

UNIVERSIDADE FEEVALE

JONAS ELIAS FLESCH

CRIAÇÃO DE APLICATIVOS PARA FACILITAR A EXECUÇÃO  
DE ENSEMBLE DOCKING

Novo Hamburgo

2015

JONAS ELIAS FLESCH

CRIAÇÃO DE APLICATIVOS PARA FACILITAR A EXECUÇÃO  
DE ENSEMBLE DOCKING

Trabalho de Conclusão de Curso  
apresentado como requisito parcial  
à obtenção do grau de Bacharel em  
Ciência da Computação pela  
Universidade Feevale

Orientador: Ricardo Ferreira de Oliveira

Novo Hamburgo

2015

## **AGRADECIMENTOS**

Gostaria de agradecer a todos os que, de alguma maneira, contribuíram para a realização desse trabalho de conclusão, em especial:

A minha noiva pela motivação que me deu e pela paciência.

A minha família que sempre me apoiou e incentivou.

Ao professor Ms. Ricardo Ferreira de Oliveira pela sabedoria transmitida.

## RESUMO

O citocromo P450 (ou CYP) é uma das enzimas mais importantes na metabolização de medicamentos. Estes muitas vezes são ativados ao se ligarem as subfamílias do citocromo P450. Como exemplo pode ser utilizado a variante CYP2E1 do citocromo P450, que realiza a metabolização do Paracetamol. O papel da Ciência da Computação no estudo das proteínas abordado neste trabalho é a simulação computacional da ligação das proteínas com outros compostos. Tendo as formas 3D das proteínas pode-se, por exemplo, realizar simulações de como uma proteína se liga a um composto, baseando-se na sua forma, o que é conhecido como Docagem Molecular. Esta é uma ferramenta importante para a descoberta de novas drogas. Esta ferramenta tenta encaixar a proteína em um ligante de várias maneiras diferentes, gerando uma energia livre para cada encaixe em uma orientação/posição diferente. Quanto menor a energia livre, melhor é a docagem. *Ensemble Docking* é uma técnica que ajuda a trabalhar as duas maiores dificuldades da docagem molecular: o sistema de pontuação e flexibilização de proteínas. Proteínas não são estruturas estáticas, nem os ligantes. A técnica de *Ensemble Docking* utiliza um corpo fixo, pois testes demonstraram que utilizar ligantes flexíveis não apresentam resultados qualitativos significantes na docagem. Para representar um receptor flexível, no *Ensemble Docking* são utilizadas múltiplas estruturas de proteínas. Uma das ferramentas desenvolvidas como resultado deste trabalho atua juntamente com o programa Gromacs e utiliza as poses geradas como parâmetro para realizar várias docagens no programa Autodock. Cada uma das poses gera uma docagem com uma energia diferente, e as melhores energias são retornadas ao usuário. Desta maneira o pesquisador pode simular a docagem de todas as poses geradas pelo Gromacs de forma automatizada. É utilizado como caso de aplicação docagens do citocromo CYP2E1.

Palavras-chave: Citocromo P450. Docagem Molecular. Ensemble Docking.

## ABSTRACT

The cytochrome P450 (or CYP) is one of the most important enzymes in the metabolism of drugs. These often are activated by binding with variants of cytochrome P450. As an example we can use the variant of the cytochrome P450 CYP2E1, which makes the metabolism of Paracetamol. The role of computer science in the study of proteins that will be addressed in this work is the computational simulation of the binding of proteins with other compounds. With the 3D shapes of the proteins you can, for example, perform simulations of how a protein binds to a compound, based on their format, which is known as Molecular Docking. This is an important tool for the discovery of new drugs. This tool tries to fit the protein in a binder in many different ways, creating a free energy for each engaging in a guide / different position. The lower the free energy, is the best docking. Ensemble docking is a technique that helps to work the two biggest problems of molecular docking, the scoring system and flexibility of proteins. Proteins are not static structures or ligands. The Ensemble Docking technique uses a fixed body, because tests have shown that using flexible ligands show no significant qualitative results in dockage. To represent a flexible receptor, the Ensemble docking of multiple protein structures are used. One of the tools that will be developed as the objective of this work is to act together with the Gromacs program and use the poses generated as a parameter to perform multiple docking in AutoDock program. Each of the generated docations have a diferente energy, and the best energies should be returned to the user. This manner the researcher can simulate the docking of all poses generated by Gromacs an automated fashion. Will be used as a case application of docking the cytochrome CYP2E1.

Key words: Citocromo P450. Docagem Molecular. Ensemble Docking.

## LISTA DE FIGURAS

Figura 1.1 – Estrutura de DNA .....	16
Figura 2.1 – Processo de autorização no PDB. ....	22
Figura 2.2 – Quantidade de estruturas disponíveis no Protein Data Bank do longo do tempo. ....	23
Figura 3.1 – Encaixe de ligante a uma proteína. ....	25
Figura 3.2 – Processos de Docagem Molecular .....	26
Figura 3.3 – Número de citações dos programas de docagem molecular .....	27
Figura 3.4 – AutoDockTools como ferramenta de avaliação do sucesso da docagem.....	29
Figura 3.5 – Encaixe de ligante após flexibilização da proteína.....	30
Figura 4.1 – Variações de proteínas e ligantes .....	32
Figura 4.2 – Trajetórias de uma proteína durante o processo de dinâmica molecular.....	33
Figura 4.3 – Simulação de dinâmica molecular utilizando GROMACS com lipídios inseridos em água.....	34
Figura 4.4 – Passos para execução de um processo de dinâmica molecular através do GROMACS .....	35
Figura 5.1 – Diagrama de Classes.....	38
Figura 5.2 – Diagrama de Atividades .....	39
Figura 5.3 – Caso de uso.....	40
Figura 5.4 – Passo a passo para instalação do ambiente de desenvolvimento .....	47
Figura 5.5 – Língua inglês com mais prioridade no navegador Chrome.....	48
Figura 5.6 – Cabeçalhos HTTP com língua inglês como prioridade no navegador Chrome ....	48
Figura 5.7 – Tela inicial da aplicação em inglês .....	49
Figura 5.8 – Tela inicial da aplicação em português.....	50
Figura 6.1 – Fluxo do algoritmo.....	52

Figura 6.2 – Arquivo PDB.....	53
Figura 6.3 – Página de seleção de arquivos de parâmetro .....	54
Figura 6.4 – Arquivo MDP.....	55
Figura 6.5 – Parâmetros de configuração da aplicação.....	56
Figura 6.6 – Fragmento do arquivo PDBQT de saída .....	57
Figura 6.7 – Arquivo PML.....	58
Figura 6.8 – Imagem de saída do PyMOL.....	58
Figura 6.9 – Energia Livre ao Longo do Tempo.....	59
Figura 7.1 – Conformação Final da docagem P450 2E1 x omega-imidazolyl-decanoic .....	62
Figura 7.2 – Energia Livre ao Longo do Tempo P450 2E1 x omega-imidazolyl-decanoic.....	62
Figura 7.3 - Comparação Resultados docagem tradicional e <i>ensemble docking</i> com P450 2E1 x omega-imidazolyl-decanoic .....	63
Figura 7.4 – Conformação Final da docagem P450 2E1 x 4-methylpyrazole.....	64
Figura 7.5 – Energia Livre ao Longo do Tempo P450 2E1 x 4-methylpyrazole.....	65
Figura 7.6 - Comparação Resultados docagem tradicional e <i>ensemble docking</i> com P450 2E1 x 4-methylpyrazole.....	66
Figura 7.7 – Conformação Final da docagem P450 2E1 x ligante Omega-Imidazolyl-Dodecanoic.....	67
Figura 7.8 – Energia Livre ao Longo do Tempo P450 2E1 x ligante Omega-Imidazolyl-Dodecanoic.....	68
Figura 7.9 - Comparação Resultados docagem tradicional e <i>ensemble docking</i> com P450 2E1 x Omega-Imidazolyl-Dodecanoic.....	69

## **LISTA DE TABELAS**

Tabela 1.1 – Colunas da seção de átomos do arquivo PDB.....	53
---	----

## LISTA DE ABREVIATURAS E SIGLAS

3D	Tridimensional
A	Adenina
APT	<i>Advanced Packaging Tool</i>
BLAST	<i>Basic Local Alignment Search Tool</i>
BNL	<i>Brookhaven National Laboratories</i>
C	Citosina
CSD	<i>Cambridge Structural Database</i>
CYP	Citocromo P450
DNA	<i>Deoxyribonucleic Acid</i>
DPF	<i>AutoDock Parameter File</i>
EDR	<i>Energy External Data Representation</i>
G	Guanina
GPMAW	<i>General Protein/Mass Analysis for Windows</i>
GRO	<i>Gromacs</i>
HTML	<i>HyperText Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IDE	<i>Integrated Development Environment</i>
JRE	<i>Java Runtime Environment</i>
JSF	<i>JavaServer Faces</i>
JSP	<i>JavaServer Pages</i>
JVM	<i>Java Virtual Machine</i>
MDP	<i>Molecular Dynamics Parameter</i>
MVC	<i>Model View Controller</i>

NCBI	<i>National Center for Biotechnology Information</i>
ORF	<i>Open Reading Frame</i>
PDB	<i>Protein Data Bank</i>
PDBQT	<i>Protein Data Bank, Partial Charge (Q), Atom Type (T)</i>
PNG	<i>Portable Network Graphics</i>
POM	<i>Project Object Model</i>
RNA	<i>Ribonucleic Acid</i>
RCSB	<i>Research Collaboratory for Structural Bioinformatics</i>
RMN	<i>Ressonância Magnética Nuclear</i>
T	<i>Timina</i>
TOP	<i>Topology</i>
TPR	<i>Topology Run</i>
TRR	<i>Trajectory File</i>
XML	<i>Extensible Markup Language</i>

## SUMÁRIO

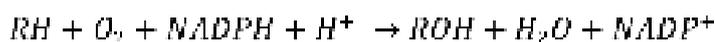
<b>INTRODUÇÃO .....</b>	<b>13</b>
<b>1 BIOINFORMÁTICA .....</b>	<b>15</b>
1.1 Introdução .....	15
1.2 Genoma .....	15
1.3 Alinhamento.....	17
1.4 Filogenia.....	17
1.5 Estrutura de Proteínas.....	18
<b>2 DESENHO RACIONAL DE FÁRMACOS.....</b>	<b>20</b>
2.1 Introdução a Desenho Racional de Fármacos.....	20
2.2 Escolhendo uma doença.....	21
2.3 Escolhendo uma droga alvo.....	21
2.4 Bancos de dados de informações.....	22
<b>3 ATRACAMENTO .....</b>	<b>25</b>
3.1 Docagem Molecular .....	25
3.2 AUTODOCK.....	27
3.3 Ensemble Docking.....	29
<b>4 DINÂMICA MOLECULAR.....</b>	<b>31</b>
4.1 Introdução .....	31
4.2 Flexibilidade de proteínas .....	31
4.3 GROMACS.....	33
<b>5 METODOLOGIA DE IMPLEMENTAÇÃO.....</b>	<b>37</b>
5.1 Tecnologias empregadas.....	37
5.2 Diagrama de classe.....	37
5.3 Diagrama de atividades .....	39
5.4 Diagrama de caso de uso.....	40
5.5 Java.....	41
5.5.1 A linguagem Java.....	41
5.5.2 Java para Web .....	41
5.5.3 Spring Boot.....	42

5.5.4	Chamada de linha de comando.....	42
5.5.5	Apache Maven.....	43
5.5.6	JFreeChart.....	44
5.6	Teste unitário .....	44
5.7	Ambiente operacional .....	45
5.8	Premissas de execução .....	45
5.9	Controle de versão.....	46
5.10	Internacionalização.....	48
<b>6</b>	<b>A FERRAMENTA.....</b>	<b>51</b>
6.1	Objetivos .....	51
6.2	Fluxo de execução.....	51
6.3	Arquivos de parâmetro.....	52
6.4	Chamada da dinâmica molecular.....	55
6.5	Chamada da docagem.....	56
6.6	Exibição dos resultados.....	57
6.6.1	Representação da melhor docagem.....	57
6.6.2	Gráfico de energia livre ao longo do tempo .....	59
6.7	Oportunidades de melhoria.....	60
<b>7</b>	<b>RESULTADOS OBTIDOS.....</b>	<b>61</b>
7.1	P450 2E1 x omega-imidazolyl-decanoic .....	61
7.2	P450 2E1 x 4-methylpyrazole.....	63
7.3	P450 2E1 x Omega-Imidazolyl-Dodecanoic.....	66
	<b>CONCLUSÃO .....</b>	<b>70</b>
	<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>72</b>

## INTRODUÇÃO

O citocromo P450 (ou CYP) é uma das enzimas mais importantes na metabolização de medicamentos. De acordo com Hollenberg (1992, p. 686), o sistema de enzimas do P450 catalisa o metabolismo de uma ampla variedade de compostos naturais e xenobióticos através de reações que requerem NADPH e O<sub>2</sub>, sendo também capaz de catalizar a hidroxilação dependente de peróxido de substratos na ausência de NADPH e O<sub>2</sub> (MARNETT; KENNEDY, 1995). As variantes do P450 catalizam diversas reações químicas incluindo-se monooxigenação, peroxidação, redução, desalquilação, expoxidação e desalogenação (GUENGERICH, 1992, p. 667; HOLLENBERG, 1992, p. 686).

As reações gerais de monooxigenase típicas do P450 são descritas com a estequiometria que se segue (HOLLENBERG, 1992, p. 686):



Na reação indicada, RH é o substrato orgânico que recebe um átomo de oxigênio de uma molécula de O<sub>2</sub> ao passo que o oxigênio restante forma uma molécula de água. O P450 atua em conjunto com outros dois componentes: a flavoproteína denominada NADPH-CYP450 redutase e a fosfatidilcolina que facilita a transferência de elétrons da NADPH-CYP450 redutase para o CYP (FOYE, 2008, p. 47). Uma característica digna de nota é a falta de especificidade das enzimas da família CYP para com substratos o que as torna capazes de oxidar milhares de compostos hidrofóbicos diferentes, convertendo-os em compostos mais hidrofílicos, facilitando sua excreção (KARP, 2008, p. 148). As isoformas individualmente são capazes de interagir com uma ampla gama de substratos químicos ocorrendo especificidades de substratos sobrepostas entre CYPs. Esta promiscuidade é vantajosa em termos de aprimoramento da defesa do organismo contra potenciais xenobióticos nocivos. Não obstante, esta interação pode levar a efeitos indesejáveis, como a rápida eliminação de fármacos e produção de compostos tóxicos (MARÉCHAL et al., 2008, p. 83).

No caso do P450, o interesse em especial é na biotransformação de medicamentos, que muitas vezes são ativados ao se ligarem as variações do citocromo P450. O P450 é considerado inclusive a família mais importante de enzimas envolvidas no metabolismo de drogas, e, conseqüentemente, no seu efeito farmacológico e toxicológico. (DE GRAAF; VERMEULEN; FEENSTRA, 2005, p. 2725-2726). Como exemplo pode ser utilizada a variante CYP2E1 do citocromo P450, que realiza a metabolização de uma grande quantidade

de compostos, em particular compostos pequenos e hidrofóbicos, incluindo potenciais carcinogênicos (RONIS, LINDROS, INGELMAN-SUNDBERG, 1996). Entre os compostos metabolizados o Paracetamol é um dos mais conhecidos. Esta variante é expressa no fígado e em vários outros tecidos. Ela transforma o composto original do medicamento em N-acetil-p-benzoquinona, que é a substância que irá atuar no organismo obtendo-se os efeitos farmacológicos desejados. (LEE, 1996, p. 12063).

O papel da Ciência da Computação no estudo das proteínas que será abordado neste trabalho é a simulação computacional da ligação das proteínas com outros compostos. Com a utilização de simulações computacionais, é possível testar candidatos a medicamentos e sua capacidade de ligar-se com proteínas existentes no ser vivo, antes da realização de testes *in-vitro* e *in-vivo*. A isto se dá o nome de teste *in-silico*. A execução de simulações computacionais pode ser realizada com grande quantidade de candidatos a ligantes, agilizando a descoberta de novos fármacos. Este processo de desenvolvimento de novos fármacos é descrito no Capítulo 2.

Uma das ferramentas que será desenvolvida deverá atuar juntamente com o programa Gromacs e utilizar as poses geradas como parâmetro para realizar várias docagens no programa Autodock. Cada uma das poses irá gerar uma docagem com uma energia diferente, e as melhores energias deverão ser retornadas ao usuário. A este processo dá-se o nome de *Ensemble Docking*, melhor descrito no Capítulo 3.2. Desta maneira o pesquisador poderá simular a docagem de todas as poses geradas pelo Gromacs de forma automatizada. Utilizar-se-á como caso de aplicação docagens do citocromo CYP2E1, isoforma do P450.

# 1 BIOINFORMÁTICA

O objetivo deste capítulo é descrever áreas de conhecimento utilizadas pelo objeto da pesquisa. Também serão introduzidas técnicas já bastante difundidas no desenvolvimento de novos fármacos e que serão utilizadas nos aplicativos que serão desenvolvidos neste trabalho. Se faz importante que os conceitos básicos de bioinformática estejam bem definidos, mesmo os que não são objeto direto deste trabalho, pois ainda assim tem importância para melhor entendimento do papel da docagem de proteínas.

## 1.1 Introdução

A bioinformática se refere ao emprego de ferramentas computacionais no estudo de problemas e questões biológicas. Estas questões biológicas podem estar relacionadas ao código DNA, a filogenia, ao dobramento e encaixe de proteínas, dentre outras áreas. A bioinformática sem dúvida é um campo de pesquisa que envolve várias áreas de conhecimento, dentre elas biologia, química, física e computação (VERLI, 2014, p. 2-12).

## 1.2 Genoma

O DNA (*deoxyribonucleic acid*, ou ácido desoxirribonucleico) é uma sequência de bases: adenina (A), citosina (C), guanina (G) e timina (T). É no DNA que estão contidas as informações necessárias para a construção de proteínas, que são a base para a vida. Conforme pode ser visto na figura 1.1, o DNA é formado por uma dupla hélice, onde adenina liga-se com timina, e citosina com guanina (SETUBAL, 1997).

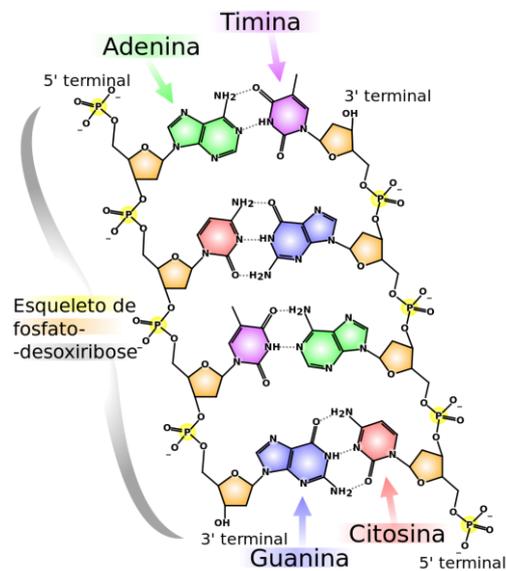


Figura 1.1 – Estrutura de DNA

Fonte: BALL, 2013.

As maneiras existentes para digitalização de códigos de DNA exigem que este seja fragmentado para sua possível leitura. Ainda não é possível realizar a leitura completa do código DNA, do seu início ao fim, sem interrupções. A técnica existente quebra o DNA em pedaços pequenos, e a leitura destes pequenos pedaços é realizada. De posse destes pequenos pedaços, se faz necessário utilizar algoritmos para sequenciamento, onde estes pedaços são juntados e validados até montar-se uma sequência próxima a original. Muitos destes algoritmos são semelhantes aos de alinhamento, pois envolvem a comparação de sequências. Existem pontos de sobreposição entre os fragmentos pois não é apenas uma sequência que é fracionada no processo de fragmentação, mas sim várias delas. Assim é possível encontrar pontos sobrepostos entre uma sequência e outra (MARGULIES, 2005, p. 376-380).

Ao realizar-se análise de sequências de DNA, um dos pontos mais relevantes é a identificação de genes. Uma das maneiras de identificar genes é buscar por uma região com uma codificação de início e uma de término, denominada de ORF (*Open Reading Frame*), que potencialmente constitui uma região codificante de uma proteína em um genoma procariótico (VERLI, 2014, p. 67-69). Já para genes de organismos eucarióticos, esta busca é bem mais complexa, envolvendo áreas conhecidas como íntrons, que são áreas do DNA que não codificam nenhum gene e podem variar bastante de tamanho, mas elas podem mudar o comportamento da sequência codificante que vem em seguida.

Além das buscas de genes baseadas na codificação, há também técnicas empíricas, que se baseiam em evidências e similaridades entre organismos. Por exemplo o gene “eyeless”, que codifica o olho em vários organismos, aparece como uma similaridade em vários organismos que tem olho ou olhos. Este gene pode ter uma série de mutações de uma espécie para a outra, mas é possível identificar as semelhanças (HALDER, CALLAERTS, GEHRING, 1995, p. 1788).

O DNA é o “código fonte da vida”, é nele que está a receita de quais proteínas são utilizadas para constituir um ser vivo. Durante o processo de síntese da proteína, o DNA não serve diretamente como molde para a síntese desta proteína, o código que é utilizado como molde é o RNA e o nome deste processo é tradução. O RNA (ribonucleic acid, ou ácido ribonucleico) ainda executa outros papéis, como componente estrutural dos ribossomos e o transporte de aminoácidos. O DNA serve como base para o RNA no processo de transcrição, onde o DNA ao invés de copiar-se, é transcrito em um novo código, o RNA, que contém apenas as informações para a síntese de uma proteína (JUNQUEIRA, 2011).

### **1.3 Alinhamento**

Na bioinformática, as sequências de DNA são representadas por uma série de caracteres, A, C, G e T. A comparação de uma sequência com outra é realizada através da comparação dos caracteres que a formam. Durante a comparação, podem surgir substituições, inserções e deleções. Os algoritmos de alinhamento buscam identificar o melhor alinhamento levando em consideração estas mutações que podem ter ocorrido entre uma sequência e outra. Pode-se, por exemplo, alinhar a sequência de dois seres da mesma espécie, para se obter a diferença dos tratos de um e de outro, ou mesmo efetuar o alinhamento de vários seres de espécies diferentes e buscar a similaridade entre eles, montando-se uma árvore filogenética (GIBAS; JAMBECK, 2001).

