PROGRAMA DE PÓS-GRADUAÇÃO MULTICÊNTRICO EM QUÍMICA DE MINAS GERAIS – UNIVERSIDADE FEDERAL DE ITAJUBÁ

FABIANA COSTA GUEDES

ProtCool: um Gerador de Protocolos para Ancoragens e Simulações de Dinâmica Molecular em Complexos Proteína-Ligante.

ITABIRA
2021

FABIANA COSTA GUEDES

ProtCool: um Gerador de Protocolos para Ancoragens e Simulações de Dinâmica Molecular em Complexos Proteína-Ligante.

Tese de doutorado submetida ao Programa de Pós-Graduação Multicêntrico em Química da rede Mineira de Química como requisito final para a obtenção do Título de Doutora em Química.

Área de Concentração: Química Teórica Computacional / Quimioinformática

Orientador: Prof. Dr. Carlos Henrique da Silveira - Unifei
Co-orientadora: Profa. Dra. Aleteia Patrícia Favacho de Araújo - UnB

ITABIRA
2021

À Deus.
À Maria Antônia, minha mãe e fonte de toda minha força!
Ao meu filho, Gabriel, com quem sempre aprendo. Obrigada por existir!
Ao meu marido, Inaldo, que entendeu meus momentos de dedicação a este trabalho, fazendo meus dias se tornarem mais leves. Obrigada por seu amor!
Aos meus irmãos, cunhados e sobrinhos. Sem vocês com certeza eu não conseguiria. Amo vocês!

AGRADECIMENTOS

“A gratidão é um fenômeno profundo e complexo que desempenha papel fundamental na felicidade e performance humana.” Emmons e McCullough, 2004, The Psychology of Gratitude.

Muita gente fala que o trabalho de um doutorado é isolado, porém, eu acredito que não é possível o desenvolvimento de um trabalho como esse sem a ajuda de diversas pessoas. E eu Graças a Deus posso dizer que tive diversas ajudas que sem elas não teria chegado aonde estou. Por isso, segue aqui os meus agradecimentos.

Primeiramente agradeço a Deus. Só devido a Ele tudo foi possível. Foi Ele que colocou as pessoas na minha vida e desenhou todos os caminhos para que tudo se concretizasse. Obrigada meu Deus!!

Agradeço ao meu filho Gabriel, que com seu jeito carinhoso sempre me ensina, mostrando muitas vezes mais maturidade do que eu própria. Ao meu marido Inaldo por estar sempre presente, mesmo distante durante o sanduíche na UnB. Só por causa de vocês dois, do apoio recebido e do carinho mesmo distante, foi possível passar esse tempo longe. Obrigada!!

Agradeço à minha família. Minha mãe que sempre foi minha fonte de inspiração. Aos meus irmãos, que estão sempre presentes, que dão forças, alegrias e segurança quando tive que estar longe. Aos meus sobrinhos que sempre trouxeram os sorrisos. Aos meus cunhados por sempre estarem próximos. Ao meu pai e ao Fernando, meu cunhado, que mesmo não estando mais presentes, fazem parte dessa história.

Ao meu orientador Carlos Henrique da Silveira, que mais que orientador, um amigo. Obrigada por compreender os momentos em que, devido a questões pessoais, me ausentei das atividades.

À Aleteia Patrícia Favacho Araújo, minha coorientadora. Primeiro pela recepção e carinho com que me recebeu em Brasília. Segundo pelos comentários e sugestões ao trabalho. Nossos encontros semanais sempre eram importantes para mim, tanto profissionalmente quanto pessoalmente. Espero levarmos a amizade ao longo da vida!

Aos professores do programa que me ajudaram na difícil tarefa de compreender a química. Todos vocês foram cruciais para a minha aprendizagem, muito obrigada. Queria fazer um agradecimento especial aos professores Daniel e Marcos, vocês dois souberam compreender a minha dificuldade em química e sempre se preocupavam se de fato estava entendendo a aula, explicando diversas vezes, se assim fosse necessário.

Agradeço ao amigo e professor Ernesto, sem você seria impossível compreender a quântica. Sempre disposto a me ajudar nas dúvidas, me atendendo diversas vezes fora do horário para que pudesse me ajudar. Já falei pessoalmente, mas não canso de agradecer, você foi essencial na minha formação. Muito obrigada sempre!

Aos alunos da Unifei Arthur, Guilherme, Pedro, Brenda, Augusto, Ian, Levi e Joana. Alunos de IC do laboratório. Que auxiliaram na preparação do ambiente e no desenvolvimento da pesquisa. Aos alunos da UnB Jeferson, Felipe, Rômulo e Pedro por me ajudarem a desvendar o BioNimbuZ.

Ao professor Leonardo Lima pelo auxílio no entendimento de dinâmica molecular. Aos pesquisadores Rafael (UFMG), Leon (aluno do Leonardo), Letícia e Pedro que me auxiliaram e “salvaram” sempre que precisava com o desenvolvimento das dinâmicas. Especial agradecimento ao Rafael, que me auxiliou sempre e se tornou um amigo.

Ao grupo de pesquisa BaBel, foi devido a esse grupo que foi possível a realização do meu sanduíche na UnB.

A todos os professores que participaram das bancas ao longo deste trabalho, professora Juliana Fedoce, professor Daniel Soares e professor Gerd Bruno da Rocha. Obrigada pelas considerações e questionamentos. Foram vocês que me fizeram crescer como pesquisadora. Aos professores Juliana Fedoce, Daniel Soares, Raquel Minardi e Lucianna Santos por terem aceito participar da banca final do doutorado.

A todos os professores da Engenharia da Computação da Unifei – Campus Itabira, que foram

sobrecarregados durante o meu afastamento. Muito obrigada!

A Rossana, Wandré e Giovani que durante a pandemia, nos encontros semanais que tínhamos devido ao trabalho, me fizeram ver que existia mais coisas além da tese, que me possibilitaram momentos de descontração e tornaram o ano de 2020 mais leve. Muito Obrigada!!

Ao amigo Rafael Francisco dos Santos, que assumiu a carga de desenvolvimentos das atas do NDE para que eu conseguisse finalizar este trabalho.

Aos meus amigos e demais familiares (tios e primos). Vocês possibilitaram as escapadas para aliviar as tensões.

À Unifei por possibilitar que ficasse afastada por dois anos para realização do doutorado.

À UnB que me recebeu como aluna do doutorado sanduíche, auxiliando na minha formação.

À CAPES que forneceu a bolsa auxílio moradia para realização do doutorado sanduíche (processo 88887.285491/2018-00).

Obrigada a todos!!

“Perder tempo em aprender coisas que não interessam, priva-nos de descobrir coisas interessantes.”
(Carlos Drummond de Andrade)

RESUMO

Nos últimos anos percebe-se uma grande evolução nas simulações em dinâmica molecular (DM), seja em precisão dos resultados quando comparado ao mundo real, seja na capacidade de representar sistemas biológicos complexos com milhares ou milhões de átomos. Apesar de todos os avanços nas teorias, algoritmos e infraestrutura computacional que dão suporte e confiança às simulações atuais, uma questão prática séria ainda persiste: a reprodutibilidade dos experimentos. Contribuem para isso tanto a falta de protocolos mais padronizados, bem como a falta de documentação abrangente do que foi realmente realizado. No sentido de tentar contribuir na solução a esses desafios, esta tese objetivou desenvolver, configurar, estruturar e verificar uma ferramenta que busca a automação de protocolos e *workflows* (chamada ProtCool) que possibilite o controle, a análise e a reprodução de experimentos em dinâmica molecular de proteínas e *docking* molecular com múltiplos ligantes. Para a realização desse objetivo principal, foi necessário percorrer pelos seguintes objetivos específicos: Modelar e implementar o *workflow* para preparação da simulação de dinâmica molecular; Modelar e implementar o *workflow* para realização de *dockings* com múltiplas moléculas; Realizar a implementação de *script* de gestão do *workflow*; Implementar a ferramenta de proveniência de dados, possibilitando que o pesquisador tenha todos os arquivos e dados gerados durante a preparação da simulação; Realizar a automatização de preparação de dinâmicas moleculares; Executar apenas parte do *workflow*; Realizar a reprodutibilidade de experimentos e de metodologias de pesquisa; Integrar tudo isso numa ferramenta chamada ProtCool; Verificar a ferramenta com estudos de casos envolvendo *docking* de ligantes com acetilcolinesterase humana e protease principal (Mpro) do SARS-Cov-2. Como estudo de casos para a validação e demonstração de uso da ferramenta, foram usados como alvos a acetilcolinesterase humana, implicada em doenças como mal *Alzheimer*, e a protease principal (Mpro) do SARS-CoV-2, o vírus responsável pela atual pandemia da COVID19. Para a acetilcolinesterase foram usados os mesmos 4 ligantes (galantamina, licoramina, sanguinina e um ligante híbrido) e alvo (4EY6) estudados por Rocha (2017) como forma de verificar a implementação do ProtCool, em que se produziu resultados equivalentes. Como forma de indicar o potencial uso do ProtCool na triagem virtual de ligantes em larga escala, num problema relevante e atual, foi realizado o *docking* molecular de múltiplos ligantes (19637 do ZINC, 8752 do *Drugbank* e 8520 do SistematX, totalizando 36909 ligantes) em múltiplos alvos (6 conformações diferentes amostradas por Metadinâmica) envolvendo a Mpro do SARS-CoV-2, tendo como resultado a geração de 4427839 poses (usando sistemas de *docking Vina* e *Smina*). Com isso, foi possível mostrar uma nova estratégia in silico de indicar ligantes inéditos como candidatos a antivirais contra COVID19.

Palavras-chaves: *Workflows*, Dinâmica Molecular, *Docking* Molecular, e-Science, ProtCool.

ABSTRACT

In recent years, a significant evolution in the simulations in molecular dynamics (DM) has been noticed either in the precision of results compared to the real world or in the capacity to represent complex biological systems with thousands or millions of atoms. Despite all the advances in theories, algorithms, and computational infrastructure that give support and confidence to the current simulations, a serious practical issue persists: the reproducibility of experiments. Other concerning aspects are the lack of more standardized protocols, the lack of comprehensive documentation of what has been accomplished. In order to try to contribute to the solution to these challenges, this thesis aimed to develop, configure, structure and verify a tool that seeks the automation of protocols and workflows (called ProtCool) that enables the control, analysis and reproduction of experiments in molecular dynamics of proteins and molecular docking with multiple ligands. To achieve this main objective, it was necessary to go through the following specific objectives: Model and implement the workflow for preparing the molecular dynamics simulation; Model and implement the workflow for performing multi-molecule dockings; Perform workflow management script implementation; Implement the data source tool, allowing the researcher to have all the files and data generated during the simulation preparation; Automate the preparation of molecular dynamics; Run only part of the workflow; Perform the reproducibility of experiments and research methodologies; Integrate all this into a tool called ProtCool; Check the tool with case studies involving docking of ligands with human acetylcholinesterase and major protease (Mpro) from SARS-Cov-2. Human acetylcholinesterase, implicated in diseases such as Alzheimer's disease, and the main protease (Mpro) of SARS-CoV-2, the virus responsible for the current pandemic of the disease COVID-19, were used as targets in case studies for validating and demonstrating the use of the tool. For acetylcholinesterase, the same 4 ligands (galantamine, lycoramine, sanguinine and a hybrid ligand) and target (4EY6) studied by Rocha (2017) were used as a way to validate the implementation of ProtCool, producing equivalent results. As a way of indicating the potential use of ProtCool in the virtual screening of ligands on a large scale, in a relevant and current problem, the molecular docking of multiple ligands (19637 from ZINC, 8752 form Drugbank, and 8520 from SistematX, totaling 36909 ligands) on multiple targets (6 different conformations sampled by Metadynamics) involving the Mpro of SARS-CoV-2 was carried out, resulting in the generation 4427839 poses (using Vina and Smina docking systems). It was possible to show a new in silico strategy to indicate new ligands as candidates for antivirals against COVID19.

