, a technique known as ensemble docking. Here, we present MDR SurFlexDock, a web tool that improves the docking experiments by computing a discrete, but representative, ensemble of contact surfaces of the receptor through clustering of molecular simulation trajectories in order to simulate the intrinsic flexibility of the ligand-contacting surface.

Results: The pipeline used in MDR SurFlexDock calculates multiple conformations of the receptor surface through: *(i)* molecular simulations in an explicit solvent, followed by gradual thermalization, *(ii)* selection of the three most representative conformations by clusterization of receptor contact surfaces, *(iii)* consecutive rigid docking of each sample for up to ten ligands. The MDR SurFlexDock pipeline is freely available as a web service at http://biocomp.uenf.br:18864.

Conclusions: MDR SurFlexDock has low a computational cost, using mainly GROMACS and AutoDock. The pipeline is highly scalable and was designed for low computational cost. The results of the interaction of each receptor-compound complex are presented in a concise tabular format to allow rapid analysis of compounds when classifying them by inhibition constant (Ki). MDR SurFlexDock can be valuable in cases of docking for new receptors obtained by homology modelling, in extensive analysis of different chemotypes on proteins with low structural information and for fast characterization of binding capacities on contact surfaces with poor structural information or only optimized for a specific ligand.

Background

Docking is one of most popular techniques in computer-aided drug design [1, 2]. This approach can be summarized as the discovery of the compounds (or ligands) that best fit into a specific part of the target protein (the receptor) [3]. As part of the experiment, structural information and interaction energy of the complex are important factors regarding the final choice for the 'best interaction model,' as most of the docking programs provide several mathematical solutions for the same problem [2], and the best results, in most cases, is the final user choice.

As proteins are flexible and dynamic macromolecules that continually alternate their surface conformations, the accurate representation of the protein surface facing the ligands is currently one of the most important challenges in docking experiments [4]. At the same time, simulating the receptor flexibility increases the computational cost of the experiment. To overcome the problem of protein flexibility several strategies are used to better sample the conformational space of the receptor [2]. Moreover, the main protein structural information of the receptor contact surface is predefined by complexes that are resolved by X-ray crystallography and stored in databases (PDB format at www.rcsb.org). As a result, protein surfaces in complexes with some ligand are typically biased to perform better in some specific contacts, and docking experiments with different chemotypes becomes a challenging task [4].

In this context, we are developing MDR SurFlexDock (Molecular Dynamics-based Receptor Surface Flexibility for Docking), a web server that allows analysis of the interaction of small protein-ligand complexes through enhanced structural sampling of receptor surfaces with low computational costs. The pipeline uses of discrete conformational search in solvated receptors, performing molecular simulations after gradual thermalization and increasing the possibilities to

better represent the physiological environment in several protein-ligand docking experiments. The receptor surface is submitted to clustering analysis over the simulation trajectories with a 1.0 Å cut-off. Finally, the three most representative conformations are used in consecutive docking experiments. At the end of the experiment, MDR SurFlexDock presents the results through a concise table of protein-ligand complexes, which are classified by the inhibition constant (Ki), as well as discrete graphical information on receptor clustering and statistical representation of ten better dockings for each compound.

Implementation

Web server

The main page of MDR SurFlexDock (Figure 1:A) provides a simple and convenient way to specify the target receptor in pdb format, up to ten ligands in the native AutoDock format (.pbdqt), an email for user notifications and simple docking parameters. The server will redirect the user to an experiment control page and, depending on the stage of experimentation, the control page will be fed graphical information of the clustering, a boxplot plot of the top ten poses of each ligand in each of the docking experiments and a list of the calculated poses by MDR SurFlexDock (Figure 1:B). In addition, structural information of each complex can be reached through a Glmol HTML 5 plugin [5]. Finally, MDR SurFlexDock provides a link to download all the experimental results to enable the user to handle them locally. The results remain available for local download for over 15 days.

The MDR SurFlexDock server was developed using the Django web framework and BioPython plug-in. In addition, Django is compatible with several Python libraries that were used in the development of this web server. When a user starts an experiment, our server places experimentation in a queue managed by the Django-celery plug-in, redirects the user to an experiment specific page and e-mails the link to this page.

Pipeline

Due to the magnitude of the conformational space in proteins, taking into account the flexibility of the receptor in docking experiments constitutes a great challenge[2]. This becomes especially complex in the case of high-throughput screening (HTS). The docking of hundreds of compounds in a flexible receptor requires, even today, computational power not always available for newly interested users. One strategy is to use some conformations of the receptor to do the docking, which is known as ensemble docking [6]. MDR SurFlexDock uses a simple approach to ensemble docking in a two-stage pipeline: (*i*) initially constructing an ensemble containing the three most representative conformations of the receptor active site on the trajectory of 5-ns molecular simulation using the GROMACS 4.6 [7] with a GROMOS96 53a6 force field [8], followed by (*ii*) molecular docking of each compound with the ensemble of representative receptor structures using AutoDock 4.2 [9] and ADT 1.6 scripts [10](Figure 1).

Molecular simulation starts with the solvation of the receptor with a SPC216 water model [11] in a triclinic box, neutralized by randomly added Cl- and Na+ ions. The system energy is minimized in 50,000 steps, or reach up to 1,000 KJ/mol/nm, using the steep descent (SD) algorithm [12]. Receptor thermalization is done by heating the system temperature from 285 K to 300 K in the NVT ensemble. Subsequently, a 5-ns molecular simulation optimizes the interaction of the solvent with the receptor, thus exploring the conformational space of the receptor. Receptor samples are obtained by clustering the conformations of the user-defined active site using 0.1 nm RMSD cut-off and Gromos algorithm [13]. In this step, the active site is composed of all the residues that are at a certain (user-defined) distance in nanometres from a central residue, also defined by the user. The central structures of the three most representative clusters over the simulation trajectory are

converted into pdb format to compose the ensemble (Figure 2). As a control, docking of the original receptor structure delivered by the user is also performed with the ligands.

The docking procedure starts with the conversion of the receptor file into the native AutoDock format (.pdbqt) through ADT's 'prepare_receptor.py' script. Subsequently, each compounds is verified by the script 'prepare_ligand4.py.' The Grid map is prepared through 'prepare_gpf4.py,' and the grid box is defined as a cube containing the residues of the active site that are a certain distance from the central residue (in nanometres). Both parameters are pre-defined by the user. This script generates the file 'grid.gpf,' which is executed by the autogrid4.2 software [9]. Subsequently, the docking experiment is configured by the script 'prepare_dpf4.py' to create the amount of poses defined by the user (10, 20 or 30) and 2.5 x 10^6 energy inferences for every docking experiment, using the Lamarckian genetic algorithm (LGA). Finally, the poses are calculated using the autodock4.2 software. During each docking, the docking log-file (.dlg) is parsed by the BioPython library to extract the inhibition constant (Ki) that will be used in the experimental output (figure 1:C). The MDR SurFlexDock pipeline is freely available as a web service at http://biocomp.uenf.br:18864. A general flowchart of the experimental pipeline is represented in Figure 2.

Results and Discussion

Here we describe MDR SurFlexDock, a virtual screening tool that addresses the problem of receptor molecular sampling through ensemble docking. One of the main goals of MDR SurFlexDock is to be a simple tool to use for exploring the initial stages of protein-ligand studies, at which point no experimental and structural information is available, and the need for an initial hypothesis of a protein-ligand structural interaction is mandatory. When implemented, even for homology modelling of the receptor, minimal information about the active site and a set of ligands in .pdbqt

format is sufficient for starting experiments. In its actual developmental stage, MDR SurFlexDock allows dockings with up to ten ligands per experiment.

The construction of extensive ensembles using molecular dynamics may be an important limitation of this approach, due to the high computational cost [6]; however, some studies indicate that positive results can be achieved if the simulation is limited to some residues [14]. In the MDR SurFlexDock pipeline, this process is performed by clustering only the user-defined contact surface under rapid molecular simulation across the receptor. In this regard, our tool constructs the most representative ensemble, with discrete structural distances between the conformations and specific differences in the ligand-interacting surface, although it does not identify large movements (such as hinged effects) due to limitations in molecular simulation time. We aimed to maintain a low computational cost, since as the number of ensemble conformations increases, the amount of dockings and experimental time will increases at the ratio of geometric progression. In addition, a greater number of ensemble conformations in a set increases the likelihood of an anti-cooperative effect [6]. In the case of MDR SurFlexDock, this would result in a waste of computer time due to docking calculations for these conformations and an unnecessary increase in complexity of experimental results.

Perspectives

Our group herein describes a simple and semiautomatic web implementation for analysis of the interaction of small protein-ligand complexes through enhanced structural sampling of the receptor surface. In principle, this technique can be used in MDR experiments using one of the several ligand databases available (http://autodock.scripps.edu/resources/databases), increasing the total number of positive results and reducing false negatives. We also foresee the integration of a better parameterization tool for ligands and the possible implementation of alternative scoring functions for docking experiments in the pipeline in a future release of MDR SurFlexDock.

Funding

This work has been supported by the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) doctoral grant [141917/2015-6] for J.L.A.F. and ProAP-CAPES support from the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), which are gratefully acknowledged.

Conflict of Interest

The authors declared no conflict of interest.

Availability and requirements

Project name: MDR SurFlexDock

Project home page: http://biocomp.uenf.br:18864

Operating system(s): Platform independent

Programming language: Python

Other requirements: Mozilla Firefox 6.0, Internet Explorer 11, Google Chrome 40

License: GNU GPL

Any restrictions to use by non-academics: licence needed.

References

1. Trott O, Olson AJ. AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. J Comput Chem. 2010;31:455–61.

2. Pagadala NS, Syed K, Tuszynski J. Software for molecular docking: a review. Biophys Rev. 2017;9:91–102.

3. Ouzounis CA. Rise and demise of bioinformatics? Promise and progress. PLoS Comput Biol. 2012;8:e1002487.

4. Guedes IA, de Magalhães CS, Dardenne LE. Receptor--ligand molecular docking. Biophys Rev. 2014;6:75–87.

5. Nakane T. GLmol-Molecular Viewer on WebGL/Javascript, Version 0.47. 2014.

6. Antunes DA, Devaurs D, Kavraki LE. Understanding the challenges of protein flexibility in drug design. Expert Opin Drug Discov. 2015;10:1301–13. doi:10.1517/17460441.2015.1094458.

7. Abraham MJ, Murtola T, Schulz R, Páll S, Smith JC, Hess B, et al. GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. SoftwareX. 2015;1:19–25.

8. Oostenbrink C, Soares TA, der Vegt NFA, Van Gunsteren WF. Validation of the 53A6 GROMOS force field. Eur Biophys J. 2005;34:273–84.

9. Norgan AP, Coffman PK, Kocher J-PA, Katzmann DJ, Sosa CP. Multilevel parallelization of AutoDock 4.2. J Cheminform. 2011;3:12.

10. Morris GM, Huey R, Lindstrom W, Sanner MF, Belew RK, Goodsell DS, et al. AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility. J Comput Chem. 2009;30:2785–91.

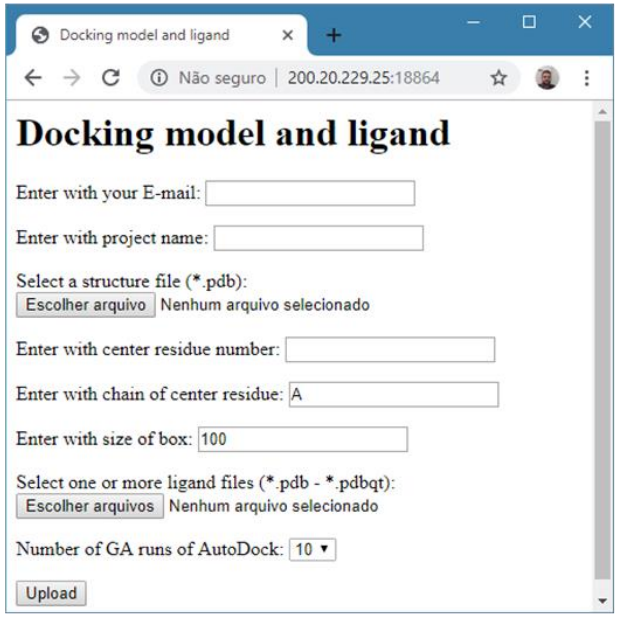
11. Teleman O, Jönsson B, Engström S. A molecular dynamics simulation of a water model with intramolecular degrees of freedom. Mol Phys. 1987;60:193–203.

12. Fletcher R. Practical methods of optimization. John Wiley & Sons; 2013.

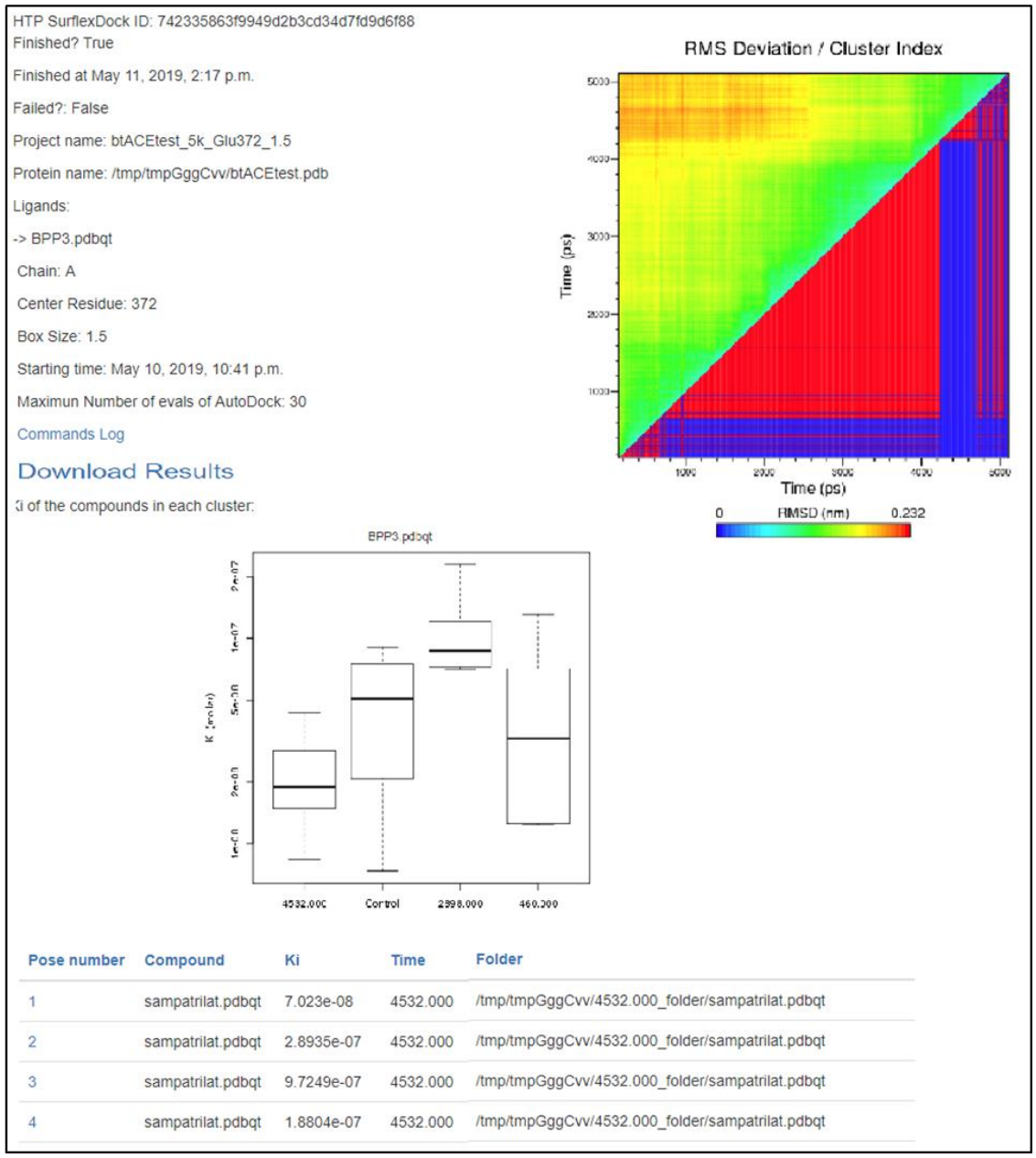
13. Daura X, Gademann K, Jaun B, Seebach D, Van Gunsteren WF, Mark AE. Peptide folding: when simulation meets experiment. Angew Chemie Int Ed. 1999;38:236–40.

14. Armen RS, Chen J, Brooks III CL. An evaluation of explicit receptor flexibility in molecular docking using molecular dynamics and torsion angle molecular dynamics. J Chem Theory Comput. 2009;5:2909–23.

(A)



(B)



(C)

Figure 1: Screenshots of MDR SurFlexDock. (A) Main page. (B) Results page containing information about the catalytic site clustering, inhibition constant and poses found. In HTP SurflexDock the results page is fed during the experiment. (C) Cubic box defined by the user for the active site used during the experiment.

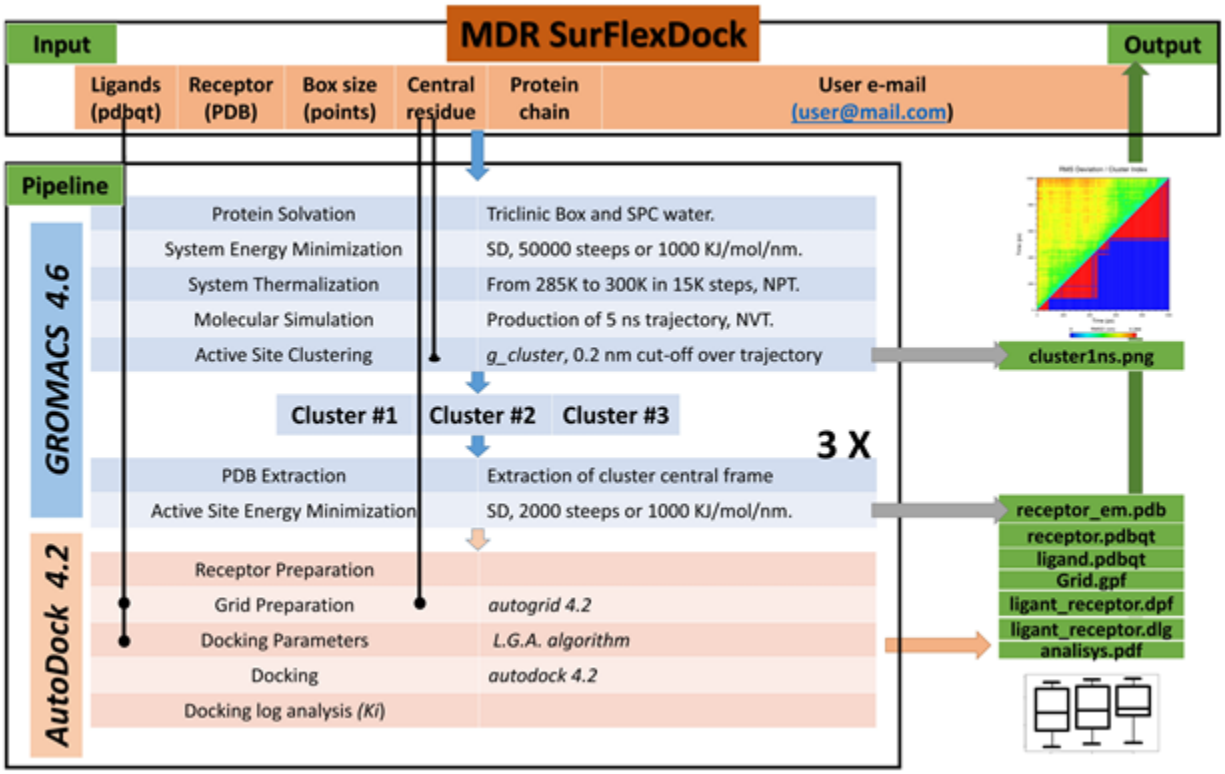


Figure 2: MDR SurFlexDock pipeline. The user inputs a receptor protein in pdb format, with one residue representing the centre of the active site, and up to ten ligands of interest in pdbqt format. The molecular simulations and clustering over trajectory is executed by GROMACS 4.6, and the docking calculations are performed using AutoDock 4.2. Each experiment has its own experimental workspace containing information about the submission, progress and a listing of the calculated poses classified by Ki constant.