### **1.4 Filogenia**

Desde o advento da Teoria da Evolução de Darwin os seres vivos vem sendo classificados em uma árvore filogenética. Nesta árvore os seres vivos são classificados de uma maneira que seja possível visualizar quais tem descendentes comuns, indo até o primeiro ser vivo monocelular do qual todos os seres vivos se originaram.

Através da bioinformática é possível aplicar-se algoritmos de comparação e alinhamento para recriar árvores filogenéticas a partir do DNA dos seres. Isto é útil para recriar a história dos seres vivos, quais mutações ocorreram e buscar inclusive estimativas temporais entre determinadas mutações. Conhecendo-se a árvore filogenética e estimativas temporais é possível inclusive mitigar migrações geográficas de indivíduos, baseando-se em antepassados comuns (PEVZNER, 2000).

Outra utilidade de conhecer-se a filogenia adequadamente é entender populações que possuem diferentes tratamentos genéticos, que inclusive podem influenciar na atuação de fármacos, que é o objetivo deste trabalho.

## 1.5 Estrutura de Proteínas

A função de uma proteína está ligada à sua estrutura tridimensional. A análise desta estrutura tem aplicações práticas de grande impacto terapêutico e biotecnológico. Está ligada a áreas como atracamento (ou docagem) proteína-ligante, aplicado ao desenho racional de fármacos, o desenho de novas proteínas com funções específicas, através da nanotecnologia e engenharia de proteínas.

Outra área de estudo relacionada a estrutura de proteínas é o enovelamento, também conhecido como *folding*, que significa “dobra” em inglês. É através do enovelamento que a proteína se “dobra” para adotar a sua forma final. Este processo é muito importante pois a proteína pode ter funções diferentes caso o resultado do enovelamento seja diferente daquele que leve a estrutura nativa.

É durante o processo de enovelamento que ocorre a estabilização da proteína. Segundo Verli (2014, p.151),

[...] a estabilidade de uma proteína não é proveniente da simples soma das energias de suas interações não covalentes. Em solução, cada grupo formador de ligações de hidrogênio na cadeia peptídica estava interagindo com moléculas de água antes da estrutura se enovelar. Então, para cada nova ligação de hidrogênio intramolecular formada quando a estrutura se enovela, uma ligação equivalente com o solvente é desfeita. Na prática, um dos principais fatores que impulsionam o enovelamento de uma proteína é o chamado efeito hidrofóbico. Resumidamente, o efeito hidrofóbico pode ser entendido como a tendência de resíduos de aminoácidos hidrofóbicos se agruparem no interior da proteína (que se torna portanto apolar) e dos resíduos hidrofílicos se exporem na superfície da mesma (que se torna portanto polar).

Este processo de enovelamento é profundamente estudado, pois é fundamental para a determinação experimental da estrutura tridimensional da proteína. A obtenção de modelos

tridimensionais através de métodos de determinação experimental, como cristalografia por difração de raios-X ou ressonância magnética nuclear de proteínas são financeiramente custosos, demorados e a estrutura final pode ainda não ser eficaz. Sendo assim, o uso de métodos computacionais para predição da estrutura com base na sequência de aminoácidos vem sendo uma alternativa muito utilizada (VERLI, 2014, p. 152).

Outro ponto forte no estudo da estrutura das proteínas é a ligação entre proteínas e outros compostos, como fármacos. Esta ligação tem como um fator determinante a estrutura da proteína, que irá contribuir para a ligação ou não da proteína com outros compostos. Muitos fármacos tem a sua função executada quando em algum momento o seu composto é ligado a uma proteína. Além disso, alguns fármacos podem não ter o efeito desejado na forma em que entram no corpo humano, mas sim depois de passar por uma biotransformação. Um exemplo deste processo é o medicamento Paracetamol, que é transformado pelo corpo em N-acetil-p-benzoquinona, que é a substância que irá atuar no organismo obtendo-se os efeitos farmacológicos desejados (LEE, 1996, p. 12063). No próximo capítulo será descrito o processo de desenho racional de fármacos.

## 2 DESENHO RACIONAL DE FÁRMACOS

Este capítulo descreve como que é possível o estudo e desenvolvimento de novos fármacos através de novos processos que se utilizam da informática. Estes novos processos são conhecidos como Desenho Racional de Fármacos, ou *Computer-Aided Drug Design*. É importante que estes conceitos estejam bem definidos, pois a ferramenta resultado da pesquisa realizada neste projeto tem como um de seus objetivos o desenvolvimento de fármacos.

### 2.1 Introdução a Desenho Racional de Fármacos

“Apesar de que muitas rotas para a descoberta de novas drogas existam, a maioria das drogas usadas clinicamente foram descobertas acidentalmente ou por triagem de encontro com produtos enzimáticos, químicos, e produtos naturais.” (CHEN; CHERN, 2007, p. 89, tradução nossa). Os primeiros passos para o estudo de desenvolvimento de uma droga tipicamente consistem em determinar um composto alvo e identificar outros compostos que são ativos comparando-os a este composto (ALMLÖF, 2007, p. 9). Embora a maioria dos fármacos terapêuticamente úteis disponíveis hoje tenham sido descobertos através de processos como esses, está cada vez mais difícil encontrar novos medicamentos desta forma, sendo necessário testar milhões de compostos em alguns casos (MORGON, 2007, p. 490).

Recentemente houve uma grande disponibilização de informações não previamente disponíveis sobre sequências de DNA e estrutura de proteínas, oriunda de projetos como o GENOMAS e o PROTEOMAS, que possibilitaram o desenvolvimento de ferramentas e métodos computacionais para o planejamento de novos fármacos (MORGON, 2007, p. 491).

Os métodos computacionais permitem encontrar melhores candidatos de pares de compostos e compostos ativos, acelerando a descobertas de novas drogas por reduzir o número de iterações de teste necessárias. Além disso, os métodos computacionais permitem testar em um período mais curto uma ampla variedade de combinações de proteínas e ligantes.

Para Alonso, Bliznyuk e Greedy (2006, p.532, tradução nossa), as proteínas são a chave para a descoberta computacional de novas drogas:

Após o projeto Genoma Humano, era esperado que um grande número de novos candidatos a drogas fossem encontrados. Entretanto, os 30.000 genes encodados no genoma humano não demonstraram uma fonte direta para o desenvolvimento de novas drogas, pois não são eles, mas sim as proteínas que eles encodam que normalmente são os alvos das drogas.

Os métodos computacionais para encontrar candidatos a novas drogas podem ser divididos em baseados em estrutura e não baseados na estrutura. Métodos não baseados na estrutura como relação quantitativa estrutura-atividade (em inglês, *Quantitative structure-activity relationship*, **QSAR**) e modelagem farmacológica se baseiam em conhecimento prévio de ligantes em potencial para prever o comportamento do candidato. Métodos baseados na estrutura utilizam informações de modelos estruturais das moléculas para analisar interações e encaixes entre elas (ALMLÖF, 2007, p. 9). Dentre as técnicas baseadas na estrutura, pode-se citar o atracamento, que testa computacionalmente a possibilidade de ligação entre uma proteína e um ligante.

## **2.2 Escolhendo uma doença**

Como uma empresa farmacêutica define qual doença será o alvo ao desenhar uma nova droga? Claramente faz mais sentido se concentrar em doenças que precisam de novas drogas. Além disto outros fatores tem que ser considerados. Para o desenvolvimento de uma nova droga um grande investimento é necessário, e as empresas farmacêuticas buscam trabalhar em projetos que garantam um bom retorno financeiro (PATRICK, 2013, p. 191).

As doenças que são o foco da maior parte da pesquisa de novos fármacos pela indústria são aquelas que atingem o “mundo desenvolvido”, como depressão, úlcera, obesidade, gripe, câncer e doenças cardiovasculares. Doenças de países tropicais recebem menos investimentos. Um exemplo disto é que a busca de drogas para o tratamento da malária vem crescendo muito, em razão de que muitos ocidentais vem contraindo a doença com o aumento do turismo para países mais exóticos. Além disso, empresas farmacêuticas vem aumentando as parcerias com governos e organizações, tais como a Fundação Bill e Melinda Gates para estudar doenças como tuberculose, malária e dengue (PATRICK, 2013, p. 191).

## **2.3 Escolhendo uma droga alvo**

Escolhida a doença alvo, é importante conhecer quais biomacromoléculas estão envolvidas. Isto permite a equipe de pesquisa identificar quais agonistas ou antagonistas devem ser desenhados para um receptor ou se inibidores devem ser desenhados para uma enzima em particular (PATRICK, 2013, p. 191).

É importante considerar também a seletividade destes compostos. Para exemplificar, pode ser utilizada a penicilina, que tem como alvo a uma enzima envolvida com a biossíntese que acontece na parede da célula. Como mamíferos não tem parede celular a penicilina tem poucos efeitos colaterais (PATRICK, 2013, p. 191). Se a droga alvo for um ligante com uma proteínas presente nos humanos, é aumentada a chance de haverem efeitos colaterais.

## 2.4 Bancos de dados de informações

As informações entram no domínio da bioinformática quando são depositados os resultados experimentais obtidos por cientistas em arquivos de bancos de dados apropriados (LESK, 2008). Atualmente, existem grandes bancos de estruturas moleculares de acesso público, como o *Protein Data Bank* (PDB; <http://www.rcsb.org/pdb/>). Existente desde 1971 na *Brookhaven National Laboratories* (BNL), iniciou com apenas sete estruturas. Foi somente na década seguinte que o número começou a crescer exponencialmente, pois a comunidade científica se tornou mais aberta a compartilhar informações e novas técnicas de criação de modelos digitais de proteínas foram desenvolvidas, como a Ressonância Magnética Nuclear (RMN) (BERMAN, 2000, p. 236).

Os dados (coordenadas atômicas, fatores estruturais e limitações de RMN) podem ser enviados via e-mail ou pela ferramenta chamada de *AutoDep Input Tool* (ADIT; <http://pdb.rutgers.edu/adit/>) desenvolvida pela RCSB. Como demonstrado na Figura 2.1, os dados são validados para garantir que seguemos dicionários de dados existentes e só então os dados se tornam disponíveis para a comunidade (BERMAN, 2000, p. 237).

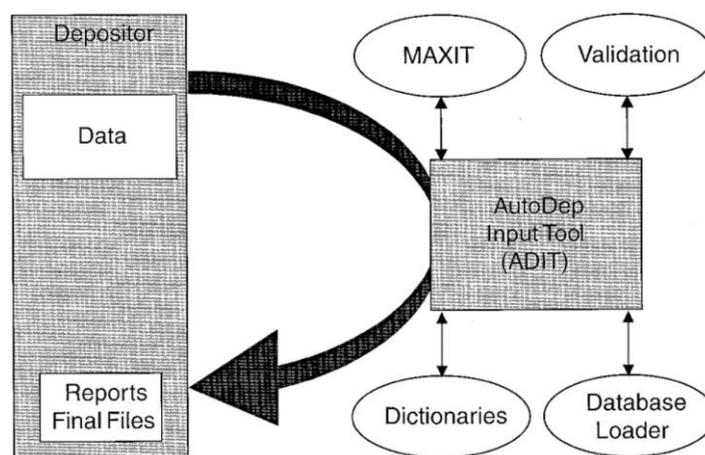


Figura 2.1 – Processo de autorização no PDB.

A popularidade do *Protein Data Bank* é crescente. Com o maior interesse e desenvolvimento de pesquisas relacionadas a bioinformática estrutural, como docagem e enovelamento, este banco de dados se torna fundamental para o compartilhamento de estruturas base para pesquisa, assim como para publicação de resultados. Na Figura 2.2 é possível ver o crescimento da popularidade do *Protein Data Bank* ao longo dos anos, demonstrando a quantidade de estruturas depositadas e disponíveis desde 1990 até 2005.

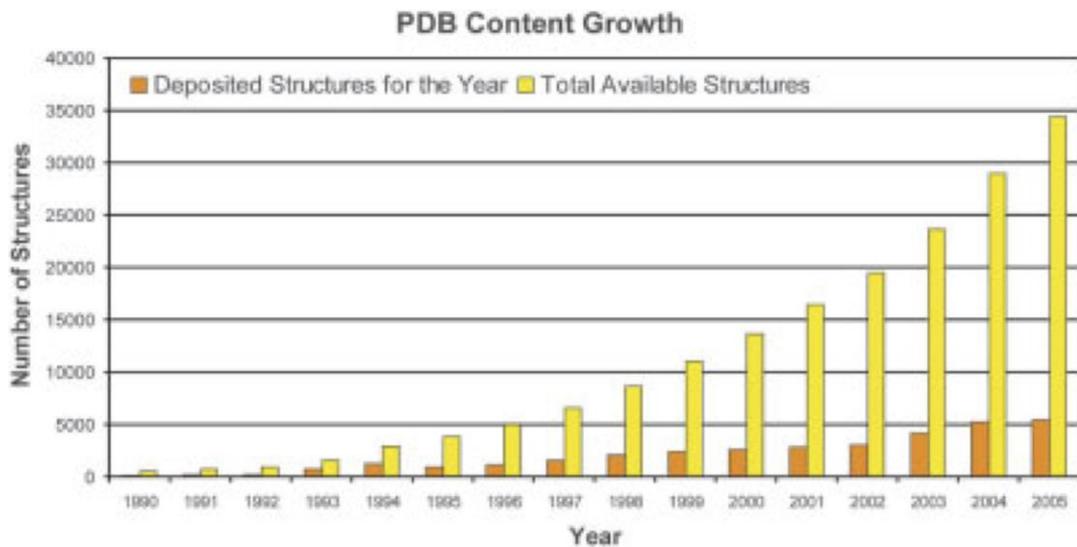


Figura 2.2 – Quantidade de estruturas disponíveis no Protein Data Bank do longo do tempo

Fonte: SOUZA; FERNANDES; RAMOS, 2006.

Quando fala-se sobre Desenvolvimento Racional de Fármacos, também é importante notar a existência de bancos de dados de compostos que podem ser utilizados como ligantes para com as proteínas. Destes pode-se citar o *Cambridge Structural Database (CSD)* (MORGON, 2007, p. 491).

O CSD continha já em 2002 mais de 250 mil modelos de pequenas moléculas. Nesta data o banco era utilizado em 109 empresas e 826 instituições acadêmicas em 58 países. O uso do CSD como uma ferramenta de pesquisa acadêmica, e o advento de métodos de química computacional em muitas empresas farmacêuticas e agroquímicas fez com que a instituição crescesse rapidamente na década de 1980, e em 2002 ela já contava com 45 colaboradores em tempo integral (ALLEN, 2002, p. 381).

O principal uso do CSD na descoberta de novas drogas é “utilizar as estruturas de pequenas moléculas lá armazenadas e como um recurso fértil de dados geométricos em fragmentos moleculares para calibração de campos de força e validação de resultados de computação química” (OPREA, 2004, p. 4 apud ALLEN; MOTHERWELL, 2002, p. 407-422, tradução nossa). Uma das principais maneiras para desenvolver novas drogas utilizando as estruturas disponíveis em bancos de dados como o CSD é através da docagem molecular, objeto do próximo capítulo.

### 3 ATRACAMENTO

Neste capítulo será abordado o tema atracamento, também conhecido como docagem, ou *Docking*, em inglês. Esta é uma ferramenta de grande importância para o planejamento racional de fármacos. É através da docagem que é avaliada a possibilidade de um ligante se encaixar em um receptor, e quando esta ligação acontece, processos de biotransformação se tornam possíveis.

#### 3.1 Docagem Molecular

Docagem Molecular é uma ferramenta importante para a descoberta de novas drogas. Esta ferramenta tenta encaixar a proteína em um ligante de várias maneiras diferentes, gerando uma energia livre para cada encaixe em uma orientação/posição diferente (HUANG; ZOU, 2007, p. 399 apud BROOIJMANS; KUNTZ, 2003, p. 335-373). Quanto menor esta energia, mais estável é a conformação, ou seja, melhor é o encaixe (ver Figura 3.1).

Para Shoichet et al. (2002, p.439, tradução nossa), a promessa da docagem é:

A promessa da docagem é que a estrutura do alvo vai prover um modelo para a descoberta de novos ligantes, diferente daqueles já conhecidos. O processo começa com um banco de dados de compostos e a estrutura do receptor de interesse e pergunta, “Dos compostos na base de dados, qual é mais suscetível a se ligar ao receptor?”.

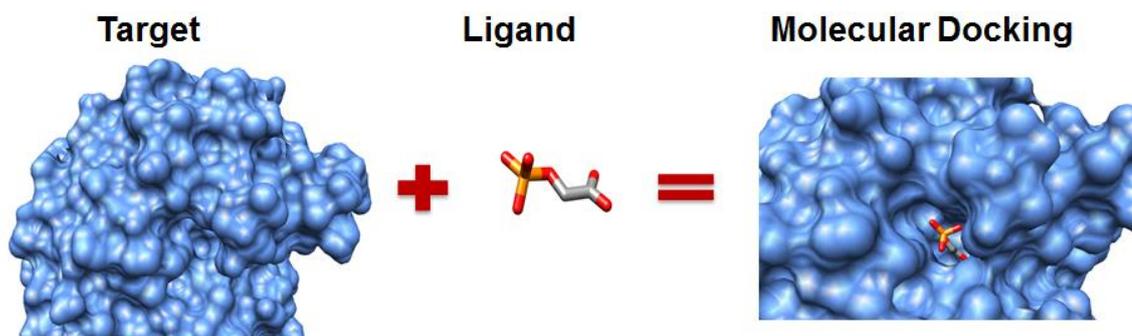


Figura 3.1 – Encaixe de ligante a uma proteína.

Fonte: HERNÁNDEZ-SANTOYO, 2013.

Huang e Zou (2010) definiram o processo de docagem:

Um programa de docagem proteína-ligante consiste de dois componentes essenciais, amostragem e pontuação. Amostragem refere-se a geração de possíveis encaixes de diferentes posições/orientações dentro de um local de encaixe da proteína e pode ser dividida em dois aspectos, amostragem do ligante e flexibilidade da proteína. Pontuação é a predição de quão firme é o encaixe para cada posição/orientação com uma função de energia física ou empírica. A melhor posição/orientação, nominalmente a com a menor pontuação energética, é o encaixe previsto.

Uma questão importante no processo de docagem é considerar a flexibilidade não só do ligante, mas do receptor (COHEN, 2010, p. 4). Conforme Najmanovich e colaboradores (NAJMANOVICH et. al., 2000), docagens com receptores rígidos predisseram ligações incorretas para 50-70% dos casos. Por outro lado, os inúmeros graus de liberdade envolvidos na incorporação da flexibilidade do receptor durante a docagem tem posto desafios computacionais de grande monta (B-RAO et. al. 2009, p. 394). Deste modo diferentes abordagens tem sido desenvolvidas para considerar a flexibilidade conformacional do receptor levando em conta estes desafios. Algumas destas são o *soft docking*, o uso de bibliotecas de rotâmeros de cadeias laterais e estruturas de múltiplos receptores (COHEN, 2010, p. 6). Esta última apresenta grande complexidade computacional. Uma abordagem que considera estruturas de múltiplos receptores mas com um custo moderado é o *ensemble docking* (B-RAO et. al. 2009, p. 395). Outras técnicas e o processo podem ser vistos na figura 3.2.

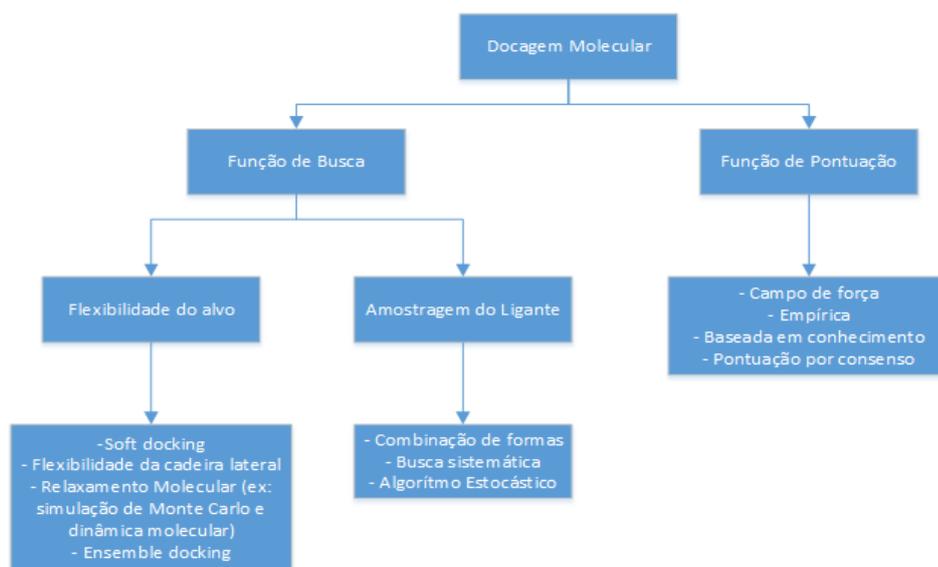


Figura 3.2 – Processos de Docagem Molecular

Fonte: adaptado de HERNÁNDEZ-SANTOYO, 2013

### 3.2 AUTODOCK

Uma das suítes utilizadas para docagem molecular é a Autodock. “Ela é desenhada para prever como pequenas moléculas, como substratos ou candidatos a medicamentos, docama um receptor com estrutura 3D conhecida” (MORRIS, 2013, tradução nossa). Além disso, dentre as ferramentas disponíveis ela é uma das mais utilizadas pela comunidade científica. Na Figura 3.3 é possível ver o percentual de citações de cada suíte de docagem, em 2005. Utilizando uma ferramenta bastante difundida, a aplicação resultado desta pesquisa tem maiores chances de aceitação na comunidade, e como será liberada com código fonte aberto, maior chance de receber contribuições futuras.