Keywords: *Workflows, Molecular Dynamics, Molecular Docking, e-Science, ProtCool.*

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

Ach	Acetilcolina - *Acetylcholine*
AchE	Acetilcolinesterase Humana – Acetylcholinesterase Human
AMBER	*Assisted Model Building with Energy Refinement*
AMD	*Advanced Micro Devices* - Processador
API	*Application Programming Interface*
^{o}C	Unidade de medida. Graus Celsius.
CHARMM	*Chemistry at Harvard Macromolecular Mechanics*
CPU	Unidade de Processamento Central
CPU/GPU	Unidade de Processamento Central / Unidade de Processamento Gráfico
DA	Doença de *Alzheimer*
DDR3	*Memory Double Data Rate type 3*
ECC	*Error-correcting code*
GB	*GigaByte* – Unidade de medida
GHz	*GigaHertz* – Unidade de medida
GNT	Galantamina - *Galantamine*
GPGPU	Unidade de Processamento Gráfico de Propósito Geral – *General Purpose Graphics Processing Unit*
GROMOS	*Groningen Molecular Simulations*
HDD	*Hard disk drive*
HETATM	Heteroátomo
HYB	Híbrido de SNG e LYC
K	Kelvin – Unidade de Temperatura
kDa	KiloDalton –Unidade de massa atômica
LBVS	Ligand-Based Virtual Screenig
LYC	Licoramina - *Lycoramine*
MD	Dinâmica Molecular – *Molecular Dynamic*
MHz	*MegaHertz* – Unidade de Medida
MMPBSA	*Molecular Mechanics Poisson-Boltzmann Surface Area*
Mpro	*Main protease* – principal protease do SARS-Cov-2
NAMD	*Not Another Molecular Dynamics*
NMR	*Nuclear Magnetic Resonance* – Ressonância Magnética Nuclear
NTF	Novelos neurofibrilares
Sars-cov-2	Corona Vírus
SSD	*Solid-state drive*
OPLS	*Optimized Potentials for Liquid Simulation*
PDB	*Protein Data Bank*
pH	Unidade de Medida
PME	*Particle Mesh Ewald*
PROV-DM	PROV *Data Model*
RAID1	*Redundant Array of Inexpensive Drives type 1*
RAM	Memória de Acesso Aleatório – *Random Access Memory*
RMSD	Deslocamento Quadrático Médio – *Root Mean Square Deviation*
RPC	Chamada de Procedimento Remoto
SFTP	*Secure File Transfer Protocol* – Protocolo seguro de transferência de arquivo
SNG	Sanguinina - *Sanguinine*
SLA	Acordo de Nível de Serviço – *Service-Level Agreement*
SO	Sistema Operacional
TB	*TeraByte* – Unidade de medida

TBVS	*Target-based Virtual Screenig*
TC	*Task Coordinator* – Coordenador de Tarefa
TE	*Task Executor* – Executores de Tarefas
TER	Término de parte do arquivo PDB
TINKER	*Software Tools for Molecular Design*
UFMG	Universidade Federal de Minas Gerais
VS	*Virtual Screening*
WfMS	*Workflows Management Systems* – Sistema Gerenciador de *Workflows*

SUMÁRIO

1. INTRODUÇÃO

No século XX, com o advento dos computadores, a Ciência investiu em um ramo com forte lastro computacional, agregando novas modelagens, simulações, análises numéricas e estatísticas intrincadas. O computador foi ampliando espaços na pesquisa científica. A consequência desse avanço é a profusão de dados e da complexidade dos experimentos, em crescimento exponencial.

Nesse contexto, surgem os *workflows* científicos que possibilitam que os experimentos sejam organizados em atividades. Além disso, eles possibilitam que ocorra a gestão e guarda de dados, a reprodutibilidade de experimentos e a facilidade de uso. Assim, os sistemas gerenciadores de *workflow* científicos, também conhecidos como *e-science*, são descritos como uma opção para que os experimentos *in silico* possam ser organizados e gerenciados (ALMEIDA *et al.*, 2017).

Neste trabalho foi possível a modelagem, implementação e verificação de *workflows* em química computacional por meio da ferramenta denominada ProtCool: um que trata sobre a preparação de experimentos de dinâmica molecular e outro que possibilita a realização de *dockings* moleculares com múltiplos ligantes, para realização de *virtual screening* (VS). Cabe destacar que em química computacional tais *workflows* se confundem com os protocolos que organizam e implementam as execuções das simulações moleculares. Esta confusão ocorre uma vez que os protocolos são definidos por ferramentas, parâmetros, passos, dados e uma sequência definida e isso também ocorre com os *workflows* científicos.

Os protocolos de dinâmicas moleculares são a metodologia empregada para que a pesquisa possa ser realizada. Assim, na definição do protocolo o pesquisador deve especificar os detalhes da preparação dos arquivos, ou seja, deve especificar as ferramentas utilizadas, bem como, os parâmetros de cada uma das ferramentas. A ProtCool permite que o pesquisador, a partir da definição do arquivo de configuração, possa gerar diferentes protocolos para a preparação de simulações de dinâmicas moleculares ou *docking* molecular. Além disso, a ProtCool consegue realizar o registro e guarda dos protocolos, bem como, dos arquivos gerados ao longo da preparação e *docking*.

A ProtCool, conforme será descrito no decorrer desta tese, trabalha com dois protocolos (*workflows*) distintos. O primeiro protocolo é o de preparação de dinâmica molecular. Este protocolo realiza a preparação de um sistema com receptor e ligante. Esta parte da ferramenta é chamada de ProtCool_Dynamic. O segundo protocolo trata de *docking* de múltiplos ligantes para realização de VS. Esta parte da ferramenta é chamada de ProtCool_Docking.

O VS ou triagem virtual é o processo de busca de um conjunto de ligantes que possam ter uma funcionalidade requerida. De forma geral, o VS pode ser classificado em: *Target-Based Virtual Screening* – TBVS e *Ligand-Based Virtual Screening* – LBVS. As primeiras levam em consideração informações tanto do alvo quanto do ligante; as segunda, somente dos ligantes e geralmente é usada quando os alvos não são conhecidos. O *docking* ou ancoragem dos ligantes a um alvo conhecido é exemplo de uma técnica TBSV (DE PINHO VELOSO, 2019).

Assim sendo, uma etapa importante e necessária é a realização de *dockings* com múltiplos ligantes e que forneçam para etapas posteriores de mineração de dados, as conformações e os *scores* alcançados por cada ligante. A *ProtCool_Docking* se insere nesse contexto, uma vez que possibilita que *dockings* sejam realizados, com capacidade de utilizar ferramentas e algoritmos distintos, oferecendo maiores chances de resultados mais abrangentes e robustos.

Para verificação dos resultados foram utilizadas as proteínas: Acetilcolinesterase Humana (AchE) (WIESNER *et al*., 2007), envolvida em neuropatologias como o mal de *Azheimer*, e a principal protease (Mpro) do SARS-Cov-2 (JIN *et al.*, 2020), o vírus responsável pela atual pandemia da COVID19.

A escolha dessas proteínas alvos deveu-se às suas importâncias biomédicas e sanitárias. A doença de *Alzheimer* (DA) é uma doença degenerativa que afeta diversas pessoas, trazendo impactos tanto para o paciente quanto para a família, devido aos fatores sociais que envolvem a doença, como perda de memória e modificações de personalidade que provoca nos pacientes (SERENIKI, VITAL, 2008;

BORGES *et al.*, 2018). Existem estudos que relatam que um possível mecanismo de ação da neurodegeneração é a hipótese colinérgica (ROCHA, 2017). Nessa hipótese a Acetilcolina é reduzida em pacientes com DA e a Acetilcolinesterase é uma das responsáveis para que isso ocorra. Medicamentos que inibem a ação da Acetilcolinesterase são utilizados de forma a regular a quantidade de Acetilcolina no organismo (WIESNER *et al.*, 2007; BORGES *et al.*, 2018). Neste trabalho será apresentado um estudo de caso com a Acetilcolinesterase, com o objetivo de demonstrar o uso do ProtCool de preparação de dinâmicas moleculares em proteínas.

O SARS-Cov-2 é um coronavírus que causa a síndrome aguda respiratória 2, responsável pela pandemia de 2020. Ainda não existem fármacos disponíveis para o tratamento eficiente da doença (JIN *et al.*, 2020). A MPro do SARS-Cov-2 será utilizada com o objetivo de demonstrar o uso do ProtCool para realização de *docking* com múltiplos ligantes.

Foi desenvolvido um *script* gerenciador de todo o sistema que permite a gestão dos *workflows* na ferramenta ProtCool. O *script* é programado com atividades em paralelo e permite o registro da proveniência de dados do sistema. Para facilitar a geração do arquivo de configuração para a execução dos *workflows*, foi desenvolvida uma interface gráfica amigável, embora o ProtCool possa ser utilizado sem ela.

Muitas são as razões que justificam uma tese como esta. Uma mudança importante foi a introdução da computação em diversas áreas de pesquisa, possibilitando estudos de simulações de fenômenos complexos. Além do uso da computação para a realização de Ciência, a exploração de grandes quantidades de dados (*e-Science*) fez com que os experimentos tivessem uma vasta quantidade de informação a ser processada e analisada. Novas tecnologias são necessárias, nesse contexto, sendo uma importante tecnologia a utilização de *workflows*, para facilitar o gerenciamento e a reprodutibilidade de experimentos (BRAGHETTO e CORDEIRO, 2017).

Purawat *et al.* (2017) afirmam que, em geral, as dinâmicas moleculares (MD) de alto desempenho envolvem longas simulações, sendo executadas por muito tempo, gerando grande número de arquivos, o que requer métodos computacionais que facilitem e organizem todo o processo. Os *workflows*, nesse sentido, podem auxiliar construindo processos completamente automatizados, facilitando, com isso, tanto a execução de novas simulações quanto a replicação de simulações já realizadas.

Além disso, em experimentos de dinâmica molecular (MD), uma das principais críticas é a reprodutibilidade de experimentos. A partir dos protocolos descritos na metodologia do trabalho, é possível refazer os experimentos, porém, nem sempre se tem todas as informações descritas adequadamente. Assim, Purawat *et al.* (2017) ressaltam essa falha afirmando que reproduzir experimentos publicados de MD pode ser bastante desafiador devido há diversos fatores, tais como: procedimentos de execução complicados que dependem de *scripts* desenvolvidos pelos usuários, reprodutibilidade técnica (compilador, biblioteca, algoritmo etc.), reprodutibilidade estatística (número de réplicas do mesmo experimento) e relato incompleto dos métodos utilizados. A geração de *workflows* pode contribuir para melhorar a reprodutibilidade de experimentos, uma vez que com o fluxo montado tem-se a estrutura para que novos experimentos sejam realizados, tanto para as mesmas moléculas quanto para moléculas distintas. Com as ferramentas de *workflows* é possível não só descrever o método utilizado, mas disponibilizar aos pesquisadores os códigos e os resultados alcançados, facilitando a reprodutibilidade de experimentos.

É importante destacar que não se trata de reproduzir os experimentos de forma a se ter os mesmos resultados. Os sistemas utilizados nesta pesquisa são não determinísticos, assim, não é possível que se chegue ao mesmo resultado. Porém, pode-se reutilizar os protocolos que são utilizados. Além disso, o armazenamento do protocolo e dos arquivos gerados durante o processo auxiliam na gestão dos laboratórios.

Dessa forma, a implementação de *workflows* possibilitará uma melhoria no dia a dia dos pesquisadores, facilitando tanto a preparação de experimentos, quanto a sua guarda. Além disso, com um controle maior da quantidade de dados gerada durante as pesquisas, será possível a realização de pesquisas com maior qualidade, trazendo ganhos para a comunidade acadêmica.