SUPPLEMENTAL MATERIAL S1

Using MDR SurFlexDock to increase positive results in SH3-peptide interactions

To demonstrate the positive impact of using the MDR SurFlexDock pipeline on challenging docking experiments in general, the human SH3 domain of vinculin, an adaptor protein that regulates focal adhesion through its interaction with other adhesion proteins (Alexandrescu , 2001), was used as a study case.

SH3 domains belongs to a group of peptide recognition domains (PRDs) which bind high cross-reactivity short linear motifs in other proteins (Pin et al., 2014). Although early studies determined a consensus motif of PxxP as the minimal consensus target for SH3 domains, this consensus was enlarged to a +xxPxxP (class I) or PxxPxx+ (class II), according to the position of a positively charged peptide residue interacting with a negative cleft at the peptide binding surface (Feng et al., 1994). Nowadays, is a common opinion that the cross-reactivity exists among SH3 domains of different subclasses, and the arbitrary classification of SH3 domains in terms of observed binding motifs is not stringent and may not be applicable for all cases (Pin et al., 2014). At this moment, vast structural information on SH3-peptide interactions can be used to define the complex and variable surface of the specificity zone, although a consensus must not be defined (Skazela et al., 2014).

The Experiment

The structure of the human SH3 domain of vinculin in complex with ELAPPKPPLPE peptide from the 4ln2 pdb (Zhao et al., 2014, Supplementary Figure S1:A) was used for initial information on the receptor (SH3 domain) and the starting ligand structure (PPKPPI peptide was used as positive control for ‘re-docking experiment’). As docking experimental results are strongly determined by the contacting surface shape of the receptor (Rentzsr et al., 2014), we considered the initial surface of peptide contact in the SH3 domain as a ‘set structure’ for PP+PPx ligand type. In order to test the impact of the enhanced structural sampling of the receptor surface technique, as implemented in MDR SurFlexDock, on this SH3 contacting surface, a full set of seven ligands were used in our experiment (Supplementary Table S1).

Supplementary Table S1: Peptide ligands used in the experiment.

Peptide ligand	Rational of the experimental ligand.
P-P-K-P-P-I	Positive control, re-docking experiment.
A-P-F-P	Negative control, small non-PxxP peptide.
A-P-S-N-P-A	Hydrophilic residues at the centre of the contacting surface.
A-P-G-P-P-A	'Class I' type ligand without charged ARG residue.
A-P-F-P-P-I	Big hydrophobic residue at the centre of the contacting surface.
A-P-R-I-P-F	'Class I' type ligand with charged ARG residue.
A-P-E-I-P-A	Negatively charged residue at the centre of the contacting surface.

The SH3 domain was used in 4ln2 pdb (Zhao et al., 2014), and peptide ligands were modelled in PyMol (DeLano, 2002) and prepared as pdbqt ligands, ready for docking in ADT (Morris et al., 2009).

Default parameters in the MDR SurFlexDock pipeline uses a GROMOS force field (Oostenbrink et al., 2005) and SPC water model for structure solvation and slow thermalization of the receptor (SH3 domain) in 15 K steps, using the GROMACS 4.6 package (Abraham et al., 2015). After equilibration on the NVT ensemble, a production molecular simulation on the NPT ensemble for 5-ns is produced. We defined the centre of the receptor contacting surface as the Trp98 of the SH3 domain, and all residues inside the box that were defined in 80 points (AutoDock 4.2 defines a point as 0.38 Å) are considered on this surface. A consequent cluster analysis of the surface structure in the 5-ns trajectory with a 0.2 nm cut-off (Supplementary Figure S1:B) was used to define the three most representative structural clusters in this trajectory. The representative structure called 'Cluster 1' was defied at the 178 ps of this simulation, and those at 2,398 ps and 3,446 ps were the representative structures of 'Cluster 2' and 'Cluster 3,' respectively. At this point, the enhanced sampling of the receptor surface is defined as all of these three structures, and they used in the docking experiment with all seven ligands in the second part of the experiment.

In docking experiments, LGA parameters and 2.5 x 10^6 inferences of energy are used, with the user defining the quantity of docking experiments for each ligand between 10 and 30. At the end

of the experiment, a box plot of every ‘best 10’ result for each ligand is presented to the user in an easy-to-compare graphical interface (Supplementary Figure S1:C and D).

Final results are presented to the user as a zip file containing the full experiment, separated by clusters and ligands. For general analysis, the result of the clustering analysis over the trajectory (Supplementary Figure S1:B) and the graphical boxplot of the ‘best 10’ calculated Ki results (Supplementary Figure S1:C and D) are provided. For specific and detailed analysis of every docking result, all docking logs with the respective receptors are provided. For this analysis, we recommend the ADT (Morris et al., 2009) or PyMOL (DeLano, 2002) programs.

Results and discussion

In our specific SH3-peptide experiment, a quick and dirty analysis is sufficient to determine a Ki of 1 x 10^4 to 1 x 10^6 kJ/mol in magnitude as insufficient or ‘bad’ ligand capacities for presented peptides in the experiment. At the same time, a calculated Ki of 1 x 10^9 kJ/mol will be considered a promising results.

As ‘negative control,’ the APFP peptide displays interactions with a calculated Ki of 1 x 10^5 kJ/mol, defining this magnitude as insufficient for SH3-peptide interactions. In the same group, we consider APFPPI, APGPPA and APEIPA peptides. For all ligands, none of the three generated surfaces (Cluster 1, Cluster 2 or Cluster 3) obtained good interaction energy, with a calculated interaction in the magnitude of 1 x 10^5 kJ/mol (Supplementary Figure S1:D) and interactions that were away from the SH3 contacting surface defined in the 4ln2 pdb (Supplementary Figure S1:A). The few results in the magnitude of 1x 10^9 kJ/mol represents ligands interacting far away from determined interaction surface and were considered false-positives.

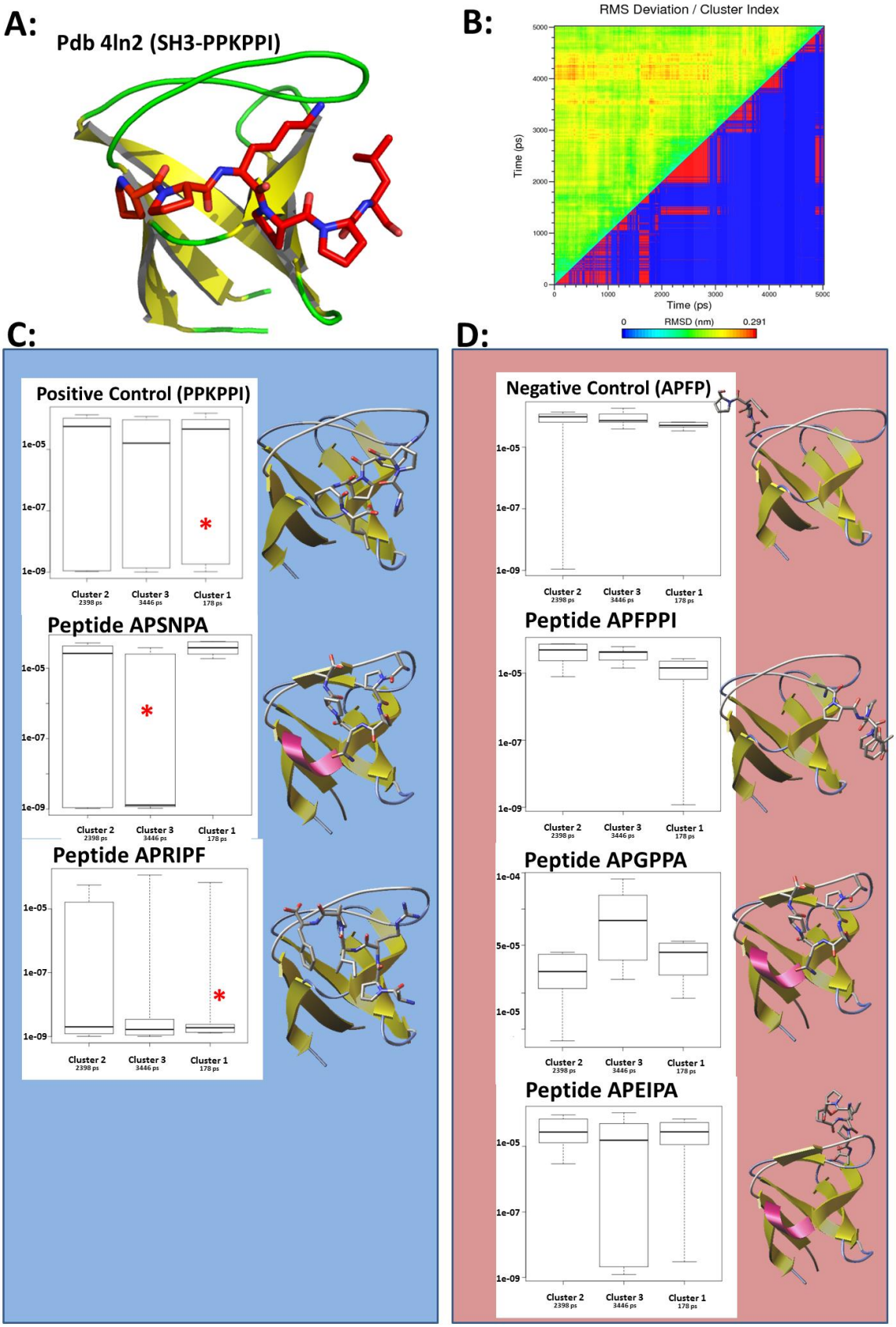
On the other hand, the PPKPPI peptide, which was considered a positive control,’ interacted in the contacting surface as 4ln2 pdb and shows a big quantity of docking results in the magnitude of 1 x 10-9 kJ/mol in the first calculated cluster (cluster 1) at 127 ps of simulations. Peptide APRIPF also shows good results in clusters 1 and 3, and is thus a promising ligand in our opinion.

The most interesting results in this experiment are those obtained for the APSNPA peptide. With a calculated Ki in the magnitude of 1 x 10^5 kJ/mol in the first cluster and the most similar to the original structure resolved in crystallography, it significantly improved its interaction magnitude when challenged with the contacting surface obtained in cluster 3, after 3,450 ps of simulation in

explicit solvent at 300 K temperature. Both the solvation and thermalization processes defined the peptide contacting surface in different shapes, enabling significantly better results in docking experiments and reaching results in the order of 1 x 10^9 kJ/mol, positioning the peptide directly in the classical peptide contacting surface of SH3 domain and defining APSNPA as a promising ligand.

Since peptide docking is a simulation experiment (Saksela & Permi, 2012; Rentzsch & Renard, 2015) and SH3 domains is not stringent in terms of observed binding motifs, but still a system of interest (Pin et al., 2014), this experiment, as proof of concept, is a challenging exercise for the MDR SurFlexDock pipeline. Enhancing the structural sampling of the receptor surface, we were able to obtain correct results for positive and negative experimental controls and were able to detect docking results of the APSNPA peptide to a SH3 domain in the nM magnitude (Ki), suggesting a new promising ligand.

As implemented, MDR SurFlexDock stands out as a valuable tool in HTP docking experimentation, reducing false negative results and increasing the quantity of promising ligands as a result of these experiments.



Supplementary Figure S1: Results obtained in MDR SurFlexDock using the second SH3 domain of human vinculin as receptor and a set of eight peptides as ligands. A: SH3 domain in contact with peptide ligand PPKPPI, as presented in 4ln2 pdb. SH3 domain is represented in secondary structure (yellow and green) and peptide ligand in wireframe (red-blue). B: Graphical representation of cluster analysis over the 5-ns trajectory obtained in molecular simulation of SH3 domain in explicit solvent at 300 K. Calculated clusters (in red) are represented in the lower part of the graphic, and calculated RMSD for every structure in the trajectory is represented in color code (green-yellow-red) in the higher part of the graphic. C: Positive results obtained in the experiments. For every ligand, the pipeline presents a graphical representation of the 'best 10' docking results for each receptor cluster. Representative structures of better results are represented, as in A. D: Negative results obtained in the experiment represented, as in C.

References

Abraham,M.J. et al. (2015) GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers

Alexandrescu A. (2001) *Modern C++ Design: Generic Programming and Design Patterens Applied.* Addision Wesley Professional, Boston.

Alexandrescu A. Modern C++ design: generic programming and design patterns applied, Boston: Addison Wesley Professional; 2001.

Alexandrescu,A. (2001) *Modern C++ Design: Generic Programming and Design Patterens Applied.* Addision Wesley Professional, Boston.

Crenshaw B, Jones WB. The future of clinical cancer management: one tumor, one chip. Bioinformatics. 2003; doi:10.1093/bioinformatics/btn000.

Dormand JR, Prince, PJ. (1980) A family of embedded Runge-Kutta formulae. J Comp Appl Math. 1980;6:19–26.

Dormand JR. and Prince PJ. (1980) A family of embedded Runge–Kutta formulae. *J. Comp. Appl. Math.,* **6**, 19–26.

Dormand,J.R. and Prince,P.J. (1980) A family of embedded Runge–Kutta formulae. *J. Comp. Appl. Math.,* **6**, 19–26.

Feng, S., et al. (1994) Two binding orientations for peptides to the Src SH3 domain: development of a general model for SH3–ligand interactions. *Science*, **266**, 1241–1247.

He P. et al. (2014) Why ligand cross-reactivity is high within peptide recognition domain families? A case study on human c-Src SH3 domain. *J Theor Biol.,* **340**, 30-37. doi: 10.1016/j.jtbi.2013.08.026.

Humphries,J.D. et al. (2007) Vinculin controls focal adhesion formation by direct interactions with talin and actin. *J. Cell Biol*., **179**, 1043-1057.

312 Morris, G. M., et al. (2009) Autodock4 and AutoDockTools4: automated docking with
313 selective receptor flexiblity*. J. Computational Chemistry*, **16**, 2785-91.

314 Rentzsch,R. and Renard,B.Y. (2015) Docking small peptides remains a great
315 challenge: an assessment using AutoDock Vina. *Brief Bioinform*., **6**, 1045-56,
316 doi:10.1093/bib/bbv008.

317 Saksela K., and Permi P. (2012) SH3 domain ligand binding: What's the consensus
318 and where's the specificity? *FEBS Lett.,* **17**, 2609-14.
319 doi:10.1016/j.febslet.2012.04.042.

320 Yoo MS, *et al*. (2003) Oxidative stress regulated genes in nigral dopaminergic
321 neuronal cell: correlation with the known pathology in Parkinson’s disease. Brain Res
322 Mol Brain Res. 2003;110:76–84.

323 Zhao, D. et al. (2014) Structural investigation of the interaction between the tandem
324 SH3 domains of c-Cbl-associated protein and vinculin. *J.Struct.Biol.,* **187**, 194-205.

Capitulo 5: HTP SurflexDock – Um *pipeline* para analises de *Virtual Screening*

5.1. Introdução:

O HTP SurFlexDock é uma ferramenta web de *Structure Based Virtual Screening* (SBVS) que utiliza a técnica de *sampling estrutural* para gerar um conjunto ampliado de conformações relevantes do receptor que podem interagir melhor com compostos que possuem estrutura ou propriedades diferentes do ligante originalmente encontrado na estrutura do receptor. O HTP SurFlexDock cria um *ensemble* com quatro conformações representativas do sitio ativo, com os quais são realizados experimentos *docking* utilizando uma biblioteca de moléculas pequenas como o ZINC (STERLING e IRWIN, 2015). O *ensemble* estrutural é composto da conformação original do receptor, em geral proveniente de uma estrutura cristalográfica e aqui chamada de '*Control'*, somada a outras três conformações obtidas de uma simulação molecular de cinco nano segundos. Estas conformações são obtidas pelo método de agrupamento (*clustering)* implementado no algoritmo GROMOS (OOSTENBRINK et al., 2005). O resultado do *virtual screening* é apresentado como uma tabela contendo a melhor pose de cada experimento de *docking* classificado pelo o ΔG calculado pelo AutoDock 4.2. Assim como no MDR SurFlexDock (Capitulo 4), a estrutura do ligante em complexo com o receptor pode ser visualizada diretamente da tabela dos resultados.

Para iniciar um experimento de VS no HTP SurFlexDock, o usuário deve inserir:

(1) um e-mail válido para receber informações sobre o andamento do seu experimento e acesso aos resultados,

(2) a estrutura da proteína-alvo (receptor) em formato pdb,

(3) a localização do sítio ativo na proteína-alvo que, assim como já explicado para o MDR SurFlexDock, é definido por um *box* cúbico. Neste o usuário deve definir o número de um resíduo que será o centro do *box* e a distância deste resíduo até o limite do sitio ativo em nanômetros,

(4) Uma biblioteca de pequenas moléculas pode ser carregada no formato .tar.gz ou .zip, onde cada composto desta biblioteca deve ser um arquivo em formato pdbqt e,

(5) A quantidade de poses geradas para cada *docking* (de 10 a 30), que é o coração da técnica de SBVS.

Ao pressionar *botão* upload, o HTP SurFlexDock inicia o *pipeline* pela obtenção do *ensemble* de conformações do receptor, seguido pelo experimentos de *docking*. O HTP SurFlexDock foi configurado para fazer a simulação molecular utilizando 10 threads de CPU. No entanto, como o AutoDock 4.2 funciona em thread de processamento única, para maximizar o desempenho da experimentação foi utilizada a biblioteca billiard.multiprocessing (*Queue, Celery: Distributed Task*), assim, são executados *dockings* simultâneos de forma que se use 90% do poder de processamento disponível no servidor. Neste contexto, em cada experimento no HTP SurFlexDock, é criado uma fila de processos do AutoDock (NORGAN et al., 2011) que irão utilizar 90% das threads da CPU do computador hospedeiro. Desta forma, Em um *virtual screening* de 90 moléculas utilizando um computador com 10 threads de processamento, serão alocados 9 processos de *docking* simultaneamente aos núcleos de processamento, enquanto 91 processos de *docking* ficam aguardando. Quando um processo de *docking* termina o processamento, o primeiro processo da fila é alocado para aquela thread que ficou ociosa. Assim, o desempenho do experimento é determinado pela velocidade de cada núcleo assim como a quantidade de threads de processamento. Ao término de cada *docking*, o HTP SurFlexDock carrega na página de resultados um ranking das melhores poses de cada *dockings* processado até o momento e ao final do experimento, o usuário é avisado via e-mail cadastrado.