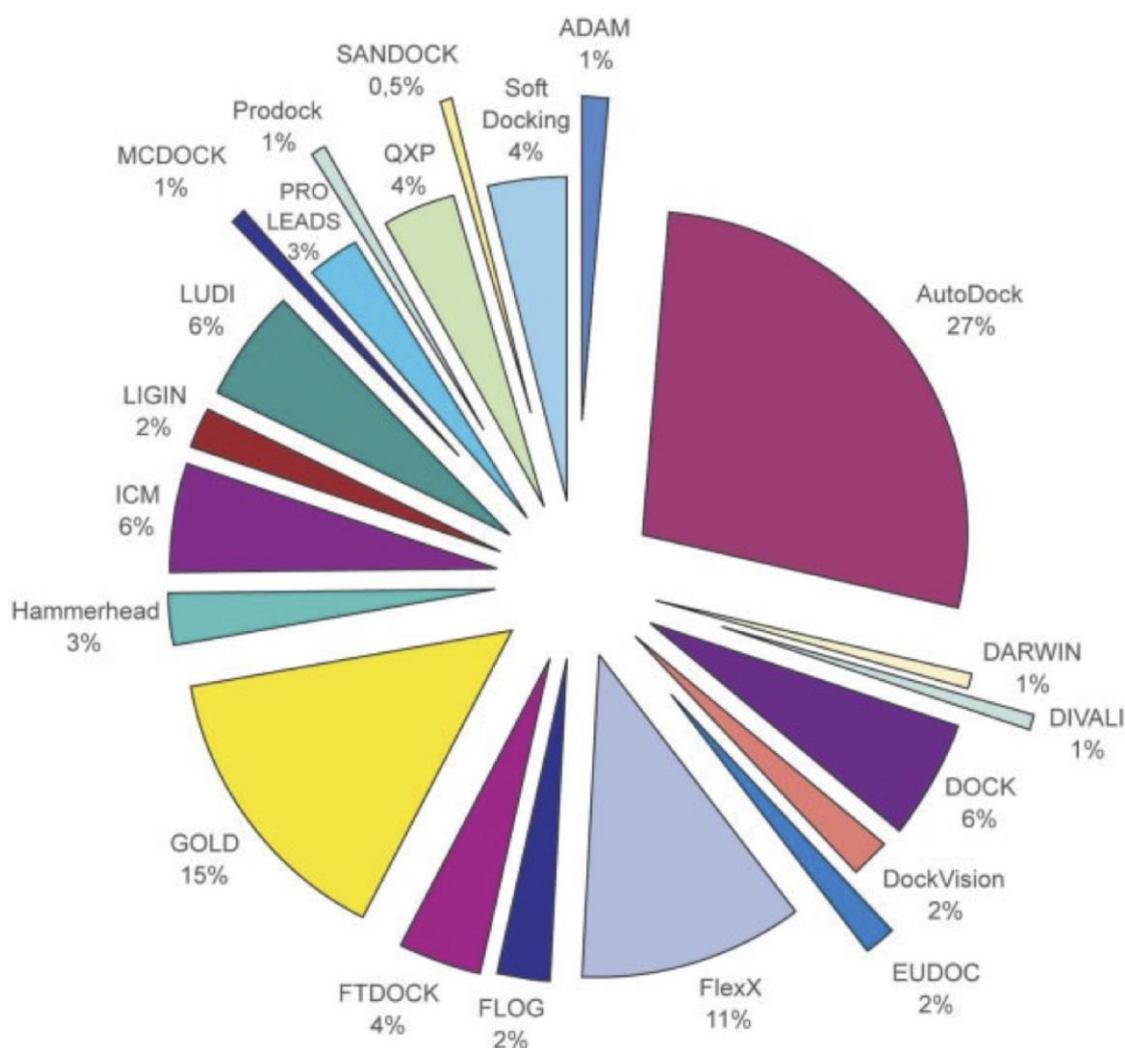


Figura 3.3 – Número de citações dos programas de docagem molecular

Fonte: SOUZA; FERNANDES; RAMOS, 2006.

O Autodock por si próprio não possui interface gráfica, apenas ferramentas de linha de comando. Existem sim outras ferramentas, como a AutoDockTools que trabalham juntamente com o Autodock e tem interface gráfica (MORRIS et al., 2009).

Segundo o manual do Autodock, a docagem é realizada em quatro passos:

Passo 1: Preparação do arquivo de coordenadas. Comumente os arquivos do ligante e do receptor estão no formato PDB. Para realizar docagens, é necessário criar um arquivo PDB estendido com informações cargas dos átomos e tipos. Este arquivo estendido tem a extensão PDBQT. A suíte de utilitários auxiliares para o Autodock, o AutoDockTools possui os utilitários *prepare\_ligand4* e *prepare\_receptor4*, que recebem arquivos PDB do ligante e do receptor respectivamente, e calculam as informações necessárias gerando como saída arquivos PDBQT.

Passo 2: Calcular AutoGrid. Para otimizar o processo de docagem, deve ser calculado previamente as afinidades para cada tipo de átomo. Dentro da suíte Autodock existe a aplicação AutoGrid, que realiza este trabalho. O parâmetro para o AutoGrid é um arquivo GPF, que contém os parâmetros para o cálculo da grid de afinidades. Para a criação deste arquivo a suíte auxiliar AutoDockTools tem a ferramenta *prepare\_gpf4*, que recebe os arquivos PDBQT do ligante e do receptor como parâmetro, gerando um arquivo GPF com valores padrão para o cálculo da grid.

Passo 3: Docagem utilizando AutoDock. Após o cálculo da grid, o próximo passo é a docagem propriamente dita. A docagem recebe uma série de parâmetros, como por exemplo quantas docagens devem ser tentadas. Estes parâmetros são passados através de um arquivo DPF. Mais uma vez a suíte AutoDockTools possui uma ferramenta para auxiliar na criação deste arquivo de parâmetros, a *prepare\_dpf4*. Assim como no caso do AutoGrid, esta ferramenta cria um arquivo DPF com valores padrão para o ligante e receptor em questão. A chamada do AutoDock dá como resultado um arquivo PDBQT com a docagem realizada. Este arquivo irá conter as coordenadas do ligante e do receptor já docados, além de informações sobre a docagem, como a energia livre.

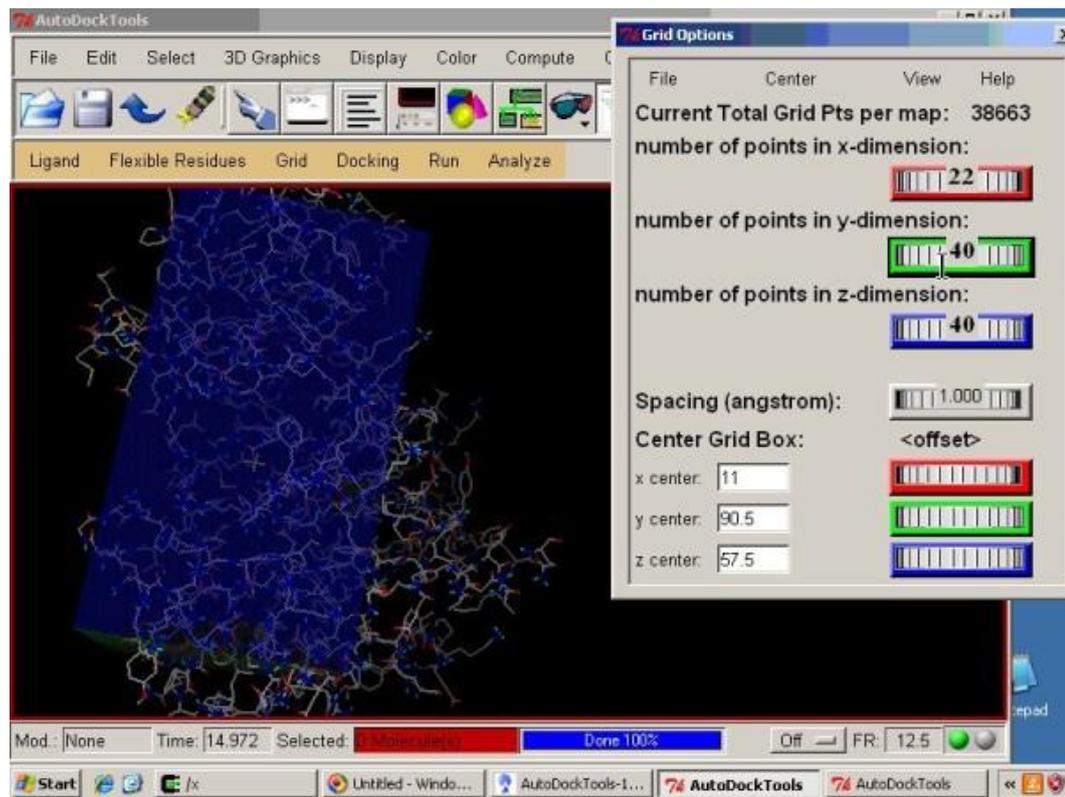


Figura 3.4 – AutoDockTools como ferramenta de avaliação do sucesso da docagem

Fonte: AUTODOCK VINA MANUAL.

Passo 4: Realizar análise do resultado utilizando AutoDockTools. Com as ferramentas inclusas no AutoDockTools, é possível visualizar representações gráficas do resultado da docagem, além de realizar análises qualitativas do resultado. Um exemplo de análise gráfica provida pelo AutoDockTools pode ser vista na Figura 3.4.

### 3.3 Ensemble Docking

*Ensemble Docking* é uma técnica que ajuda a trabalhar a solução das duas maiores dificuldades da docagem molecular: o sistema de pontuação e flexibilização de proteínas. Proteínas não são estruturas estáticas, nem os ligantes. A técnica de *Ensemble Docking* utiliza um corpo fixo, pois testes demonstraram que utilizar ligantes flexíveis não apresentam resultados qualitativos significantes na docagem. Para representar um receptor flexível, no *Ensemble Docking* são utilizadas múltiplas estruturas de proteínas (HUANG; ZOU, 2007, p. 399-401).

O Autodock incorpora apenas um tipo de flexibilização: a torção de cadeias laterais. Estas sim são incorporadas durante a simulação, girando dentro dos níveis de liberdade

permitidos. O que este trabalho visa alcançar, é um outro tipo de flexibilidade: incorporar possíveis mudanças na estrutura da proteína, pelo fato de ela mudar sua estrutura por estar inserida dentro de um ambiente aquoso. Existem ferramentas para calcular a hidrofobicidade de uma proteína, como a GPMAW (PERI; STEEN; PANDEY, 2001), mas nenhuma delas foi incluída na aplicação construída, pois o objetivo da pesquisa é construir uma ferramenta que aplique a dinâmica molecular independente da hidrofobicidade.

Uma maneira de realizar-se *Ensemble Docking* é utilizar o programa Gromacs para executar a dinâmica molecular, simulando a flexibilidade da proteína. A saída dada pelo Gromacs são várias poses, geradas através da simulação. Cada uma destas poses representa um possível estado da proteína, pois como já foi dito ela não é uma estrutura estática, e sim flexível. Para que o processo de *Ensemble Docking* esteja completo, esta série de poses será utilizada como entrada para o programa Autodock, que irá simular a docagem para cada uma das posições. Cada uma destas poses poderá se docar, ou encaixar, de maneira melhor ou pior com o ligante. O resultado esperado pode ser visto na imagem Figura 3.5. O processo de dinâmica molecular é descrito com maior detalhamento no próximo capítulo.

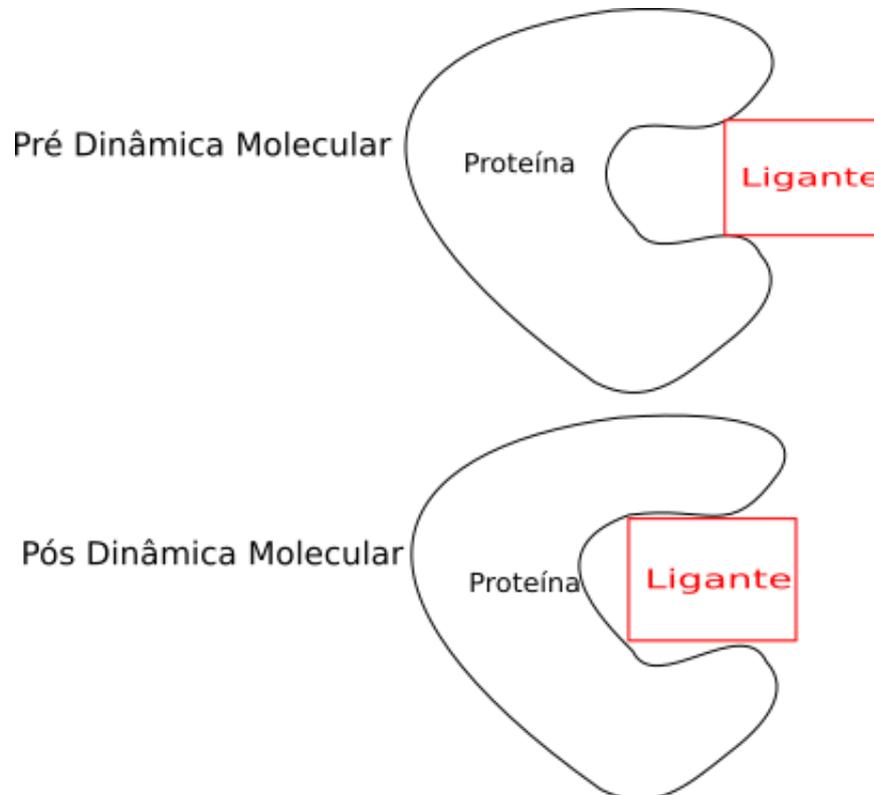


Figura 3.5 – Encaixe de ligante após flexibilização da proteína

## **4 DINÂMICA MOLECULAR**

Dinâmica Molecular é a simulação computacional dos movimentos físicos de átomos e moléculas. O interesse em especial neste trabalho é a simulação do comportamento de uma proteína ao ser imersa em um fluido. Como já falado anteriormente, proteínas não são estruturas rígidas, e o seu formato muda ao ser imersa em um líquido. Este capítulo trata de como a dinâmica molecular é utilizada para simular este comportamento.

### **4.1 Introdução**

Dinâmica molecular dinamiza a simulação de sistemas moleculares, envolvendo a solução da equação de movimento de Newton em pequenos espaços de tempo. Estas simulações nos permitem ter modelos de moléculas muito mais próximos da realidade biológica, pois inclui diretamente características como a flexibilidade molecular (através da variação temporal de propriedades) e a temperatura (através da aceleração dos átomos). A área de especial interesse para a pesquisa aqui realizada é a química computacional, que estuda química experimental, onde os problemas são resolvidos através de métodos computacionais.

Aplicando-se as leis da Natureza de movimento dos átomos é possível se chegar a um conjunto (ou ensemble) de configurações. A Dinâmica Molecular gera este conjunto de configurações para uma trajetória no tempo, através da solução das equações de Newton (VAN GUNSTEREN; BERENDSEN; JC, 1990, p. 992-1000).

Uma das principais ferramentas para simulação de dinâmica molecular é o pacote de programas GROMACS (VAN DER SPOEL, 2005). Ele provê uma ferramenta versátil, eficiente e de código aberto para estas simulações, especialmente dirigida para simulação de moléculas biológicas, como proteínas, em ambientes aquosos.

### **4.2 Flexibilidade de proteínas**

Segundo Voet, Voet e Pratt (2014), “a forma estática que as estruturas das proteínas são normalmente retratadas pode deixar a falsa impressão de que proteínas têm estruturas fixas e rígidas”. Na verdade proteínas são estruturas flexíveis, podendo ter várias estruturas com a mesma estabilidade. Estas variações de formação possibilitam mais oportunidades de

um ligante encaixar-se a proteína, pois as proteínas e ligantes são por natureza estruturas tridimensionais, que não podem ocupar o mesmo espaço físico ao mesmo tempo. A Figura 4.1 mostra as variações na conformação da proteína e de dois ligantes.



Figura 4.1 – Variações de proteínas e ligantes

Fonte: VOET, VOET, PRATT, 2014.

Um outro exemplo desta mobilidade pode ser visto no trabalho apresentado por Schroeder et al. (2005), que demonstra a flexibilidade da enzima InhA. A Figura 4.2 demonstra uma representação desta flexibilidade, onde cada cor de conformação da InhA representa a conformação da proteína em um instante de tempo diferente.

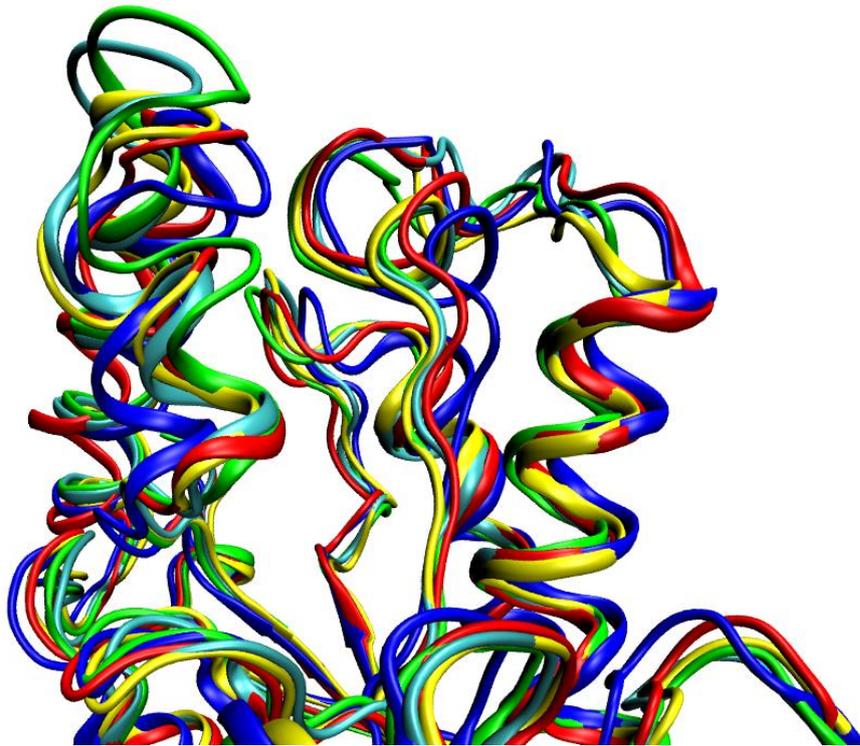


Figura 4.2 – Trajetórias de uma proteína durante o processo de dinâmica molecular

Fonte: COHEN, 2010.

Nesta figura, há um exemplo da flexibilidade da proteína InhA, através da sobreposição de diferentes poses tiradas ao longo de uma simulação por dinâmica molecular. Em azul a estrutura cristalina (tempo zero = início da trajetória, código PDB: 1ENY); em vermelho aos 500 ps da trajetória; em amarelo aos 1000 ps; em verde aos 1500 ps e em ciano 2000 ps. Para melhor representação das diferentes conformações, a figura mostra apenas a porção correspondente ao “encosto” da estrutura (COHEN, 2010).

### 4.3 GROMACS

A ferramenta GROMACS (VAN DER SPOEL, 2005) é uma das mais conhecidas para docagem molecular. O seu nome vem do inglês *GRO*ningen *MA*chine for *C*hemical *S*imulation, que em português se traduz como “máquina para simulação química de Groningen”. Groningen é a universidade onde o GROMACS foi construído.

O principal objetivo do GROMACS é prover um programa versátil e eficiente para execução de dinâmica molecular, direcionado especialmente para simulações de moléculas biológicas em ambiente aquoso (van der Spoel, 2005). A Figura 4.3 é um bom exemplo do que é possível alcançar com simulações de comportamentos de substâncias hidrofóbicas utilizando GROMACS. Neste simulação, lipídios (DPPC) foram inseridos dentro de água, numa proporção de uma molécula de lipídio para cada 12 moléculas de água, em posições aleatórias. Ao longo da simulação é possível observar que as moléculas de lipídio diminuem a superfície de contato com a água agrupando-se, comportamento esperado para uma substância hidrofóbica.

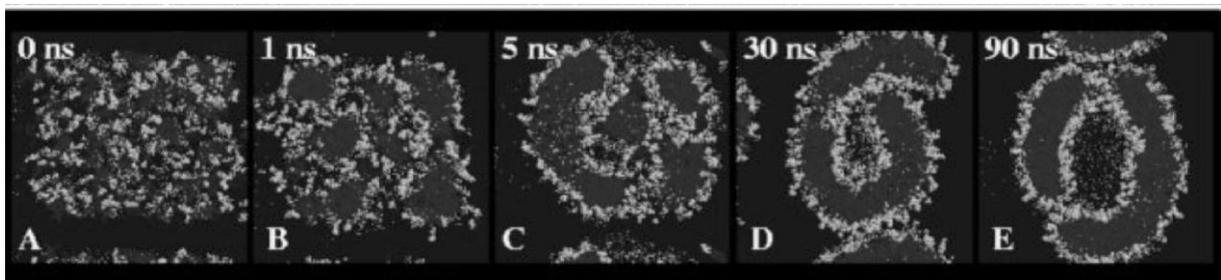


Figura 4.3 – Simulação de dinâmica molecular utilizando GROMACS com lipídios inseridos em água

Fonte: adaptado de VAN DER SPOEL, 2005.

Na prática, o GROMACS consiste em uma série de utilitários chamáveis por linha de comando. Estes utilitários devem ser chamados em uma ordem já estabelecida, pois os arquivos de entrada do passo posterior são gerados pelo utilitário do passo anterior.