A QwikMD, que é uma ferramenta que possibilita a preparação, execução e análise de dinâmicas moleculares, utilizando para isso os sistemas VMD e NAMD (RIBEIRO *et al.*, 2016), se assemelha com a ProtCool, principalmente no que diz respeito ao *workflow* da dinâmica (ProtCool_Dynamic). Porém, a QwikMD realiza a preparação de proteínas, enquanto o ProtCool_Dynamic realiza a preparação de proteínas complexadas a ligantes. Além disso, a ProtCool_Dynamic possui um conjunto de ferramentas que são associadas a ela permitindo que acertos na estrutura da proteína sejam realizados. Assim, o Modeller, que é uma ferramenta de modelagem que realiza os acertos dos *gaps* identificados, é utilizado. O ProtCool utiliza, também, um sistema de verificação da protonação, o H++. A questão das restrições harmônicas é estabelecida, o que possibilita que o protocolo de dinâmica seja devidamente especificado pelo pesquisador. Isso garante que o pesquisador possa determinar quantas MD são realizadas para minimização e relaxamento, definindo todo o protocolo. A ProtCool_Dynamic também realiza o *docking* da proteína com diversos ligantes utilizando duas ferramentas de *docking*. Isso possibilita a dinâmica de complexos. Podem ser preparadas diversas moléculas ao mesmo tempo. Estes pontos não foram identificados na QwikMD.

É importante destacar que esta pesquisa se enquadra na interface da área de química, ou seja, ela não está pautada em apresentar resultados puramente químicos, mas sim, de apresentar uma nova ferramenta que auxilia o químico na realização das suas pesquisas. O objetivo desta tese é a de oferecer um ferramental para controle de experimentos, contribuindo para a gestão eficiente dos laboratórios de pesquisa.

Na Seção 1 deste trabalho é apresentada a introdução, com a descrição dos objetivos. A Seção 2 é a responsável pelo referencial teórico do trabalho. Essa seção está dividida nos seguintes temas: *workflows* científicos, proveniência dos dados, *BioNimbuZ* e dinâmica molecular. A Seção **Erro! Fonte de referência não encontrada.** apresenta a ProtCool e descreve a metodologia utilizada para o desenvolvimento da ferramenta e as duas ferramentas ProtCool_Dynamic e ProtCool_Docking. A Seção **Erro! Fonte de referência não encontrada.** apresenta a Acetilcolinesterase humana, mostrando um estudo de caso de utilização da ProtCool_Dynamic. A Seção 4 apresenta o estudo de caso da ProtCool_Docking que foi realizado com a MPro da SarCoV-2. Finalmente a Seção **Erro! Fonte de referência não encontrada.** apresenta as considerações finais da pesquisa, mostrando os próximos passos a serem desenvolvidos.

1.1.Objetivos

Desenvolver, configurar, estruturar e verificar uma ferramenta que busca a automação de protocolos e *workflows* (chamada ProtCool) que possibilite o controle, a análise e a reprodução de experimentos em dinâmica molecular de proteínas e *docking* molecular com múltiplos ligantes.

Objetivos específicos:

i. Modelar e implementar o *workflow* para preparação da simulação de dinâmica molecular;
ii. Modelar e implementar o *workflow* para realização de *dockings* com múltiplas moléculas;
iii. Realizar a implementação de *script* de gestão do *workflow*;
iv. Implementar a ferramenta de proveniência de dados, possibilitando que o pesquisador tenha todos os arquivos e dados gerados durante a preparação da simulação;
v. Realizar a automatização de preparação de dinâmicas moleculares;
vi. Executar apenas parte do *workflow*;
vii. Realizar a reprodutibilidade de experimentos e de metodologias de pesquisa;
viii. Integrar tudo isso numa ferramenta chamada ProtCool;
ix. Verificar a ferramenta com estudos de casos envolvendo *docking* de ligantes com acetilcolinesterase humana e protease principal (Mpro) do SARS-Cov-2.

2. REVISÃO BIBLIOGRAFICA

2.1. *Workflows* Científicos

A quantidade de informações e processos existentes em experimentos científicos faz com que sejam desenvolvidos *workflows* para gerenciar todo o processo. Um *workflow* pode ser entendido como um conjunto de tarefas bem definidas que são ordenadas de forma a se atingir um objetivo específico, além disso, é importante que esses passos possam ser realizados de forma idêntica em uma segunda execução (YU e BUYYA, 2005; SALDANHA, 2012, PURAWAT *et al.*, 2017). Os *workflows* são compostos de: grupos de dados, fases de análise, fluxos e ferramentas (SALDANHA, 2012). O *workflow* científico pode ser entendido como a automatização de um experimento, em que ocorre destaque às tarefas que devem ser executadas, aos dados que serão processados, e as dependências existentes entre os dados que estão em processamento (BRAGHETTO e CORDEIRO, 2014).

Os *workflows* científicos executam uma grande quantidade de informação, e as conexões entre as atividades mostram o fluxo que os dados devem percorrer a fim de gerar a informação ao final do processo (SALDANHA, 2012). Dessa forma, o fluxo de controle passa a ser apenas uma representação complementar de todo o processo (BRAGHETTO e CORDEIRO, 2014). *Workflows* científicos possuem algumas características e propósitos particulares, sendo que esses devem respeitar uma ordem para serem executados e precisam de uma modelagem e gerenciamento *ad-hoc* (HONDO *et al.*, 2017).

Assim, os *workflows* científicos devem ser preparados para execução de longa duração, com grande quantidade de dados, recursos heterogêneos, vários domínios de aplicação e disponibilidade de recursos dinâmicos (YU e BUYYA, 2005). São adequados para processos com múltiplos passos de análise, e que utilizam diversas ferramentas de software, realizando reutilização, tanto do fluxo quanto dos dados (GULER *et al.*, 2016).

Os Sistemas de Gerenciamento de *Workflows* (*Workflows Management Systems* – WfMS, Sistema Gerenciador de *Workflows* – também chamados de *e-Sciences*) surgiram para facilitar essa atividade, tendo como função a automatização da execução de *workflows*, e, também, possibilita o monitoramento das fases de um *workflow* (SALDANHA, 2012). Os WfMS são sistemas computacionais que possibilitam a execução de aplicações científicas que foram modeladas como *workflows*, ou seja, que possuem atividades que tem uma ordem lógica (BRAGHETTO e CORDEIRO, 2014). Alguns exemplos de sistemas gerenciadores de *workflow* são: *Discovery Net* (ROWE *et al.*, 2003), *Kepler* (LUDÄSCHER *et al.*, 2006), *Taverna* (HULL *et al.*, 2006), *Triana* (TAYLOR *et al.*, 2005), *e-science Central* (HIDEN *et al.*, 2013), *BioNimbuz* (SALDANHA, 2012).

Alguns requisitos que devem ser levados em consideração por WfMS, são eles: alta taxa de processamento – os *workflows*, principalmente os científicos, possuem uma grande quantidade de dados, grande quantidade de tarefas e que precisam de muito tempo de processamento; facilidade de uso – deve ser fácil e intuitivo de se utilizar, com interface gráfica amigável; flexibilidade – possibilitar a inclusão de ferramentas, já que cada *workflow* poderá tratar de assuntos diferentes; modularidade – os *workflows* devem ser tratados de forma a que cada tarefa seja um módulo, podendo ser reexecutado apenas parte do *workflow* que foi afetado; tolerância a falhas – o sistema deve se recuperar de falhas, reiniciando o *workflow* na fase em que ocorreu o problema, sem a necessidade de se executar todo o *workflow* novamente; reprodutibilidade – o sistema deve possibilitar a reprodução dos experimentos (SALDANHA, 2012).

Purawat *et al.* (2017) ressaltam a importância dos *workflows* científicos para realizar a reprodutibilidade de experimentos. Essa reprodutibilidade é tão importante para refazer experimentos de outros pesquisadores, quanto para validação de seus próprios experimentos (PURAWAT *et al.*, 2017). Além disso, deve-se atentar para aqueles experimentos que são realizados para diversas moléculas diferentes. A reutilização dos *workflows* acontece nos seguintes níveis: reutilização com parâmetros e dados diversos; modificação do *workflow* para refinar o método de pesquisa;

compartilhamento com outros pesquisadores, visando o trabalho em grupo (CUEVAS-VICENTTÍN *et al.*, 2012). Estas três formas de reutilização são realizadas pelo ProtCool.

Diversos pesquisadores, para facilitar o trabalho em experimentos, desenvolvem *scripts* e vão utilizando-os em cada etapa do experimento (PURAWAT *et al.*, 2017). Os *workflows* podem facilitar, ordenando e automatizando o processo, e com isso facilitará a reprodutibilidade, o desenvolvimento de métodos e o treinamento de pesquisadores (PURAWAT *et al.*, 2017). Nos *workflows* científicos é adequado considerar o desenvolvimento de *scripts* para a realização das tarefas que devem ser realizadas, mas eles podem obedecer a um conjunto de regras, tais como: o *script* pode ser interrompido e, posteriormente, retornar à sua execução; eles devem ser recuperáveis, mesmo que tenha que retornar a um ponto anterior à sua execução; os *scripts* devem ser executados remotamente e em paralelo, caso existam condições técnicas para isto; a entrada de um *script* é realizada a partir de saídas de outros elementos do processo (BARGA e GANNON, 2007).

Vantagens na utilização de *workflows* científicos nas pesquisas *in silico*: os *workflows* são normalmente desenvolvidos de tarefas repetitivas, possibilitando que os cientistas se concentrem mais nas análises do resultado, do que no processo de geração dos dados; os *workflows* documentam todo o processo científico, garantindo uma melhor divulgação, colaboração e reprodutibilidade; é realizada toda a proveniência de dados, o que auxilia em diversos momentos da pesquisa; os processos científicos são mais eficientes, trazendo maior agilidade; os sistemas de *workflow* facilitam e incentivam a reutilização dos diversos artefatos, facilitando a criação e a modificação de *workflows* (LUDÄSCHER *et al.*, 2009). Assim, um item importante a respeito dos *workflows* é a sua modelagem, ou seja, a construção do modelo de *workflow*, sua execução e verificação. Isso será tratado na Seção 2.1.1.

2.1.1. Modelagem, Execução e Verificação de *Workflows* Científicos

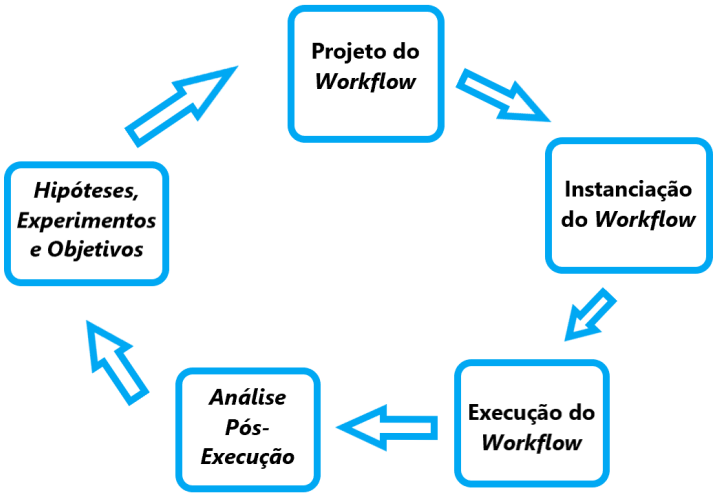
A Figura 1 apresenta o ciclo de vida de um *workflow* científico, que é formado de quatro fases principais, as quais são: projeto, instanciação, execução, e análise pós-execução (BRAGHETTO e CORDEIRO, 2014). Além dessas quatro fases, existe uma fase de hipóteses, experimentos e objetivos, que é a responsável pela definição do *workflow* como um todo (BRAGHETTO e CORDEIRO, 2014). Internamente, é importante ter a infraestrutura necessária para que o *workflow* possa ser executado (fontes de dados, repositório de *workflows*, dados de proveniência e recursos computacionais), e, também, o monitoramento em tempo de execução, a partir dos dados de proveniência (BRAGHETTO e CORDEIRO, 2014).

Deelman *et al.* (2009) e Ludäscher *et al.* (2009) trazem uma divisão um pouco diferente em seus ciclos de vida do desenvolvimento do *workflow*, mas na verdade, analisando os processos verifica-se que é apenas uma questão notacional, todas as atividades são realizadas nos três fluxos, em momentos distintos. Para Deelman *et al.* (2009) as fases do ciclo de vida são: composição, mapeamento, execução, metadados e proveniência. Deelman *et al.* (2009) também consideram que pode ocorrer a realimentação da modelagem, gerando novos *workflows* tanto das partes de um já existente, como da criação de versões a partir dos resultados alcançados nas pesquisas. Ludäscher *et al.* (2009) propõem as seguintes fases: *design* e composição do *workflow*, planejamento dos recursos do *workflow*, execução dos *workflows*, análise da execução dos *workflows*, compartilhamento dos resultados, e funções dos usuários.