A validação de softwares de SBVS geralmente é feita com a utilização de métricas de avaliação de desempenho, isto é, avaliar o quanto um software é eficiente no enriquecimento de melhores ligantes (TRUCHON e BAYLY, 2007). Normalmente, esta avaliação pode ser realizada de duas formas. (1) através do cálculo do RMSD (*Root Mean Square deviation*), isto é, o valor médio da distância entre os átomos do complexo proteína-ligante calculado, relativo a estrutura cristalizada ou (2) avaliando se o software é capaz de enriquecer os ligantes em uma biblioteca contendo ligantes e moléculas sem atividade com a proteína-alvo por meio da metodologia ROC (*Receiver Operating Characteristic*) (BOEHM, 2011; FRADERA e BABAOGLU, 2018).

Para avaliar o HTP SurFlexDock, utilizamos a metodologia ROC para verificar se a ferramenta é capaz de enriquecer ligantes de propriedades químicas distintas do ligante que a enzima conversora de angiotensina I foi cristalizada. Para este experimento, utilizamos uma

biblioteca que contém moléculas ativas e inativas para alguns ACE, obtida do banco de dados DUD (MYSINGER et al., 2012).

5.2. Avaliando enriquecimentos de *virtual screening* através da metodologia ROC

Em um experimento de SBVS estamos interessados em diferenciar as moléculas que são ativas das inativas, desta forma, um software de VS pode ser analisado por métodos de avaliação de classificações binárias como, por exemplo, o ROC. Uma curva ROC é um método que permite avaliar o quanto um determinado parâmetro é eficiente para uma classificação binária (GRAU et al., 2015). Este método foi criado por militares americanos durante a segunda grande guerra para avaliar o quão eficiente seus radares eram na detecção de objetos inimigos (CARTER et al., 2016). Hoje, as curvas ROC são utilizadas em diversas áreas como Psicologia, Medicina, métodos de inteligência artificial e mineração de dados. A curva ROC consiste em um gráfico bidimensional onde se representa o eixo y indicando a taxa de positivos verdadeiros (TPR - *True Positive Rate*) e o eixo x, a taxa de falsos positivos (FPR - *False Positive Rate*) na classificação da amostra em estudo (**Figura 9**). Neste contexto, uma classificação perfeita, isto é sem falsos positivos, possui uma curva ROC como indicado na Figura 9:A. Além da curva, outro parâmetro normalmente calculado é a área abaixo da curva (AUC) que representa numericamente a eficiência da classificação de um dado parâmetro.

No gráfico ROC de um *virtual screening*, o eixo y indica a quantidade de ligantes encontrados enquanto o eixo x representa a quantidade de moléculas inativas. Do mesmo modo, o parâmetro AUC indica o quão eficiente o *virtual screening* é para encontrar os ligantes. Outro parâmetro bastante discutido é o enriquecimento precoce (TRUCHON e BAYLY, 2007) – quantização dos primeiros ligantes enriquecidos que aparecem em um gráfico ROC, tendo em vista que apenas a fração das primeiras moléculas enriquecidas de um experimento de *virtual screening* será utilizada para futuros testes de laboratório. Isto é, em termos farmacológicos é interessante que os ligantes escolhidos tenham claramente uma forte interação com o alvo proteico (TRUCHON e BAYLY, 2007).

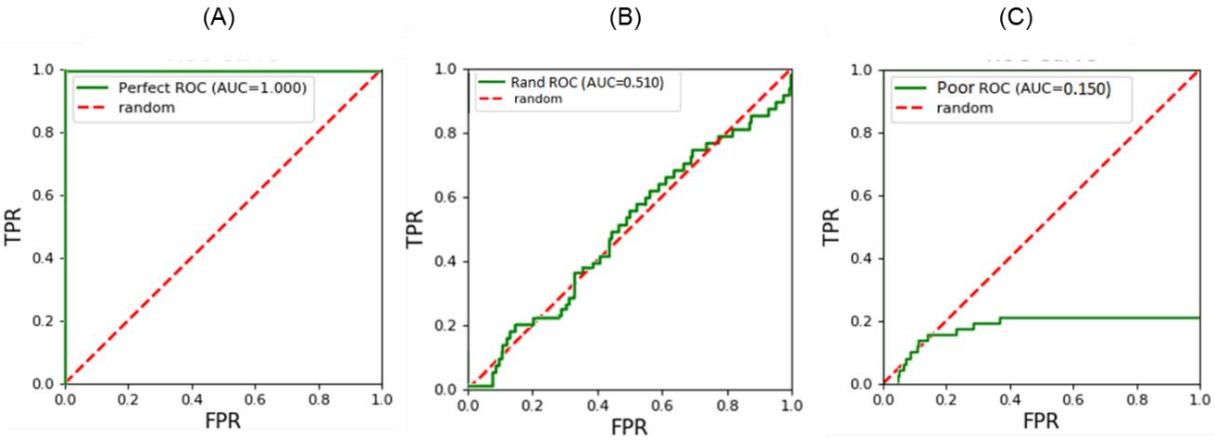


Figura 9 - Exemplos de curvas ROC. (A) gráfico ROC de uma classificação perfeita com AUC= 1.0. (B) Quando a curva se aproxima da reta TPR=FPR, tracejada em vermelho, temos uma classificação aleatória, com a AUC próxima a 0.5. (C) Caso a curva fique abaixo da reta TPR=FPR, temos um resultado ruim. TPR = True Positive Rate (Taxa de verdadeiros positivo), FPR = False Positive Rate (Taxa de Falsos positivo).

5.3. Enzima conversora de angiotensina I como candidata a experimentos de *virtual screening*

A enzima conversora de angiotensina I humana (hACE, EC: 3.14.15.1) é uma metaloprotease de grande relevância terapêutica, pois, é alvo no tratamento de vários problemas vasculares que incluem hipertensão arterial, infarto do miocárdio e nefropatia diabética (BATEMAN et al., 2017; EVANS et al., 2016). No corpo humano, a hACE atua no sistema renina-angiotensina (RAS), de forma catalisar a conversão da angiotensina I em angiotensina II, este último um importante vasoconstritor (MASUYER et al., 2012). Por outro lado, a hACE é também um importante participante do sistema calicreína-cinina, uma vez que degrada o hormônio bradicinina, o qual possui atividade hipotensiva. Em humanos foram identificados dois tipos de hACE, duas produzidas em células somáticas (sACE) e o outro, em células do testículo (tACE) (JIANG et al., 2019), ademais, estudos apontam que a tACE está relacionado à fertilidade masculina (GALANIS et al., 2016; GIANZO et al., 2018).

Molecularmente, a sACE é um dímero cujos os domínios são conhecidos como C-terminal (hACEc) e N-terminal (hACEn) (Figura 10) (NATESH et al., 2003). A similaridade entre os dois domínios é de aproximadamente 55% e ambos possuem a capacidade de converter a angiotensina I em angiotensina II (BURNIER e BRUNNER, 2000; ZHANG et al., 2013). A estrutura tridimensional desta enzima foi determinada por cristalografia de raios X em 2003 (NATESH et al., 2003), sendo identificado em seu sítio catalítico um íon zinco (II) que é ligado a 2 histidinas e 1 glutamato (GUAN et al., 2016; JALKUTE et al., 2013; ZHANG et al., 2013), é importante destacar que sem este cofator a enzima não funciona

adequadamente. Além disto, existem 2 cofatores cloro que estão aproximadamente a uma distância de 10Å e 20Å do íon zinco. Há uma grande discussão sobre a função destes íons cloro, porém estudos sugerem que a presença destes aumenta a capacidade catalítica das hACE (ZHANG et al., 2013).

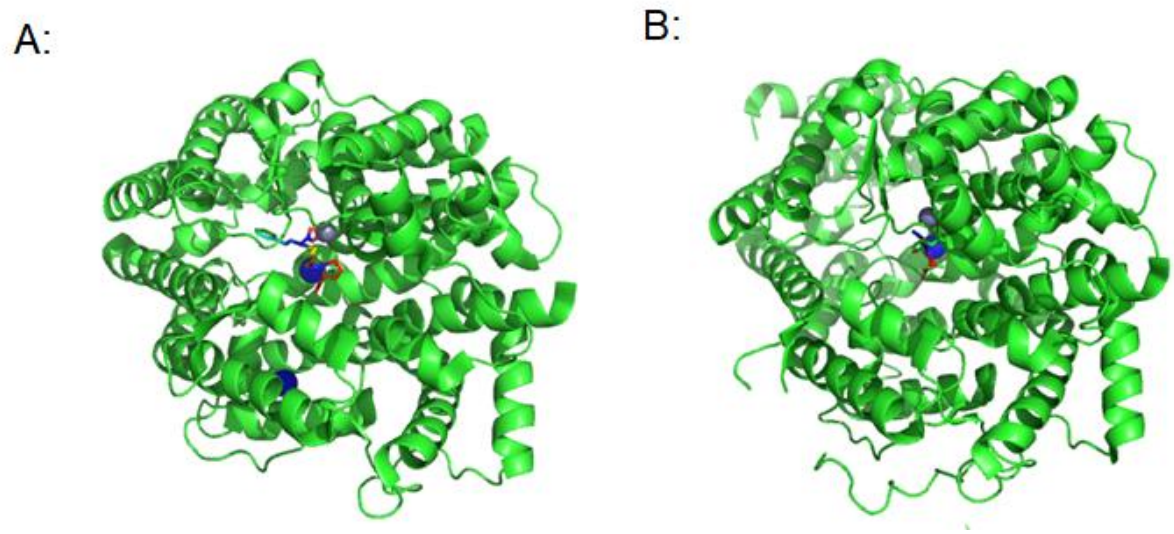


Figura 10 - Estrutura dos domínios C-terminal e N-Terminal da sACE. (A) Domínio C-terminal com o inibidor lisinopril PDB id 1o86. (B) Domínio N-terminal, cadeia A do PDB id 5amb com ligante P6G.

Uma família importante de inibidores da enzima conversora da angiotensina I foi identificada e isolada no veneno da serpente *Bothrops jararaca* a partir dos trabalhos de Maurício Rocha e Silva (1949) e Sérgio Henrique Ferreira (1970). Esta família é denominada Peptídeos Potenciadores de Bradicinina (*Bradykinin-Potentiating Peptides* - BPP), pois foi descoberto que estes peptídeos evitavam a degradação do hormônio bradicinina (FERREIRA et al., 1970; SCIANI e PIMENTA, 2017). Estudos utilizando as BPP's demonstraram que elas apresentavam grande atividade hipotensiva, e posteriormente, foi descoberto que a atividade hipotensiva de algumas das BPP's estava relacionadas com a inibição da enzima conversora de angiotensina I ao competir com os seus ligantes naturais (CAMARGO et al., 2012).

Baseando-se em um modelo hipotético da hACE, com o sitio catalítico semelhante ao da carboxipeptidase A, David Cushman e Miguel Ondetti utilizaram um peptídeo análogo a BPP que, em experimentos, descobriu-se que apresentaram atividade inibitória com a hACE. Ao referido peptídeo foi incluído um grupamento tiol (-SH), para potencializar o efeito hipotensor (CUSHMAN e ONDETTI, 1999). Tempos mais tarde, este inibidor foi comercializado como medicamento para hipertensão arterial com o nome de Captopril, o primeiro fármaco desenhado para um alvo molecular (CAMARGO et al., 2012) (Figura 11:A).

Outros inibidores de hACE foram patenteados, subsequentemente, entre eles o Lisinopril (LANCASTER e TODD, 1988), o Enalapril (TODD e HEEL, 1986) e o Fosinopril (MURDOCH e MCTAVISH, 1992). Estes fármacos foram criados com o intuito de sanar os efeitos colaterais do Captopril, especialmente relacionados ao grupamento tiol (BRYAN, 2009). Desta forma, estes compostos utilizam o grupamento carboxílico (Lisinopril e Enalapril) ou fosfínico (Fosinopril) para a interação direta com o cofator zinco (GUTHRIE, 1993; NATESH et al., 2004) , além de outras alterações estruturais com o intuito de manter o mesmo desempenho do Captopril (MENARD e PATCHETT, 2001) (Figura 12:B).

As interações dos ligantes em hACE ocorrem em quatro poços denominados S1, S2, S1’ e S2’ presentes no sitio ativo da enzima. Moléculas peptídicas de diferentes tamanhos podem acomodar suas cadeias laterais P1, P2, P1’ e P2’ nos poços da enzima através de pontes de hidrogênio e interações hidrofóbicas (Figuras 11 e 12) (FERNANDEZ et al., 2004; MASUYER et al., 2012).

Além disso, o íon zinco está localizado os poços S1 e S1’(NATESH et al., 2004), onde os ligantes fazem uma interação direta com grupamentos carboxílicos, tiólicos, oxos, fosfínicos entre outros (Tabela 2:A) (NATESH et al., 2003; YOUNG, 1980). A conversão da Angiotensina I em Angiotensina II é feita pela hACE ao clivar os dois últimos resíduos a parte c-terminal da angiotensina I. Neste sentido, estudos utilizando simulações moleculares indicam que a catalise do substrato peptídico na hACE é feita por um ataque nucleofílico realizado por uma molécula de água que é coordenada pelo cofator zinco e cadeia lateral de um glutamato próximo (Figura 12:A) (MU et al., 2016). Por outro lado, o bloqueio da enzima ocorre, quando os compostos inibidores interagem com a hACE inativam o íon zinco como é o caso dos compostos farmacêuticos ou alterando contatos chave necessários para atividade enzimática (Figura 12:B-C) (FERNANDEZ et al., 2004). Estas características terapêuticas e moleculares da hACE, além da quantidade de informação estrutural de interação de diferentes inibidores com seus sítios ativos (pdbs: 1O86; 1UZE; 1UZF; 2XY9; 4BZR; 4BZS; 4CA5; 6F9T; 6F9U; 6F9V; 4UFB; 2XYD; 2C6N; 6H5W; 6H5X; 6EN6; e 3NXQ), torna esta enzima um potencial candidato para avaliação e validação de testes de *virtual screening*.

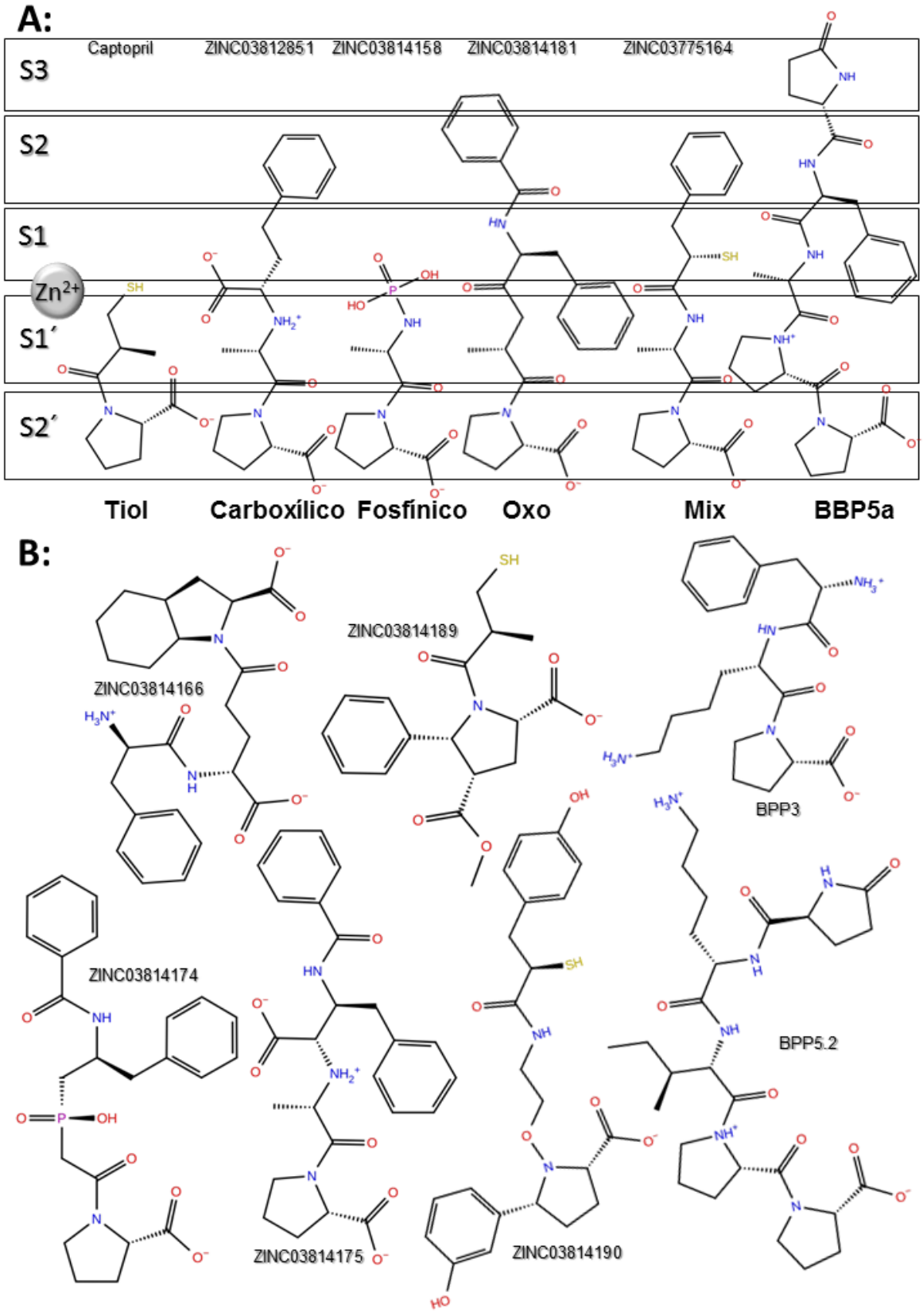


Figura 11 - Diferentes classes de inibidores para hACE. (a) Esquema Interação das classes de inibidores com os poços do sitio catalítico da hACE. A classificação destes inibidores é dada através do radical que interage no íon Zn2+ presente no sítio catalítico da enzima. (B) Inibidores de diferentes classes e tamanhos que foram utilizados no experimento de enriquecimento de ligantes.