Segue abaixo fluxo de chamada:

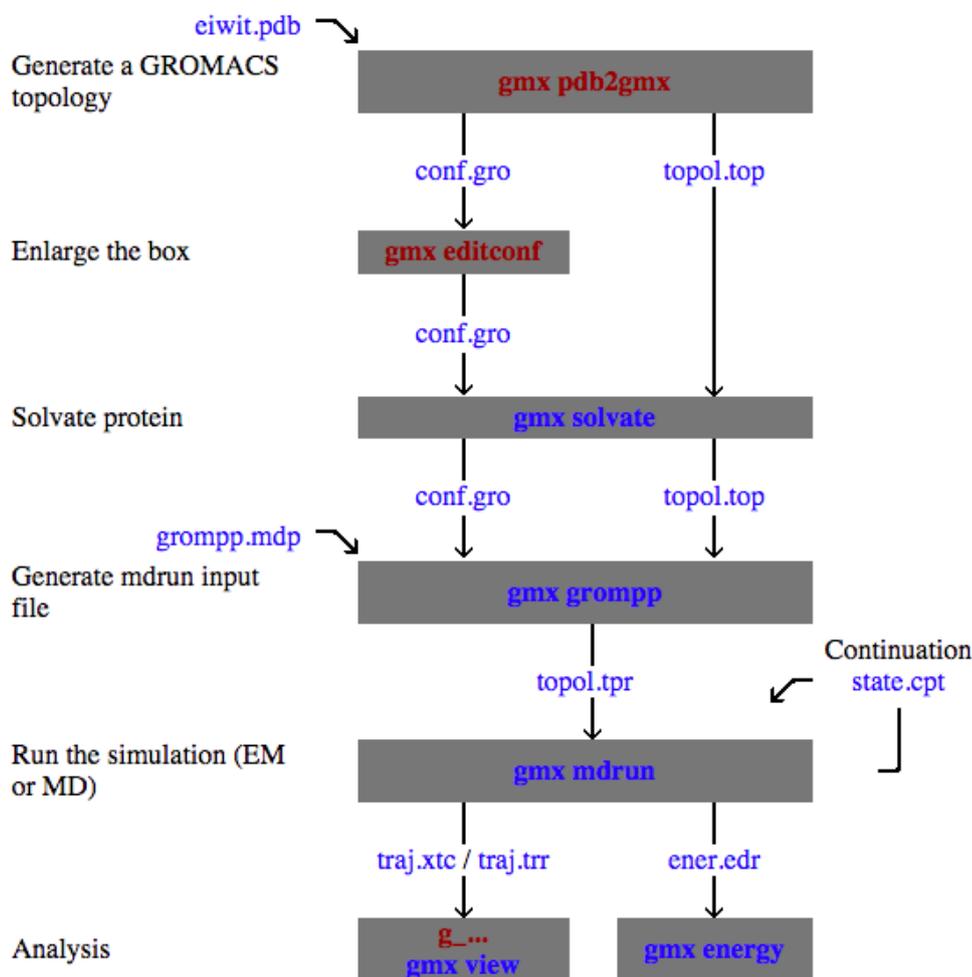


Figura 4.4 – Passos para execução de um processo de dinâmica molecular através do GROMACS

Fonte: VAN DER SPOEL, 2014.

Conforme a Figura 4.4, o primeiro passo para realizar a dinâmica molecular é a chamada do utilitário `pdb2gmX`. Este utilitário lê um arquivo no formato *PDB* (*Protein Data Bank*), obtido por exemplo no site <http://www.rcsb.org/>, e gera dois arquivos de saída: um arquivo GRO com as coordenadas de cada átomo, e um arquivo TOP com a topologia.

O arquivo GRO gerado pelo utilitário `pdb2gmX` é utilizado como parâmetro para o utilitário `editconf`, que por sua vez aumenta o tamanho da caixa em que a dinâmica molecular

irá trabalhar. Um dos principais parâmetros para este utilitário é a distância entre o a solução (no nosso caso a proteína), e as paredes da caixa.

O próximo passo é chamar o utilitário *solvate* (ou *genbox* no Gromacs 4). Este aplicativo serve para solvatar o composto em um solvente, como a água. O solvente pode ser especificado e é o parâmetro mais importante deste utilitário.

Após o composto ter sido solvatado, o utilitário *grompp* deve ser chamado. É no *grompp* que os principais parâmetros da dinâmica molecular devem ser passados, pois o seu objetivo é compilar estes parâmetros em um arquivo TPR, que será utilizado para a execução da dinâmica molecular. O utilitário *grompp* tem vários parâmetros, a maioria deles sendo passados através de um arquivo de parâmetros com extensão *MDP*.

Para executar a dinâmica molecular propriamente dita, é utilizado o programa *mdrun*. Ele recebe como parâmetro o arquivo TPR gerado pelo *grompp*. Como a maior parte dos parâmetros para a simulação foi compilada no arquivo TPR, existem poucos parâmetros significativos. A saída do *mdrun* são os arquivos TRR (trajetória da simulação), GRO (conformação final), e EDR (energia).

Após realizada a dinâmica molecular, é necessário analisá-la para obter informações sobre o seu resultado. Existem utilitários do pacote GROMACS como o *view* e o *energy* para este fim. Outro utilitário muito importante é o *trjconv*, que exporta as poses da trajetória da dinâmica molecular para formato PDB.

A dinâmica molecular é um dos componentes principais do processo de *ensemble docking*. Os outros componentes utilizados na ferramenta estão descritos no próximo capítulo, juntamente com a metodologia utilizada.

## 5 METODOLOGIA DE IMPLEMENTAÇÃO

Neste capítulo serão descritos os métodos, ferramentas e tecnologias utilizadas no desenvolvimento da aplicação. Serão descritos detalhes, como *frameworks* utilizados, e diagramas UML sobre a ferramenta construída. Aqui serão também detalhados os motivos da utilização das tecnologias empregadas.

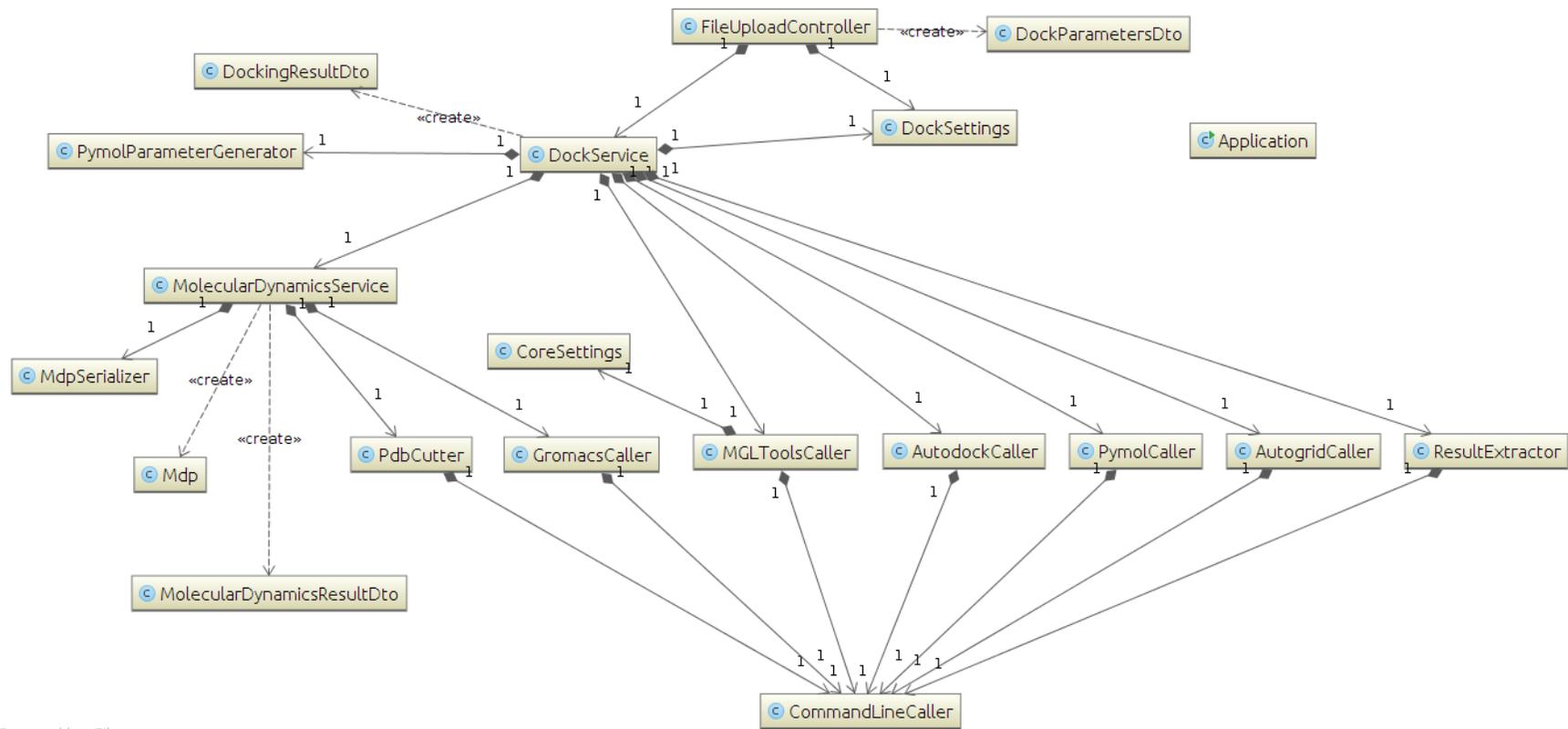
### 5.1 Tecnologias empregadas

Para cumprir o objetivo de ser uma ferramenta genérica e que seja de fácil utilização, foi criada uma aplicação Java com interface web. O principal motivo para a escolha do desenvolvimento de uma ferramenta web, e não *desktop*, é a facilidade de uso para o usuário. Utilizando tecnologias como a biblioteca Swing do Java, seria necessário que o usuário instalasse a ferramenta no seu próprio computador, além de impor certos requisitos como a instalação da JRE Java. Através de uma ferramenta web, é possível criar-se uma interface rica, de fácil utilização, e acessível da web caso necessário no futuro.

Além disso, é possível observar o sucesso da ferramenta online *BLAST: Basic Local Alignment Search Tool* (ALTSCHUL et al, 1997). Ela permite realizar alinhamento locais em uma ferramenta disponível a toda a comunidade online. Basta realizar o *upload* dos arquivos através do browser que o processo é iniciado. É com este nível de facilidade de uso que a ferramenta de *ensemble docking* está sendo construída.

### 5.2 Diagrama de classe

O diagrama de classe é a representação da estrutura na forma gráfica das classes implementadas e/ou a serem desenvolvidas. Na Figura 5.1 há o modelo gráfico das classes implementadas na aplicação para realização de *Ensemble Docking*. Este modelo não inclui as classes de teste unitário.



Powered by yFiles

Figura 5.1 – Diagrama de Classes

Fonte: Próprio autor

### 5.3 Diagrama de atividades

O diagrama de atividades é a representação do fluxo de execução do programa. A Figura 5.2 é um modelo do fluxo a ser executado em um processo de *Ensemble Docking*.

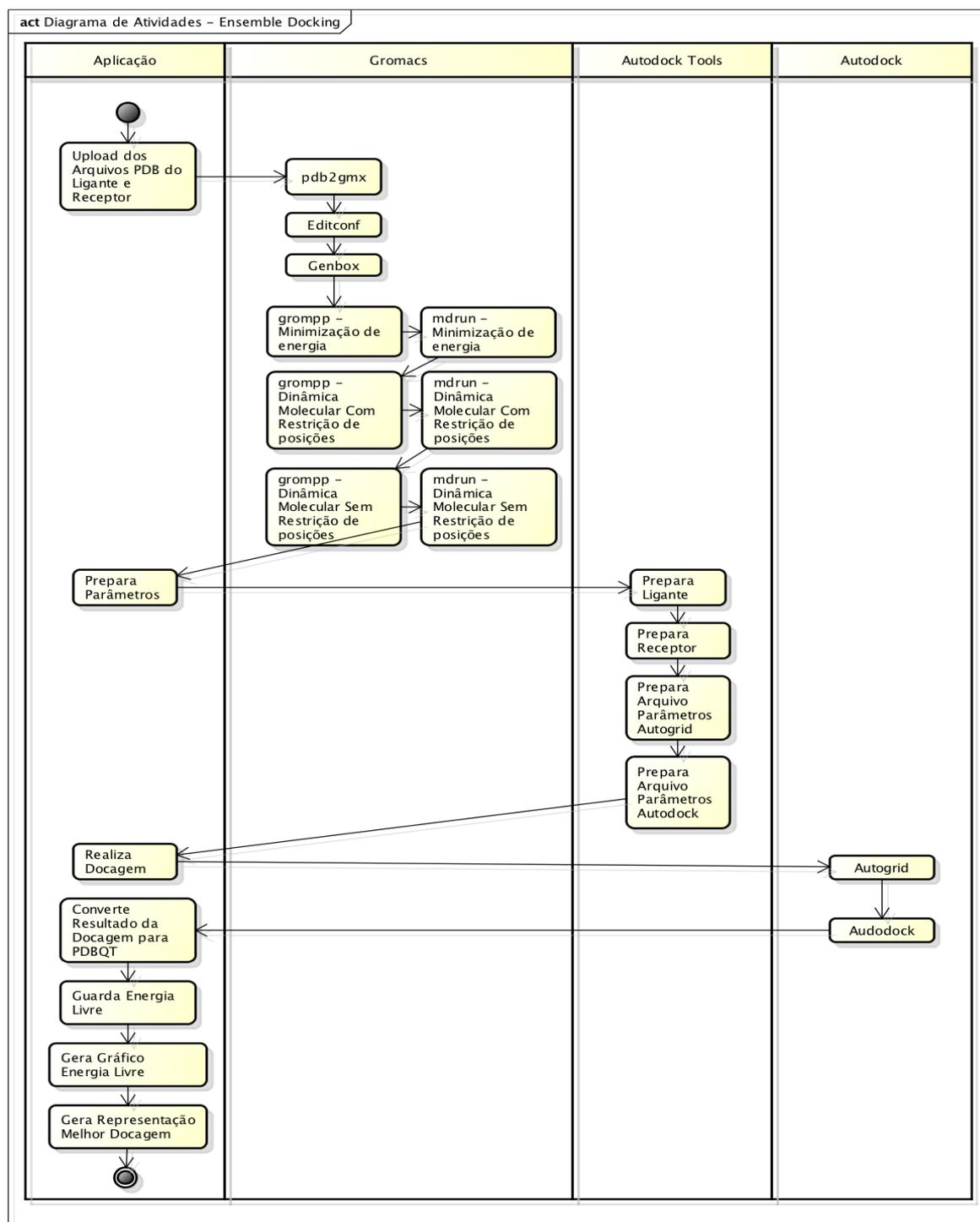


Figura 5.2 – Diagrama de Atividades

Fonte: Próprio autor

## 5.4 Diagrama de caso de uso

O diagrama de caso de uso descreve a funcionalidade de um sistema, sendo uma ferramenta adequada para descrever os requisitos funcionais de um sistema. A Figura 5.3 é um modelo do caso de uso para execução de *Ensemble Docking*. O modelo tem poucas interações por parte do pesquisador, mas a chamada inclui várias funcionalidades, desde a entrada de parâmetros, passando pela realização da dinâmica molecular e das docagens, até a geração dos gráficos para exibição dos resultados.

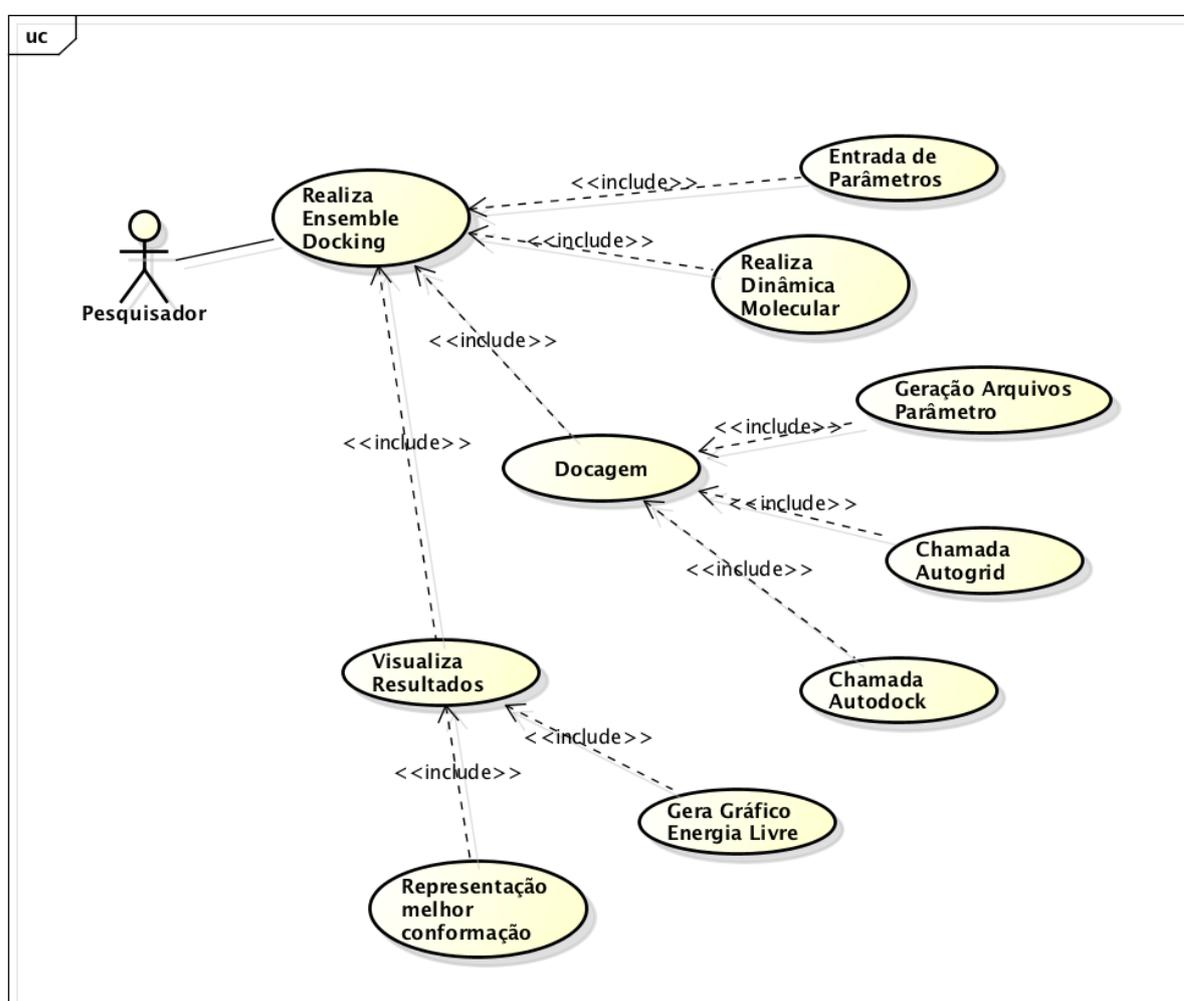


Figura 5.3 – Caso de uso

Fonte: Próprio autor

## 5.5 Java

No trabalho será utilizada a linguagem Java para o desenvolvimento da interface de interação do usuário. Especificamente será utilizado Java para WEB. Nesta seção serão descritas as vantagens de utilizar esta linguagem, além das estratégias utilizadas.

### 5.5.1 A linguagem Java

A linguagem Java iniciou no ano de 1991. Uma das suas principais promessas era possibilitar que o mesmo código rodasse em várias plataformas. A primeira versão foi lançada pela Sun em 1995. Desde lá a linguagem evoluiu até a sua última versão, Java SE 8, que foi lançada em 18 de março de 2014.

Java é uma linguagem orientada a objeto e fortemente tipada. Ao executar a compilação, é gerado um código chamado de Bytecode. Este código é executado pela *Java Virtual Machine* (JVM). Cada sistema operacional ou plataforma possui a sua JVM que irá interpretar e executar o código em Bytecode, possibilitando assim que o código compilado rode em qualquer plataforma (DEITEL, 2014).

A linguagem Java pode ser dividida em várias plataformas, dentre as quais as de interesse para este trabalho são a Java SE, que é a base da linguagem Java, e a Java EE, que permite dentre outras coisas que o código Java seja utilizado como um servidor web provendo páginas dinâmicas. Para isso, conforme é definido na especificação da plataforma, é necessário utilizar um container web, tal como Apache Tomcat ou Jetty (HANS, 2003).

### 5.5.2 Java para Web

A linguagem Java permite a criação de conteúdo dinâmico na web através de *Servlets* e *Java Server Pages* (JSP). Como a utilização dos Servlets nativos é bastante onerosa, existem várias bibliotecas que estendem os Servlets nativos da linguagem Java, facilitando o desenvolvimento, dentre eles pode-se citar *JSF*, *Struts* e *Spring MVC* (HANS, 2003). Dentre os citados, neste trabalho será utilizado Spring MVC, pois é sabido que ele tem a maior facilidade de configuração e a melhor performance.

Mesmo com a utilização de Spring MVC, a utilização da tecnologia *JSP* ainda é possível. JSP contém *tags* de marcação padrão, como *tags* HTML, mas além disso possui *tags* que permitem a inserção de conteúdo dinâmico na página. É possível, por exemplo, realizar

uma busca em um banco de dados no servidor Java, e exibir o resultado da consulta para o usuário através de atributos inseridos na página JSP (HANS, 2003). A relação entre JSP e Spring MVC acontece no momento em que o Spring MVC mapeia endereços para certos controladores, que por sua vez possuem este tipo de lógica e podem indicar qual JSP deve ser renderizado para o usuário, substituindo valores previamente marcados no JSP por conteúdo dinâmico.

Com o Spring MVC, é possível utilizar JSPs para a renderização do HTML e chamadas Ajax para futuras atualizações de dados na tela sem a necessidade de recarregar a página. Estas chamadas Ajax nada mais são do que chamadas HTTP através de código Javascript que é executado no navegador do usuário (GARRET, 2005). Estas chamadas HTTP buscam informações no servidor, no caso processadas por controladores do Spring MVC, e recebem o resultado sem a necessidade de carregar a página novamente. Desta forma é possível atualizar o conteúdo em tela com estas informações, proporcionando uma resposta mais rápida ao usuário e uma melhor experiência de uso.

### **5.5.3 Spring Boot**

Uma das principais bibliotecas utilizadas na aplicação foi a Spring Boot. Ela inclui uma série de bibliotecas previamente testadas em conjunto como sua dependência. Além disso, ela possui um grande enfoque em diminuir o tempo de configuração de um projeto Java Web, o que ajudou o projeto a cumprir mais objetivos dentro do prazo previsto.

Uma das maiores funcionalidades do Spring Boot é a facilidade em incluir um servidor *Java Servlet* embutido na aplicação. Através desta técnica não é necessário que a aplicação Java web seja incluída dentro de um recipiente web, o que acontece é que o recipiente está embutido dentro do pacote final, possibilitando ser rodado como um executável. O Spring Boot permite a utilização do Apache Tomcat ou do Jetty como servidor de aplicação embutido, na ferramenta construída optou-se pelo Jetty.

### **5.5.4 Chamada de linha de comando**

Uma dos motivos para a escolha da linguagem Java para implementação da ferramenta é a sua integração com ferramentas de linha de comando. Como é necessário

chamar várias aplicações através da linha de comando, tais como Gromacs, Autodock e Pymol, este tipo de integração é muito importante.

A integração de um programa com outros programas que rodam em linha de comando traz alguns desafios. O primeiro é a sincronização da saída de console destes programas. Foi identificado durante o desenvolvimento que alguns programas precisam ter a sua saída lida para que a sua execução siga normalmente, pois provavelmente efetuam a saída para o console de maneira síncrona. Já outros programas não apresentaram esta dificuldade.