O *workflow* científico é gerado a partir de hipóteses científicas a serem testadas ou objetos de pesquisa específicos que se deseja estudar (BRAGHETTO e CORDEIRO, 2014). A partir dessa etapa é gerado o projeto de *workflow* que será desenvolvido e testado. O projeto de *workflow* é um modelo que será seguido para que a hipótese de pesquisa possa ser testada (BRAGHETTO e CORDEIRO, 2014). Os modelos se baseiam muitas vezes na metodologia de pesquisa que será desenvolvida. Alguns projetos podem ser derivados de outros modelos já testados, ou podem usar resultados de *workflows* já existentes (BRAGHETTO e CORDEIRO, 2014). Para Deelman *et al.* (2009), essa etapa é chamada

de composição, e os autores destacam que esse processo pode ser iterativo, podendo ocorrer a execução de partes do *workflow* para que o restante seja modelado. Os modelos de *workflows* são gerados pela especificação de requisitos dos cientistas (LUDÄSCHER *et al.*, 2009). Nesse momento é realizada a programação de cada tarefa do *workflow*, de forma que elas atendam à especificação dos requisitos, não é uma programação de um sistema tradicional, uma vez que os programadores devem se preocupar com cada tarefa isoladamente, e não com o sistema como um todo (LUDÄSCHER *et al.*, 2009).

Figura 1 – Fases do ciclo de vida de um *workflow* científico. Mostra todas as etapas importantes para que um *workflow* possa ser modelado, passando pelo projeto, instanciação, execução e análise para prever melhorias no projeto. O ProtCool foi gerado a partir desse modelo, seguindo essas etapas e se preocupando com os devidos armazenamentos.



Fonte: Adaptado de Braghetto e Cordeiro (2014, p. 3).

A etapa de instanciação é responsável pela correta preparação do ambiente para que ocorram as execuções do *workflow*, que compreende a indicação de arquivos (dados) de entrada e a definição de parâmetros específicos por parte do desenvolvedor (BRAGHETTO e CORDEIRO, 2014). Nesse momento existe a alocação e a seleção de recursos computacionais necessários, e toda a parte de transferência de arquivos iniciais (BRAGHETTO e CORDEIRO, 2014). Deelman *et al.* (2009) determinam uma etapa de mapeamento (similar à instanciação), em que as partes do *workflow* são tratadas isoladas, e enviadas para a execução, nesse momento, são realizadas as otimizações e o escalonamento de recursos. Ludäscher *et al.* (2009) realizam as atividades da instanciação na etapa de planejamento dos recursos do *workflow*, nessa etapa são determinados os recursos, que podem ser estabelecidos tanto manualmente, quanto automaticamente por um sistema. Os autores também destacam que nessa etapa toda a parte de validação do *workflow* deve ser realizada de forma a garantir que os requisitos dos pesquisadores sejam atendidos (LUDÄSCHER *et al.*, 2009).

A fase de execução é a responsável por processar o *workflow*. É o momento em que cada atividade será executada a fim de gerar os devidos resultados (BRAGHETTO e CORDEIRO, 2014; DEELMAN *et al.*, 2009; LUDÄSCHER *et al.*, 2009). Toda a execução das atividades gera arquivos intermediários e finais, que podem servir como entrada para a próxima tarefa (BRAGHETTO e CORDEIRO, 2014; DEELMAN *et al.*, 2009; LUDÄSCHER *et al.*, 2009). Todo o percurso, resultados e informações a respeito de cada execução de tarefa devem ser guardados, visando tanto uma reprodução do experimento, quanto a recuperação em caso de falha (BRAGHETTO e CORDEIRO, 2014; DEELMAN *et al.*, 2009; LUDÄSCHER *et al.*, 2009). É importante a coleta dos dados para a realização de uma proveniência de dados adequada e para a recuperação em caso de falhas (LUDÄSCHER *et al.*, 2009).

A funcionalidade de guarda dos dados é chamada de proveniência de dados, e será descrita na Seção 2.2. Durante a execução das atividades deve existir um monitoramento constante da execução e os dados de proveniência também auxiliam nesse momento, podendo o cientista analisar a qualquer momento o andamento da sua execução e redefinir parâmetros para ajuste da execução (BRAGHETTO e CORDEIRO, 2014). São os dados da proveniência que podem ser analisados para gerar ou editar *workflows*, de modo a facilitar a melhoria e a geração de pesquisas (DEELMAN *et al.*, 2009).

A etapa de análise pós-execução é a etapa que após o *workflow* ser executado, o cientista terá em mãos um conjunto de resultados que deve ser analisado (BRAGHETTO e CORDEIRO, 2014). A partir dessa análise, podem surgir novas hipóteses que trarão a necessidade de nova geração de *workflow* (BRAGHETTO e CORDEIRO, 2014). Assim sendo, é possível que a partir de uma análise o cientista verifique que o resultado esperado não foi alcançado, fazendo com que o pesquisador modifique o *workflow* e faça novas execuções do experimento (BRAGHETTO e CORDEIRO, 2014). Além disso, o pesquisador ainda pode utilizar o mesmo *workflow* para gerar novos experimentos, com dados e parâmetros de entrada distintos, gerando novas análises facilmente (BRAGHETTO e CORDEIRO, 2014). O cientista tem o papel nessa etapa de responsável pela análise dos dados e pela tomada de decisão a partir dos dados coletados (LUDÄSCHER *et al.*, 2009).

Os *workflows* científicos tornaram-se mais complexos, o que requer métodos para que sejam verificados de acordo com a sua correção (SILVA *et al.*, 2010). Muitas vezes apenas se considera que os *workflows* estão de acordo com os controles e as dependências definidas pelo pesquisador, mas os *workflows* devem ser confiáveis, o que requer que sejam especificados corretamente (SILVA *et al.*, 2010). Dessa forma, quando se garante a correção de um *workflow*, existe uma diminuição do trabalho do pesquisador, logo, técnicas de verificação são importantes e devem ser aplicadas em *workflows* (SILVA *et al.*, 2010).

Segundo Silva *et al.* (2010) os problemas de especificação que podem ser pesquisados são: *deadlock* (dois ou mais processos são impedidos de continuarem as suas execuções), falhas de sincronização, tempo de execução, todas as atividades são executadas pelo menos uma vez e ordem de execução das atividades (existem algumas atividades que sempre devem ser executadas em uma determinada ordem).

Após a modelagem do *workflow* é necessário realizar o seu escalonamento, que é a etapa de definição dos recursos necessários para que cada atividade do *workflow* seja realizada, além da etapa de codificação das atividades, de forma a possibilitar que o *workflow* seja executado (BRAGHETTO e CORDEIRO, 2014). Nesse caso, o escalonamento deve ser redigido de acordo com a demanda, a prioridade, a oferta de recursos e o tempo de execução (BRAGHETTO e CORDEIRO, 2014). A computação em nuvem pode ser utilizada para que se possa estabelecer uma maior quantidade de recursos, mesmo que isso venha com algum custo adicional (BRAGHETTO e CORDEIRO, 2014).

2.2.Proveniência de Dados

A proveniência de dados pode ser entendida como o processo de geração de um dado, desde a sua origem, até quando ele é armazenado em um banco de dados (ALMEIDA *et al.*, 2017). É importante, pois é ela que possibilita que um pesquisador possa revisar os detalhes de um experimento, permitindo a sua análise, a validação, a revisão e a reexecução (ALMEIDA *et al.*, 2017).

Os *e-sciences* tais como Kepler, *e-Science Central* e BioNimbuZ têm como sua principal característica a capacidade de armazenar e de compartilhar os dados que foram gerados, e os experimentos que foram desenvolvidos (MISSIER *et al.*, 2012). Para que haja a geração de conhecimentos, e confecção de artigos, diversos experimentos precisam ser gerados e os artigos apresentam apenas algumas visões do que foi gerado (MISSIER *et al.,* 2012). Os resultados, mesmo os parciais, incorretos ou que não foram selecionados devem permanecer organizados, e deve-se permitir que sejam refeitos a qualquer momento (MISSIER *et al.*, 2012). Hondo *et al.* (2017) acreditam que o papel da ciência é gerar conhecimento por meio de experimentos que possam ser de fato

reproduzidos. *In silico* essa tarefa nem sempre é corriqueira, pois muitas vezes os experimentos não são adequadamente gerenciados (HONDO *et al.*, 2017).

Além disso, na pesquisa *in silico*, existem diversos programas e parâmetros que podem ser utilizados, e a definição e a utilização dos parâmetros e programas afetam os dados gerados e os resultados alcançados. As configurações computacionais são importantes para que ocorra a reprodutibilidade (HONDO *et al.*, 2017). A geração da proveniência de dados em *workflows* possibilita a reprodução de experimentos, com os mesmos parâmetros de entrada, ou a mudança de alguns parâmetros, com o objetivo de analisar quais são mais importantes para o problema proposto, sem a perda de dados anteriores.

Os *e-Sciences* devem possuir a habilidade de proveniência de dados. Purawat *et al.* (2017) reforçam a importância da reprodutibilidade de experimentos, que é alcançada por meio da proveniência de dados. A reprodutibilidade realizada pela proveniência de dados é conseguida a partir do conhecimento da trajetória dos dados ao longo da execução do *workflow* (HONDO *et al.*, 2017).

A reutilização de *workflows* é eficaz em diversos níveis: podem ser reutilizados com parâmetros e dados diferentes, podem modificar o fluxo, os fluxos podem ser compartilhados com outros cientistas e grupos de pesquisa para realizarem trabalhos semelhantes, e podem utilizar parte do fluxo para geração de outras pesquisas (LUDÄSCHER *et al.*, 2009). Todos estes tipos de reprodutibilidade são realizados pela ProtCool.

Os fatores que possibilitam a reutilização de *workflows* são: descrições das funções e finalidade do fluxo, documentação dos serviços realizados (se possível com exemplos de dados de entrada e saída), proveniência adequada dos dados, restrições de propriedade e permissões de utilização, qualidade do fluxo, dependências de outros fluxos, dados, componentes (LUDÄSCHER *et al.*, 2009).

É importante na proveniência, não só o dado de entrada, mas o processo que o transformou no produto que foi armazenado (GUIMARÃES *et al.*, 2015). Além de informar sobre o uso de dados, a proveniência permite a tomada de decisões a partir de execuções anteriores do *workflow*, possibilitando que novas pesquisas sejam realizadas e com maior qualidade de resultados (LUDÄSCHER *et al.*, 2009).

Algumas informações de proveniência que devem ser armazenadas são: dados de entrada, parâmetros definidos pelo cientista, registro de atividades executadas, tempo de início e término de cada atividade, recursos empregados na execução das atividades, referências para dados de entrada e saída de cada atividade, dentre outras (BRAGHETTO e CORDEIRO, 2014). A proveniência não ocorre apenas em relação aos dados, mas podem ocorrer a gravação de versões distintas de *workflows*, apresentando as diversas modificações que podem ter ocorrido ao longo do processo de construção da pesquisa (CUEVAS-VICENTTÍN *et al.*, 2012).

2.3. BioNimbuZ

E-science é um termo que foi introduzido pela primeira vez por John Taylor em 2001 (APPEL *et al*., 2016), com o objetivo de denominar a infraestrutura necessária para possibilitar as colaborações e o trabalho multidisciplinar em áreas da ciência. Existem dimensões importantes para o desenvolvimento de *um e-science*, que são: a construção de uma infraestrutura computacional para uso distribuído ou para processamento de larga escala; a produção e o uso intensivo de dados; e a colaboração entre atores da ciência, pelo compartilhamento de esforços, dados e/ou recursos computacionais (APPEL *et al.*, 2016). Contudo, é importante ressaltar que o uso de *e-sciences* é para a pesquisa científica, e os recursos são compartilhados para a exploração dos dados (APPEL *et al.*, 2016). Os *e-sciences* são responsáveis por possibilitar apoio técnico para os pesquisadores, sem que eles precisem se preocupar com esses problemas, mas apenas com a sua pesquisa (BRAGHETTO e CORDEIRO, 2014). A ProtCool consegue realizar este apoio ao cientista, eliminando os problemas técnicos e permitindo que o pesquisador se preocupe apenas com os problemas da pesquisa.