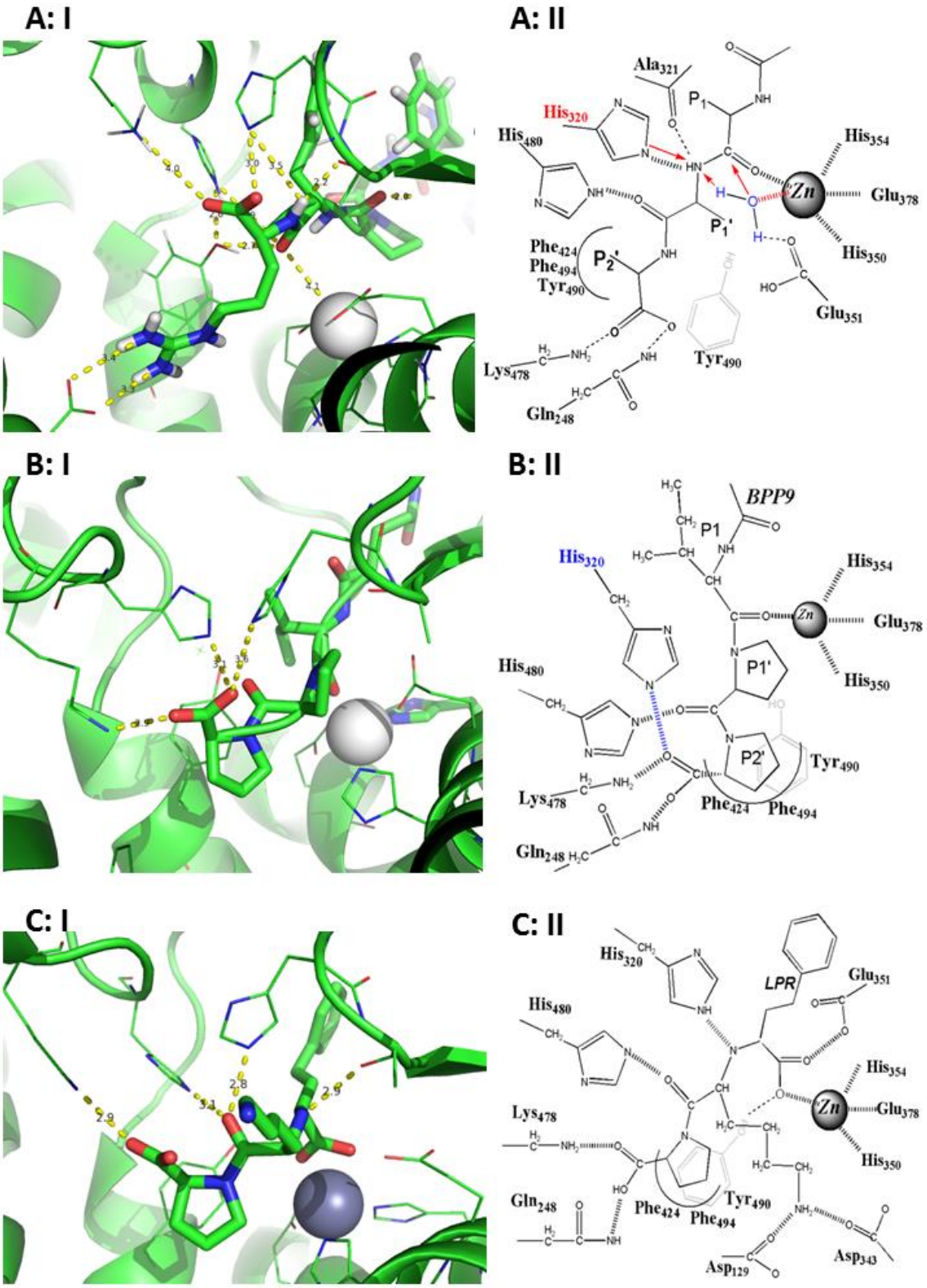


Figura 12 - Representação da interferência de diferentes moléculas com as cadeias laterais da hACE. (I) Representação tridimensional (II). Representação bidimensional. (A) hACE em conformação para hidrolise de um peptídeo. (B) Interferência causada pelo inibidor BPP9. (C) Interferência causada pelo inibidor Lisinopril

5.4. Materiais e métodos

5.4.1. Obtenção das estruturas dos receptores

A estrutura cristalográfica da hACE foi obtida no site RSCB PDB (BERMAN et al., 2000) (https://www.rcsb.org/) por meio do PDB ID 1O86 (NATESH et al., 2003) para o domínio C-terminal e PDB ID 5amb (LARMUTH et al., 2016) para o domínio N-terminal. Ambos os domínios foram modelados pelo *pipeline* de modelagem para completar resíduos ou cadeias laterais faltantes da estrutura cristalográfica utilizando as próprias estruturas como *template*, sendo que do pdb 5amb (N-terminal) foi selecionada para fazer o modelo a cadeia A. Dos modelos gerados foram removidas moléculas de águas estruturais e respectivos inibidores, permanecendo apenas na proteína os íons de cloro e zinco.

5.4.2. Seleção compostos para criação da biblioteca de moléculas

Para os experimentos foram utilizados uma biblioteca de compostos para a enzima convertedora de angiotensina I disponível no banco de dados DUD (*Database of useful Decoys*) (MYSINGER et al., 2012)

Nesta biblioteca de 51 ligantes com atividade reconhecida e 1797 *decoys* foram adicionados 3 ligantes com grupamentos fosfínicos extraídos dos pdb 2OC2 (ligante RX3) (ANTHONY et al., 2010), 4CA6 (ligante 3EF) (MASUYER et al., 2014), 2X97 (ligante RX4) (AKIF et al., 2010) disponíveis no RSCB PDB e 6 ligantes peptídicos naturais do tipo BPPs (Peptídeos potenciadores de bradicinina) (Tabela I). A relação final das bibliotecas usadas foi aproximadamente de 1/39 (ligante/decoys).

Todas as moléculas escolhidas foram descompactadas e convertidas para o formato pdbqt utilizando o script do AutoDockTools pdb_to_pdbqt.py (MORRIS et al., 2009). Durante a conversão, as moléculas foram nomeadas como ‘ligandxx’ e ‘decoyxxxx’ seguidos de um número sequencial, de acordo com sua classificação. Para a utilização no HTP SurFlexDock, os compostos foram organizados em pequenas bibliotecas sendo compactados pelo o software GNU-tar.

5.4.3. Utilização do HTP SurflexDock

Nos experimentos que utilizaram o domínio C-terminal da hACE foi carregado no HTP SurflexDock a estrutura da hACEc.pdb e como resíduo central para o *box* foi utilizado o resíduo Glu411 que coordena o íon de Zn. Cada experimento foi selecionado para realizar 30 experimentos de *docking* por composto. Por outro lado, nos experimentos com o domínio N-

terminal, foi carregada a estrutura hACEn.pdb, selecionado o resíduo Glu311, os outros parâmetros foram escolhidos os mesmos dos experimentos hACEc.pdb.

Em HTP SurFlexDock, os experimentos de enriquecimento foram realizados em conformações representativas do receptor obtidas de simulações comparativas de 2 e 5 ns. Cada um destes experimentos foi realizado utilizando os domínios N- e C-terminal. Em ambos os experimentos foram utilizados 46 ligantes e 1797 compostos *decoys* (proporção aproximada de 1/39) como descrito anteriormente. Para cada *cluster* estrutural do receptor calculado, foi criada uma tabela contendo a melhor pose para cada *docking* e seu **Δ**G. Esta tabela foi utilizada junto com o script *roc2py* para gerar o gráfico ROC e calcular a área abaixo da curva. Na execução do HTP SurflexDock e análise de dados foram usados um servidor Intel(R) Xeon(R) CPU E5-2670 v2 @ 2.50GHz e notebooks Acer Nitro5-AN515.

5.4.4. Avaliação dos experimentos por meio de gráfico ROC

Ao final de cada experimento, o HTP SurflexDock cria uma tabela com a melhor pose de cada *docking* ranqueado pelo **Δ**G. Esta tabela somada a uma lista dos ligantes foi utilizado no script *open-source* roc2py[2] (OECHEM, 2012) para gerar o gráfico ROC e calcular a área embaixo da curva (parâmetro AUC). Este programa utiliza a informação de cada linha da tabela e insere um ponto na curva. Uma versão do script roc2py foi alterada para avaliar o enriquecimento por meio do gráfico ROC dos primeiros 20% dos resultados obtidos no experimento.

5.5. Resultados e discussão

Desenvolvemos o HTP SurflexDock para realizar experimentos de *virtual screeening* com o objetivo de uma melhor capacidade de enriquecimento a diversidade de ligantes através do uso de *ensemble docking*. Em HTP SurflexDock, o *ensemble* é composto da conformação inicial da proteína alvo, somada a 3 conformações mais representativas obtidas da simulação molecular de 5ns. Neste contexto, avaliamos o desempenho de nossa ferramenta com o *virtual screening* nos domínios N- e C-terminal da enzima conversora da angiotensina I (hACE) e uma biblioteca com cerca de 1800 compostos contendo 46 ligantes para a enzima e 1797 compostos supostamente sem atividade (*decoys*). Avaliamos também o impacto da alteração de diferentes parâmetros utilizados pelo software no resultado do experimento.

[2] Disponível em: https://docs.eyesopen.com/toolkits/cookbook/python/plotting/roc.html

5.5.1 Seleção de ligantes da hACE para a criação da biblioteca de compostos utilizada no experimento.

A biblioteca disponível no banco de dados DUD possui 1797 moléculas *decoys* somado a 49 ligantes. Destes ligantes, um não funcionou (ligand_39; ZINC03814190) com o HTP SurflexDock e foi eliminado. Outros 10 ligantes foram eliminados devido ao desempenho insatisfatório em experimentos preliminares (Apêndice A). Para melhorar a proporção e a variedade de quimiotipos de ligantes no experimento, adicionamos a lista de ligantes 3 compostos fosfínicos obtidos de estruturas cristalográficas de hACE depositadas no RCSB PDB e 6 ligantes do tipo BPP (peptídeos potenciadores de bradicinina) (Tabela I).

Tabela I – Modificações feitas no grupo de ligantes obtidos no banco de dados DUD.

Nome do ligante no experimento	Procedência	Tipo de composto	Referencia
Compostos retirados do experimento			
ligand_5	ZINC03812885	Tiol	DUD
ligand_6	ZINC03814157	Tiol	DUD
ligand_11	ZINC03814162	Tiol	DUD
ligand_16	ZINC03814167	Tiol	DUD
ligand_20	ZINC03814171	Tiol	DUD
ligand_26	ZINC03814177	Tiol	DUD
ligand_28	ZINC03814179	Tiol	DUD
ligand_38	ZINC03814189	Tiol	DUD
ligand_39	ZINC03814190	Tiol	DUD
ligand_40	ZINC03814191	Tiol	DUD
Compostos adicionados no experimento			
RX3	pdb 2OC2	Fosfínico	ANTHONY et al., 2010
3EF	pdb 4CA6	Fosfínico	MASUYER et al., 2014
RX4	pdb 2X97	Fosfínico	AKIF et al., 2010
BPP3 (F-K-P)	Este trabalho	BPP	---
BPP4 (F-A-P-P)	Este trabalho	BPP	---
BPP5a (Pyr-K-W-A-P)	UniProtQ6LEM5	BPP	IANZER et al., 2011
BPP5.2 (Pyr-K-I-P-P)	Este trabalho	BPP	---
BPP5.3 (Pyr-F-A-K-P)	Este trabalho	BPP	---
BPP5.4 (Pyr-F-A-P-P)	Este trabalho	BPP	---

Neste contexto, a biblioteca final utilizada nos experimentos foi composta de 1797 moléculas *decoys* e 46 ligantes classificados como (1) 16 Carboxilicos, (2) 5 Fosfínicos (3) 19 Tiól e (4) 6 BPPs. A relação final foi aproximadamente de 1/39 (ligante/*decoys*).

5.5.2 Impacto da escolha do tamanho do *box* no *screening* com a hACE

Para determinar o tamanho de *box* mais eficiente na definição do sítio ativo da hACE, avaliamos os domínios N- e C-terminal da enzima com *box* de dimensões de meia aresta d=0.75 nm, d=1.0 nm e d=1.2 nm. Nestes experimentos, foi utilizada uma biblioteca contendo um total de 100 compostos, destes 10 eram os melhores ligantes para cada domínio determinados em experimento preliminar (Anexo A) e 90 moléculas *decoys* selecionadas aleatoriamente do total de 1790 *decoys* obtidos no DUD.

De forma geral o incremento do tamanho do *box* conduziu a um melhor enriquecimento de ligantes (Figura 13), sendo que o *box* com o tamanho de meia aresta d=1.2 nm obteve a melhor performance tanto no domínio C-terminal, com parâmetro AUC de aproximadamente de 97% (Figura 13:C) quanto no N-terminal, obtendo o parâmetro AUC de aproximadamente de 98% (Figura 13:F).

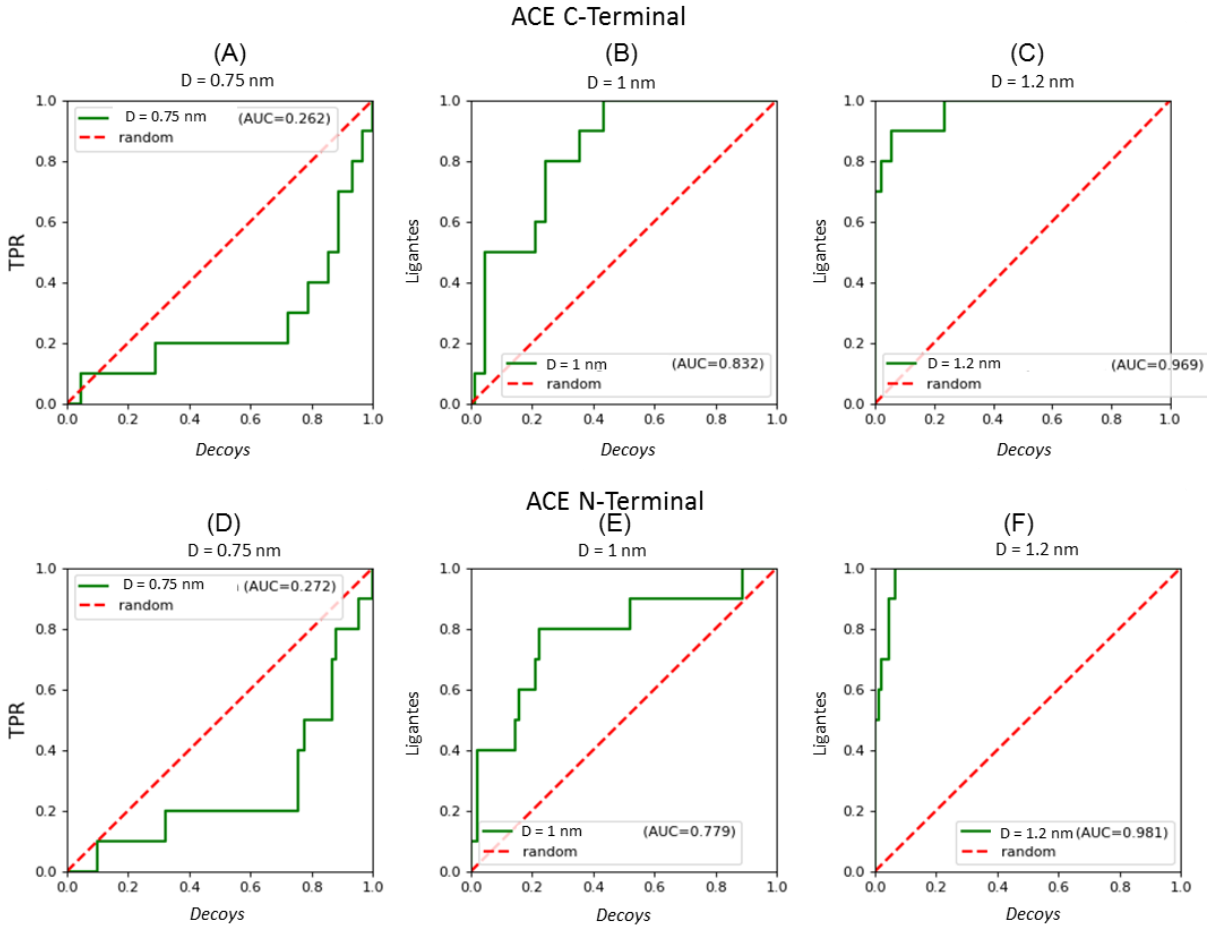


Figura 13 - Avaliação do impacto tamanho da *box* de solvatação no *docking* da hACE. Para este experimento foi utilizado um set com os 10 melhores ligantes de cada domínio (Anexo A) e 90 decoys escolhidos randomicamente. As distancias assinaladas com 'D' indicam a distância entre o carbono alfa do resíduo central definido pelo o usuário no início do experimento até os limites do *box* (metade da aresta do *box*). (A) Gráfico ROC no domínio C-terminal da hACE utilizando *box* com meia aresta d=0.75 nm. (B) Gráfico ROC no domínio C-terminal na hACE utilizando *box* com meia aresta d=1.0 nm. (C) Gráfico ROC no domínio C-terminal da hACE utilizando *box* com meia aresta d=1.2 nm. (D) Gráfico ROC no domínio N-terminal da hACE utilizando *box* com meia aresta d=0.75 nm. (E) Gráfico ROC no domínio N-terminal da hACE utilizando *box* com meia aresta d=1.0 nm. (E) Gráfico ROC no domínio N-terminal da hACE utilizando *box* com meia aresta d=1.2 nm.

5.5.3 Impacto de diferentes tempos de simulação nos resultados de *ensemble docking* utilizado no HTP SurflexDock.

A simulação da flexibilidade do receptor no HTP SurflexDock é feita por meio da criação de um *ensemble* contendo a conformação original do receptor mais 3 conformações mais representativas da simulação molecular do receptor. Neste experimento, avaliamos o impacto na criação do *ensemble* e no resultado dos experimentos utilizando simulações moleculares de 2ns e 5ns.

Neste contexto, utilizando o *box* de meia aresta d=1.2nm, o experimento de *virtual screening* com o uso do *ensemble docking* criado a partir de uma simulação de 2 ns demonstrou no domínio C-terminal da hACE, uma piora no enriquecimento dos ligantes em todas as conformações como pode ser observado pelos gráficos ROC e parâmetros AUC da Figura 14. Em relação ao enriquecimento inicial, nas conformações t=718 ps e t=1000 ps a quantidade de ligantes do tipo BPP enriquecidos foi superior do que a conformação controle. Além disso, a conformação t=718 ps houve o enriquecimento inicial de 2 ligantes do tipo fosfínicos enquanto as outras conformações houve o enriquecimento somente de 1 ligante (Tabela II). No entanto, não foram observadas mudanças de tipos de ligante no enriquecimento dos primeiros 15 compostos.

No domínio N-terminal de hACE foi observado uma piora do enriquecimento geral dos ligantes que pode ser visualizada pela Figura 15, além disso, a Tabela III indica um menor número de ligantes detectados nos primeiros 15 compostos do resultado. Por outro lado, uma discreta mudança no perfil de ligantes pode ser observada nas conformações dos *clusters* t=314 ps e t=966. Estas conformações enriqueceram um ligante fosfínico a mais do que a conformação controle. Outro aspecto que pode ser observado nesta mesma tabela é o enriquecimento de um composto tiol na conformação t=592. Contudo, acreditamos que esse hit seja uma exceção no experimento devido ao péssimo desempenho geral de todos os ligantes do tipo tiól nos experimentos com o HTP SurflexDock.