É sabido que fazer a aplicação Java depender de aplicações instaladas na máquina gera uma série de dependências, como o caminho de sua instalação e versão. Alguns programas chamados, como o Autodock por exemplo, já foram construídos com enfoque em retrocompatibilidade. Neste caso, dentro do arquivo DPF de parâmetros para a docagem existe um atributo com a versão dos parâmetros, para possibilitar retrocompatibilidade em novas versões do Autodock.

Além disso, foi criado dentro da aplicação um arquivo de parâmetros com extensão *properties*, que é utilizado pela aplicação para uma série de configurações, dentre elas o local de instalação de ferramentas externas.

### **5.5.5 Apache Maven**

A ferramenta de automação de compilação Apache Maven (MAVEN, 2011) foi utilizada para a compilação e resolução de dependências. Conforme Fuxreiter (2010, p.2, tradução nossa), ele é definido da seguinte maneira:

Apache Maven é uma ferramenta de gerenciamento de projetos e compreensão. Baseado no conceito de modelo de Project Object Model (POM). Maven pode gerenciar a compilação do código fonte, do processo de construção do projeto, reportação e documentação à partir de um arquivo de configuração central. [...] Projetos Java podem ser automaticamente construídos de uma maneira a cumprir os requisitos OSGi através do plugin de empacotamento do Maven. Além disso, Maven suporta gerenciamento de dependência através da descarga de artefatos externos de repositórios públicos ou privados por demanda durante o processo de construção e pode publicar aplicações geradas em locais remotos.

No projeto sendo executado, o Apache Maven foi especialmente útil, pois todas as bibliotecas foram gerenciadas através do seu sistema de gerenciamento de dependências. Não foi necessário descarregar manualmente as bibliotecas e incluí-las manualmente no projeto. Para a inclusão das bibliotecas, foram incluídas entradas no arquivo *pom*, que é definido no

formato XML (Extensible Markup Language). Todas as bibliotecas estavam disponíveis no repositório central do Maven, fazendo com que o processo de construção se tornasse agnóstico de ambiente, podendo ser executado em qualquer ambiente com as ferramentas necessárias e acesso à internet.

### 5.5.6 JFreeChart

A do gráfico com coordenadas bidimensionais é feita através da biblioteca JFreeChart (Gilbert, 2002). Para a geração do gráfico, explicado e listrado no Capítulo 6.6.2, foi utilizada a versão 1.0.19 da biblioteca. A biblioteca fornece gráficos de pizza, de barras, de Gantt, plotagem X Y e combinações destes.

Durante o desenvolvimento foram avaliadas outras opções, como bibliotecas de geração de gráficos através de *javascript*, mas a geração no lado servidor foi escolhida pela natureza síncrona da aplicação. Apesar destas bibliotecas terem maiores opções de interatividade, foi avaliado se objetivo principal que é a análise da docagem, seria atingido sem ônus através da utilização desta biblioteca.

## 5.6 Teste unitário

Durante o desenvolvimento foi utilizada a técnica Desenvolvimento Guiado por Testes (Beck, 2003). Foram primeiramente criados testes unitários para cada uma das chamadas a outras ferramentas, então implementado o código de chamada de cada uma das funcionalidades. O resultado foi um código desacoplado e de fácil manutenção. Conforme visto no diagrama de classes, as classes tem papéis bem definidos e estão organizadas por funcionalidades.

A utilização de testes unitários permitiu também diminuir o tempo de desenvolvimento. O processo completo de *ensemble docking* pode levar algumas horas, dependendo dos parâmetros utilizados. Para que não fosse necessário rodar todo este processo no desenvolvimento de uma nova funcionalidade, foram criados arquivos de exemplo de entrada para cada um dos processos, então o teste unitário é rodado para aquele processo em específico, garantindo a sua correta implementação verificando se a saída tem o resultado esperado.

Para chamadas de utilitários externos, como o Gromacs e o Autodock, arquivos de exemplo para a sua chamada foram armazenados dentro do pacote de testes. Estes arquivos eram utilizados como parâmetro para estas chamadas, e ao final passou a ser verificado se o arquivo de saída esperado foi gerado.

## 5.7 Ambiente operacional

Para o desenvolvimento foi utilizado o sistema operacional *Ubuntu*, na sua versão 14.10. A escolha deste sistema operacional como base para o desenvolvimento se deveu a disponibilidade das ferramentas de bioinformática pra ele, em comparação com as alternativas OS X e Windows. As ferramentas Autodock, Gromacs e Pymol podem ser instaladas através do seu gerenciados de pacotes padrão APT: Advanced Packaging Tool.

Como IDE (Integrated Development Environment, ou Ambiente de Desenvolvimento Integrado), foi utilizado o IntelliJ (JETBRAINS, 2003). Esta ferramenta possui integração completo com a linguagem Java, a ferramenta de automação de compilação Apache Maven (MAVEN, 2011), e ao controle de versão Git.

## 5.8 Premissas de execução

Como a aplicação tem uma série de dependências, além do ambiente operacional para desenvolvimento, o ambiente em que a aplicação será instalada deve seguir as seguintes premissas:

Ser utilizado em um sistema Unix com o *shell* no caminho `/bin/sh`.

Ter Java versão 8 ou superior instalado.

Ter a ferramenta Autodock na versão 4.2 instalada.

Ter Gromacs 4.6.5 instalado.

Ter AutoDockTools 1.4.5 instalado e o seu caminho configurado no arquivo de propriedades.

Ter o Pymol versão 1.7.4 instalado.

## 5.9 Controle de versão

Como repositório de código foi utilizado o Git (TORVALDS; HAMANO, 2010), hospedado no serviço Github. O Github foi escolhido por ser um dos mais conhecidos serviços de hospedagem de código aberto. A página do Github é <https://github.com/jonasflesch/ensebledocking>.

A ferramenta está disponibilizada para a comunidade através da licença GNU *General Public License* (Free Software Foundation, 1991). Esta licença foi escolhida pela liberdade que ela dá para utilização da ferramenta e do seu código fonte em projetos futuros, tanto de código aberto como de código fechado.

Para facilitar a contribuição de outras grupos de pesquisa e da comunidade, foi criada uma página com informações sobre como importar o projeto para desenvolvimento. Esta página informativa foi escrita na linguagem *markdown*, que permite criar páginas de documentação com marcação de texto, de maneira a ficar disponível para o público através da web. O resultado pode ser visto na Figura 5.4.

README.md

# Ensemble Docking

Trabalho de conclusão do aluno Jonas Flesch

## Instruções para importação

Este é um projeto Maven. Sempre importe como tal na sua IDE. Passo a passo para importação:

### 1) Instalação das ferramentas de Bioinformática

A aplicação se integra com várias ferramentas, a maioria delas instaláveis através do gerenciados de pacotes da distribuição:

- Gromacs 4.6.5
- Autodock 4.2.5.1
- MGLTools 1.5.6
- Pymol 1.7

### 2) Importar o código fonte

Para importar o código fonte, será necessário ter um cliente git instalado. Para importar à partir do cliente linha de comando, execute o seguinte comando dentro da pasta em que deseja realizar a importação:

```
git clone https://github.com/jonasflesch/enssembledocking.git
```

### 3) Importar para a IDE de Desenvolvimento

#### 3.1) Eclipse

A sua versão do Eclipse deve conter o plugin do Maven e de desenvolvimento Web. Para isso, ao realizar o download do site selecione a opção compatível com Java EE. Após abrir o Eclipse Java EE, importe o projeto como um projeto Maven:

File -> Import -> Maven -> Existing Maven Projects -> Busque o diretório dos fontes criado no passo 2.

#### 3.2) IntelliJ

Para abrir o projeto no IntelliJ, basta ir em Open Project e buscar o diretório onde o projeto foi importado no passo 2.

### 4) Rodar a aplicação

A aplicação foi desenvolvida utilizando o framework Spring Boot, fazendo uso do servidor de aplicação embarcado Jetty. Desta maneira, para rodar a aplicação basta rodar a classe Application, pois ela tem um método main que irá subir a aplicação.

A porta padrão é a 8080, então após subido o servidor basta acessá-la pela url abaixo:

```
http://localhost:8080
```

## Ambiente operacional

A aplicação foi desenvolvida utilizando:

- Ubuntu 14.04 x64
- Java 1.8
- Maven 3.0.5
- IntelliJ IDEA 14
- Gromacs 4.6.5
- Autodock 4.2.5.1
- MGLTools 1.5.6
- Pymol 1.7

Figura 5.4 – Passo a passo para instalação do ambiente de desenvolvimento

Fonte: Próprio autor

## 5.10 Internacionalização

Como um dos principais objetivos da ferramenta é servir para a comunidade científica e ser utilizada como um serviço aberto, se faz coerente a internacionalização da mesma. Foi implementada uma maneira que torna transparente a seleção da linguagem por parte do usuário: a linguagem de preferência do navegador web é utilizada como base para a linguagem em que a aplicação exibe os seus textos.

Na Figura 5.5 é demonstrada a configuração de linguagem do navegador Chrome, onde a língua inglês tem mais prioridade que a língua português.

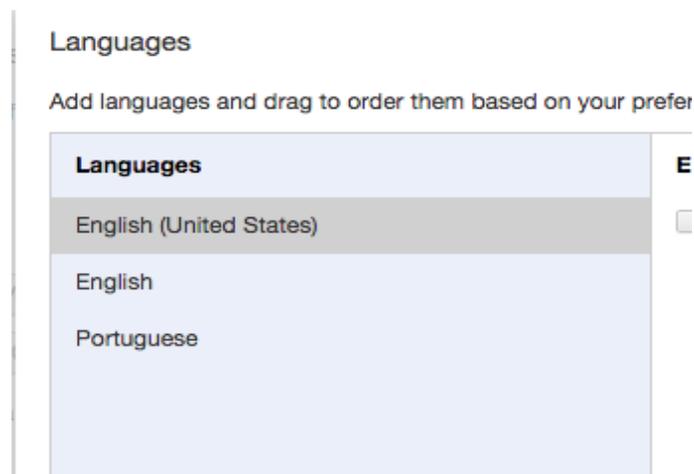


Figura 5.5 – Língua inglês com mais prioridade no navegador Chrome

Fonte: Próprio autor

Um cabeçalho HTTP é passado pelo navegador em todas as requisições com as configurações de linguagem. O cabeçalho em questão é o “Accept-Language”, que passa a ordem de prioridade de cada uma das línguas configuradas pelo usuário. Os cabeçalhos resultantes da configuração demonstrada na Figura 5.5 podem ser vistos na Figura 5.6.

```

▼ Request Headers view source
Accept: text/html,application/xhtml+xml,application/javascript;q=0.9,*/*;q=0.8
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8,pt;q=0.6
Cache-Control: max-age=0
Connection: keep-alive
Cookie: JSESSIONID=k7045gxt3vn149mzejeh2h42
Host: localhost:8080
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/42.0.2866.90 Safari/537.36

```

Figura 5.6 – Cabeçalhos HTTP com língua inglês como prioridade no navegador Chrome

Fonte: Próprio autor

A aplicação foi desenvolvida para considerar o cabeçalho HTTP “Accept-Language”, e conseqüentemente, a configuração realizada pelo usuário no seu navegador. Foram incluídas as línguas inglês e português. Se a língua com mais prioridade não for nem inglês nem português, a aplicação é exibida como textos em inglês. Na Figura 5.7 é possível visualizar a tela inicial da aplicação em inglês, e na figura 5.8 a tela inicial da aplicação em português.

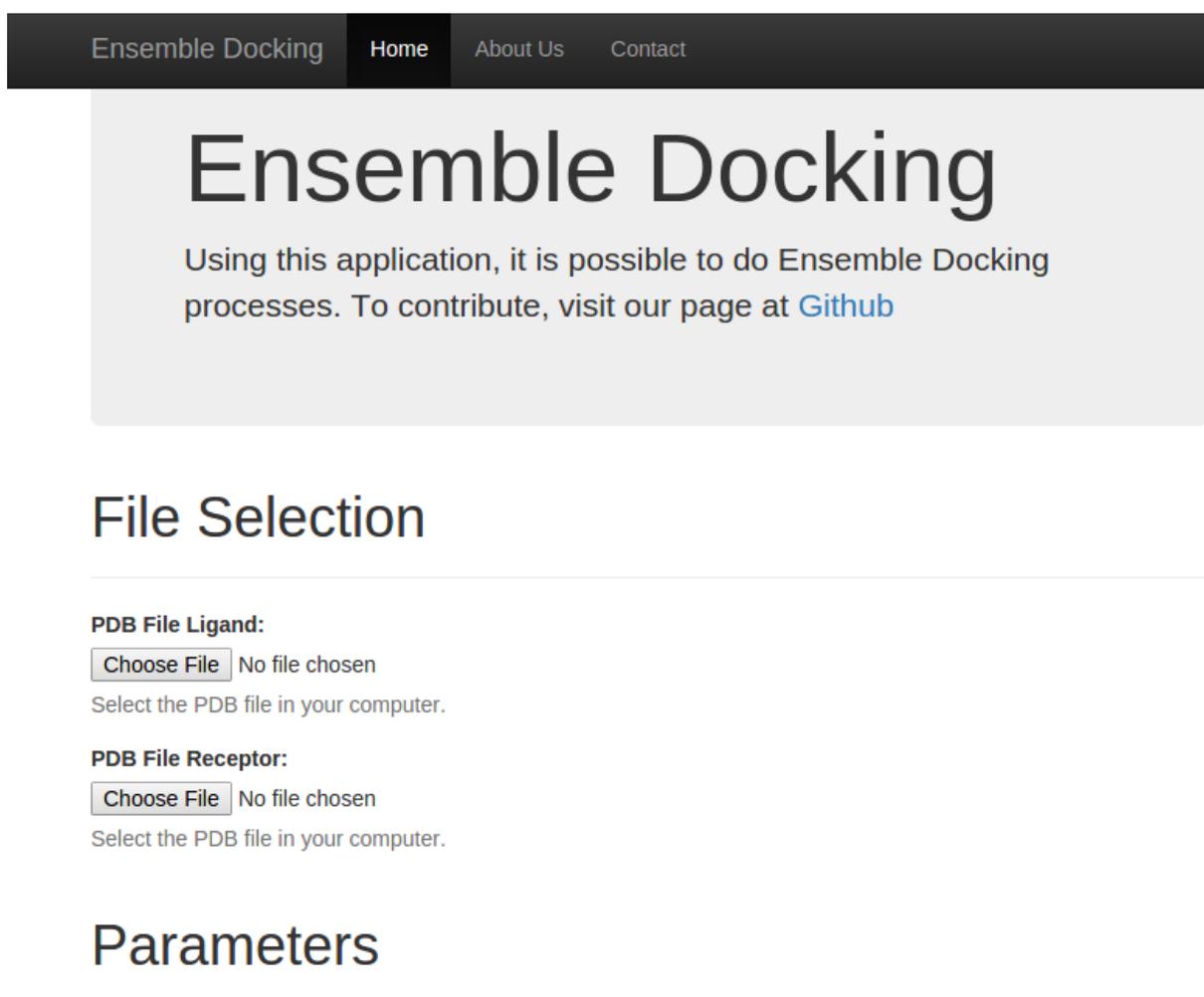


Figura 5.7 – Tela inicial da aplicação em inglês

Fonte: Próprio autor

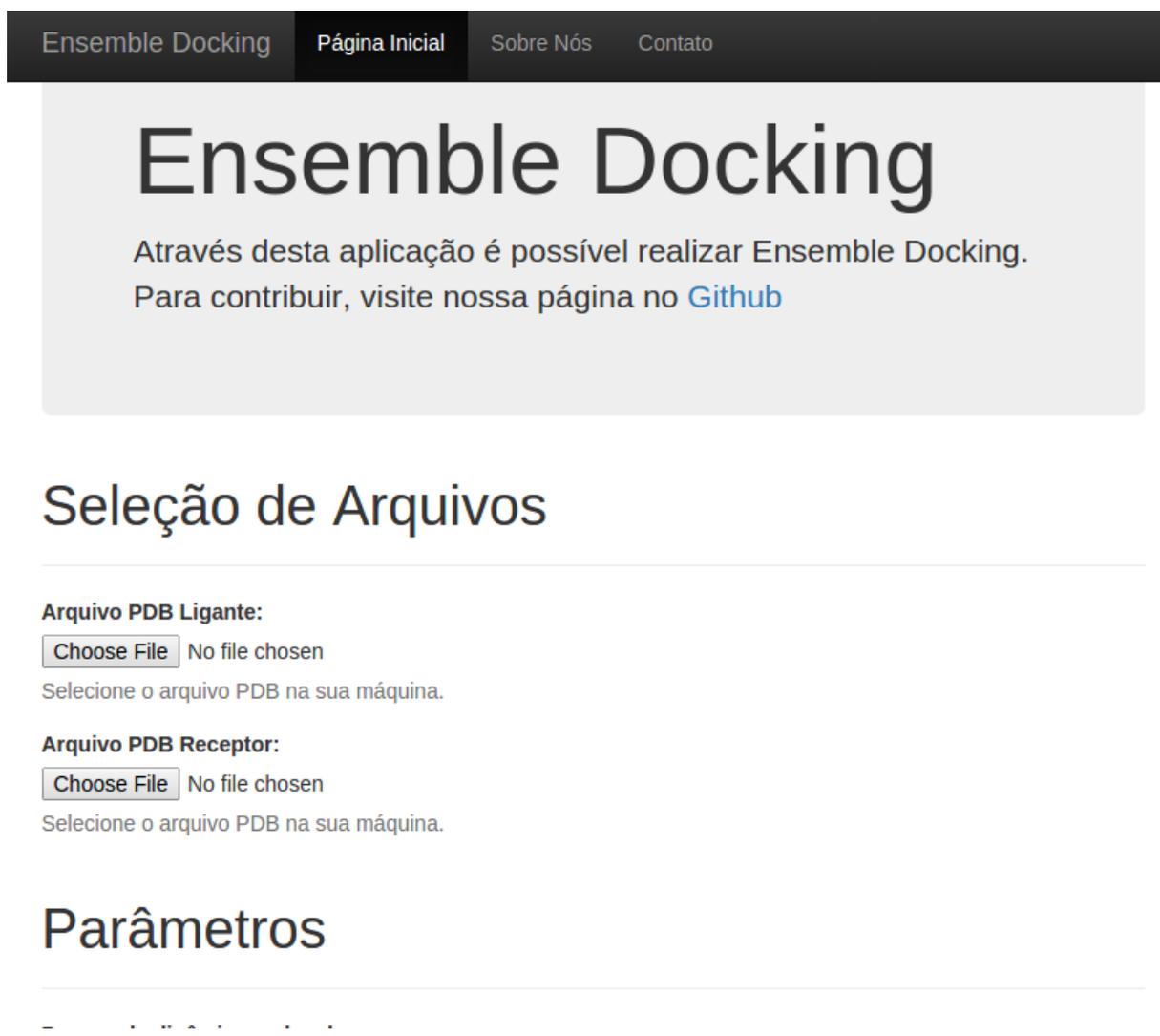


Figura 5.8 – Tela inicial da aplicação em português

Fonte: Próprio autor

Na Figura 5.8 é possível visualizar que o texto da página está em português, mas o botão para realização do *upload* das imagens está em inglês. Isso se deve ao fato de que o texto do botão é dependente da linguagem do navegador, e não da preferência de linguagem do navegador. A versão do navegador utilizado para realizar as capturas de tela era a versão em inglês, por isso o texto é “Choose File”.

No próximo capítulo será detalhada a ferramenta com enfoque nos seus objetivos, detalhando também o fluxo de execução e os resultados esperados. Serão apresentados também os arquivos de entrada a serem utilizados, e a saída esperada.

## 6 A FERRAMENTA

Neste capítulo será detalhada a ferramenta que foi desenvolvida. Aqui haverão os objetivos da ferramenta, descrevendo o porquê dela ser única e o que será atingido integrando as ferramentas já descritas até aqui. Haverá um maior nível de detalhamento sobre o fluxo de execução a ferramenta e como ela se integra com ferramentas como o Autodock e o Gromacs.

### 6.1 Objetivos

O principal objetivo desta ferramenta é facilitar o processo de *Ensemble Docking*. Como descrito anteriormente, para a realização de *Ensemble Docking* é necessário gerar várias “poses” através de dinâmica molecular, logo esta suíte irá integrar-se com a ferramenta de dinâmica molecular “Gromacs”.

Para a realização dos atracamentos, será utilizada a ferramenta Autodock, descrita no capítulo 3.1. Como parâmetro para o Autodock, serão utilizadas as poses geradas pelo Gromacs ao longo da simulação. Para cada docagem realizada, será guardado o seu resultado para avaliação da melhor docagem, aquela com menor energia livre, ou seja, o melhor encaixe.

As ferramentas relacionadas a bioinformática normalmente são programas executáveis à partir da linha de comando, ou *scripts* como o Biopython (BIOPYTHON, 2014). O objetivo aqui também é propor a aplicação de uma interface web para a realização de processos de bioinformática, neste caso focado na técnica de *ensemble docking*.

### 6.2 Fluxo de execução

Na Figura 6.1 estão representados os passos para do *script* de docagem. O papel da interface em Java será chamar este *script* passando os parâmetros informados pelo usuário e retornar os resultados de forma amigável para o usuário, que estará acessando através de um navegador de internet.