As plataformas de *e-science* web existentes possuem uma arquitetura em três camadas. Na primeira camada está o *frontend* da aplicação que irá fornecer a interface com o usuário do sistema; a segunda camada possui a parte administrativa do sistema, com o gerenciamento de tarefas, configurações e gerenciamento de usuários; e a terceira camada é a que controla o banco de dados da aplicação. As plataformas possuem segurança sofisticada (Polyakov *et al.*, 2015).

Existem diversos sistemas gerenciadores de *workflows*, que podem ser chamados de *e-Sciences*, tais como Kepler (ALTINTAS *et al.*, 2004), e-Science Central (HIDEN *et al.*, 2011) e BioNimbuZ (MENDES, 2018).

O BioNimbuZ (SALDANHA, 2012) é uma plataforma de nuvem federada, proposta inicialmente por Saldanha (2012), na UnB. Foi desenvolvido com o objetivo de suprir a necessidade de se ter uma maior capacidade de processamento, armazenamento e outros recursos necessários a aplicações de Bioinformática (MOURA, 2017). Aceita a integração de nuvens públicas, privadas e comunitárias, sendo que cada provedor mantém as suas políticas e características internas (MOURA, 2017). Uma característica importante do BioNimbuZ é que o usuário pode utilizar os diversos serviços de todas as nuvens federadas de forma transparente, ou seja, ele não tem conhecimento de qual nuvem está fazendo uso (MOURA, 2017).

O BioNimbuZ foi projetado com o objetivo de facilitar a execução de *workflows* em plataformas de nuvens federadas. Foi desenvolvido visando simplicidade, velocidade e eficiência na entrada de um provedor na federação (SALDANHA *et al.*, 2012).

A submissão de *Jobs* (serviços dos *workflows*) é a principal atividade do BioNimbuZ. Cada *job* passa a ferramenta que será utilizada e quais os parâmetros para a sua execução. No BioNimbuZ é possível enviar um ou mais *Jobs* em uma única submissão, isso possibilita, por exemplo, que um mesmo serviço seja realizado com parâmetros diferentes ou que diversos serviços de um mesmo tipo sejam realizados diversas vezes. O BioNimbuZ não realiza a associação entre os vários *Jobs* executados simultaneamente, isso indica que cada *job* será considerado como se tivesse sido realizado de forma individual (SALDANHA, 2012).

A arquitetura da plataforma é composta de quatro camadas: aplicação, integração, núcleo e infraestrutura. A Camada de Aplicação é a responsável pela integração entre a aplicação do usuário e os serviços disponibilizados pela plataforma que ficam na Camada de Núcleo. Para fazer a interface entre as duas camadas, existe a Camada de Integração, que é responsável pelo gerenciamento dos dados que são transferidos entre as duas camadas. Na camada de Infraestrutura, estão os *plug-ins* de cada provedor. A camada de Infraestrutura é formada por diversos provedores de nuvem, cada um com a sua infraestrutura e um *Plug-in* que possibilita a comunicação e o gerenciamento do BioNimbuZ. Na camada de Núcleo se encontram os serviços controladores do BioNumbuZ. Na Camada de Aplicação estão as interfaces do usuário e os *workflows* dos usuários (SALDANHA, 2012; SALDANHA *et al.*, 2012; MOURA, 2017; ROSA, 2017; ROSA *et al.*, 2016).

A existência do *plug-in* de integração, presente na Camada de Infraestrutura é responsável pela flexibilidade e escalabilidade da plataforma, pois possibilita a inclusão de novos provedores, uma vez que é esse *plug-in* que realiza o mapeamento das demandas para o devido provedor (MOURA, 2017).

As aplicações de usuários (quarta camada) podem ser de vários tipos, como páginas web, linhas de comando, *interface* gráfica, sistemas gerenciadores de *workflow*, entre outros (SALDANHA, 2012). Esses serviços coletam as ações que os usuários desejam realizar e passam para que o núcleo do BioNimbuZ possa realizar a sua execução, escolhendo a nuvem em que será processada (SALDANHA, 2012).

A Camada de Aplicação possui uma *interface* web que permite ao usuário a entrada das ações que deseja executar na federação (MOURA, 2017; VERGARA, 2017). A ferramenta possibilita o gerenciamento de *workflows*, com a criação e o acompanhamento do *status* de execução, e, também, a gestão do armazenamento de arquivos, possibilitando a entrada de arquivos e todo o seu controle (MOURA, 2017; VERGARA, 2017).

O BioNimbuZ permite a integração da arquitetura AProvBio, que executa a proveniência de dados, utilizando o modelo PROV-DM (PROV *Data Model*) e um banco de dados de grafos (ALMEIDA *et al.*, 2017; HONDO *et al.*, 2017). No modelo apresentado pode-se executar a proveniência prospectiva (busca os passos a serem seguidos para a realização de uma pesquisa), retrospectiva (busca os dados que são gerados a partir da execução das atividades) e a definida pelo usuário (usuário determina as informações pertinentes para a sua pesquisa) (ALMEIDA *et al.*, 2017). Além disso, pode-se dizer que a proveniência é realizada em nível de captura executada, pois busca informações de atividades realizadas no *workflow*; ou em nível de atividade, pois busca os dados gerados pela execução do processo (ALMEIDA *et al.*, 2017).

No modelo proposto pelo BioNimbuZ, pode-se considerar os seguintes benefícios (ALMEIDA *et al.*, 2017, p. 2120): gerenciamento da implementação dos resultados dos experimentos, capacidade de armazenamento e reexecução de cada fase, maior confiabilidade em experimentos posteriores, usuários podem revisar as suas conclusões e fazer novas descobertas.

A proveniência de dados do BioNimbuZ utiliza a solução de banco de dados NoSQL, que traz como vantagens: as soluções são escaláveis, permitindo processamento distribuído; alta disponibilidade; consideram um esquema flexível (podem tratar dados estruturados ou não-estruturados) (HONDO et. al., 2017). A utilização de banco para o armazenamento da proveniência tem como benefícios: controle de acesso, segurança, processos de transação além da existência de outros recursos (GUIMARÃES *et al.*, 2015).

As limitações do BioNimbuZ são: devem ser gerada imagens de máquina virtual que serão instaladas nas máquinas virtuais, já que o BioNimbuZ não possui funcionalidade para o gerenciamento e a distribuição das aplicações; pode ser necessária a instalação manual de aplicações nas nuvens; o mecanismo de instalação do BioNimbuZ também não é bem definido, necessitando de diversos procedimentos para a implantação de seus componentes; todos os serviços do BioNimbuZ são desenvolvidos em uma aplicação monolítica, sendo necessária toda a sua execução, mesmo que seja necessário apenas parte dos serviços do sistema (ALVES, 2017). Para tratar esse último ponto, Alves (2017) propôs o desenvolvimento do BioNimbuZBox que utiliza a tecnologia de *containers* para resolver essas limitações.

Contudo, uma das vantagens do BioNimbuZ é que como a plataforma foi projetada para ser independente de ferramentas e banco de dados, com isso outras aplicações podem ser adaptadas e utilizar a infraestrutura proposta, bem como, novos *workflows* podem ser propostos (ROSA *et al.*, 2016). Os novos *workflows* podem ser facilmente implementados, já que a ferramenta possibilita a execução de qualquer *workflow* científico (ROSA *et al.*, 2016).

Mendes (2018) propôs uma remodelagem para o BioNimbuZ, gerando uma nova versão para a plataforma, o BioNimbuZ 2 (ou BioNimbuZ versão 2). Essa nova versão foi proposta de maneira hierárquica e distribuída, possuindo camadas bem divididas e *plugins* de nuvens como microsserviços, visando total independência, o que garante maior tolerância a falhas (MENDES, 2018). A nova arquitetura permitiu uma diminuição de custos, uma vez que possibilita uma maior flexibilidade na execução das tarefas e pela divisão do núcleo em diversas partes (GOMES, 2018). Outra grande vantagem é com relação à manutenção, já que é possível realizar modificações de forma independente em cada camada da plataforma (GOMES, 2018).

Essa nova proposta de arquitetura foi projetada para incorporar as principais características da nova versão. Na proposta, a arquitetura foi estruturada em quatro camadas, as quais são: Camada de Aplicação (Serviços Web para que os usuários possam interagir com a plataforma); Camada de Federação (possibilita a criação de federações de nuvens), Camada de Coordenação (coordena o fluxo e realiza o monitoramento de execução das tarefas) e Camada de Execução (processa a tarefa e realiza o monitoramento do seu estado e de recursos utilizados) (MENDES, 2018, GOMES, 2018).

Na Camada de Aplicação estão disponíveis os seguintes serviços: Serviço Web – disponibiliza páginas web para que os usuários possam interagir com a plataforma; Serviço de Segurança – responsável pelo armazenamento dos dados de autenticação e credenciais de serviço de rede, além de

possibilitar o acesso de múltiplos usuários, com isolamento de dados; Serviço de Armazenamento – responsável pelo armazenamento de arquivos de entrada e saída das tarefas executadas; Serviço de Predição – realiza a distribuição dos serviços de nuvem pelas credenciais cadastradas no sistema; Controlador de SLA – responsável pelo controle dos SLAs, garantindo que os acordos não serão violados (MENDES, 2018, GOMES, 2018).

A Camada de Federação é a responsável por gerenciar a federação de nuvens, o que é realizado pelo ambiente ZooKeeper (ZOOKEEPER, 2018) e pelos *plugins* que implementam as funcionalidades dos diversos provedores de nuvem. Os *plugins* possuem serviços e controladores, os quais são: Serviço de Descoberta – possui informações de tabelas de preços e tipos de recursos de computação e armazenamento da plataforma de nuvem; Serviço de Provisionamento – responsável por alocar/desalocar máquinas da nuvem; e Controlador de Credenciais – realiza o controle das credenciais cadastradas, passando as informações necessárias ao sistema para que a nuvem possa ser utilizada (MENDES, 2018, GOMES, 2018).

A Camada de Coordenação controla as tarefas e as atividades criadas pelos usuários por meio do TC (*Task Coordinator* – Coordenador de Tarefa). Os TCs solicitam a execução das tarefas para o TE (*Task Executor* – Executores de Tarefas) que fazem parte da Camada de Execução. A execução das tarefas é realizada pelo TE por meio de três passos: *download* de arquivos, execução da aplicação e *upload* dos arquivos gerados (MENDES, 2018).

A Camada de Coordenação possui quatro serviços para que as suas atividades possam ser realizadas, que são: Serviço de Monitoramento – realiza o monitoramento dos TEs, o consumo de recursos do TC e do TE; Serviço de Dependências – monitora as dependências existentes entre as tarefas, de forma a definir as próximas atividades e a verificar se as atividades foram encerradas; Serviço de Elasticidade – define quando aumentar ou diminuir os recursos alocados; Serviço de Escalonamento – divide as tarefas que devem ser executadas na nuvem (MENDES, 2018, GOMES, 2018).

Já a Camada de Execução possui três serviços, que são: Serviço de Aquisição de Recursos – responsável por obter os arquivos de entrada necessários ao processamento; Serviço de Execução – responsável pelo ciclo de vida das tarefas e pelo seu monitoramento; e, Serviço de Persistência – responsável pelo *upload* dos arquivos gerados (MENDES, 2018).

Todas as camadas possuem a tolerância a falhas. Isso é importante, uma vez que a comunicação envolve todas as camadas da Arquitetura, o que requer uma tolerância a falhas efetiva (MENDES, 2018). Uma vantagem de fazer a implementação de tolerância a falhas em cada uma das camadas é a possibilidade de se definir níveis de tolerância diferentes para cada uma das camadas (GOMES, 2018).