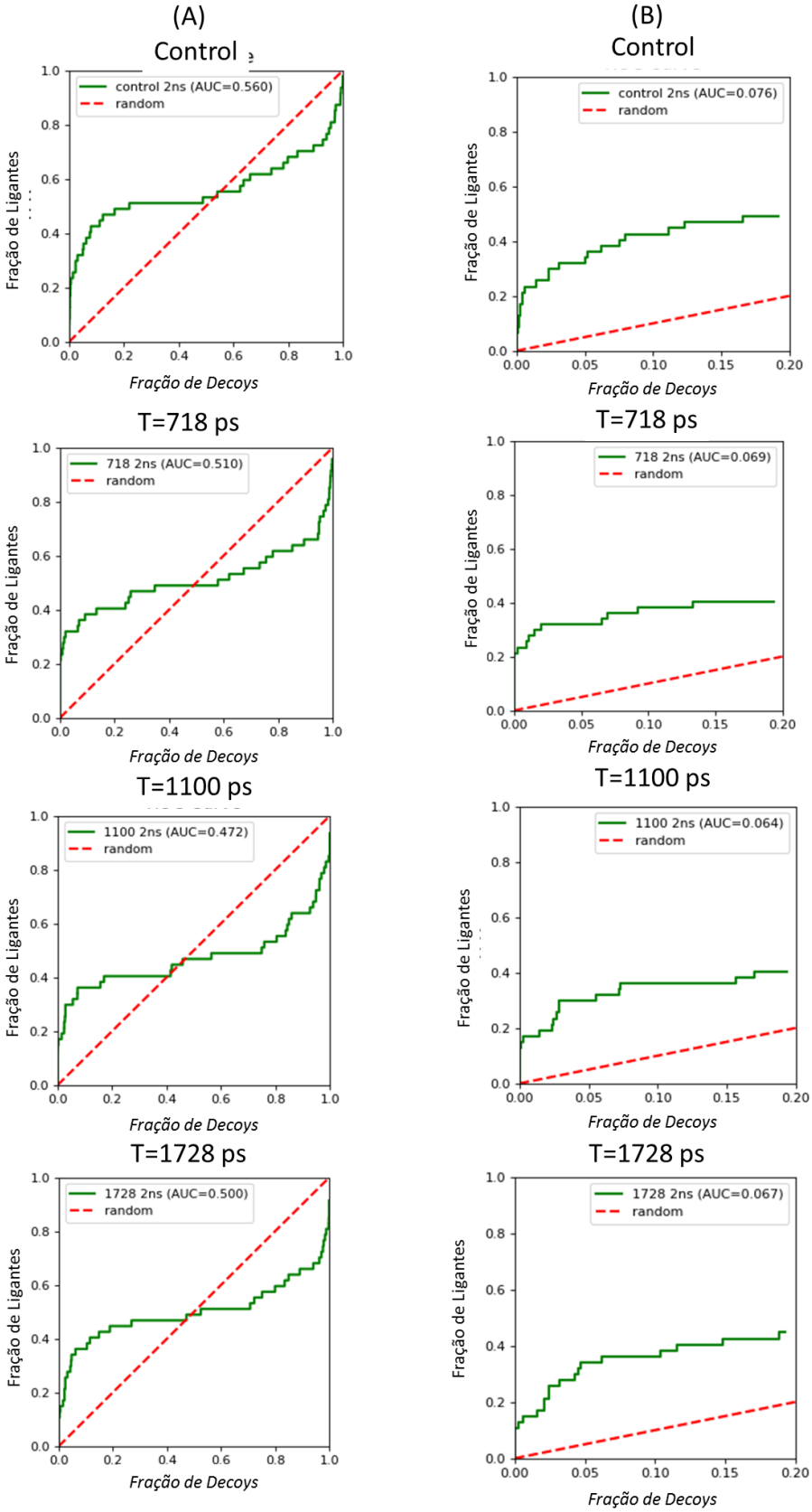


Figura 14 - Curvas ROC do domínio C-Terminal da hACE (hACEc) para cada conformação representativa obtida da simulação de 2ns . (A) Curva ROC referente a 100% da biblioteca testada. (B) Curva ROC referente aos primeiros 20% da biblioteca testada.

Tabela II– Enriquecimento inicial dos 15 melhores *dockings* utilizando conformações obtidas da simulação de 2ns do domínio C-Terminal da hACE

Control			***Cluster* t=718**			***Cluster* t=1100**			***Cluster* t=1728**		
Compound	**ΔG (kcal/mol)**	***Chemotype***	***Compound***	**ΔG (kcal/mol)**	***Chemotype***	***Compound***	**ΔG (kcal/mol)**	***Chemotype***	***Control***	**ΔG (kcal/mol)**	***Chemotype***
BPP5a	-11,03	BPP	BPP5_2	-11,81	BPP	BPP5_3	-10,39	BPP	BPP3	-11,47	BPP
ligand_37	-10,8	Carboxilico	BPP5a	-11,52	BPP	BPP3	-10,36	BPP	BPP5_3	-11,08	BPP
RX3	-10,41	Fosfinico	RX3	-11,3	Fosfinico	BPP5a	-10,21	BPP	BPP5a	-10,92	BPP
decoys_600	-10,39		BPP3	-10,43	BPP	ligand_15	-9,96	Carboxilico	ligand_15	-9,92	Carboxilico
ligand_34	-10,37	Carboxilico	ligand_15	-10,4	Carboxilico	BPP5_2	-9,92	BPP	RX3	-9,76	Fosfinico
decoys_619	-10,34		BPP5_3	-10,18	BPP	RX3	-9,39	Fosfinico	decoys_600	-9,48	
decoys_1321	-10,32		ligand_24	-9,35	Carboxilico	decoys_522	-9,35		decoys_695	-9,43	
BPP5_3	-10,3	BPP	BPP4	-9,33	BPP	BPP4	-9,16	BPP	decoys_584	-9,39	
ligand_24	-10,28	Carboxilico	ligand_23	-9,3	Fosfinico	decoys_600	-8,95		decoys_811	-9,36	
decoys_156	-10,26		BPP5_4	-9,18	BPP	decoys_1775	-8,94		decoys_619	-9,29	
ligand_15	-10,16	Carboxilico	decoys_695	-9,15		decoys_1699	-8,94		ligand_24	-9,23	Carboxilico
ligand_33	-10,1	Carboxilico	decoys_1715	-9,08		ligand_42	-8,91		decoys_1693	-9,12	
decoys_1274	-10,08		decoys_939	-9,01		decoys_1715	-8,88	Carboxílico	decoys_554	-9,09	
decoys_554	-10,07		decoys_1204	-8,99		decoys_320	-8,84		decoys_534	-9,07	
decoys_1269	-10,02		ligand_30	-8,98	Carboxilico	decoys_323	-8,82		decoys_1083	-9,02	

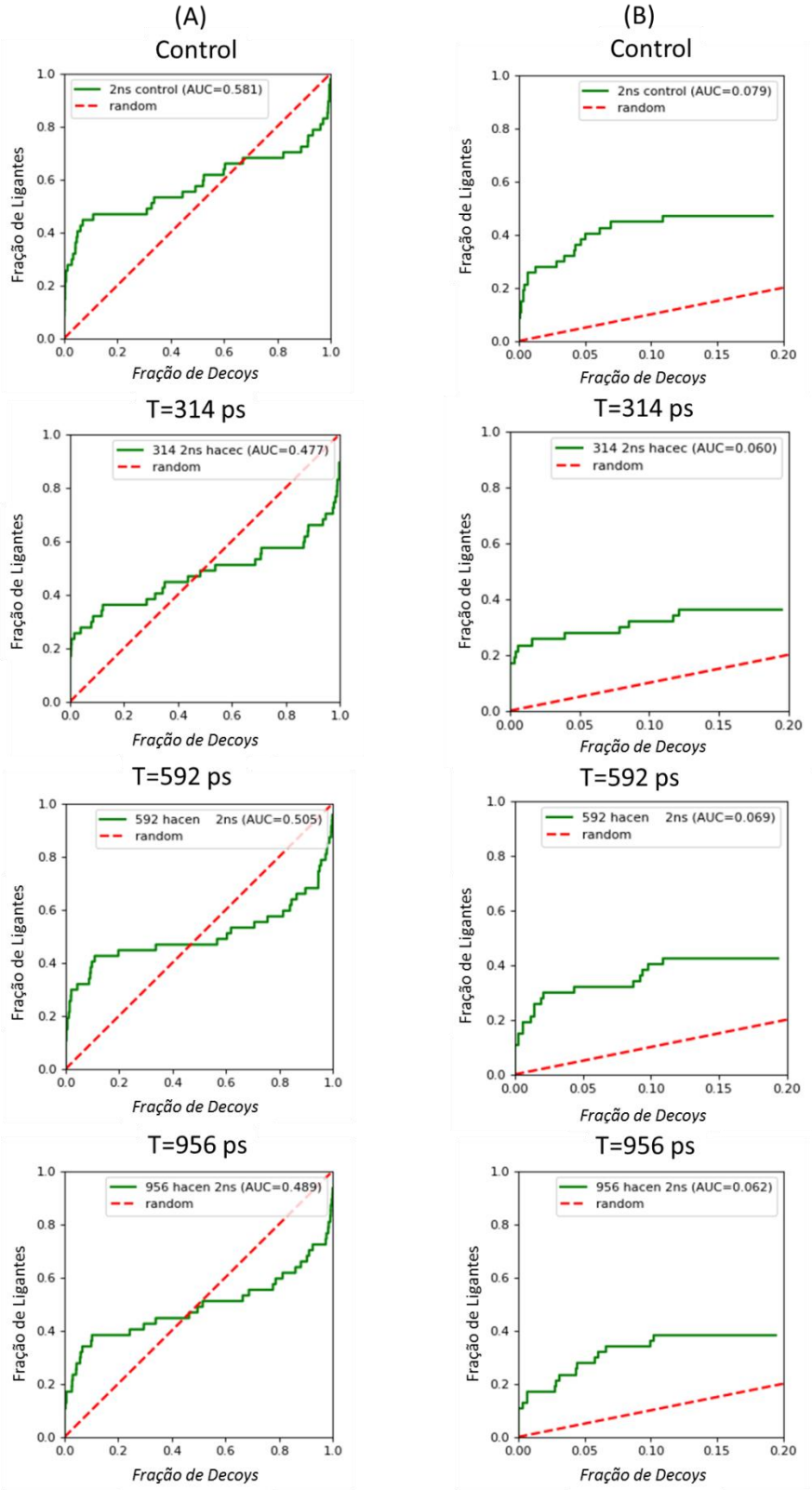


Figura 15 – Curvas ROC do domínio N-Terminal da hACE (hACEn) para cada conformação representativa da simulação de 2ns. (A) Curva ROC referente a 100% da biblioteca testada. (B) Curva ROC referente aos primeiros 20% da biblioteca testada.

Tabela III– Enriquecimento inicial dos 15 melhores *dockings* conformações obtidas da simulação de 2ns do domínio N-Terminal da hACE.

Control			***Cluster* t=314**			***Cluster* t=592**			***Cluster* t=956**		
Molécula	**ΔG (kcal/mol)**	***Chemotype***	**Molécula**	**ΔG (kcal/mol)**	***Chemotype***	**Molécula**	**ΔG (kcal/mol)**	***Chemotype***	**Molécula**	**ΔG (kcal/mol)**	***Chemotype***
BPP5_3	-11,79	BPP	RX3	-12,37	Fosfinico	BPP5a	-11,36	BPP	BPP5a	-12,34	BPP
BPP5a	-11,7	BPP	BPP5_3	-11,71	BPP	BPP5_3	-11,1	BPP	ligand_15	-10,82	Carboxilico
BPP3	-11,52	BPP	ligand_15	-11,48		BPP3	-10,77	BPP	RX3	-10,5	Fosfinico
RX3	-11,42	Fosfinico	BPP5_2	-10,94	BPP	RX3	-10,26	Fosfinico	BPP5_3	-10,3	BPP
decoys_796	-10,83		BPP5a	-10,82	BPP	ligand_15	-10,12	Carboxilico	BPP3	-9,91	BPP
decoys_848	-10,74		ligand_23	-10,62	Fosfinico	decoys_338	-9,62		decoys_326	-9,75	
ligand_24	-10,74	Carboxilico	ligand_42	-10,58	Carboxilico	decoys_1608	-9,28		decoys_522	-9,74	
decoys_1574	-10,74		BPP3	-10,4	BPP	decoys_554	-9,25		decoys_1564	-9,72	
ligand_15	-10,7	Carboxilico	decoys_1672	-10,35		decoys_165	-9,14		decoys_613	-9,63	
ligand_33	-10,59	Carboxilico	decoys_526	-10,34		BPP4	-9,14	BPP	decoys_320	-9,61	
decoys_1278	-10,56		decoys_619	-10,31		ligand_36	-9,12	Tiol	decoys_1779	-9,56	
decoys_1270	-10,43		decoys_323	-10,29		decoys_49	-9,09		ligand_23	-9,55	Fosfinico
decoys_1277	-10,43		decoys_320	-10,28		decoys_541	-9,08		decoys_267	-9,51	
BPP5_4	-10,4	BPP	decoys_229	-10,24		decoys_1699	-9,06		decoys_1545	-9,51	
ligand_34	-10,35	Carboxilico	ligand_34	-10,21	Carboxilico	decoys_473	-9,05		decoys_518	-9,5	

Por outro lado, utilizando o HTP SurflexDock e o *ensemble* gerado a partir da simulação molecular de 5ns do domínio C-terminal de hACE aponta pelo menos uma conformação, do *cluster* t=2182 ps, conseguiu obter um enriquecimento dos ligantes melhor do que a conformação controle (Figura 16). Além disso, no enriquecimento inicial dos 15 primeiros compostos, este *cluster* obteve a maior mudança de perfil dos *quimiotipos* com a inclusão de 4 compostos carboxílicos e 1 composto do tipo BPP, o que equivale um aumento de aproximadamente 83% de ligantes nesta fração do enriquecimento (Tabela IV). No entanto, não foi observada alteração significativa dos tipos de ligantes.

No mesmo experimento repetido para o domínio N-Terminal revelou uma melhora de cerca de 4% no enriquecimento dos ligantes entre o pior e melhor resultado de AUC (Figura 17:A), somado a um incremento de ligantes do tipo fosfínico no enriquecimento inicial dos primeiros 15 compostos (Tabela V). Porém, assim como no experimento com o domínio C-terminal não foi observado nessa fração grandes alterações dos tipos de ligantes enriquecidos.

Por outro lado ao examinar o enriquecimento inicial de diferentes frações de compostos observa-se que uma conformação obtida da simulação de 2ns utilizando o domínio C-Terminal conseguiu enriquecer uma maior quantidade de ligantes que o controle até os primeiros 15 compostos (0.8% dos compostos; Tabela VI). Por outro lado, uma conformação obtida da simulação de 5 nanosegundos do mesmo domínio permitiu o enriquecimento inicial de uma maior quantidade de ligantes em relação ao controle até os primeiros 92 compostos (5% dos compostos; Tabela VII). Já na unidade N-Terminal, uma conformação obtida da simulação de 2ns só conseguiu enriquecer mais ligantes do que controle até os primeiros 10 compostos (0.55% dos compostos; Tabela VI), enquanto a conformação do *cluster* t=4460 obtida na simulação de 5 ns conseguiu um melhor enriquecimento inicial até os primeiros 369 compostos (20% dos compostos; Tabela VII).

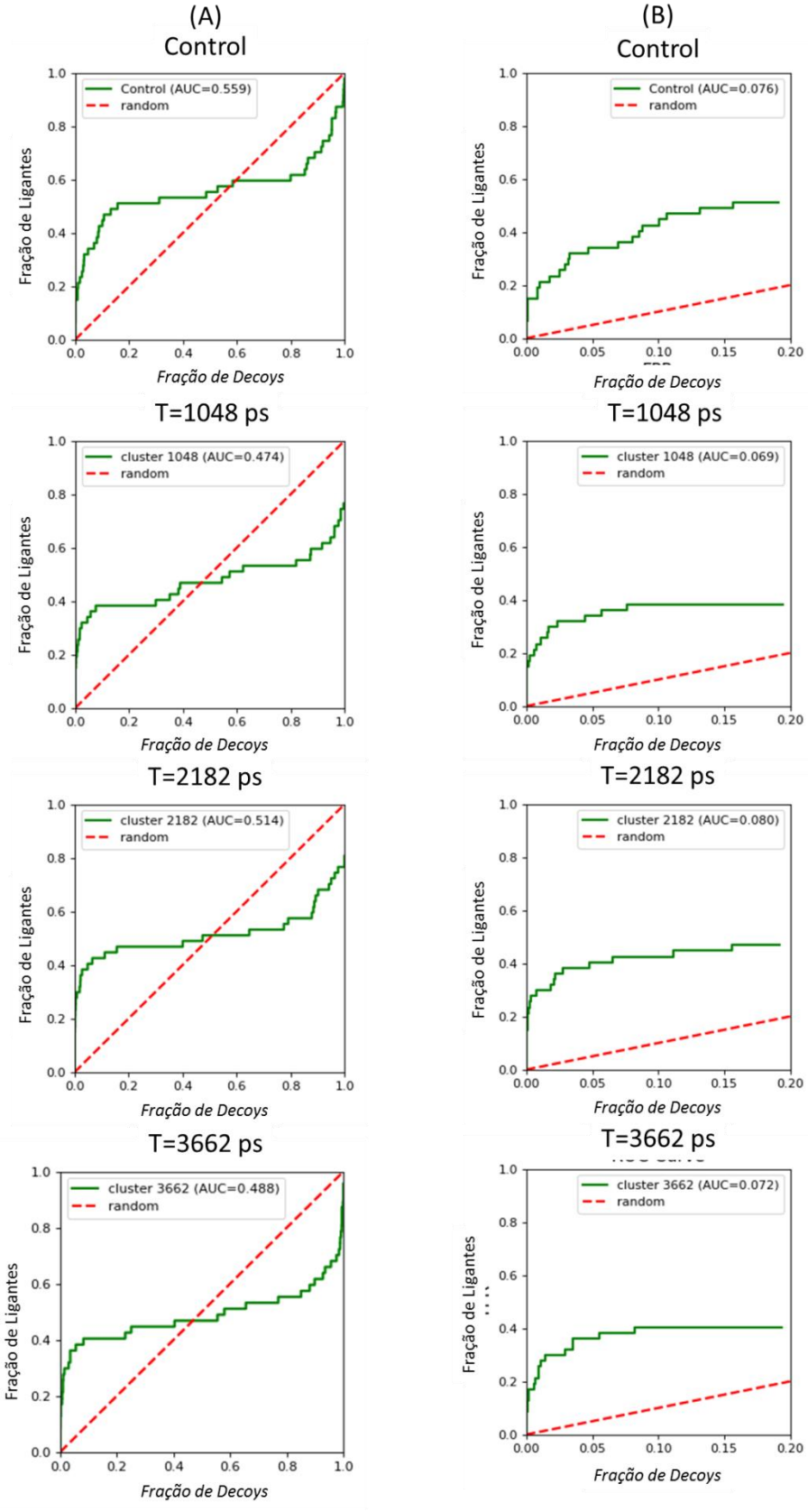


Figura 16 - Curvas ROC do domínio C-Terminal da hACE (hACEc) para cada conformação representativa da simulação de 5ns. (A) Curva ROC referente a 100% da biblioteca testada. (B) Curva ROC referente aos primeiros 20% da biblioteca testada.

Tabela IV– Enriquecimento inicial dos 15 melhores *dockings* utilizando conformações obtidas da simulação de 5ns do domínio C-Terminal da hACE.