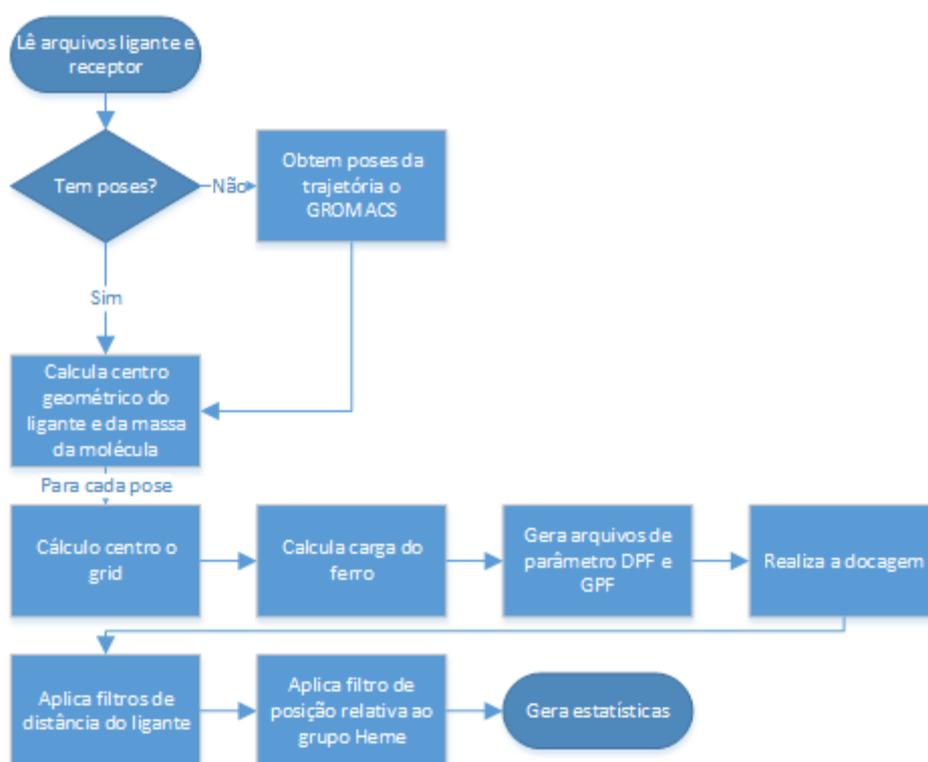


Figura 6.1 – Fluxo do algoritmo

Fonte: Próprios autores

A interface deverá permitir que os parâmetros necessários para a execução do *script* sejam passados pelo usuário, com validações mínima para que os dados estejam de acordo com o que o *script* precisa para executar a docagem. Caso aconteça algum erro ou alguma crítica mais avançada quanto as ferramentas com que o *script* se integra, este erro deverá ser exibido ao usuário. Em caso de sucesso, o resultado deverá ser retornado para o usuário, com informações suficientes para que ele possa seguir a sua pesquisa.

### 6.3 Arquivos de parâmetro

Os principais parâmetros da ferramenta são os arquivos do ligante e do receptor. O formato dos arquivos de parâmetro é PDB (*Protein Data Bank*). Este formato de input foi escolhido pois, segundo SUSSMAN et al. (1998), ele é o maior padrão de formatos de arquivo com coordenadas tridimensionais de proteínas, sendo utilizado por centenas de programas. Conforme descrito no capítulo 2.4, o repositório central de arquivos PDB continua crescendo, com utilização muito superior as suas alternativas. Cada linha do arquivo representa a coordenada de um átomo da proteína, conforme pode ser visto na Figura 6.2.

receptor_old.pdb x											
ATOM	3779	N	LYS	B	31	-4.762	-32.519	49.736	1.00	52.13	N
ATOM	3780	CA	LYS	B	31	-5.755	-32.498	48.600	1.00	52.15	C
ATOM	3781	C	LYS	B	31	-5.927	-31.155	47.825	1.00	51.25	C
ATOM	3782	O	LYS	B	31	-5.887	-30.087	48.433	1.00	51.07	O
ATOM	3783	CB	LYS	B	31	-7.105	-32.929	49.103	1.00	52.22	C
ATOM	3784	CG	LYS	B	31	-8.158	-32.659	48.058	1.00	53.40	C
ATOM	3785	CD	LYS	B	31	-9.136	-33.794	47.975	1.00	56.89	C
ATOM	3786	CE	LYS	B	31	-10.157	-33.756	49.092	1.00	57.72	C
ATOM	3787	NZ	LYS	B	31	-11.307	-34.617	48.700	1.00	59.51	N
ATOM	3788	N	LEU	B	32	-6.139	-31.209	46.499	1.00	50.26	N
ATOM	3789	CA	LEU	B	32	-6.115	-29.972	45.680	1.00	48.67	C
ATOM	3790	C	LEU	B	32	-7.372	-29.127	45.796	1.00	47.33	C
ATOM	3791	O	LEU	B	32	-8.464	-29.676	46.060	1.00	48.02	O
ATOM	3792	CB	LEU	B	32	-5.854	-30.270	44.210	1.00	49.08	C
ATOM	3793	CG	LEU	B	32	-4.456	-30.711	43.739	1.00	49.25	C
ATOM	3794	CD1	LEU	B	32	-4.576	-30.876	42.237	1.00	50.57	C
ATOM	3795	CD2	LEU	B	32	-3.285	-29.757	44.077	1.00	48.26	C
ATOM	3796	N	PRO	B	33	-7.242	-27.783	45.633	1.00	45.12	N
ATOM	3797	CA	PRO	B	33	-8.421	-26.927	45.587	1.00	43.44	C
ATOM	3798	C	PRO	B	33	-9.373	-27.420	44.509	1.00	41.86	C
ATOM	3799	O	PRO	B	33	-8.920	-27.992	43.512	1.00	42.05	O
ATOM	3800	CB	PRO	B	33	-7.828	-25.579	45.219	1.00	42.87	C
ATOM	3801	CG	PRO	B	33	-6.559	-25.597	45.848	1.00	43.33	C
ATOM	3802	CD	PRO	B	33	-6.033	-26.969	45.538	1.00	44.89	C
ATOM	3803	N	PRO	B	34	-10.682	-27.207	44.677	1.00	40.94	N

Figura 6.2 – Arquivo PDB

Fonte: Próprio autor

O arquivo PDB é um arquivo com informações em posições fixas, com colunas designadas para cada informação. Ao mesmo tempo, existem vários tipos de informação especificadas dentro do arquivo PDB, como uma seção com informações sobre aquele arquivo de coordenadas, seção para designar o autor, dentre outras (WWPDB, 2011). A seção de maior interesse é a com as coordenadas de cada átomo, com prefixo “ATOM”. O conteúdo de cada coluna desta seção pode ser visto na Tabela 1.1.

Tabela 1.1 – Colunas da seção de átomos do arquivo PDB.

Colunas	Tipo de dado	Definição
1 - 6	Nome do registro	
7 - 11	Inteiro	Número único do átomo
13 - 16	Átomo	Nome do átomo

17	Caractere	Indicação de localização alternativa
18 - 20	Nome do resíduo	Nome do resíduo
22	Caractere	Identificador da cadeia
23 - 26	Inteiro	Número sequencial do resíduo
27	Caractere	Código para inserção de resíduos
31 - 38	Real(8.3)	Coordenadas ortogonais para X em Ångströms
39 - 46	Real(8.3)	Coordenadas ortogonais para Y em Ångströms
47 - 54	Real(8.3)	Coordenadas ortogonais para Z em Ångströms
55 - 60	Real(6.2)	Ocupação
61 - 66	Real(6.2)	Fator de temperatura
77 - 78	String(2)	Símbolo do elemento, justificado a direita
79-80	String(2)	Carga do átomo

Fonte: Adaptado de WWPDB, 2011.

Como entrada para o ligante e o receptor a serem utilizados no processo *de ensemble docking*, o usuário realiza o *upload* dos dois arquivos “PDB” pela tela inicial da aplicação, conforme a Figura 6.3.

## Seleção de Arquivos

**Arquivo PDB Ligante:**  
 No file chosen  
 Selecione o arquivo PDB na sua máquina.

**Arquivo PDB Receptor:**  
 No file chosen  
 Selecione o arquivo PDB na sua máquina.

Figura 6.3 – Página de seleção de arquivos de parâmetro

Fonte: Próprio autor

Durante o processo de *upload*, a aplicação não armazena o nome do arquivo enviado pelo usuário, pois isso caracterizaria uma falha de segurança (OWASP, 2014). Um novo nome é utilizado para gravar cada um dos arquivos no disco. A gravação no disco é importante pois as ferramentas que serão chamadas, como GROMACS e Autodock, trabalham com arquivos em disco como parâmetro.

## 6.4 Chamada da dinâmica molecular

Para realizar a chamada do programa Gromacs, a aplicação executa todos os passos descritos no capítulo 5.3. Dentre os mais complexos está a criação do arquivo de parâmetros MDP. Na Figura 6.4 pode ser visto o arquivo MDP da execução da dinâmica molecular sem restrição de posições.

```

1  title           = fws
2  cpp             = /usr/bin/cpp
3  constraints     = all-bonds
4  integrator     = md
5  tinit          = 0
6  dt              = 0.002      ; ps !
7  nsteps         = 50000      ; total 100 ps.
8  nstcomm        = 1
9  nstxout        = 500
10 nstvout        = 0
11 nstfout        = 0
12 nstlog         = 100
13 nstenergy      = 100
14 nstlist        = 10
15 ns_type        = grid
16 rlist          = 1.0
17 coulombtype    = PME
18 rcoulomb       = 1.0
19 vdwtype        = Cut-off
20 rvdw           = 1.4
21 fourierspacing = 0.12
22 fourier_nx     = 0
23 fourier_ny     = 0
24 fourier_nz     = 0
25 pme_order      = 4
26 ewald_rtol     = 1e-5
27 optimize_fft   = yes
28 ; Berendsen temperature coupling is on in three groups
29 Tcoupl         = berendsen
30 tau_t          = 0.1         0.1
31 tc_grps        = Protein    SOL
32 ref_t          = 300        300
33 ; Pressure coupling is on
34 Pcoupl        = parrinello-rahman
35 pcoupltype    = isotropic
36 tau_p         = 0.5
37 compressibility = 4.5e-5
38 ref_p         = 1.0
39 ; Generate velocities is on at 300 K.
40 gen_vel       = yes
41 gen_temp      = 300.0
42 gen_seed      = 173529

```

Figura 6.4 – Arquivo MDP

Os parâmetros passados na tela inicial da aplicação influenciam diretamente em como os parâmetros do arquivo MDP são gerados. Na Figura 6.5 é possível ver os parâmetros passados na tela inicial. O parâmetro “Passos da dinâmica molecular” será a saída do atributo *nsteps* do arquivo. Já o parâmetro “Output a cada”, será a saída do atributo *nstxout*.

**Parâmetros**

---

**Passos da dinâmica molecular**

O número de passos a serem executados na dinâmica molecular. Cada passo tem 0.002 picosegundos.

**Output a cada**

A cada quantos passos deve ser executada uma docagem.

Figura 6.5 – Parâmetros de configuração da aplicação

Fonte: Próprio autor

## 6.5 Chamada da docagem

A quantidade de docagens a serem realizadas é o resultado da divisão dos passos da dinâmica molecular pelo “Output a cada”. Conforme descrito no capítulo 6.2, as conformações resultado de cada passo são os parâmetros para docagem. A aplicação carrega para a memória o PDB resultante da execução do programa *trjconv*, e quebra este arquivo em cada uma das conformações, buscando pela âncora “ENDMDL”, que define o final de uma conformação no PDB.

Cada uma das conformações é utilizada para uma docagem completamente distinta, onde será feita uma chamada para o Autodock passando como ligante o arquivo enviado pelo usuário no início do processo, e como receptor a conformação extraída como resultado da dinâmica molecular.

O resultado de cada docagem é um arquivo PDBQT. A aplicação busca pela âncora "Estimated Free Energy of Binding =", que é o ponto do arquivo que indica a energia livre daquela docagem. No exemplo da Figura 6.6, a energia livre foi de -4.24 kcal/mol.

```

MODEL          1
USER          Run = 1
USER          DPF = /tmp/dock2062357385588378132/fileLigand.dpf
USER
USER          Estimated Free Energy of Binding      = -4.24 kcal/mol [(1)+(2)+(3)-(4)]
USER          Estimated Inhibition Constant, Ki    = 774.61 uM (micromolar) [Temperature = 298.15 K]
USER
USER          (1) Final Intermolecular Energy     = -4.24 kcal/mol
USER          vdW + Hbond + desolv Energy        = -4.20 kcal/mol
USER          Electrostatic Energy               = -0.04 kcal/mol
USER          (2) Final Total Internal Energy     = +0.00 kcal/mol
USER          (3) Torsional Free Energy          = +0.00 kcal/mol
USER          (4) Unbound System's Energy        = +0.00 kcal/mol
USER
USER          NEWDPF move fileLigand.pdbqt
USER          NEWDPF about 5.575000 -3.855000 35.608700
USER          NEWDPF tran0 41.184774 47.849396 48.903404
USER          NEWDPF quaternion0 0.537905 -0.561580 0.354695 -0.519112
USER          NEWDPF axisangle0 0.629345 -0.657045 0.414990 -117.454561
USER          NEWDPF quat0 0.629345 -0.657045 0.414990 -117.454561
USER          keepresnum = 1
USER

```

Figura 6.6 – Fragmento do arquivo PDBQT de saída

Fonte: Próprio autor

## 6.6 Exibição dos resultados

Para exibição dos resultados, são geradas duas saídas, a representação tridimensional da melhor docagem, e o gráfico de energia livre ao longo do tempo.

### 6.6.1 Representação da melhor docagem

Para representar a melhor docagem, é exibida uma representação tridimensional da docagem. Uma representação tridimensional é muito importante, conforme afirmado por Seeliger, 2010: "O exame visual da geometrias previstas de ligação (poses de ancoragem), assim, contribui decisivamente para o desenvolvimento de um composto de líder". Para a geração da representação em formato de imagem a aplicação utiliza o programa Pymol (DELANO, 2002).

O primeiro passo é transformar o arquivo PDBQT de saída do Autodock para PDB. Este processo é realizado cortando todos os valores após a sexagésima sexta coluna do arquivo, que são valores adicionais ao PDB clássico e relevantes apenas para o Autodock.

Após a conversão do arquivo PDBQT em PDB, a aplicação gera um arquivo PML, que será utilizado como parâmetro para o Pymol. Conforme pode ser visto na Figura 6.7, o arquivo PML contém instruções da linguagem de *script* própria do Pymol.

```
load /tmp/dock1713675739417169333/fileLigand_docked.pdb, ligand
load /tmp/dock1713675739417169333/308360213.pdb, receptor
hide lines
show lines, receptor
show sticks, ligand
png /home/jonasflesch/bio/docked_images/1424990815.png, dpi=300
log_close
quit
```

Figura 6.7 – Arquivo PML

Fonte: Próprio autor

No arquivo de instruções para o Pymol existem instruções para carregar os arquivos do ligante resultado da docagem, e do receptor, resultado da dinâmica molecular. Há também instruções para exibir o receptor no formato de linhas e o ligante no formato de gravetos, o que facilita a visualização da posição do ligante no resultado final. Por fim, o resultado é exportado para uma imagem no formato PNG e o programa é fechado. Um exemplo de resultado pode ser visto na Figura 6.8.

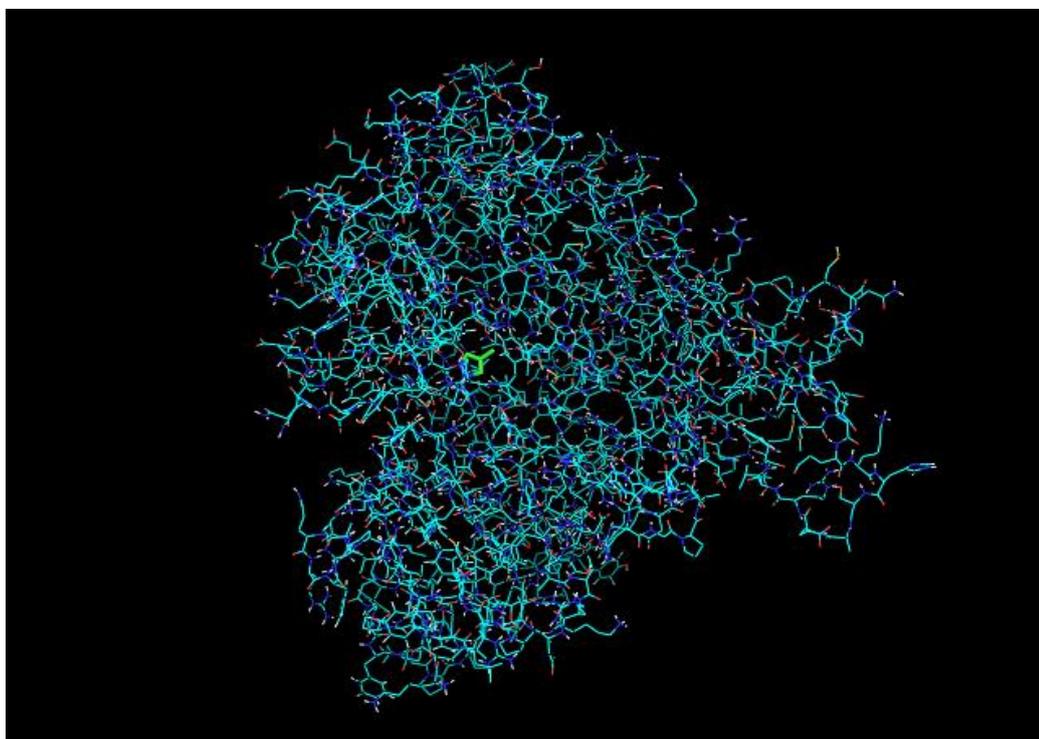


Figura 6.8 – Imagem de saída do PyMOL

Fonte: Próprio autor

## 6.6.2 Gráfico de energia livre ao longo do tempo

Uma saída importante para análise de qualidade do *ensemble docking* é um gráfico de energia livre ao longo do tempo. Como a dinâmica molecular é a simulação da exposição da molécula em um líquido, ao longo desta simulação a estrutura da molécula é alterada. Conforme a estrutura vai sendo alterada, o resultado das docagens pode variar para melhor ou para pior.

Para a geração deste gráfico é utilizada a energia livre de cada docagem. A forma de obtenção da energia livre pode ser vista no Capítulo 3.1. O eixo “x” do gráfico é o tempo, ou quantidade de passos da dinâmica molecular, e o eixo “y” é a energia livre naquele momento. Um exemplo de gráfico de energia livre ao longo do tempo pode ser obtido abaixo, na Figura 6.9.

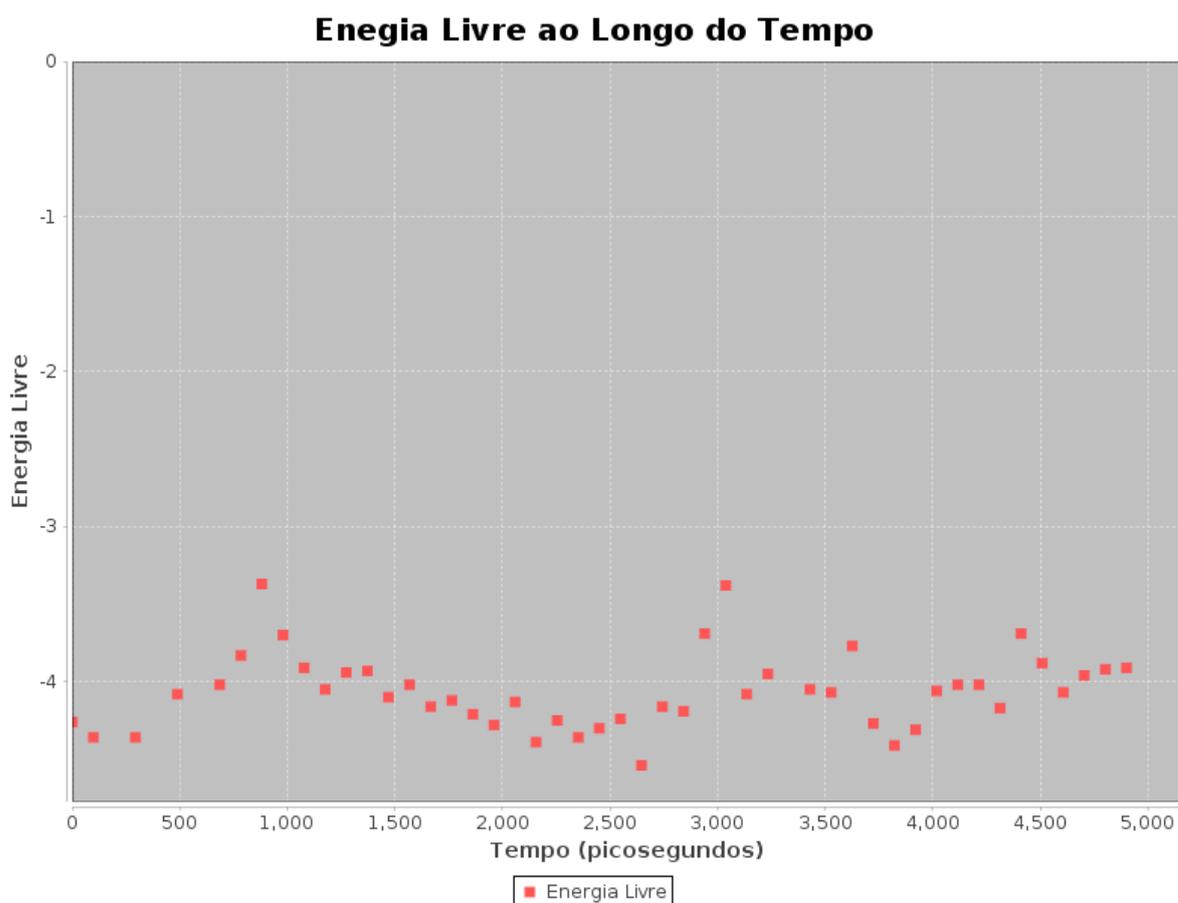


Figura 6.9 – Energia Livre ao Longo do Tempo

Fonte: Próprio autor

## 6.7 Oportunidades de melhoria

O trabalho que foi executado foi adequado para um Trabalho de Conclusão de Curso, mas para que a ferramenta se torne mais completa e atenda melhor usuários sem conhecimento de linguagens de programação algumas melhorias podem ser realizadas.

Uma destas melhorias é a realização de requisições assíncronas. Foi identificado, durante a construção da ferramenta de *ensemble docking*, que dependendo dos parâmetros o processo pode levar várias horas. Como a aplicação foi criada sem nenhum tipo de requisição assíncrona, se o usuário fechar o navegador de internet ou tiver uma perda de conexão, ele pode não ver os resultados da docagem. Para solucionar este problema, pode ser utilizado como exemplo uma ferramenta muito utilizada em outro campo da bioinformática, o alinhamento de sequências, a *BLAST: Basic Local Alignment Search Tool* (ALTSCHUL et al, 1997). Durante a utilização desta ferramenta, o usuário solicita um alinhamento e recebe uma chave da sua solicitação. De posse desta chave ele pode consultar a situação do alinhamento.

Outro ponto da aplicação que pode ser melhorado é sua interface visual. Poderiam ser incorporadas páginas de explicação ao usuário de quais processos estão sendo chamados e como cada parâmetro influencia na execução. Poderiam também ser dadas dicas para interpretar o resultado, com mais informações sobre os gráficos gerados.