2.4.Dinâmica Molecular

A química computacional é um ramo interdisciplinar da química que utiliza softwares dedicados para resolver problemas químicos e bioquímicos (FERNANDES, 2011) Alguns campos são: Modelagem Molecular, Simulação Molecular, Quimiometria, Quimioinformática e Bioinformática (FERNANDES, 2011). Alguns métodos computacionais existentes são: *ab initio* e DFT; mecânica e dinâmica molecular; Monte Carlo; *docking*; entre outros.

A dinâmica molecular (MD, do inglês *Molecular Dynamic*) é uma simulação que permite que diversos estudos possam ser realizados em moléculas. Estes estudos fornecem informações a respeito do comportamento dinâmico microscópico das moléculas em estudo, levando em consideração o tempo em que estes comportamentos ocorrem (NAMBA *et al.*, 2008). Para a realização de simulações de dinâmicas moleculares é necessária uma preparação do sistema que está sendo estudado, sendo para isso utilizados protocolos neste processo (PURAWAT *et al.*, 2017). Devido à quantidade de etapas necessárias para a preparação destes sistemas, e da quantidade de opções disponíveis para que uma preparação seja efetuada, um tempo considerável do pesquisador acaba por ser exigido. Pensando-se em diminuir esse tempo e melhorar o processo de pesquisa dos laboratórios, foi idealizada a ferramenta

ProtCool que trabalha com a definição de protocolos. Essa ferramenta será melhor descrita neste trabalho.

A dinâmica molecular possibilita a visão temporal e espacial a nível molecular (ABRAHAM *et al.*, 2015). Tem seu início na década de 70, com a química de polímeros e a biologia estrutural, e era utilizada para realizar estudos de propriedades moleculares como flexibilidade, distorção e estabilização (PRONK *et al.*, 2013).

As simulações utilizam campos de força (veja definição mais adiante), que possibilitam por exemplo, o enovelamento de proteínas; predizer as interações existentes entre receptores e ligantes e previsão de propriedades funcionais de receptores, dentre outras funcionalidades (ABRAHAM *et al.*, 2015; PRONK *et al.*, 2013).

A dinâmica molecular pode ser utilizada como uma ponte entre a teoria e o laboratório. Ela possibilita que experimentos sejam testados e validados com dados da dinâmica, ao mesmo tempo em que dados da dinâmica podem ser validados por dados experimentais (ALLEN, 2004).

Segundo Martínez *et al.* (2007, p. 414) "a Dinâmica Molecular é uma técnica computacional em que se determinam os movimentos das partículas de qualquer sistema, do qual se conhecem o potencial de interação entre as partículas e as equações que regem seu movimento". A dinâmica molecular gera uma série de conformações instantâneas, que juntas mostram a trajetória da molécula ao longo do tempo (MACHADO *et al.*, 2007).

A dinâmica molecular pode tanto seguir a mecânica clássica de Newton, quanto a mecânica quântica (GEORG e CANUTO, 2007). A mecânica quântica possibilita estudos mais fundamentais das moléculas, no nível atômico (estruturas eletrônicas) (GEORG e CANUTO, 2007). Hoje em dia existem métodos que seguem puramente a dinâmica quântica, a dinâmica clássica e existem também métodos híbridos. Nesta tese, a atual versão do ProtCool opera apenas com protocolos e *workflows* de dinâmicas moleculares clássicas.

A dinâmica molecular clássica permite seguir a trajetória de cada átomo de uma molécula, considerando a mecânica molecular clássica de Newton (GU e BOURNE, 2009). A MD mostra as movimentações que as moléculas efetuam, determinando as diversas configurações de cada constituinte, e as propriedades da molécula (MARTÍNEZ *et al.*, 2007). A dinâmica molecular é responsável por rastrear as posições dos átomos nas moléculas ao longo do tempo, gerando uma grande quantidade de dados que deve ser analisada (MCGIBBON *et al*., 2015). É possível considerar em uma MD a movimentação de íons e moléculas individuais (Shen *et al.*, 2016).

Para que isso possa ser realizado a MD – a partir de condições iniciais, possuindo as coordenadas espaciais e velocidade de cada átomo, considerando as condições do sistema (temperatura, pressão e volume) e o campo de força desejado – determina a trajetória dos átomos, de acordo com a evolução temporal. Além de todos esses pontos, há que considerar a influência dos demais átomos e moléculas pertencentes ao sistema (SOARES, 2009).

Para que os softwares de dinâmica molecular possam ser executados e gerar os resultados descritos acima, eles executam os seguintes passos, a partir da estrutura de partida do sistema: caracterização dos parâmetros do campo de força escolhido para todos os átomos do sistema; definição de velocidades iniciais para todos os átomos do sistema, respeitando as condições termodinâmicas estabelecidas; depois de um determinado intervalo de tempo previsto, atualização dos dados de posicionamento dos átomos; realização de cálculos da energia potencial total, que é incidida sobre esse átomo por todos os demais átomos, de acordo com o campo de força definido; cálculo da energia total em cada átomo; cálculo da aceleração de cada átomo e então volta-se a atualizar o posicionamento dos átomos e segue-se os demais passos, até que o tempo de simulação ainda não tenha sido alcançado (ROCHA, 2017).

Assim, os usuários de MD observam a trajetória molecular, a partir de definições iniciais, e é por essa trajetória que realizam as suas inferências a respeito do sistema (ABRAHAM *et al.*, 2015). É possível, com a dinâmica molecular realizar estudos de sistemas biológicos e químicos para complementar os experimentos existentes, além de possibilitar a definição de técnicas experimentais

a serem utilizadas (SALOMON-FERRER *et al.*, 2013). Para realizar essas análises são necessários novos sistemas, ferramentas, análises, mais hardware e mais visualizações de todos os dados que foram gerados (MCGIBBON *et al.*, 2015).

Além disso, um fator importante da MD é que os efeitos de temperatura das moléculas não podem ser desconsiderados. É importante destacar que nas simulações clássicas de MD, as ligações químicas não são rompidas, não existem interações entre orbitais e não existem ressonâncias sendo representadas (MARTÍNEZ *et al.*, 2007).

Alguns exemplos de pacotes de simulação são: AMBER[1] (*Assisted Model Building with Energy Refinement*) (SALOMON-FERRER *et al.*, 2013); CHARMM[2] (*Chemistry at Harvard Macromolecular Mechanics*) (BROOKS *et al.*, 1983); NAMD[3] (*Not Another Molecular Dynamics*) (PHILLIPS *et al*., 2020); TINKER[4] (*Software Tools for Molecular Design*) (RACKERS *et al*., 2018); GROMOS[5] (*Groningen Molecular Simulations*) (GUNSTEREN *et al*., 1996); GROMACS[6] (ABRAHAM *et al.*, 2015); DL_POLY[7] (*Daresburg Laboratory Polyatomic Simulator*) (SMITH *et al*., 2002) (MARTÍNEZ *et al.*, 2007) .

2.4.1. Etapas da Dinâmica Molecular

Quando se realiza uma dinâmica molecular envolvendo alvos e ligantes (o objeto de estudo desta tese), existem diversas etapas necessárias para a preparação das moléculas, de forma que a dinâmica possa ser realizada com sucesso (PURAWAT *et al.*, 2017; MARTÍNEZ *et al.*, 2007). Dentre as atividades realizadas algumas se destacam, são elas: configuração inicial das moléculas; *docking* molecular (atracamento molecular) dos ligantes; modelagem molecular do alvo (proteínas); protonação; definição do campo de força; solvatação e ionização; realização das simulações da dinâmica molecular. Para cada um desses itens, uma seção irá descrever melhor os passos.

2.4.1.1.Configuração inicial das moléculas

Na configuração inicial do sistema, as moléculas são preparadas para que possa ocorrer a simulação. Nesse processo, deve-se, entre outras coisas, recuperar o arquivo com as coordenadas atômicas da molécula alvo, no caso desta tese, sempre uma proteína (ROCHA, 2017). O principal repositório de coordenadas atômicas biomoleculares é o PDB[8] (*Protein Data Bank)*. Esse banco de dados possui um conjunto de moléculas resolvidas por difração de raio-x e outras técnicas (ROCHA, 2017). No caso de proteínas, os arquivos PDB fornecidos possuem diversas informações, além das coordenadas atômicas, como a qual resíduo de aminoácido o átomo pertence, número do resíduo, tipo de estrutura secundária, fatores de temperatura, ocupâncias, entre outras informações (ROCHA, 2017).

Uma limpeza desse arquivo é realizada nas fases iniciais do *workflow*, de forma a possibilitar que o arquivo esteja livre de ocupâncias, que a numeração dos resíduos e átomos seja adequada e que ele esteja preparado para as demais fases do processo de preparação da dinâmica molecular.

2.4.1.2. *Docking* Molecular

[1] http://ambermd.org
[2] https://www.charmm.org
[3] https://www.ks.uiuc.edu/Research/namd/
[4] https://dasher.wustl.edu/tinker/
[5] http://www.gromos.net
[6] https://www.gromacs.org
[7] https://www.scd.stfc.ac.uk/Pages/DL_POLY.aspx
[8] https://www.rcsb.org

O conceito existente no *docking* é o de chave-fechadura, ou seja, duas moléculas encaixam, para se complementarem (DE PINHO VELOSO, 2019). O *docking* é composto de: representação das moléculas (arquivos do receptor e do ligante); definição do local onde o ligante será inserido (sítio ativo, definido por um *pocket*); algoritmo que realiza a busca pelas melhores poses de inserção no *pocket* definido; método de pontuação (*scores*) que avalia a melhor pose encontrada (DE PINHO VELOSO, 2019).

O *docking* molecular é necessário na dinâmica molecular de estudos de complexos proteína-ligante para se definir uma pose inicial. O *docking* molecular serve para a realização do atracamento de uma molécula ligante em um local do receptor, por exemplo em um sítio ativo. Eles são muito utilizados em estudos de otimização de poses quanto para seleção de potenciais inibidores entre uma quantidade de moléculas (*virtual screening*) (ROCHA, 2017). Para que consiga recuperar uma pose, essas ferramentas amostram uma quantidade de poses em um local determinado no receptor. Essas amostras são ranqueadas e comparadas de acordo com uma função de avaliação (*scoring function*) (ROCHA, 2017). Nesse estudo são consideradas características de afinidade eletrônica, as interações intermoleculares entre o ligante e complexo, além das especificidades do complexo formado, permitindo uma compreensão das propriedades físico-químicas e das interações energéticas do sistema (DE MEDEIROS FILHO *et al.*, 2020).

Para a realização de *docking*, além da molécula do receptor que foi recuperada na fase anterior, é necessária a busca da molécula do ligante. Essa busca é realizada em bases de dados específicas, tais como: ZINC[9] (STERLING; IRWIN, 2015), *DrugBank*10 (WISHART *et al.*, 2018), dentre outras bases.

Os itens importantes para que o *docking* ocorra são: as moléculas do receptor e ligante que serão estudadas; o *pocket* alvo que será utilizado (local que será considerado para a inserção do ligante no receptor); o algoritmo que fará a busca pela conformação do ligante dentro do *pocket* definido; e o *score*, que é um método de pontuação que permite avaliar a pose alcançada pelo ligante (DE PINHO VELOSO, 2019). Os dois processos principais são a docagem (encaixe) da molécula no seu alvo (*pocket* escolhido) e a pontuação que determinará o quão fortemente a interação entre o receptor e ligante foi estabelecida (BALLESTER, 2010).

Existem diversos softwares para realização de *docking*, tais como: *Autodock Vina*[11] (TROTT; OLSON, 2010), *Smina*[12] (KOES; BAUMGARTNER; CAMACHO, 2013), Molegro[13] (THOMSEN; CHRISTENSEN, 2006). Na presente versão do ProtCool, é possível trabalhar com *Autodock Vina* e *Smina*.

2.4.1.3.Modelagem Molecular

A modelagem molecular, no contexto desta tese, é usada para tentar resolver as lacunas existentes na proteína alvo, como cadeias laterais incompletas ou alças com resíduos faltantes. Uma forma de se resolver essas lacunas é por meio da modelagem por homologia, utilizando-se como modelo estruturas similares às que estão sendo modeladas. Para isso, os softwares normalmente necessitam das posições faltantes para buscar as sequências corretas que estariam presentes no local (ROCHA, 2017).