Control			*Cluster* t=1048			*Cluster* t=2182			*Cluster* t=3662		
Molécula	**ΔG (kcal/mol)**	**Quimiotipo**	**Molécula**	**ΔG (kcal/mol)**	**Quimiotipo**	**Molécula**	**ΔG (kcal/mol)**	**Quimiotipo**	**Molécula**	**ΔG (kcal/mol)**	**Quimiotipo**
BPP5_3	-11,05	BPP	BPP5_3	-11,33	BPP	BPP5a	-12,05	BPP	BPP3	-11,12	BPP
RX3	-10,5	Fosfínico	BPP3	-10,63	BPP	RX3	-11,14	Fosfínico	ligand_15	-10,34	Carboxílico
ligand_15	-10,49	Carboxílico	ligand_15	-10,43	Carboxílico	BPP3	-11,05	BPP	BPP5a	-10,16	BPP
decoys_600	-10,49		BPP5_4	-10,04	BPP	ligand_15	-10,74	Carboxílico	BPP5_3	-10	BPP
BPP3	-10,47	BPP	ligand_24	-9,92	Carboxílico	ligand_24	-10,69	Carboxílico	decoys_554	-9,46	
ligand_33	-10,47	Carboxílico	RX3	-9,71		BPP5_3	-10,46	BPP	RX3	-9,46	Fosfínico
BPP5_2	-10,28	BPP	ligand_31	-9,63	Carboxílico	ligand_34	-9,91	Carboxílico	ligand_23	-9,27	Fosfínico
BPP5a	-10,28	BPP	decoys_53	-9,6		decoys_564	-9,61		decoys_600	-9,19	
decoys_156	-10,27		decoys_1699	-9,56		ligand_42	-9,6	Carboxílico	decoys_534	-9,1	
decoys_534	-10,24		decoys_564	-9,51		decoys_156	-9,57		ligand_42	-8,95	Carboxílico
decoys_1281	-10,21		BPP5a	-9,5	BPP	ligand_37	-9,35	Carboxílico	BPP4	-8,93	BPP
decoys_1321	-10,2		decoys_939	-9,49		BPP5_2	-9,34	BPP	decoys_815	-8,87	
decoys_1205	-10,1		BPP5_2	-9,29	BPP	decoys_1699	-9,32		decoys_522	-8,83	
decoys_554	-9,92		decoys_165	-9,27		ligand_33	-9,32	Carboxílico	decoys_1773	-8,8	

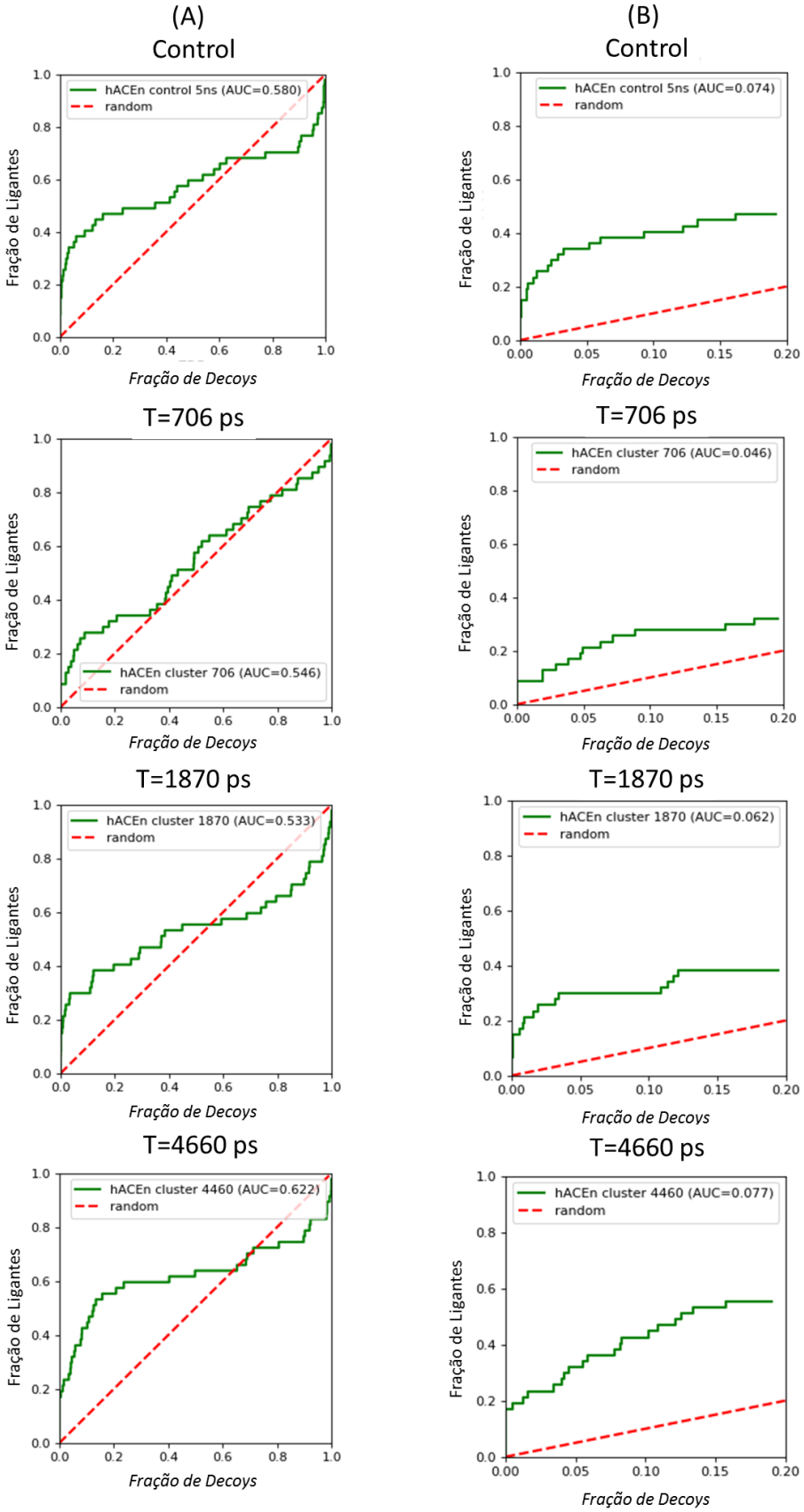


Figura 17 – Curvas ROC do domínio N-Terminal da hACE (hACEn) para cada conformação representativa de uma simulação de 5ns. (A) Curva ROC referente a 100% da biblioteca testada. (B) Curva ROC referente aos primeiros 20% da biblioteca testada.

Tabela V– Enriquecimento inicial dos 15 melhores *dockings* utilizando conformações obtidas da simulação de 5ns do domínio N-Terminal da hACE

Control			***Cluster* t=706**			***Cluster* t=1870**			***Cluster* t=4460**		
Molécula	**ΔG (kcal/mol)**	***Chemotype***	**Molécula**	**ΔG (kcal/mol)**	***Chemotype***	**Molécula**	**ΔG (kcal/mol)**	***Chemotype***	**Molécula**	**ΔG (kcal/mol)**	***Chemotype***
BPP3	-12,33	BPP	BPP3	-10,75	BPP	RX3	-11,25	Fosfinico	BPP5_3	-11,59	BPP
ligand_15	-11,33	Carboxilico	RX3	-10,6	Fosfinico	ligand_15	-11,1	Carboxilico	RX3	-11,4	Fosfinico
RX3	-10,99	Fosfinico	ligand_15	-10,42	Carboxilico	BPP5_3	-10,93	BPP	BPP3	-10,51	BPP
ligand_24	-10,8	Carboxilico	ligand_27	-10	Fosfinico	decoys_338	-9,86		ligand_42	-10,44	Carboxilico
decoys_1574	-10,74		decoys_1270	-9,98		BPP5_2	-9,85	BPP	BPP5a	-10,33	BPP
decoys_1398	-10,7		decoys_1716	-9,88		decoys_599	-9,84		BPP5_2	-10,3	BPP
BPP4	-10,64	BPP	decoys_608	-9,82		BPP3	-9,76	BPP	ligand_15	-10,16	Carboxilico
BPP5_3	-10,59	BPP	decoys_320	-9,79		ligand_24	-9,71	Carboxilico	ligand_23	-9,9	Fosfinico
ligand_34	-10,5	Carboxilico	decoys_284	-9,75		ligand_23	-9,69	Fosfinico	decoys_1398	-9,8	
decoys_1269	-10,5		decoys_1398	-9,71		decoys_1773	-9,61		decoys_1773	-9,79	
decoys_1270	-10,39		decoys_913	-9,69		decoys_913	-9,53		decoys_338	-9,75	
decoys_284	-10,38		decoys_286	-9,58		decoys_619	-9,5		decoys_619	-9,74	
decoys_1152	-10,37		decoys_619	-9,58		decoys_1270	-9,46		decoys_310	-9,68	
decoys_1320	-10,34		decoys_310	-9,53		decoys_1276	-9,45		decoys_534	-9,65	
decoys_848	-10,33		decoys_613	-9,49		decoys_705	-9,42		decoys_531	-9,64	

Tabela VI - Quantidade de ligantes enriquecidos por conformação do *ensemble* obtido da simulação de 2 ns

2ns of receptor structural sampling					
hACEc					
Ranked experimental subset	**0.55%** (first 10)	**0.8%** (first 15)	**5%** (first 92)	**10%** (first 184)	**20%** (first 369)
Original structure (Control)	6	8	15	20*	23*
Cluster I (718 ps) **X**	8*	11*	15*	17	19
Cluster II (1100 ps)	7	8	14	17	19
Cluster III (1728 ps)	5	6	13	17	21
hACEn					
Ranked experimental subset	**0.55%** (first 10)	**0.8%** (first 15)	**5%** (first 92)	**010%** (first 184)	**20%** (first 369)
Original structure (Control) **X**	7	9*	15*	21*	22*
Cluster I (314 ps)	8*	9	13	15	17
Cluster II (592 ps)	6	7	14	17	20
Cluster III (956 ps)	5	6	12	16	18

Tabela VII - Quantidade de ligantes enriquecidos por conformação do *ensemble* obtido da simulação de 5 ns

5ns of receptor structural sampling					
hACE-C					
Ranked experimental subset	**0.55%** (first 10)	**0.8%** (first 15)	**5%** (first 92)	**10%** (first 184)	**20%** (first 369)
Original structure (Control)	8	8	16	21*	23*
Cluster I (1048 ps)	7	9	16	18	18
Cluster II (2188 ps) **X**	8*	11*	18*	19	20
Cluster III (3662 ps)	7	8	16	18	18
hACE-N					
Ranked experimental subset	**0.55%** (first 10)	**0.8%** (first 15)	**5%** (first 92)	**010%** (first 184)	**20%** (first 369)
Original structure (Control)	4	6	11	16	17
Cluster I (706 ps)	4	4	9	14	16
Cluster II (1870 ps)	7	7	14	14	18
Cluster III (4460 ps) **X**	8*	8*	14*	19*	25*

Nesse sentido, o desempenho do *virtual screening* utilizando os *ensembles* de conformações obtidas das simulações de 2ns foram inferiores se comparados com os mesmos experimentos realizados com conformações obtidas das simulações de 5ns. Neste caso, o maior tempo de simulação permite que os receptores experimentem uma maior exploratória conformacional em contato com o solvente, assim obtendo uma superfície de contato do sítio ativo mais diversa se comparada as obtidas na simulação de 2ns. Comparativamente, a superfície de contato das conformações dos *ensembles* obtidas das simulações de 2ns aparentam ser mais similares aos cristais (conformação controle) do que as superfícies de contato dos *ensembles* obtidos das simulações de 5ns, (Figura 18).

Em relação aos tipos de ligante utilizados este estudo de caso, o HTP SurflexDock conseguiu classificar e enriquecer de forma eficiente ligantes do tipo Carboxílicos, Peptídicos e Fosfínicos utilizando *ensembles* obtidos da simulação de 5ns. Entretanto, observando-se os gráficos ROC, pode-se perceber que existe um grupo de ligantes que possuem o enriquecimento inferior aos compostos *decoys*. Este grupo é constituído por compostos predominantemente de ligantes do tipo tiól (Anexo I). Acredita-se que o problema está relacionado ao cálculo da energia de interação do radical tiol destes ligantes com o íon zinco, realizado pelo o AutoDock 4.2. Não se removeu este *quimiotipo* de ligantes nestes experimentos com objetivo de demonstrar as dificuldades em se realizar *docking* com metaloprotease do tipo ACE e ainda da dificuldade de alguns metais serem corretamente interpretados pelo AutoDock 4.2 na composição de ligantes.

5.5.4 O uso de *ensemble docking* no *virtual screening* influi no resultado experimental e no enriquecimento de ligantes.

Conforme já demonstrado em outros estudos com *virtual screening*, a utilização de *ensemble docking* possui a capacidade de produzir um maior enriquecimento de ligantes (ELLINGSON et al., 2015), assim como melhora a diversidade de compostos que são enriquecidos durante o *docking* (CAVASOTTO et al., 2005; CAVASOTTO e ABAGYAN, 2004; FERRARI et al., 2004). Do mesmo modo, os nossos resultados utilizando a simulação molecular de 5ns para construir o *ensemble*, demonstram claramente que para ambos domínios da hACE o HTP SurflexDock foi capaz de favorecer, em alguma das estruturas obtidas no *ensemble*, um maior enriquecimento inicial de ligantes.

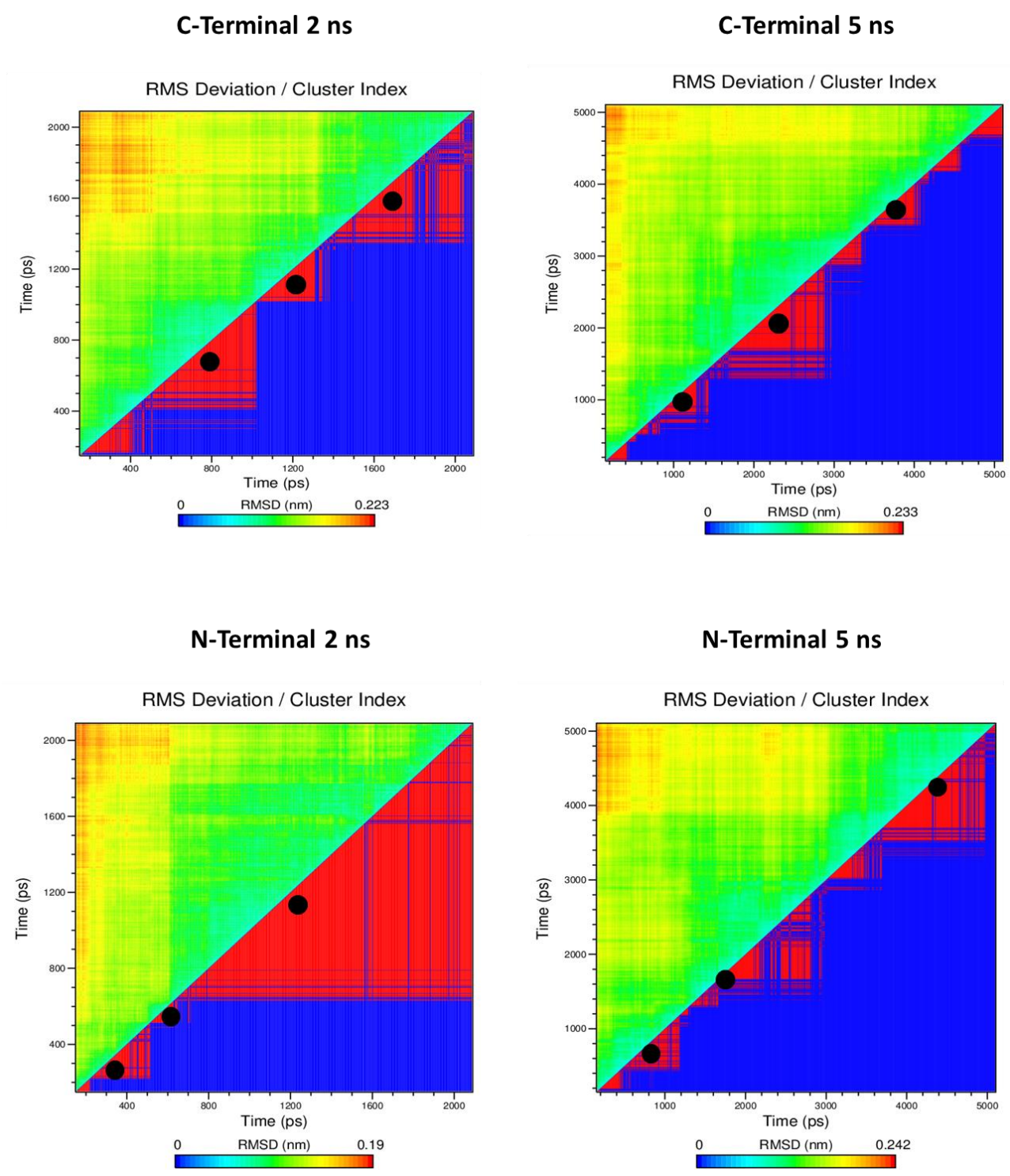


Figura 18 - Gráficos do RMSD em função do tempo das simulações de 2ns e 5ns do sítio ativo dos domínios C-terminal e N-terminal da hACE. Os círculos pretos indicam os clusters dos quais foram extraídos as conformações mais representativa de cada experimento

Isto e claramente observado se comparados em relação à conformação original do cristal (conformação controle), e em todos os casos até os primeiros 5% (Tabela VI).

Além disso, no domínio C-terminal houve uma mudança clara no perfil de enriquecimento inicial do tipo de ligantes como descrito na seção anterior. Estes resultados apontam que, mesmo com abordagem simplificada e uso discreto de recursos computacionais, a abordagem metodológica implementada no *pipeline* do HTP SurflexDock entrega ao usuário o resultado inicialmente pretendido.

5.5.5 Houve diferenças no enriquecimento dos domínios N-terminal e C-terminal de hACE?

Nos gráficos de enriquecimento e nas tabelas de enriquecimento inicial obtidos no HTP SurflexDock para domínios N- e C-terminal de hACE observamos que possuem diferentes perfis de enriquecimento de ligantes.

No domínio C-terminal derivado da estrutura cristalográfica 1O86 que foi determinada com um ligante do tipo carboxílico (Lisinopril), observamos uma maior dificuldade em alterar o padrão de enriquecimento inicial com inibidores de estrutura diferente. Já o domínio N-terminal foi cristalizado com um ligante fosfínico (pdb 5amb**,** ligante P6G). O parâmetro AUC obtido no experimento do domínio N-terminal é cerca de 6% maior do que o obtido para o domínio C-terminal. Além disto, o domínio N-terminal da hACE teve maior facilidade em enriquecer diferentes perfis de ligantes em suas conformações do *ensemble* (Tabela V).

Estes resultados apontam para variações importantes de enriquecimento nos resultados obtidos para diferentes domínios, mesmo com atividade catalítica parecida e identidade de sequencia de 60%. Isto reforça o caráter caso-dependente deste tipo de resultado como já apontado por outros grupos na literatura (ELLINGSON e colab, 2015).

Capitulo 6: Conclusões

1- Rigel fornece um ambiente de desenvolvimento para *pipelines* de biologia computacional como modelagem molecular, *docking* e *virtual screening*.

2- O desenvolvimento do HTP SurflexDock e do MDR SurflexDock foi facilitado pela plataforma Rigel e a biblioteca DynDock. O uso destes dois softwares e o uso da interface web permitiram que a interface de usuário destes *pipelines* fosse simples e o uso transparente para o usuário final.

3- Utilizando os domínios N-terminal e C-terminal da enzima conversora da angiotensina I (hACE) e uma biblioteca contendo 46 ligantes e 1797 decoys, os resultados dos testes apontam que o HTP SurflexDock consegue apresentar ao usuário um perfil de enriquecimento de ligantes muito mais abrangente do que se fosse feito somente com a conformação obtida do cristal.

4- No HTP SurflexDock os experimentos que utilizaram o *ensemble* obtido da simulação de 5ns obtiveram um melhor desempenho no enriquecimento inicial dos ligantes do que os experimentos utilizando conformações do *ensemble* obtidos da simulação de 2ns.

5- Entre os ligantes utilizados no experimento com o HTP SurflexDock, o tipo que obteve melhor desempenho foram os do tipo BPP (peptídeos potenciadores de bradicinina), onde a maioria dos inibidores utilizados foram enriquecidos entre as 15 primeiras moléculas de grande parte das conformações dos *ensembles*.

6- Os ligantes do tipo Tiol obtiveram um péssimo desempenho em todos os experimentos de *virtual screening* utilizando o HTP SurflexDock. Creditamos isto a uma dificuldade do software AutoDock 4.2 em calcular corretamente a energia de interação entre o radical Tiol e o íon zinco.