No próximo capítulo serão detalhados resultados obtidos através do uso da ferramenta. Será possível ver resultados de docagens realizadas utilizando a ferramenta bom como a comparação destes resultados com uma docagem tradicional.

## 7 RESULTADOS OBTIDOS

Durante este capítulo serão analisados resultados de execuções de *ensemble docking* à partir da ferramenta. Foram escolhidos como casos de teste ligantes que tenham encaixe relevante com o citocromo P450, na sua variante 2E1. Para obtenção destes ligantes, foi utilizado o *Protein Data Bank*, uma base de dados pública de proteínas e ligantes já definida anteriormente neste trabalho.

### 7.1 P450 2E1 x omega-imidazolyl-decanoic

O primeiro *ensemble docking* que será analisado será utilizando o ligante *omega-imidazolyl-decanoic* sendo docado no receptor P450 2E1. Para obter os arquivos PDB necessários, foi-se utilizado o *Protein Data Bank* como fonte de dados. Foi utilizada a conformação 3GPH (PORUBSKY; BATTALIE; SCOTT, 2010), que possui um CYP e um *omega-imidazolyl-decanoic*. O arquivo foi analisado e separado manualmente para obter-se dois arquivos PDB distintos: um para o receptor (CYP), um para o ligante (*omega-imidazolyl-decanoic*). Na Figura 7.1 pode ser vista a conformação da docagem, e na Figura 7.2 o gráfico de energia livre ao longo do tempo.

Os arquivos foram selecionados na aplicação utilizando a interface web. Como parâmetros foram passados:

Passos da dinâmica molecular: 5000

Output a cada: 100

A simulação levou uma hora e trinta minutos em uma máquina virtual com três cores, virtualizados à partir de um processador Intel Core i5 com 2.5 Ghz.

Analisando a Figura 7.1, que demonstra a representação tridimensional da docagem, é possível visualizar que o ligante ficou posicionado em uma região de encaixe da conformação do CYP, que é o resultado esperado. Já observando a figura 7.2, que contém a energia livre ao longo do tempo, é possível visualizar que a energia livre varia ao longo da simulação da dinâmica molecular. Isto demonstra que a capacidade do ligante se docar, ou encaixar, no receptor de fato varia com a aplicação da dinâmica molecular.

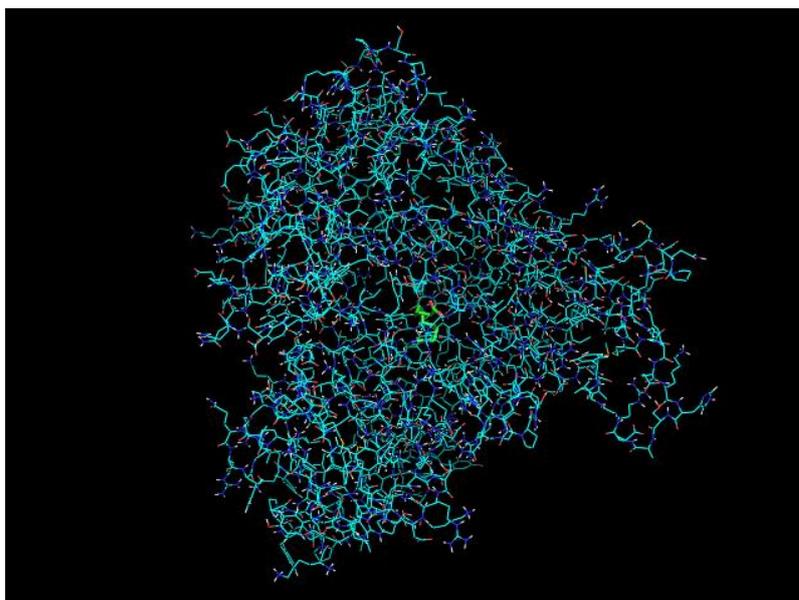


Figura 7.1 – Conformação Final da docagem P450 2E1 x omega-imidazolyl-decanoic

Fonte: Próprio autor

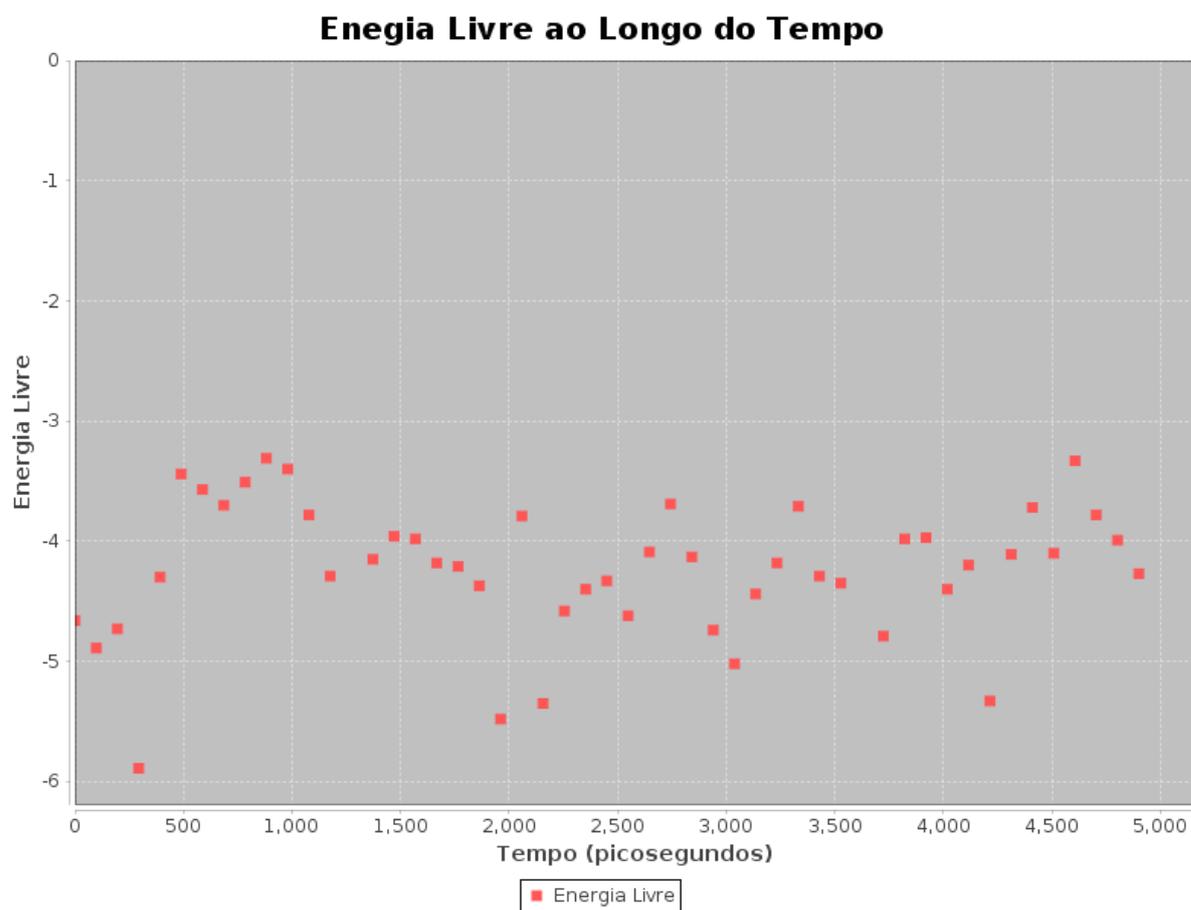


Figura 7.2 – Energia Livre ao Longo do Tempo P450 2E1 x omega-imidazolyl-decanoic

Fonte: Próprio autor

Para uma melhor análise da eficiência do atracamento, foi realizada uma docagem manualmente utilizando o mesmo ligante e receptor sem o uso da dinâmica molecular. Para isso, foi-se utilizado o Autodock Tools e o Autodock. Como parâmetros foram utilizados os mesmos arquivos PDB que forma utilizados para *ensemble docking*. Como resultado foi obtida uma energia livre de -5,21 kcal/mol. No gráfico a energia livre foi multiplicada por -1 para obter-se valores positivos e ter-se uma melhor visualização dos resultados. A comparação deste resultado com o do *ensemble docking* pode ser vista na figura 7.3.

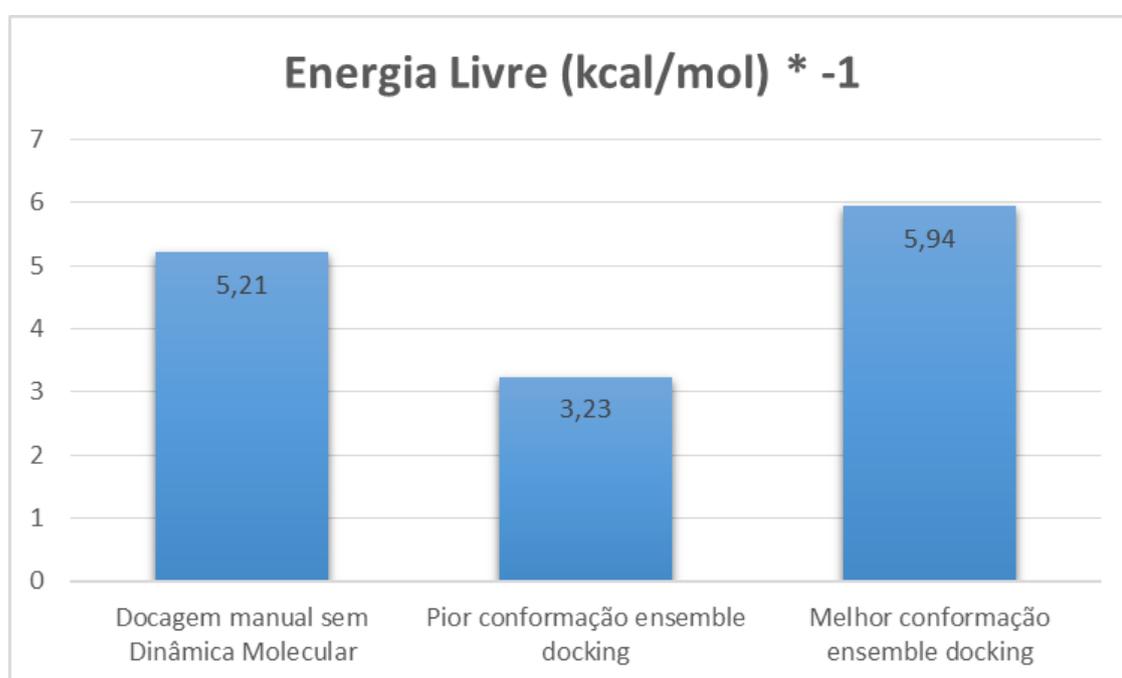


Figura 7.3 - Comparação Resultados docagem tradicional e *ensemble docking* com P450 2E1 x omega-imidazolyl-decanoic

Fonte: Próprio autor

Pode ser observado através da comparação dos resultados que aplicada a dinâmica molecular, há conformações com resultados melhores e conformações com resultados piores do que a conformação inicial.

## 7.2 P450 2E1 x 4-methylpyrazole

A segunda docagem que será analisada será utilizando o ligante *4-methylpyrazole* sendo docado no receptor P450 2E1. Novamente foi utilizado o *Protein Data Bank* como fonte de dados. Foi utilizada a conformação 3E4E (PORUBSKY; MENEELY; SCOTT,

2008), que possui dois CYP e um *4-methylpyrazole*. Foi realizada uma separação manual do arquivo PDB obtido através do banco de dados previamente citado, sendo o resultado dois arquivos, um para o receptor (CYP), um para o ligante (*4-methylpyrazole*). Na Figura 7.4 pode ser vista a conformação da docagem, e na Figura 7.5 o gráfico de energia livre ao longo do tempo.

Os arquivos foram selecionados na aplicação utilizando a interface web. Como parâmetros foram passados:

Passos da dinâmica molecular: 5000

Output a cada: 100

A simulação levou duas horas e quarenta minutos em uma máquina virtual com três cores, virtualizados à partir de um processador Intel Core i5 com 2.5 Ghz.

A análise aqui feita tem resultados muito semelhantes aos da análise realizada no capítulo 7.1. Assim como na docagem entre o CYP 2E1 com *omega-imidazolyl-decanoic*, a docagem entre CYP 2E1 e o *4-methylpyrazole* foi realizada com sucesso, encontrando uma região suscetível para docagem dentro do CYP. Além disso, a energia livre ao longo do tempo mostrou um resultado bastante claro de que ao longo do tempo a conformação oscila entre momentos em que a docagem se torna menos suscetível (menor energia livre), com momentos com docagem de melhor qualidade (energia livre mais negativa).

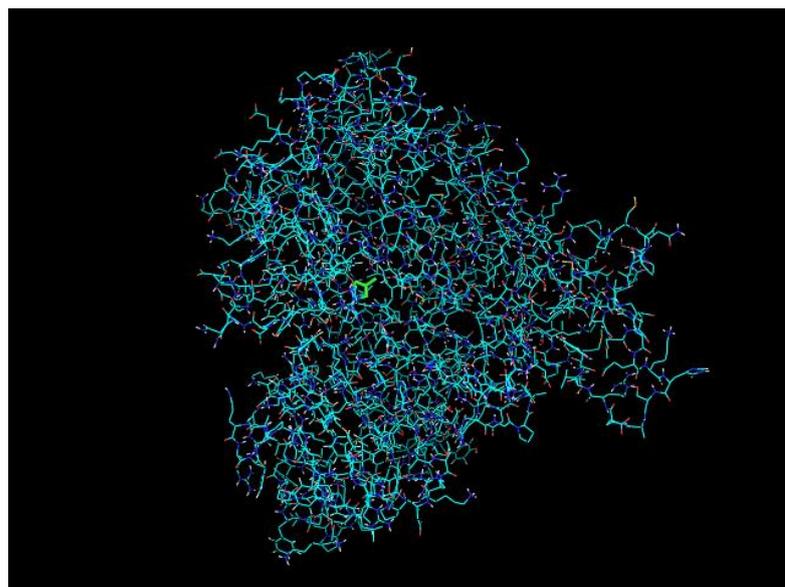


Figura 7.4 – Conformação Final da docagem P450 2E1 x 4-methylpyrazole

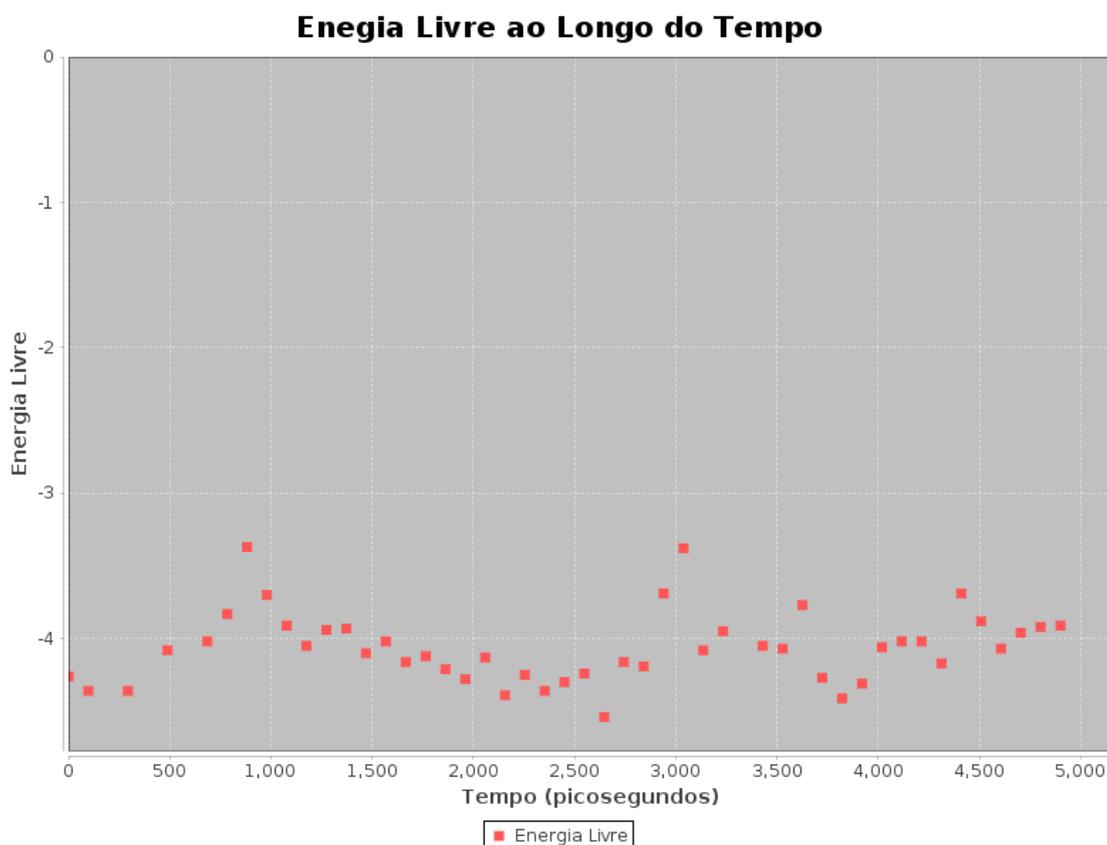


Figura 7.5 – Energia Livre ao Longo do Tempo P450 2E1 x 4-methylpyrazole

Fonte: Próprio autor

Novamente para uma melhor análise da eficiência do atracamento, foi realizada outra docagem manual utilizando o mesmo ligante e receptor sem o uso da dinâmica molecular. Como parâmetros também foram utilizados os mesmos arquivos PDB que foram utilizados para *ensemble docking*. Como resultado foi obtida uma energia livre de -2,84 kcal/mol. Novamente no gráfico a energia livre foi multiplicada por -1 para obter-se valores positivos e ter-se uma melhor visualização dos resultados. A comparação pode ser vista na figura 7.6.

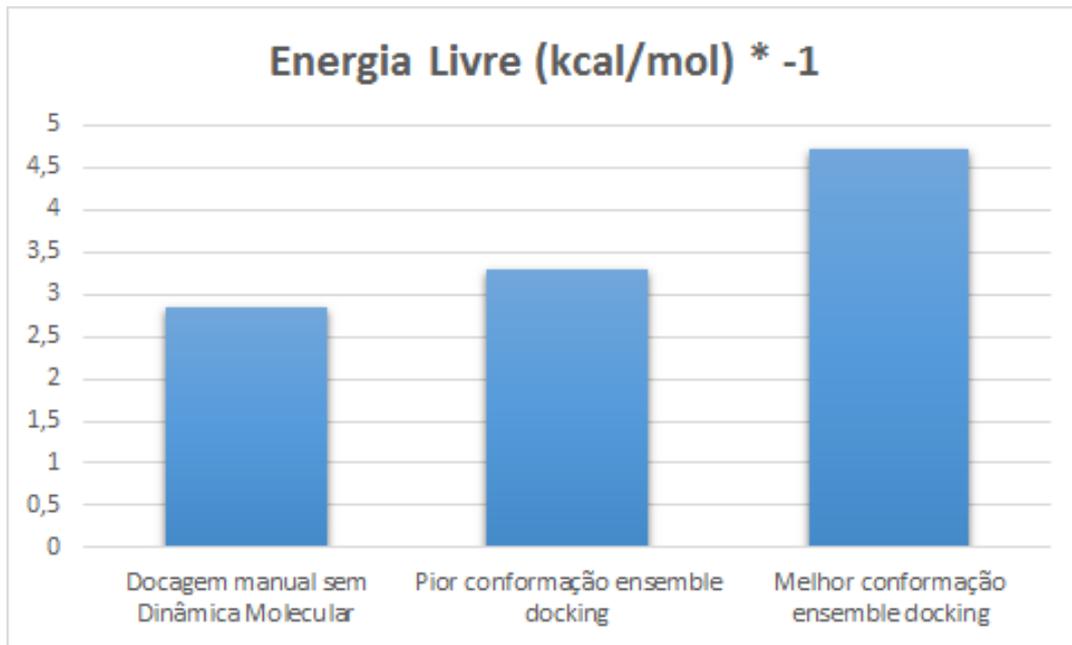


Figura 7.6 - Comparação Resultados docagem tradicional e *ensemble docking* com P450 2E1 x 4-methylpyrazole

Fonte: Próprio autor

Observa-se nesta segunda comparação que a pior energia de uma conformação utilizando *ensemble docking* foi superior a conformação da docagem manual. Um dos fatores que podem influenciar neste resultado é a aleatoriedade de uma docagem realizada através do Autodock. A cada docagem, valores pseudoaleatórios são gerados e utilizados na simulação, a fim de simular a aleatoriedade existente nos acontecimentos reais.

### 7.3 P450 2E1 x Omega-Imidazolyl-Dodecanoic

A terceira docagem que será analisada será utilizando o ligante *Omega-Imidazolyl-Dodecanoic* sendo docado novamente no receptor P450 2E1. Novamente foi utilizado o *Protein Data Bank* como fonte de dados. Foi utilizada a conformação 3LC4 (PORUBSKY; BATTAILE; SCOTT, 2010), que possui dois CYP e um *Omega-Imidazolyl-Dodecanoic*. Como anteriormente, foi realizada uma separação manual do arquivo PDB obtido através do banco de dados previamente citado, separando o ligante do receptor. Na Figura 7.7 pode ser vista a conformação da docagem, e na Figura 7.8 o gráfico de energia livre ao longo do tempo.

Os arquivos foram selecionados na aplicação utilizando a interface web. Como parâmetros foram passados:

Passos da dinâmica molecular: 50000

Output a cada: 250

A simulação levou três horas e trinta minutos em uma máquina virtual com três cores, virtualizados à partir de um processador Intel Core i5 com 2.5 Ghz.

A análise aqui feita tem resultados muito semelhantes aos das análises dos capítulos anteriores. A energia livre de cada conformação desta docagem ficou mais dispersa no gráfico, em razão do maior tempo de dinâmica molecular em relação as simulações anteriores. Nas simulações anteriores foram utilizados 5000 passos, e nesta 50000, um número dez vezes superior. Ao mesmo tempo, foram realizadas o dobro de docagens, visto que o output foi realizado a cada 250.