Para que o processo de modelagem seja realizado ele deve passar pelos seguintes passos: identificação do homólogo que será utilizado como modelo; alinhamento entre a sequência que se deseja modelar e o modelo que mostra como deveria ser a estrutura dessa sequência; geração do *backbone*; modelagem dos *loops* existentes; modelagem das cadeias laterais; otimização do modelo; e validação do modelo (KUNTAL *et al.*, 2010).

[9] http://zinc15.docking.org
[10] https://go.drugbank.com
[11] http://vina.scripps.edu
[12] https://sourceforge.net/projects/smina/
[13] http://molexus.io/molegro-virtual-docker/

A modelagem comparativa é uma técnica em que se prevê a estrutura 3D de uma sequência de proteína, realizando a comparação e o seu alinhamento com proteínas mais conhecidas que servem de modelos (WEBB; SALI, 2016). Um software bastante utilizado para esse fim é o Modeller[14] (WEBB; SALI, 2016). O Modeller é um programa para modelagem comparativa de estruturas de proteínas que utiliza o alinhamento de uma sequência a ser modelada com as estruturas do modelo (WEBB; SALI, 2016).

2.4.1.4.Hidrogênios e Protonação

As estruturas cristalográficas de proteínas que são recuperadas no PDB nem sempre possuem informações sobre os átomos de hidrogênio presentes na estrutura e, com isso, não podem ser utilizados para estudos de estados de protonação. Com isso, é necessário se utilizar métodos teóricos para a busca dos estados de protonação de estruturas cristalográficas (SOUZA; SANT'ANNA, 2012). Algumas ferramentas que fazem esse estudo são: H++[15] (ANANDAKRISHNAN; AGUILAR; ONUFRIEV, 2012), PropKa[16] (SONDERGAARD *et al*., 2011), Epik[17] (SHELLEY *et al*., 2007), entre outras. Na presente versão do ProtCool a verificação e acerto da protonação é realizada usando-se o H++.

O estado de protonação é importante pois ele interfere na distribuição de cargas e pode afetar a solubilidade e permeabilidade, além de afetar as conformações previstas para a molécula, os modos de interação e afinidade de ligantes, o que afeta as interações proteína-ligante (SHELLEY, 2007).

Os estados de protonação (ionização) dos diferentes grupos ácido e básico modificam a estrutura e a função das macromoléculas. Uma das condições que podem ser afetadas são as afinidades dos ligantes para com as proteínas, logo a protonação dos sítios ativos é importante nesse contexto (SITE H++, 2021). Fazer o estudo computacional da protonação acaba por ser mais barato, uma vez que a determinação da protonação de todos os grupos por meio de RMN acaba por ser inviável (SITE H++, 2021).

O H++ leva em consideração a posição do grupo de estudo dentro da molécula e as características do solvente para conseguir identificar o estado de protonação de cada elemento da estrutura. Adiciona hidrogênios que estejam ausentes na estrutura de acordo com o pH do sistema. O H++ utiliza a abordagem baseada na eletrostática clássica e na mecânica estatística básica, realizando assim, aproximações da realidade. Mas ainda não realiza ajustes heurísticos e nem aproximações empíricas para grandes conjuntos de dados.

2.4.1.5.Definição de Campos de Força

Existe um conjunto de parâmetros e equações que devem ser ajustados, de forma a garantir o comportamento adequado das moléculas durante as dinâmicas (MARTÍNEZ *et al.*, 2007). A esse conjunto de parâmetros e equações dá-se o nome campo de força. O campo de força descreve as equações envolvidas na energia potencial, ou seja, como é realizada a interação entre os diversos átomos e quais são os seus parâmetros (GARGANO, 2009).

A MD é responsável por resolver um conjunto de equações de Newton para um conjunto de átomos, utilizando para isso campos de força convencionais ou customizados (SOMOGYI *et al.*, 2016). Os campos de força são compostos por termos independentes que podem identificar: a vibração linear de um estiramento, a deformação angular, as torções de ângulos diedros e as interações entre pares de átomos não ligados covalentemente (interações de van der Waals e eletrostáticas) (GARGANO, 2009).

[14] https://salilab.org/modeller/
[15] http://biophysics.cs.vt.edu – versão 3.2
[16] https://github.com/jensengroup/propka
[17] https://www.schrodinger.com/products/epik

As contribuições de cada uma dessas forças são diferentes para cada átomo (ou conjunto de átomos) existente na molécula simulada, e elas são determinadas por um conjunto de parâmetros ajustados de acordo com dados experimentais (GARGANO, 2009). Logo, cada campo de força é preparado para a simulação de um tipo de molécula de estudo e cada parâmetro do campo de força é determinado a partir de dados experimentais e/ou por aproximações quânticas (GEORG e CANUTO, 2007).

A escolha do campo de força é uma etapa essencial para o estudo, pois ele que determinará as forças que agirão sobre cada átomo, indicando como será a evolução no tempo da dinâmica. Existem diversos campos de força disponíveis para proteínas, lipídios, sacarídeos e outras moléculas orgânicas, como: OPLS[18] (*Optimized Potentials for Liquid Simulation*) (LEAVER-FAY *et al*., 2013); *Amber* (SALOMON-FERRER *et al.*, 2013); *Gromos* (GUNSTEREN *et al*., 1996); CHARMM (BROOKS *et al.*, 1983). Na presente versão do ProtCool permite-se trabalhar com os campos de força do tipo AMBER.

Para que simulações de proteínas estejam mais próximas do seu contexto biológico, é necessário especificar o solvente, seja de forma implícita ou explícita (AGUILAR, 2009). Nesse sentido, deve-se determinar a caixa de simulação (cúbica, ortorrômbica, ou outra geometria), que conterá o sistema proteína-ligante-solvente. Além disso, é necessário determinar as dimensões da caixa; as condições periódicas de contorno; as posições iniciais das partículas na caixa de simulação e quantidade mínima de partículas a serem tratadas (ABRAHAM *et al.*, 2015).

No caso das proteínas, deve-se buscar a sua estrutura no PDB (*Protein Data Bank)* e deve-se adicionar as moléculas do solvente e *contra-íons* (MARTÍNEZ *et al.*, 2007). Contra-íons são introduzidos de forma a tornar o sistema proteína-ligante-solvente eletricamente neutro. O cálculo de distribuição de cargas deve ser realizado, pois essa distribuição terá papel fundamental na energia do sistema (AGUILAR, 2009).

2.4.1.6.Realização da dinâmica molecular

A realização da dinâmica molecular necessita que alguns passos sejam realizados, tais como a minimização de energia, aquecimento, equilíbrio, para só depois colocar o sistema em produção. Isso é necessário, uma vez que todo o processo de preparação da dinâmica realiza uma modificação na proteína cristalográfica (se for o caso), alterando o seu ambiente original (ROCHA, 2017). Alguns processos visam, então, diminuir possíveis artefatos ou situações irrealistas que poderiam afetar os resultados das análises, tais como um par de átomos colocados a uma distância proibitiva ou improvável. Com isso, algumas pequenas simulações são realizadas de forma a garantir que o sistema seja colocado em um estado inicial em que os efeitos de transição podem ser desprezados (ROCHA, 2017). Essas etapas iniciais são a minimização e o relaxamento.

Na etapa de minimização o sistema é colocado energeticamente em um mínimo local. Nessa etapa a energia potencial é o maior parâmetro de mudança do sistema (PURAWAT *et al.*, 2017). A minimização de energia é uma tarefa realizada antes que as simulações possam ser realizadas e ela busca calcular as coordenadas de forma a encontrar a energia potencial mínima do sistema (DA CRUZ *et al.*, 2009). Quando o sistema está minimizado, ele possui forças pequenas entre os átomos e, com isso, é uma boa estrutura de partida para que as simulações de dinâmica molecular possam ser realizadas (NAMBA *et al.*, 2008).

O equilíbrio inicial é responsável por equilibrar o solvente e os íons dispostos ao redor da molécula (CALIXTO *et al.*, 2015). Essa etapa garante que o sistema não entre em colapso ou num estado absurdo. O equilíbrio busca, a partir de passos graduais, o equilíbrio térmico do sistema sob determinadas condições. A etapa de equilibração é realizada a partir de uma configuração inicial e

[18] https://www.sciencedirect.com/topics/biochemistry-genetics-and-molecular-biology/opls

deve ser acompanhada de forma a garantir que o sistema saiu da configuração inicial (MARTÍNEZ *et al.*, 2007).

O período de equilíbrio vai depender do sistema que está em estudo, mas ele busca coincidir com o equilíbrio termodinâmico desse sistema (NAMBA *et al.*, 2008). O equilíbrio é realizado de forma a evitar variações bruscas da energia total do sistema e que ele pode ser dividido em duas fases: uma com restrições no movimento das moléculas e outra sem as restrições (NUNES, 2015).

O processo de relaxamento é realizado aplicando-se restrições harmônicas ao sistema (ROCHA, 2017). Essas restrições são aplicadas tanto em ligantes quanto na proteína. A ideia da restrição harmônica é fazer com que um átomo tenha a sua mobilidade reduzida, com isso, forças fazem com que o átomo permaneça próximo da sua posição inicial ou ideal. Um processo de relaxamento, com isso, é realizado por meio da diminuição gradual e planejada das restrições do sistema. Ao final desse processo planejado, são iniciadas as MDs produtivas (ROCHA, 2017).

A produção executa as simulações em um ambiente fisiológico simulado, em sistemas biológicos. É a etapa que possibilitará a avaliação do problema que se deseja estudar (PURAWAT *et al.*, 2017). O tempo de simulação necessário vai depender tanto dos processos dinâmicos que estão em estudo, como da convergência estatística das propriedades que estão sendo consideradas. Na produção alguns valores são registrados ao longo da dinâmica para posterior análise, tais como: coordenadas e velocidade atômicas; propriedades físico-químicas de interesse. A definição do tempo do passo em uma dinâmica é uma importante informação que o pesquisador deve se preocupar, além de se definir o tempo total de simulação desejado (MARTÍNEZ *et al.*, 2007).

Deve-se realizar várias rodadas de simulação para que se obtenha um conjunto de dados e trajetórias que de fato representem o sistema que está sendo simulado. É o pesquisador que determina o número de rodadas, o que pode ser realizado observando-se o valor médio das propriedades de interesse, verificando se o valor convergiu para os valores que se deseja estudar (MARTÍNEZ *et al.*, 2007).