Referências

ABASCAL, Jose L F e VEGA, Carlos. **A general purpose model for the condensed phases of water: TIP4P/2005**. The Journal of chemical physics, v. 123, n. 23, p. 234505, 2005.

ABRAHAM, Mark James e colab. **GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers**. SoftwareX, v. 1, p. 19–25, 2015.

AKIF, Mohd e colab. **High-resolution crystal structures of Drosophila melanogaster angiotensin-converting enzyme in complex with novel inhibitors and antihypertensive drugs**. Journal of molecular biology, v. 400, n. 3, p. 502–517, 2010.

ALLEN, Michael P e OTHERS. **Introduction to molecular dynamics simulation**. Computational soft matter: from synthetic polymers to proteins, v. 23, p. 1–28, 2004.

ALMEIDA FILHO, J. L. DE; REAL TAMARIZ, A. DEL; FERNANDEZ, J. H. **AutoModel: A Client-Server Tool for Intuitive and Interactive Homology Modeling of Protein-Ligand Complexes.** ALVES, R. (Org.). In: Advances in Bioinformatics and Computational Biology. Cham: Springer International Publishing, 2018. ISBN: 978-3-030-01722-4.

AMARO, Rommie E e colab. **Ensemble Docking in Drug Discovery**. Biophysical Journal, 2018.

ANDERSEN, Hans C. **Molecular dynamics simulations at constant pressure and/or temperature**. The Journal of chemical physics, v. 72, n. 4, p. 2384–2393, 1980.

ANTHONY, Colin S e colab. **The N Domain of Human Angiotensin-I-converting Enzyme THE ROLE OF N-GLYCOSYLATION AND THE CRYSTAL STRUCTURE IN COMPLEX WITH AN N DOMAIN-SPECIFIC PHOSPHINIC INHIBITOR, RXP407**. Journal of Biological Chemistry, v. 285, n. 46, p. 35685–35693, 2010.

ANTUNES, Dinler A e DEVAURS, Didier e KAVRAKI, Lydia E. **Understanding the challenges of protein flexibility in drug design**. Expert Opinion on Drug Discovery, v. 10, n. 12, p. 1301–1313, 2015. Disponível em: <http://www.tandfonline.com/doi/full/10.1517/17460441.2015.1094458>.

BATEMAN, Brian T e colab. **Angiotensin-converting enzyme inhibitors and the risk of congenital malformations**. Obstetrics and gynecology, v. 129, n. 1, p. 174, 2017.

BERENDSEN, Herman J C e colab. **Molecular dynamics with coupling to an external bath**. The Journal of chemical physics, v. 81, n. 8, p. 3684–3690, 1984.

BERMAN, Helen M e colab. **The Protein Data Bank www.rcsb.org**. Nucleic acids research, v. 28, n. 1, p. 235–242, 2000.

BIASINI, Marco e colab. **SWISS-MODEL: modelling protein tertiary and quaternary structure using evolutionary information**. Nucleic acids research, p. gku340, 2014.

BOEHM, M. **Virtual screening: Principles, challenges and practical guidelines**. . [S.l.]: Wiley: Nova York. , 2011

BRAUN, Efrem e MOOSAVI, Seyed Mohamad e SMIT, Berend. **Anomalous effects of velocity rescaling algorithms: The flying ice cube effect revisited**. Journal of chemical theory and computation, v. 14, n. 10, p. 5262–5272, 2018.

BROOKS, Bernard R e colab. **CHARMM: the biomolecular simulation program**. Journal of computational chemistry, v. 30, n. 10, p. 1545–1614, 2009.

BRYAN, Jenny. **From snake venom to ACE inhibitor--The discovery and rise of captopril**. Pharmaceutical Journal, v. 282, n. 7548, p. 455, 2009.

BUCK, Ian e colab. **CAMPAIGN: an open-source library of GPU-accelerated data clustering algorithms**. Bioinformatics, v. 27, n. 3, p. S2, 2014.

BURNIER, M. e BRUNNER, H. R. **Angiotensin II receptor antagonists**. Lancet, v. 355, n. 9204, p. 637–645, 2000.

BUSSI, Giovanni e DONADIO, Davide e PARRINELLO, Michele. **Canonical sampling through velocity rescaling**. The Journal of chemical physics, v. 126, n. 1, p. 14101, 2007.

CAMARGO, Antonio C M e colab. **Bradykinin-potentiating peptides: beyond captopril**. Toxicon, v. 59, n. 4, p. 516–523, 2012.

CARTER, Jane V e colab. **ROC-ing along: Evaluation and interpretation of receiver operating characteristic curves**. Surgery, v. 159, n. 6, p. 1638–1645, 2016.

CAVASOTTO, Claudio N e ABAGYAN, Ruben A. **Protein flexibility in ligand docking and virtual screening to protein kinases**. Journal of molecular biology, v. 337, n. 1, p. 209–225, 2004.

CAVASOTTO, Claudio N e ORRY, Andrew J W e ABAGYAN, Ruben A. **The challenge of considering receptor flexibility in ligand docking and virtual screening**. Current Computer-Aided Drug Design, v. 1, n. 4, p. 423–440, 2005.

CAVASOTTO, Claudio N e PHATAK, Sharangdhar S. **Homology modeling in drug discovery: current trends and applications**. Drug discovery today, v. 14, n. 13–14, p. 676–683, 2009.

CHANG, Jia-Ming e DI TOMMASO, Paolo e NOTREDAME, Cedric. **TCS: a new multiple sequence alignment reliability measure to estimate alignment accuracy and improve phylogenetic tree reconstruction**. Molecular biology and evolution, p. msu117, 2014.

CHAUDHURY, Sidhartha e GRAY, Jeffrey J. **Conformer selection and induced fit in flexible backbone protein--protein docking using computational and NMR ensembles**. Journal of molecular biology, v. 381, n. 4, p. 1068–1087, 2008.

CUSHMAN, David W e ONDETTI, Miguel A. **Design of angiotensin converting enzyme inhibitors**. Nature medicine, v. 5, n. 10, p. 1110, 1999.

D'ANGELO, Gianni e RAMPONE, Salvatore. **Towards a HPC-oriented parallel**

implementation of a learning algorithm for bioinformatics applications. BMC bioinformatics, v. 15, n. Suppl 5, p. S2, 2014.

DAURA, Xavier e colab. **Peptide folding: when simulation meets experiment**. Angewandte Chemie International Edition, v. 38, n. 1–2, p. 236–240, 1999.

DE OLIVEIRA, Saulo H P e SHI, Jiye e DEANE, Charlotte M. **Building a better fragment library for de novo protein structure prediction**. PloS one, v. 10, n. 4, p. e0123998, 2015.

DE VIVO, Marco e colab. **Role of molecular dynamics and related methods in drug discovery**. Journal of medicinal chemistry, v. 59, n. 9, p. 4035–4061, 2016.

DORN, Márcio e colab. **Three-dimensional protein structure prediction: Methods and computational strategies**. Computational Biology and Chemistry, v. 53, p. 251–276, 2014.

DÖRR, Mark e colab. **Fully automatized high-throughput enzyme library screening using a robotic platform**. Biotechnology and bioengineering, v. 113, n. 7, p. 1421–1432, 2016.

DOS SANTOS MUNIZ, Heloisa e NASCIMENTO, Alessandro S. **Ligand-and receptor-based docking with LiBELa**. Journal of computer-aided molecular design, v. 29, n. 8, p. 713–723, 2015.

DU, Q e XIE, N e HUANG, R. **Recent Development of Peptide Drugs and Advance on Theory and Methodology of Peptide Inhibitor Design.** Medicinal chemistry (Shariqah (United Arab Emirates)), 2014.

E SILVA, M Rocha e BERALDO, Wilson T e ROSENFELD, G. **Bradykinin, a hypotensive and smooth muscle stimulating factor released from plasma globulin by snake venoms and by trypsin**. American Journal of Physiology-Legacy Content, v. 156, n. 2, p. 261–273, 1949.

EDGAR, Robert C. **MUSCLE: a multiple sequence alignment method with reduced time and space complexity**. BMC bioinformatics, v. 5, n. 1, p. 113, 2004.

ESWAR, Narayanan e colab. **Comparative protein structure modeling using Modeller**. Current protocols in bioinformatics, p. 5–6, 2006.

EVANS, Marie e colab. **Angiotensin-converting enzyme inhibitors and angiotensin receptor blockers in myocardial infarction patients with renal dysfunction**. Journal of the American College of Cardiology, v. 67, n. 14, p. 1687–1697, 2016.

EWING, Todd J A e colab. **DOCK 4.0: search strategies for automated molecular docking of flexible molecule databases**. Journal of computer-aided molecular design, v. 15, n. 5, p. 411–428, 2001.

FERNANDEZ, Jorge Hernandez e colab. **Using bradykinin-potentiating peptide structures to develop new antihypertensive drugs**. Genet Mol Res, v. 3, n. 4, p. 554–563, 2004.

FERRARI, Anna Maria e colab. **Soft docking and multiple receptor conformations in virtual screening**. Journal of medicinal chemistry, v. 47, n. 21, p. 5076–5084, 2004.

FERREIRA, Sergio H e BARTELT, Diana C e GREENE, Lewis J. **Isolation of bradykinin-potentiating peptides from Bothrops jararaca venom**. Biochemistry, v. 9, n. 13, p. 2583–2593, 1970.

FLETCHER, Roger. **Practical methods of optimization**. [S.l.]: John Wiley & Sons, 2013.

FRADERA, Xavier e BABAOGLU, Kerim. **Overview of methods and strategies for conducting virtual small molecule screening**. Current protocols in chemical biology, v. 9, n. 3, p. 196–212, 2018.

FROHMBERG, W e colab. **G-PAS 2.0--an improved version of protein alignment tool with an efficient backtracking routine on multiple GPUs**. Bulletin of the Polish Academy of Sciences: Technical Sciences, v. 60, n. 3, p. 491–494, 2012.

GALANIS, Athanassios S e colab. **Structural studies through 1 H NMR spectroscopy of somatic angiotensin converting enzyme (ACE) active sites and comparison with testis ACE crystal structure**. Collection of Czechoslovak Chemical Communications, v. 6, p. 29–31, 2016.

GANESAN, Aravindhan e COOTE, Michelle L e BARAKAT, Khaled. **Molecular dynamics-driven drug discovery: leaping forward with confidence**. Drug discovery today, v. 22, n. 2, p. 249–269, 2017.

GEORGE, Boby e WILLIAMS, Laurie. **A structured experiment of test-driven development**. Information and software Technology, v. 46, n. 5, p. 337–342, 2004.

GIANZO, Marta e colab. **Human sperm testicular angiotensin-converting enzyme helps determine human embryo quality**. Asian journal of andrology, v. 20, n. 5, p. 498, 2018.

GILISSEN, Christian e colab. **Disease gene identification strategies for exome sequencing**. European Journal of Human Genetics, v. 20, n. 5, p. 490–497, 2012.

GOPAL, Srinivasa M e KUHN, Alexander B e SCHÄFER, Lars V. **Systematic evaluation of bundled SPC water for biomolecular simulations**. Physical Chemistry Chemical Physics, v. 17, n. 13, p. 8393–8406, 2015.

GOTOH, Osamu. **An improved algorithm for matching biological sequences**. Journal of molecular biology, v. 162, n. 3, p. 705–708, 1982.

GRAU, Jan e GROSSE, Ivo e KEILWAGEN, Jens. **PRROC: computing and visualizing precision-recall and receiver operating characteristic curves in R**. Bioinformatics, v. 31, n. 15, p. 2595–2597, 2015.

GUAN, Shan-shan e colab. **Insight into the interactive residues between two domains of human somatic Angiotensin-converting enzyme and Angiotensin II by MM-PBSA calculation and steered molecular dynamics simulation**. Journal of Biomolecular Structure and Dynamics, v. 34, n. 1, p. 15–28, 2016.

GUNARATHNE, Thilina e colab. **Cloud computing paradigms for pleasingly parallel biomedical applications**. Concurrency and Computation: Practice and Experience, v. 23, n. 17, p. 2338–2354, 2011.

GUTHRIE, Robert. **Fosinopril: an overview**. The American journal of cardiology, v. 72, n. 20, p. H22--H24, 1993.

HAILE, James M. **Molecular dynamics simulation**. Elementary methods, 1992.

HAIRER, Ernst e LUBICH, Christian e WANNER, Gerhard. **Geometric numerical integration illustrated by the Störmer--Verlet method**. Acta numerica, v. 12, p. 399–450, 2003.

HASSAN BAIG, Mohammad e colab. **Computer aided drug design: success and limitations**. Current pharmaceutical design, v. 22, n. 5, p. 572–581, 2016.

HOOVER, William G. **Canonical dynamics: Equilibrium phase-space distributions**. Physical review A, v. 31, n. 3, p. 1695, 1985.

HWANG, Howook e VREVEN, Thom e WENG, Zhiping. **Binding interface prediction by combining protein--protein docking results**. Proteins: Structure, Function, and Bioinformatics, v. 82, n. 1, p. 57–66, 2014.

IANZER, Danielle e colab. **BPP-5a produces a potent and long-lasting NO-dependent antihypertensive effect**. Therapeutic advances in cardiovascular disease, v. 5, n. 6, p. 281–295, 2011.

JAIN, Ajay N. **Bias, reporting, and sharing: computational evaluations of docking methods**. Journal of computer-aided molecular design, v. 22, n. 3–4, p. 201–212, 2008.

JALKUTE, Chidambar Balbhim e colab. **Molecular dynamics simulation and molecular docking studies of angiotensin converting enzyme with inhibitor lisinopril and amyloid beta peptide**. Protein Journal, v. 32, n. 5, p. 356–364, 2013.

JIANG, Zhenyan e colab. **Insight into the binding of ACE-inhibitory peptides to angiotensin-converting enzyme: a molecular simulation**. Molecular Simulation, v. 45, n. 3, p. 215–222, 2019.

JORGENSEN, William L. **Transferable intermolecular potential functions for water, alcohols, and ethers. Application to liquid water**. J. Am. Chem. Soc.;(United States), v. 103, n. 2, 1981.

JORGENSEN, William L e MAXWELL, David S e TIRADO-RIVES, Julian. **Development and testing of the OPLS all-atom force field on conformational energetics and properties of organic liquids**. Journal of the American Chemical Society, v. 118, n. 45, p. 11225–11236, 1996.

KANDATHIL, Shaun M e colab. **Improved fragment-based protein structure prediction by redesign of search heuristics**. Scientific reports, v. 8, n. 1, p. 13694, 2018.

KARPLUS, Martin e PETSKO, Gregory A. **Molecular dynamics simulations in biology**. Nature, v. 347, n. 6294, p. 631, 1990.

KELLEY, Lawrence A e colab. **The Phyre2 web portal for protein modeling, prediction and analysis**. Nature protocols, v. 10, n. 6, p. 845–858, 2015.

KIM, David E e CHIVIAN, Dylan e BAKER, David. **Protein structure prediction**

and analysis using the Robetta server. Nucleic acids research, v. 32, n. suppl_2, p. W526--W531, 2004.

KOZA, John R e colab. **Genetic programming III: Darwinian invention and problem solving**. [S.l.]: Morgan Kaufmann, 1999. v. 3.

KRIEGER, Elmar e NABUURS, Sander B e VRIEND, Gert. **Homology modeling**. Methods of biochemical analysis, v. 44, p. 509–524, 2003.

LAMBERT, Christophe e colab. **ESyPred3D: Prediction of proteins 3D structures**. Bioinformatics, v. 18, n. 9, p. 1250–1256, 2002.

LANCASTER, Simon G e TODD, Peter A. **Lisinopril**. Drugs, v. 35, n. 6, p. 646–669, 1988.

LARMUTH, Kate M e colab. **Kinetic and structural characterization of amyloid-β peptide hydrolysis by human angiotensin-1-converting enzyme**. The FEBS journal, v. 283, n. 6, p. 1060–1076, 2016.

LEE, Jooyoung e FREDDOLINO, Peter L e ZHANG, Yang. Ab initio protein structure prediction. From protein structure to function with bioinformatics. [S.l.]: Springer, 2017. p. 3–35.

LEIMKUHLER, Ben e MATTHEWS, Charles. **Molecular Dynamics.** [S.l.]: Springer, 2016.

LI, Qingliang e SHAH, Salim. Structure-based virtual screening. Protein Bioinformatics. [S.l.]: Springer, 2017. p. 111–124.

MACALINO, Stephani Joy Y e colab. **Role of computer-aided drug design in modern drug discovery**. Archives of pharmacal research, v. 38, n. 9, p. 1686–1701, 2015.

MANDAL, Soma e MOUDGIL, Mee'nal e MANDAL, Sanat K. **Rational drug design**. European journal of pharmacology, v. 625, n. 1–3, p. 90–100, 2009.

MASUYER, Geoffrey e colab. **Crystal structures of highly specific phosphinic tripeptide enantiomers in complex with the angiotensin-I converting enzyme**. The FEBS journal, v. 281, n. 3, p. 943–956, 2014.

MASUYER, Geoffrey e colab. **Molecular recognition and regulation of human angiotensin-I converting enzyme (ACE) activity by natural inhibitory peptides**. Scientific Reports, v. 2, p. 1–10, 2012.

MAZUR, Alexy K. **Common molecular dynamics algorithms revisited: accuracy and optimal time steps of Störmer--Leapfrog integrators**. Journal of Computational Physics, v. 136, n. 2, p. 354–365, 1997.

MENARD, Joel e PATCHETT, Arthur A. **Angiotensin-converting enzyme inhibitors**. Advances in protein chemistry, v. 56, p. 13–75, 2001.

MIHĂŞAN, Marius. **What in silico molecular docking can do for the `bench-working biologists'**. Journal of Biosciences, v. 37, n. 1, p. 1089–1095, 2012. Disponível em: <https://doi.org/10.1007/s12038-012-9273-8>.

MOROY, Gautier e colab. **Sampling of conformational ensemble for virtual screening using molecular dynamics simulations and normal mode analysis**. Future medicinal chemistry, v. 7, n. 17, p. 2317–2331, 2015.

MORRIS, Garrett M e colab. **AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility**. Journal of computational chemistry, v. 30, n. 16, p. 2785–2791, 2009.

MOULT, John e colab. **Critical assessment of methods of protein structure prediction (CASP)—round x**. Proteins: Structure, Function, and Bioinformatics, v. 82, n. S2, p. 1–6, 2014.

MU, Xia e ZHANG, Chunchun e XU, Dingguo. **QM/MM investigation of the catalytic mechanism of angiotensin-converting enzyme**. Journal of molecular modeling, v. 22, n. 6, p. 132, 2016.

MURDOCH, David e MCTAVISH, Donna. **Fosinopril**. Drugs, v. 43, n. 1, p. 123–140, 1992.

MYSINGER, Michael M e colab. **Directory of useful decoys, enhanced (DUD-E): better ligands and decoys for better benchmarking**. Journal of medicinal chemistry, v. 55, n. 14, p. 6582–6594, 2012.

NAMBA, Adriana Mieco e DA SILVA, V B e DA SILVA, CHTP. **Dinâmica molecular: teoria e aplicações em planejamento de fármacos**. Eclética Química, v. 33, n. 4, 2008.

NATESH, Ramanathan e colab. **Crystal structure of the human enzyme – lisinopril complex**. Nature, v. 421, n. 1995, p. 1427–1429, 2003.

NATESH, Ramanathan e colab. **Structural details on the binding of antihypertensive drugs captopril and enalaprilat to human testicular angiotensin I-converting enzyme**. Biochemistry, v. 43, n. 27, p. 8718–8724, 2004.