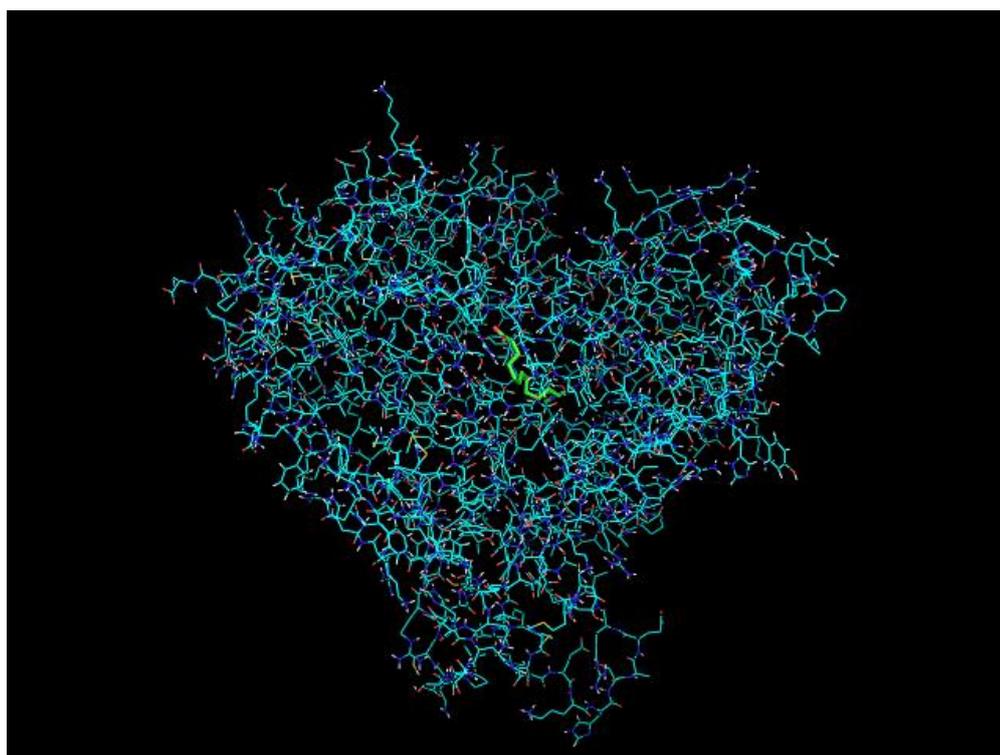


Figura 7.7 – Conformação Final da docagem P450 2E1 x ligante Omega-Imidazolyl-Dodecanoic

Fonte: Próprio autor

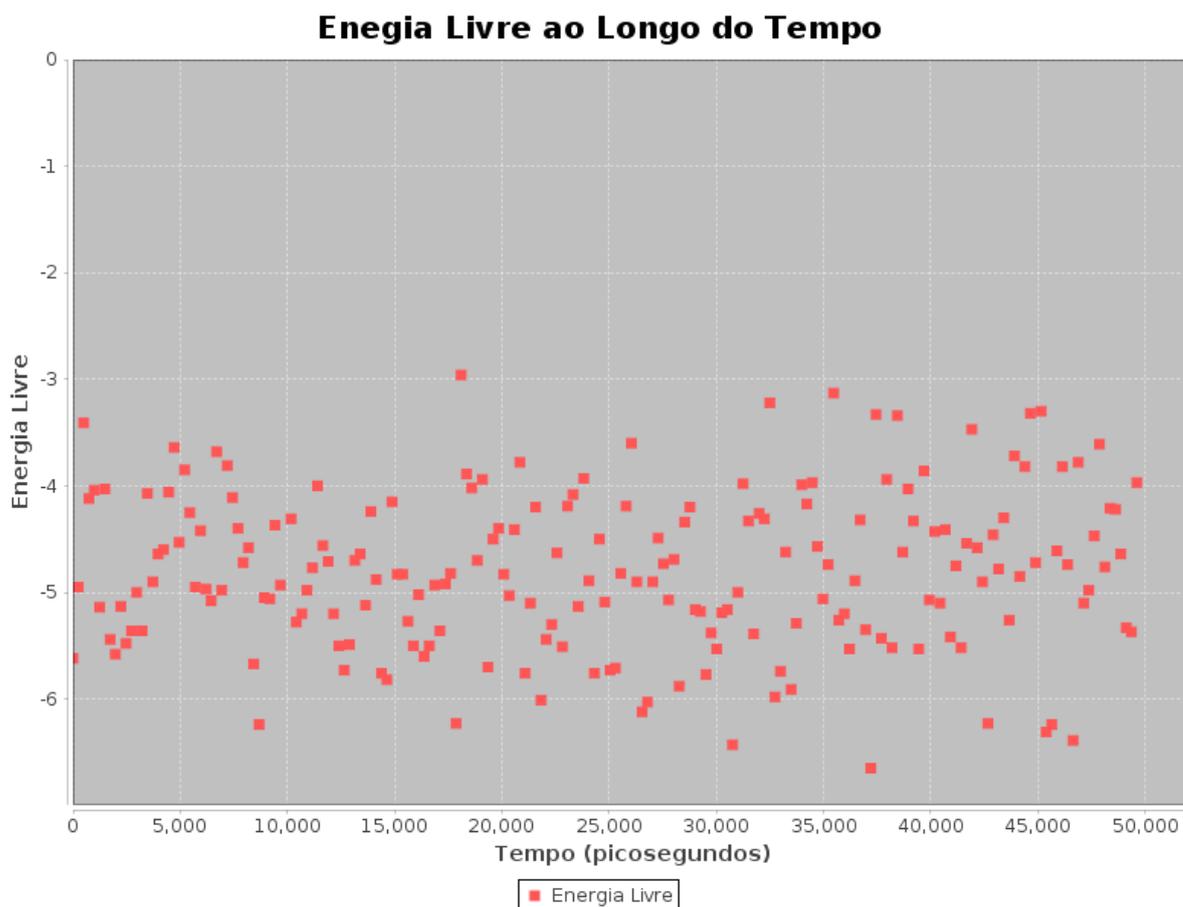


Figura 7.8 – Energia Livre ao Longo do Tempo P450 2E1 x ligante Omega-Imidazolyl-Dodecanoic

Fonte: Próprio autor

Assim como nas análises anteriores, foi realizada outra docagem manual utilizando o mesmo ligante e receptor sem o uso da dinâmica molecular. Como parâmetros também foram utilizados os mesmos arquivos PDB que foram utilizados para *ensemble docking*. Como resultado foi obtida uma energia livre de -5,91 kcal/mol. Novamente no gráfico a energia livre foi multiplicada por -1 para obter-se valores positivos e ter-se uma melhor visualização dos resultados. A comparação pode ser vista na figura 7.9.

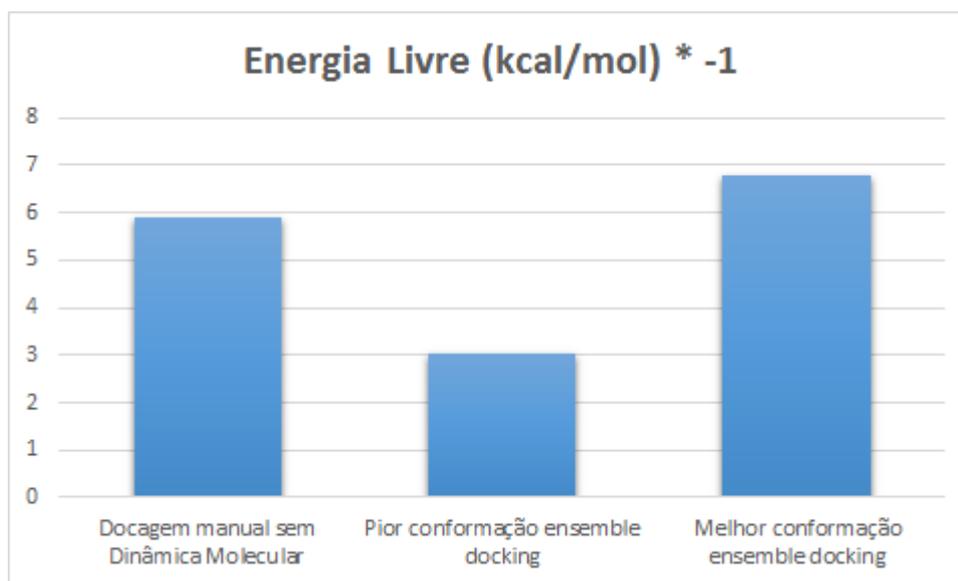


Figura 7.9 - Comparação Resultados docagem tradicional e *ensemble docking* com P450 2E1 x Omega-Imidazolyl-Dodecanoic

Fonte: Próprio autor

Aqui foram encontradas docagens melhores e docagens piores utilizando dinâmica molecular, em comparação a docagem tradicional. Isto significa que ao executar a dinâmica molecular o receptor foi alterado para estruturas que não receberam tão bem ao ligante, mas ao mesmo tempo também foi alterado para estruturas que recebem bem o ligante.

## CONCLUSÃO

Através do estudo que foi realizado durante o trabalho e o que foi demonstrado nos resultados obtidos, conclui-se que os objetivos do trabalho foram totalmente obtidos.

Um dos principais objetivos do trabalho era estudar conceitos e aplicações da bioinformática. Foi realizado um levantamento de várias áreas diferentes da bioinformática, desde fundamentos, passando por aplicações de sequenciamento, até chegar na análise e simulação estrutural de proteínas, que é o objetivo deste trabalho. Quanto a docagem de proteínas, que é a principal área de conhecimento que circunda o objeto do trabalho, foram entendidas e utilizadas as principais ferramentas, sendo integradas com sucesso na aplicação desenvolvida.

Com o uso da ferramenta construída, a realização de *ensemble docking* se tornou possível com a utilização de apenas uma interface. Como foi demonstrado durante o trabalho, sem a utilização da ferramenta construída haveria um grande trabalho manual do pesquisador para executar todos os passos necessários no Gromacs. Não obstante, a chamada manual da docagem via Autodock para cada uma das poses geradas pelo Gromacs é de enorme esforço. Em alguns exemplos analisados mais de cem poses foram geradas, e para cada uma destas poses precisam ser chamados comandos para a execução do Autogrid e do Autodock, além de no final consolidar os resultados para extrair informações úteis.

O trabalho não tinha como objetivo encontrar um novo candidato a fármaco ou analisar como diferenças genéticas e conseqüentemente de estrutura em alguma proteína afetam a sua docagem com algum ligante, mas sem dúvida a ferramenta resultado do desenvolvimento da pesquisa auxiliará a pesquisas futuras. Com o uso da ferramenta será possível agilizar pesquisas futuras que tenham estes objetivos. Utilizando como ligante um composto que até hoje sofreu apenas tentativas de docagem com receptores (proteínas) rígidas, será possível analisar se há novas possibilidades de docagem, agora com receptores flexíveis que passam pelo processo de dinâmica molecular. Além disso, se identificado que em uma população existem diferentes variações de uma proteína com diferentes estruturas, será possível testar, em várias simulações, como cada uma destas variações se liga com um ligante.

Sugere-se como trabalho futuro a utilização da ferramenta para avaliação de candidatos a fármacos. Através deste uso, poderão ser identificados mais pontos de melhoria e

trabalhos futuros, como mais controle para o utilizador sobre os parâmetros do Autodock e do Gromacs, ou cálculos a serem executados pela ferramenta para geração destes parâmetros.

Nas três simulações realizadas, os resultados ficaram dentro dos parâmetros necessários: as docagens de cada conformação foram realizadas com sucesso, e pode-se observar diferentes energias livres ao longo de cada conformação da dinâmica molecular, que é o um parâmetro do sucesso na realização de *ensemble docking*.

Tendo em mente que um dos objetivos do trabalho era analisar as docagens realizadas, considera-se este como um objetivo cumprido. Nas duas saídas da ferramenta, a representação tridimensional da melhor docagem e o gráfico de energia livre ao longo do tempo, será possível analisar a qualidade de uma docagem. De posse dessas duas saídas, é possível também pensar em novos parâmetros para a docagem, como tempos maiores de dinâmica molecular.

## REFERÊNCIAS BIBLIOGRÁFICAS

ALLEN, Frank H. The Cambridge Structural Database: a quarter of a million crystal structures and rising. **Acta Crystallographica Section B: Structural Science**, v. 58, n. 3, p. 380-388, 2002.

ALLEN, Frank H.; MOTHERWELL, W D Samuel. Applications of the Cambridge Structural Database in organic chemistry and crystal chemistry. **Acta Crystallographica Section B: Structural Science**, v. 58, n. 3, p. 407-422, 2002.

ALTSCHUL, Stephen F. et al. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. **Nucleic acids research**, v. 25, n. 17, p. 3389-3402, 1997.

ALONSO, Hernan; BLIZNYUK, Andrey A.; GREASY, Jill E. Combining docking and molecular dynamic simulations in drug design. **Medicinal research reviews**, v. 26, n. 5, p. 531-568, 2006.

ALMLÖF, Martin. Computational Methods for Calculation of Ligand-Receptor Binding Affinities Involving Protein and Nucleic Acid Complexes. 2007.

AUTODOCK MANUAL. Disponível em: < <http://vina.scripps.edu/manual.html>>. Acesso em: 10 maio 2015.

BALL, Madeleine Price. **DNA chemical structure**, 2013. Disponível em: < [http://commons.wikimedia.org/wiki/File:DNA\\_chemical\\_structure.svg](http://commons.wikimedia.org/wiki/File:DNA_chemical_structure.svg)>. Acessado em 10 nov. 2014.

BECK, Kent. **Test-driven development: by example**. Addison-Wesley Professional, 2003.

BERMAN, Helen M. et al. The protein databank. **Nucleic acids research**, v. 28, n. 1, p. 235-242, 2000.

BIOPYTHON. In: WIKIPÉDIA, a enciclopédia livre. Flórida: Wikimedia Foundation, 2014. Disponível em: <<http://en.wikipedia.org/w/index.php?title=Biopython&oldid=630893144>>. Acesso em: 13 nov. 2014.

B-RAO, Chandrika; SUBRAMANIAN, Jyothi; SHARMA, Somesh D. Managing protein flexibility in docking and its applications. **Drug discovery today**, v. 14, n. 7, p. 394-400, 2009.

CHEN, Grace Shiahuy; CHERN, Ji-Wang. **Computer-aided drug design**. Wiley, NY, USA, 2007.

COHEN, Elisângela Machado Leal. Um estudo do efeito da flexibilidade explícita da enzima InhA de *M. tuberculosis* na docagem molecular dos inibidores etionamida, triclosano e isoniazida-pentacionoferrato II. 2010.

DE GRAAF, Chris; VERMEULEN, Nico PE; FEENSTRA, K. Anton. Cytochrome P450 in silico: an integrative modeling approach. **Journal of medicinal chemistry**, v. 48, n. 8, p. 2725-2755, 2005.

DEITEL, Paul; DEITEL, Harvey. **JAVA Programming**. Prentice Hall, 2014.

DELANO, Warren L. The PyMOL molecular graphics system. 2002.

FOYE, William O. **Foye's principles of medicinal chemistry**. Lippincott Williams & Wilkins, 2008.

GIBAS, Cynthia; JAMBECK, Per. **Desenvolvendo Bioinformática**, Rio de Janeiro: Campus, 2001.

GILBERT, David. The jfreechart class library. **Developer Guide. Object Refinery**, v. 7, 2002.

GUENGERICH, F. P. Cytochrome P450: advances and prospects. **The FASEB journal**, v. 6, n. 2, p. 667-668, 1992.

HALDER, Georg; CALLAERTS, Patrick; GEHRING, Walter J. Induction of ectopic eyes by targeted expression of the eyeless gene in Drosophila. **Science**, v. 267, n. 5205, p. 1788-1792, 1995.

HANS, Bergsten. JavaServer pages. 2003.

HERNÁNDEZ-SANTOYO, Alejandra et al. Protein-Protein and Protein-Ligand Docking. 2013.

HOLLENBERG, P. F. Mechanisms of cytochrome P450 and peroxidase-catalyzed xenobiotic metabolism. **The FASEB journal**, v. 6, n. 2, p. 686-694, 1992.

HUANG, Sheng-You; ZOU, Xiaoqin. Advances and challenges in protein-ligand docking. **International journal of molecular sciences**, v. 11, n. 8, p. 3016-3034, 2010.

HUANG, Sheng-You; ZOU, Xiaoqin. Ensemble docking of multiple protein structures: considering protein structural variations in molecular docking. **Proteins: Structure, Function, and Bioinformatics**, v. 66, n. 2, p. 399-421, 2007.

JETBRAINS, I. IntelliJ idea. **On-line at [www.intellij.com](http://www.intellij.com)**, 2003.

JUNQUEIRA, Luis C.; CARNEIRO, José. **Biologia celular e molecular**. 8 ed. Rio de Janeiro: Guanabara-Koogan, 2011

KARP, Gerald; PRUITT, Nancy L. Cell and molecular biology: concepts and experiments. J. Wiley, 2008.

LEE, Susanna ST et al. Role of CYP2E1 in the hepatotoxicity of acetaminophen. **Journal of Biological Chemistry**, v. 271, n. 20, p. 12063-12067, 1996.

LESK, Arthur M. **Introdução à bioinformática**. Artmed, 2008.

MANGALAM, Harry. The Bio\* toolkits—a brief overview. **Briefings in bioinformatics**, v. 3, n. 3, p. 296-302, 2002.

MARÉCHAL, J.-D. et al. Insights into drug metabolism by cytochromes P450 from modelling studies of CYP2D6-drug interactions. **British journal of pharmacology**, v. 153, n. S1, p. S82-S89, 2008.

MARGULIES, Marcel et al. Genome sequencing in microfabricated high-density picolitre reactors. **Nature**, v. 437, n. 7057, p. 376-380, 2005.

MARNETT, Lawrence J.; KENNEDY, Todd A. Comparison of the peroxidase activity of hemoproteins and cytochrome P450. In: **Cytochrome P450**. Springer US, 1995. p. 49-80.

MAVEN, In. Apache maven project. 2011.

MORGON, Nelson H.; COUTINHO, Kaline. **Métodos de química teórica e modelagem molecular**. Editora Livraria da Física, 2007.

MORRIS, Garret M. **What is AutoDock?**. Disponível em: <<http://autodock.scripps.edu/>>. Acesso em: 01 set. 2013.

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

NAJMANOVICH, Rafael et al. Side-chain flexibility in proteins upon ligand binding. **Proteins: Structure, Function, and Bioinformatics**, v. 39, n. 3, p. 261-268, 2000.

OPREA, Tudor. Chemoinformatics in Drug Discovery. Weinheim, 2005.

OWASP. **Unrestricted File Upload**. 2014. Disponível em: <[https://www.owasp.org/index.php/Unrestricted\\_File\\_Upload](https://www.owasp.org/index.php/Unrestricted_File_Upload)>. Acesso em: 07 maio 2015.

PATRICK, Graham L. **An introduction to medicinal chemistry**. Oxford university press, 2013.

PERI, Suraj; STEEN, Hanno; PANDEY, Akhilesh. GPMAW—a software tool for analyzing proteins and peptides. **Trends in biochemical sciences**, v. 26, n. 11, p. 687-689, 2001.

PEVZNER, Pavel. **Computational molecular biology: An algorithmic approach**. Cambridge: The MIT Press, 2000.

PORUBSKY, P.R.; BATTALIE, K.P.; SCOTT, E.E. Human cytochrome P450 2E1 structures with fatty acid analogs reveal a previously unobserved binding mode. **Journal of Biological Chemistry**, v. 285, p. 22282-22290, 2010.

PORUBSKY, P.R.; MENEELY, K.M.; SCOTT, E.E. Structures of human cytochrome P-450 2E1. Insights into the binding of inhibitors and both small molecular weight and fatty acid substrates. **Journal of Biological Chemistry**, v. 283, p. 33698-33707, 2008.

RONIS, M.J.J.; LINDROS, K.O.; INGELMAN-SUNDBERG, M.; The CYP2E Subfamily. In: IOANNIDES, Costas (Ed.). **Cytochromes P450: metabolic and toxicological aspects**. CRC Press, 1996.

SCHROEDER, Evelyn Koeche et al. Molecular dynamics simulation studies of the wild-type, I21V, and I16T mutants of isoniazid-resistant Mycobacterium tuberculosis enoyl reductase (InhA) in complex with NADH: toward the understanding of NADH-InhA different affinities. **Biophysical journal**, v. 89, n. 2, p. 876-884, 2005.

SEELIGER, Daniel; DE GROOT, Bert L. Ligand docking and binding site analysis with PyMOL and Autodock/Vina. **Journal of computer-aided molecular design**, v. 24, n. 5, p. 417-422, 2010.

SETUBAL, Carlos; MEIDANIS, João. **Introduction to Computational Molecular Biology**. Boston: PWS, 1997.

SHOICHET, Brian K. et al. Lead discovery using molecular docking. **Current opinion in chemical biology**, v. 6, n. 4, p. 439-446, 2002.

SOUSA, Sergio Filipe; FERNANDES, Pedro Alexandrino; RAMOS, Maria Joao. Protein–ligand docking: current status and future challenges. **Proteins: Structure, Function, and Bioinformatics**, v. 65, n. 1, p. 15-26, 2006.

SUSSMAN, Joel L. et al. Protein Data Bank (PDB): database of three-dimensional structural information of biological macromolecules. **Acta Crystallographica Section D: Biological Crystallography**, v. 54, n. 6, p. 1078-1084, 1998.

TORVALDS, Linus; HAMANO, Junio. Git: Fast version control system. URL <http://git-scm.com>, 2010.

VAN DER SPOEL, D. et al. The GROMACS Manual (version 5.0.4). 2014. Disponível em: <[http://manual.gromacs.org/online/getting\\_started.html](http://manual.gromacs.org/online/getting_started.html)>. Acesso em: 10 maio 2015

VAN GUNSTEREN, Wilfred F.; BERENDSEN, Herman JC. Computer simulation of molecular dynamics: Methodology, applications, and perspectives in chemistry. **Angewandte Chemie International Edition in English**, v. 29, n. 9, p. 992-1023, 1990.

VERLI, Hugo. Bioinformática da Biologia à flexibilidade molecular. Porto Alegre, 2014.

VOET, Donald; VOET, Judith G.; PRATT, Charlotte W. **Fundamentos de Bioquímica: A Vida em Nível Molecular**. Artmed Editora, 2014

WWPDB. **Protein Data Bank Contents Guide: Atomic Coordinate Entry Format Description.** 2011. Disponível em: <<http://www.wwpdb.org/documentation/file-format-content/format33/v3.3.html>>. Acesso em: 10 maio 2015.