NESTER, Karina e GAWEDA, Karolina e PLAZINSKI, Wojciech. **A GROMOS Force Field for Furanose-Based Carbohydrates**. Journal of chemical theory and computation, v. 15, n. 2, p. 1168–1186, 2019.

NORGAN, Andrew P e colab. **Multilevel parallelization of AutoDock 4.2**. Journal of cheminformatics, v. 3, n. 1, p. 12, 2011.

NOSÉ, Shuichi. **A molecular dynamics method for simulations in the canonical ensemble**. Molecular physics, v. 52, n. 2, p. 255–268, 1984.

OECHEM, T K. **OpenEye Scientific Software**. Inc., Santa Fe, NM, USA, 2012.

OOSTENBRINK, Chris e colab. **Validation of the 53A6 GROMOS force field**. European Biophysics Journal, v. 34, n. 4, p. 273–284, 2005.

SANTOS FILHO, O. A, Ricardo Bicca de Alencastro. **Modelagem de proteínas por homologia**. Química Nova. [S.l.]: scielo. , 2003

PAGADALA, Nataraj S e SYED, Khajamohiddin e TUSZYNSKI, Jack. **Software for molecular docking: a review**. Biophysical reviews, v. 9, n. 2, p. 91–102, 2017.

PARRINELLO, M_ e RAHMAN, A. **Crystal structure and pair potentials: A molecular-dynamics study**. Physical Review Letters, v. 45, n. 14, p. 1196, 1980.

PARRINELLO, Michele e RAHMAN, Aneesur. **Polymorphic transitions in single crystals: A new molecular dynamics method**. Journal of Applied physics, v. 52, n. 12, p. 7182–7190, 1981.

PYZER-KNAPP, Edward O e colab. **What is high-throughput virtual screening? A perspective from organic materials discovery**. Annual Review of Materials Research, v. 45, p. 195–216, 2015.

Queue, Celery: Distributed Task. . [S.l: s.n.]. , [S.d.]

RAPAPORT, D C e RAPAPORT, D C R. **The Art of Molecular Dynamics Simulation**. [S.l.]: Cambridge University Press, 2004. Disponível em: <https://books.google.com.br/books?id=iqDJ2hjqBMEC>.

RASHID, Mahmood A e colab. **An enhanced genetic algorithm for ab initio protein structure prediction**. IEEE Transactions on Evolutionary Computation, v. 20, n. 4, p. 627–644, 2015.

RENTZSCH, Robert e RENARD, Bernhard Y. **Docking small peptides remains a great challenge: an assessment using AutoDock Vina**. Briefings in bioinformatics, v. 16, n. 6, p. 1045–1056, 2015.

REYMOND, Jean-Louis e colab. **Chemical space as a source for new drugs**. MedChemComm, v. 1, n. 1, p. 30–38, 2010.

SADEDIN, Simon P. e POPE, Bernard e OSHLACK, Alicia. **Bpipe: a tool for running and managing bioinformatics pipelines**. Bioinformatics, v. 28, n. 11, p. 1525–1526, 1 Jun 2012. Disponível em: <https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/bts167>. Acesso em: 5 jun 2019.

SALI, Andrej e BLUNDELL, Tom L. **Definition of general topological equivalence in protein structures: A procedure involving comparison of properties and relationships through simulated annealing and dynamic programming**. Journal of molecular biology, v. 212, n. 2, p. 403–428, 1990.

SCHERAGA, Harold A e KHALILI, Mey e LIWO, Adam. **Protein-folding dynamics: overview of molecular simulation techniques**. Annu. Rev. Phys. Chem., v. 58, p. 57–83, 2007.

SCIANI, Juliana Mozer e PIMENTA, Daniel Carvalho. **The modular nature of bradykinin-potentiating peptides isolated from snake venoms**. Journal of Venomous Animals and Toxins including Tropical Diseases, v. 23, n. 1, p. 45, 2017.

SEDDON, Gavin e colab. **Drug design for ever, from hype to hope**. Journal of computer-aided molecular design, v. 26, n. 1, p. 137–150, 2012.

SHEN, M. **Statistical potential for assessment and prediction of protein structures**. Protein Science. [S.l.]: Wiley-Blackwell. , 2006

SINKO, William e LINDERT, Steffen e MCCAMMON, J Andrew. **Accounting for**

Receptor Flexibility and Enhanced Sampling Methods in Computer-Aided Drug Design. Chemical biology & drug design, v. 81, n. 1, p. 41–49, 2013.

SMITH, Temple F e WATERMAN, Michael S. **Identification of common molecular subsequences**. Journal of molecular biology, v. 147, n. 1, p. 195–197, 1981.

SONG, Lin Frank e colab. **Using AMBER18 for Relative Free Energy Calculations**. Journal of Chemical Information and Modeling, 2019.

STERLING, Teague e IRWIN, John J. **ZINC 15--ligand discovery for everyone**. Journal of chemical information and modeling, v. 55, n. 11, p. 2324–2337, 2015.

SZYMAŃSKI PAWEŁAND MARKOWICZ, Magdalena e MIKICIUK-OLASIK, Elżbieta. **Adaptation of high-throughput screening in drug discovery toxicological screening tests**. International journal of molecular sciences, v. 13, n. 1, p. 427–452, 2012.

TELEMAN, Olle e JÖNSSON, Bo e ENGSTRÖM, Sven. **A molecular dynamics simulation of a water model with intramolecular degrees of freedom**. Molecular Physics, v. 60, n. 1, p. 193–203, 1987.

The Web framework for perfectionists with deadlines | Django. Disponível em: <https://www.djangoproject.com/>. Acesso em: 3 jun 2019.

TODD, Peter A e HEEL, Rennie C. **Enalapril**. Drugs, v. 31, n. 3, p. 198–248, 1986.

TRUCHON, Jean-François e BAYLY, Christopher I. **Evaluating virtual screening methods: good and bad metrics for the *early recognition* problem**. Journal of chemical information and modeling, v. 47, n. 2, p. 488–508, 2007.

VERLI, Hugo. **Bioinformática: da Biologia à Flexibilidade Moleculares**. 3. ed. Porto Alegre: The name of the publisher, 2014.

WASSERMAN, Philip D. **Neural computing: theory and practice**. [S.l.]: Van Nostrand Reinhold Co., 1989.

WEBB, Benjamin e SALI, Andrej. **Comparative protein structure modeling using Modeller**. Current protocols in bioinformatics, p. 5–6, 2014a.

WEBB, Benjamin e SALI, Andrej. Protein structure modeling with MODELLER. Protein Structure Prediction. [S.l.]: Springer, 2014b. p. 1–15.

WEBB, Benjamin e SALI, Andrej. Protein structure modeling with MODELLER. Functional Genomics. [S.l.]: Springer, 2017. p. 39–54.

XU, Jun e HAGLER, Arnold. **Chemoinformatics and drug discovery**. Molecules, v. 7, n. 8, p. 566–600, 2002.

YANG, Jianyi e ZHANG, Yang. **Protein structure and function prediction using I-TASSER**. Current protocols in bioinformatics, v. 52, n. 1, p. 5–8, 2015.

YOUNG, G. **of Human**. Biology of Reproduction, v. 9, p. 826–835, 1980.

ZHANG, Chunchun e WU, Shanshan e XU, Dingguo. **Catalytic Mechanism of Angiotensin-Converting Enzyme and E ff ects of the Chloride Ion**. n. Md, 2013.

Apêndice A:

Seleção dos 10 melhores ligantes de hACE para o teste de impacto da escolha do tamanho de *box* no *screening* com a hACE (seção 4.5.2):

O HTP SurflexDock foi utilizado com as estruturas tridimensionais dos domínios C-terminal (pdb id: 1o86) e N-terminal (pdb id: 5amb) da enzina conversora da angiotensina-I e uma biblioteca contendo somente ligantes desta enzima que foi obtida do site DUD[3]. A pasta dud_ligands2006/ace_ligands.mol2.gz da biblioteca obtida foi descompactada e as moléculas em formato mol2 foram convertidas para o formato pdbqt usando o script do AutoDockTools pdb_to_pdbqt.py. Subsequentemente as moléculas em formato pdbqt foram compactadas como ligand.tar.gz para a criação de uma nova biblioteca.

O HTP SurflexDock foi configurado para usar um *box* com tamanho de meia aresta 1.5 nm, sendo este *box* centrado no resíduos Glu384 para a seleção dos 10 melhores ligantes do domínio N-terminal e Glu372 para a seleção dos 10 melhores ligantes do domínio C-terminal. Para os resultados foram considerado somente os *dockings* realizados com o cristal (conformação control). Os resultados foram classificados pelo ΔG calculado pelo AutoDock 4.2 e uma tabela com os 10 melhores ligantes de cada domínio foi criada.

Resultados

Os 10 melhores ligantes do domínio C-terminal apresentaram ΔG entre -12.25 e -11.0 kcal/mol (Tabela VIII) e no N-terminal apresentaram ΔG entre -11.69 e -9.3 kcal/mol (Tabela X). Os arquivos em formato pdbqt desses ligantes foram utilizados na criação as duas bibliotecas (uma para cada domínio da hACE) e utilizadas no experimento de avaliação do impacto de diferentes tamanhos de box, conforme descrito na seção 2.5.2 deste documento.

[3] http://dud.docking.org/r2/ace.tar.gz

Tabela VIII - Lista dos 10 ligantes que tiveram o melhor desempenho com o domínio C-terminal de hACE

Compound	Ki (molar)	ΔG (kcal/mol)	*Zinc Code*
ligand_42	1.06e-09	-12.25	ZINC03814193
ligand_27	1.22e-09	-12.16	ZINC03814178
ligand_22	2.27e-09	-11.79	ZINC03814173
ligand_33	2.41e-09	-11.76	ZINC03814184
ligand_25	2.43e-09	-11.75	ZINC03814176
ligand_34	3.03e-09	-11.62	ZINC03814185
ligand_30	3.78e-09	-11.49	ZINC03814181
ligand_37	5.28e-09	-11.29	ZINC03814188
ligand_15	5.5e-09	-11.27	ZINC03814166
ligand_32	8.65e-09	-11.0	ZINC03814183

Tabela IX - Lista dos 10 ligantes que tiveram o melhor desempenho com o domínio N-terminal de hACE

Compound	Ki (Molar)	ΔG (kcal/mol)	*Zinc Code*
ligand_15	2.72e-09	-11.69	ZINC03814166
ligand_27	2.83e-08	-10.3	ZINC03814178
ligand_24	3,60e-08	-10.15	ZINC03814175
ligand_31	4,09e-08	-10.08	ZINC03814182
ligand_25	6,65e-08	-9.79	ZINC03814176
ligand_37	9,00e-08	-9.61	ZINC03814188
ligand_33	1,08e-07	-9.5	ZINC03814184
ligand_34	1,23e-07	-9.43	ZINC03814185
ligand_36	1,24e-07	-9.42	ZINC03814187
ligand_30	1,52e-07	-9.3	ZINC03814181

Apêndice B: Tabelas com os piores enriquecimentos do virtual *screening* da Enzima conversora de Angiotensina I utilizando o HTP SurflexDock (Capitulo II).

Tabela X - Classificação dos 10 piores ligantes no enriquecimento dos ligantes utilizando o domínio C-Terminal da hACE e sampling de 2 ns.

Control			*Cluster* t=718			*Cluster* t=1100			*Cluster* t=1728		
Compound	ΔG (kcal/mol)	*Chemotype*	*Compound*	ΔG (kcal/mol)	*Chemotype*	*Compound*	ΔG (kcal/mol)	*Chemotype*	*Compound*	ΔG (kcal/mol)	*Chemotype*
ligand_14	-6,88	Tiol	ligand_21	-5,81	Tiol	ligand_07	-5,49	Fosfinico	ligand_35	-5,61	Carboxilico
ligand_01	-6,83	Tiol	ligand_02	-5,72	Tiol	ligand_35	-5,38	Carboxilico	ligand_01	-5,5	Tiol
ligand_17	-6,77	Tiol	ligand_12	-5,66	Carboxilico	ligand_19	-5,2	Tiol	ligand_43	-5,19	Tiol
RX4	-6,63	Fosfinico	ligand_43	-5,62	Tiol	ligand_21	-5	Tiol	ligand_12	-5,12	Carboxilico
ligand_46	-6,6	Tiol	ligand_19	-5,59	Tiol	ligand_43	-4,8	Tiol	ligand_49	-4,96	Tiol
ligand_43	-6,32	Tiol	ligand_49	-5,54	Tiol	ligand_10	-4,76	Tiol	ligand_19	-4,95	Tiol
ligand_13	-6,31	Tiol	ligand_46	-5,44	Tiol	ligand_17	-4,74	Tiol	ligand_10	-4,94	Tiol
ligand_08	-6,25	Tiol	ligand_13	-5,36	Tiol	ligand_01	-4,72	Tiol	ligand_13	-4,85	Tiol
ligand_10	-5,84	Tiol	ligand_01	-5,29	Tiol	ligand_49	-4,61	Tiol	ligand_17	-4,83	Tiol
ligand_49	-5,76	Tiol	ligand_10	-4,65	Tiol	ligand_46	-4,47	Tiol	ligand_46	-4,81	Tiol

Tabela XI -- Classificação dos 10 piores ligantes no enriquecimento dos ligantes utilizando o domínio N-Terminal da hACE e sampling de 2 ns.

Control			*Cluster* t=314			*Cluster* t=592			*Cluster* t=956		
Molécula	**ΔG (kcal/mol)**	***Chemotype***	**Molécula**	**ΔG (kcal/mol)**	***Chemotype***	**Molécula**	**ΔG (kcal/mol)**	***Chemotype***	**Molécula**	**ΔG (kcal/mol)**	***Chemotype***
ligand_08	-7,15	Tiol	ligand_08	-6,3	Tiol	ligand_21	-6,12	Tiol	ligand_14	-6,02	Tiol
ligand_17	-7	Tiol	ligand_49	-6,28	Tiol	ligand_43	-6,01	Tiol	ligand_35	-6,01	Carboxilico
ligand_21	-6,9	Tiol	ligand_12	-6,26	Carboxilico	ligand_49	-5,93	Tiol	ligand_07	-5,92	Fosfinico
ligand_46	-6,68	Tiol	ligand_07	-6,06	Fosfinico	ligand_46	-5,87	Tiol	ligand_43	-5,83	Tiol
ligand_13	-6,65	Tiol	ligand_01	-6,05	Tiol	ligand_19	-5,76	Tiol	ligand_01	-5,74	Tiol
ligand_01	-6,61	Tiol	ligand_35	-6	Carboxilico	ligand_35	-5,56	Carboxilico	ligand_46	-5,66	Tiol
ligand_43	-6,5	Tiol	ligand_17	-5,65	Tiol	ligand_13	-5,54	Tiol	ligand_49	-5,61	Tiol
ligand_19	-6,47	Tiol	ligand_19	-5,5	Tiol	ligand_12	-5,4	Carboxilico	ligand_19	-5,37	Tiol
ligand_49	-6,44	Tiol	ligand_13	-5,48	Tiol	ligand_10	-5,36	Tiol	ligand_10	-4,97	Tiol
ligand_10	-6,3	Tiol	ligand_10	-5,23	Tiol	ligand_17	-5,12	Tiol	ligand_17	-4,72	Tiol

Tabela XII - Classificação dos 10 piores ligantes no enriquecimento dos ligantes utilizando o domínio C-Terminal da hACE e *sampling* de 5 ns.

Control			*Cluster* t=1048			*Cluster* t=2182			*Cluster* t=3662		
Molécula	ΔG (kcal/mol)	*Chemotype*	Molécula	ΔG (kcal/mol)	*Chemotype*	Molécula	ΔG (kcal/mol)	*Chemotype*	Molécula	ΔG (kcal/mol)	*Chemotype*
ligand_41	-6,82	Tiólico	ligand_46	-5,39	Tiólico	ligand_46	-5,18	Tiólico	ligand_49	-5,49	Tiólico
ligand_18	-6,77	Tiólico	ligand_43	-5,38	Tiólico	ligand_19	-5,16	Tiólico	ligand_46	-5,41	Tiólico
ligand_17	-6,77	Tiólico	ligand_01	-5,29	Tiólico	ligand_01	-5,11	Tiólico	ligand_07	-5,39	Fosfínico
ligand_01	-6,76	Tiólico	ligand_49	-5,22	Tiólico	ligand_07	-5,11	Fosfínico	ligand_01	-5,38	Tiólico
ligand_19	-6,67	Tiólico	ligand_13	-5,22	Tiólico	ligand_17	-5,08	Tiólico	ligand_35	-5,38	Carboxílico
ligand_43	-6,65	Tiólico	ligand_07	-5,03	Fosfínico	ligand_49	-5,03	Tiólico	ligand_08	-5,25	Tiólico
ligand_10	-6,11	Tiólico	ligand_35	-4,88	Carboxílico	ligand_43	-4,99	Tiólico	ligand_17	-5,02	Tiólico
ligand_13	-6,09	Tiólico	ligand_19	-4,8	Tiólico	ligand_13	-4,79	Tiólico	ligand_19	-4,61	Tiólico
ligand_08	-6	Tiólico	ligand_10	-4,77	Tiólico	ligand_35	-4,65	Carboxílico	ligand_13	-4,51	Tiólico
ligand_46	-5,81	Tiólico	ligand_17	-4,51	Tiólico	ligand_10	-4,44	Tiólico	ligand_10	-4,08	Tiólico

Tabela XIII - Classificação dos 10 piores ligantes no enriquecimento dos ligantes utilizando o domínio N-Terminal da hACE e sampling de 5 ns.

Control			*Cluster* t=706			*Cluster* t=1870			*Cluster* t=4460		
Molécula	ΔG (kcal/mol)	*Chemotype*	Molécula	ΔG (kcal/mol)	*Chemotype*	Molécula	ΔG (kcal/mol)	*Chemotype*	Molécula	ΔG (kcal/mol)	*Chemotype*
ligand_21	-7,05	Tiolico	ligand_12	-7,24	Carboxilico	ligand_14	-6,74	Tiolico	ligand_02	-6,84	Tiolico
ligand_17	-7,01	Tiolico	ligand_41	-7,14	Tiolico	ligand_43	-6,52	Tiolico	ligand_43	-6,75	Tiolico
ligand_01	-6,92	Tiolico	ligand_46	-7,02	Tiolico	ligand_35	-6,46	Carboxilico	ligand_12	-6,74	Carboxilico
ligand_08	-6,89	Tiolico	ligand_01	-6,98	Tiolico	ligand_01	-6,44	Tiolico	ligand_01	-6,38	Tiolico
ligand_46	-6,71	Tiolico	ligand_19	-6,81	Tiolico	ligand_46	-6,4	Tiolico	ligand_49	-6,36	Tiolico
ligand_49	-6,55	Tiolico	ligand_13	-6,7	Tiolico	ligand_19	-6,18	Tiolico	ligand_13	-6,36	Tiolico
ligand_13	-6,52	Tiolico	ligand_43	-6,57	Tiolico	ligand_07	-6,15	Fosfinico	ligand_46	-6,22	Tiolico
ligand_43	-6,52	Tiolico	ligand_49	-6,21	Tiolico	ligand_13	-6	Tiolico	ligand_10	-5,98	Tiolico
ligand_19	-6,47	Tiolico	ligand_17	-6,06	Tiolico	ligand_10	-5,64	Tiolico	ligand_19	-5,84	Tiolico
ligand_10	-6,17	Tiolico	ligand_10	-5,97	Tiolico	ligand_17	-5,5	Tiolico	ligand_17	-5,83	Tiolico