3. PROTCOOL

3.1.Metodologia

Pelo que foi descrito na Seção 2.1.1, existe um modelo que descreve como deve ser realizada a modelagem, projeto, inserção, execução e análise de *workflows* (Figura 1). Essas etapas são:

i. Hipóteses, Experimentos e Objetivos – Estabelecer os *workflows* a serem desenvolvidos;
ii. Projeto do *workflow*
    a. Modelagem do *workflow*
        i. Definir Tarefas do *workflow*
        ii. Definir arquivos de entrada
        iii. Validar *workflow*
iii. Instanciação do *worklflow*
    a. Definir parâmetros do *workflow*
    b. Desenvolver *scripts*
    c. Preparar arquivos de entrada
    d. Preparar ambiente de execução do *workflow*
iv. Execução do *workflow*
v. Análise pós-execução
    a. Analisar resultados do *workflow*
    b. Definir próximos passos

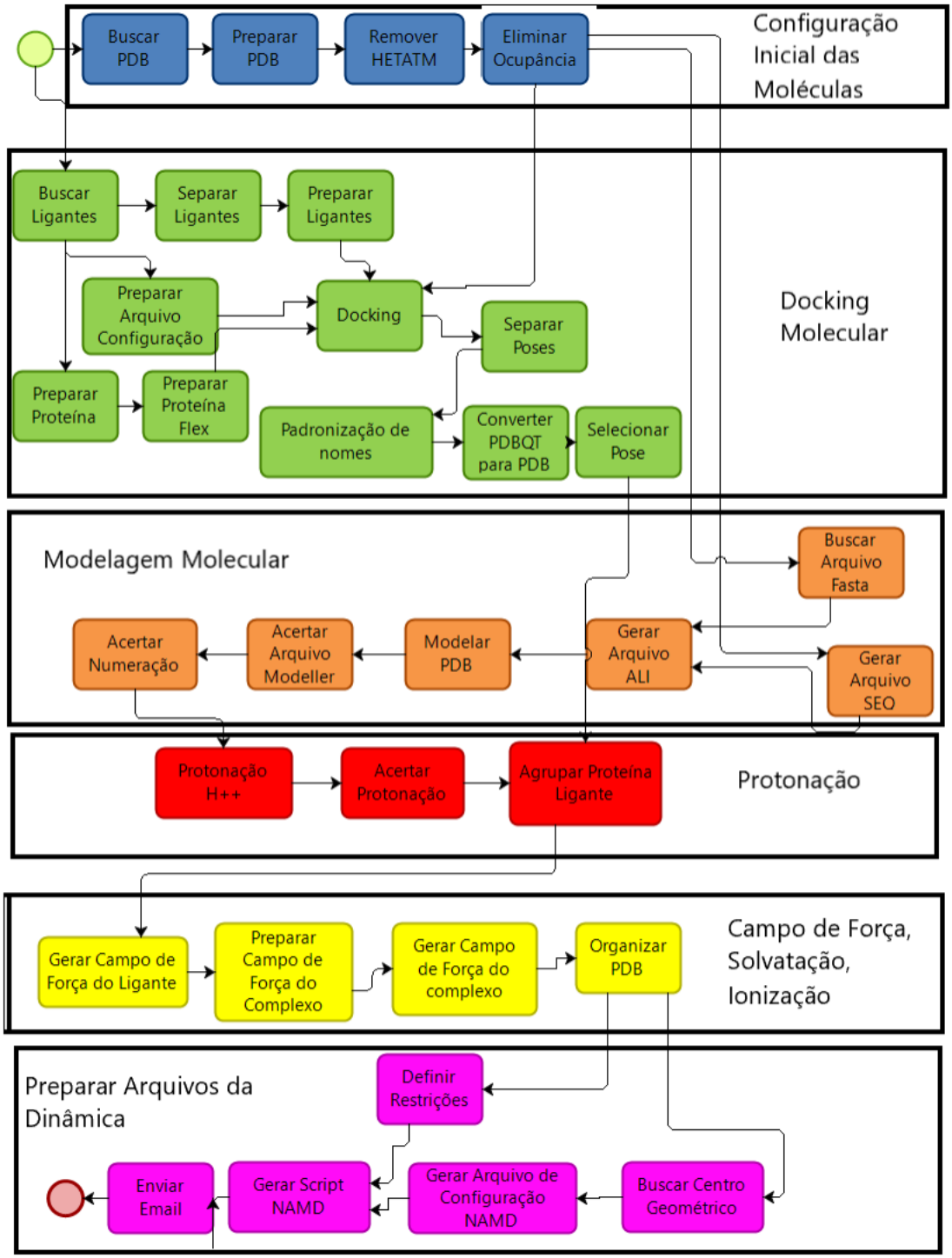
A primeira tarefa é a de "Hipóteses, Experimentos e Objetivos". Nesta etapa partiu-se da hipótese de que era possível o desenvolvimento de um *workflow* em que fosse possível a automatização de todo o processo de preparação de dinâmicas moleculares de complexos (proteína-ligante). Nesse caso, o problema é o de desenvolver o *workflow* de preparação de dinâmicas moleculares clássicas, de forma que o pesquisador possa ser liberado dessa tarefa rotineira e passe a dedicar mais tempo para realização de análises de resultados.

Partindo desta hipótese, era necessário o desenvolvimento do projeto do *workflow*. A etapa de Projeto necessitava da modelagem do *workflow*, com a definição de tarefas, arquivos de entrada e a validação do *workflow*. Para a definição das tarefas foi necessário o estudo de ferramentas utilizadas em uma preparação de dinâmica molecular, além de se estabelecer a ordem de execução de cada uma destas tarefas. A Seção 2.4.1 apresenta as etapas de uma dinâmica molecular e esta informação foi o ponto de partida para que um *workflow* fosse modelado, de forma a atender à hipótese levantada. Neste trabalho foram seguidas as etapas propostas, porém, cada uma das etapas teve que ser detalhada em diversas tarefas, de forma a que fossem adequadamente realizadas. Nessa fase de projeto do *workflow* foi necessário realizar o desenho do *workflow* apresentado na Figura 2.

O *workflow* da dinâmica foi dividido em seis partes: Configuração Inicial das Moléculas; *Docking* Molecular; Modelagem Molecular; Protonação; Campo de Força, Solvatação, Ionização; Preparar Arquivos da Dinâmica. Cada uma destas etapas é formada por uma série de *scripts* que permitirão a execução de todos os passos necessários para que a preparação seja realizada. Conforme pode ser observado, esta definição não só estabelece o *workflow* que será seguido, bem como a base para a geração de protocolos de pesquisa a serem executados pelo sistema.

Após a definição de cada uma das tarefas propostas no *workflow*, era necessário definir os arquivos de entrada e saída de cada um dos *scripts*. Isso foi realizado a partir do estudo de cada uma das tarefas que deveriam ser executadas. Houve o estudo tanto do objetivo de cada parte, bem como dos sistemas que estavam em cada uma das tarefas definidas. Cada um dos sistemas requer e gera arquivos específicos, o que auxiliou como ponto de partida para a definição inicial de cada etapa.

Figura 2 – Visão do protocolo completo do ProtCool_Dynamic. É possível visualizar o protocolo completo da ProtCool_Dynamic.



Fonte: Autores.

Com todos os dados, foi necessário realizar a validação dos resultados, a partir do modelo. Essa validação é importante uma vez que foram desenvolvidos *scripts* e a validação evita retrabalho. Toda a validação foi realizada observando-se se todas as etapas possuem as suas entradas e saídas definidas e se todas as tarefas do *workflow* estão devidamente encadeadas, sendo permitida a execução de cada uma das partes do sistema.

Com o *workflow* definido pode-se realizar a Instanciação. A instanciação é a etapa do desenvolvimento completo do sistema. Uma informação importante que foi descrita anteriormente é que, diferente de um desenvolvimento de software comum, em que se pensa em todo o sistema de uma única vez, na instanciação de *workflows*, deve-se tratar de cada um dos *scripts* de forma isolada. Isso garante que cada passo fará apenas uma única atividade e uma tarefa possa ser utilizada em outros *workflows*, garantindo uma das formas de reprodutibilidade. O desenvolvimento de *scripts*, desta forma, foi realizado um a um, garantindo que todos os arquivos de entrada e saída do sistema fossem adequadamente encadeados no processo.

Aqui é importante uma série de definições a respeito do *workflow*, tais como, o que é trabalho da ferramenta de gestão do *workflow* e o que deve ser gerado pelas tarefas modeladas; quais tarefas do *workflow* serão automáticas, quais precisam de auxílio do usuário e quais serão completamente realizadas pelo usuário; quais tarefas requerem o uso de ferramentas externas ao processo e quais são essas ferramentas. A preparação dos arquivos de entrada e de saída de cada uma das tarefas é responsabilidade dessa etapa do processo. Além disso, existe a definição de quais das tarefas deveriam ser implementadas em paralelo, de forma a garantir agilidade ao processo e, ao mesmo tempo, utilização de todos os recursos disponíveis.

Para a efetivação dessa etapa, foram assim desenvolvidos:

i. *Scripts* de cada tarefa do *workflow*;
ii. Definições de entradas e saídas de cada processo, bem como a definição do fluxo a ser seguido;
iii. Criação do arquivo de configuração para que as opções do usuário pudessem ser disponibilizadas;
iv. Desenvolvimento de interface gráfica para preenchimento dos arquivos de configuração, de forma a facilitar o uso do sistema;
v. Desenvolvimento dos *scripts* principais de cada *workflow*, que controlam o fluxo e acesso a cada um dos *scripts*.

Outro cuidado durante o desenvolvimento do sistema é a criação de *scripts* com processamento paralelo, o que garante rapidez ao processo de execução do sistema. Também foi estabelecida toda a forma de proveniência de dados do sistema, garantindo que todos os dados gerados ao longo do processo sejam adequadamente armazenados.

Após o desenvolvimento do software, foi necessário se pensar na estrutura dos arquivos de entrada e saída do sistema, garantindo todo o processo de execução do sistema. Com a instanciação, e preparação dos arquivos, é importante que todos os *scripts* e dados de entrada estejam disponíveis para a execução do sistema.

Para que os processos sejam executados, foi necessário o desenvolvimento de *scripts* de gestão do *workflow*. Este *script* é responsável por fazer toda a gestão das execuções e realizar todo o processamento do sistema. O controlador é o responsável pela gerência de todo o processo, desde o envio dos arquivos e a execução dos *scripts* desenvolvidos. Ele controla toda a execução, podendo o pesquisador verificar o *status* do processo a qualquer momento e, ao final, disponibiliza os arquivos para que o pesquisador possa recuperar e fazer as devidas análises. Para facilitar a gestão de arquivos do sistema, todos os arquivos são disponibilizados em pasta definida pelo pesquisador, não necessitando que o pesquisador ao final do processo realize o *download* dos arquivos gerados. Além disso, o recurso de paralelismo é utilizado de forma a garantir maior rapidez na execução dos *workflows*.

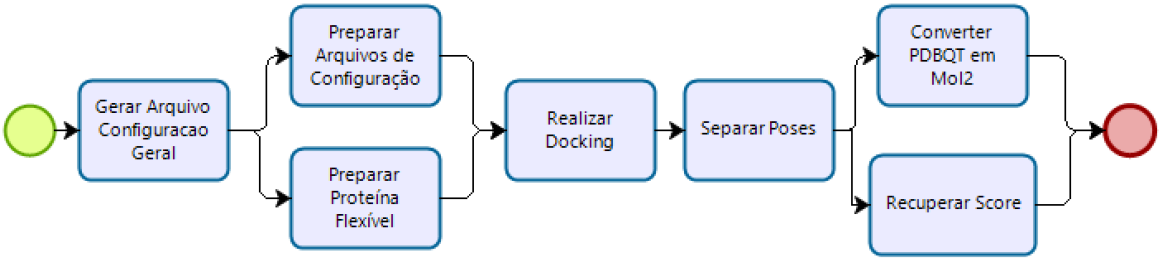
Para efetuar a execução do *workflow* é importante garantir que os arquivos iniciais estejam presentes no sistema. No caso do ProtCool, tem que ser garantido que o arquivo de configuração esteja com os dados da dinâmica que se deseja executar, com todos os seus elementos. Foram gerados dois *scripts* iniciais que validam o arquivo de configuração gerado. Estes *scripts* são responsáveis por

checar se todas as *tags* obrigatórias estão presentes no arquivo, além disso, eles também verificam se as *tags* estão preenchidas e se possuem valores válidos. Nem todas as *tags* possuem um padrão de digitação, porém, aquelas que possuem padrão serão validadas. Isso possibilitará a redução considerável de erros de usuários do sistema, pois garante que a entrada de dados foi realizada de maneira correta. Também foi desenvolvida uma interface de preenchimento do arquivo de configuração, que auxilia neste processo.

A execução do sistema foi realizada com o desenvolvimento de dois estudos de caso, mostrando como se prepara todos os arquivos e os resultados que foram alcançados.

Após a execução, foi realizada a análise pós-execução. Nessa análise pode-se perceber que o *workflow* atendia ao proposto. Além disso, percebeu-se que seria possível desenvolver outros *workflows* a partir do *workflow* gerado inicialmente. Com isso, foi desenvolvido o *workflow* de *docking* para múltiplos ligantes. O processo foi derivado do *workflow* original, senso passado por todo o processo de geração de modelagem *workflow*, para garantir que o sistema fosse definido com qualidade. A Figura 3 apresenta o modelo do *workflow* de *docking* gerado.

Figura 3 – ProtCool_Docking – *virtual screenig*. O processo inteiro é formado por 7 tarefas. As tarefas são desenvolvidas em paralelo, bem como, existem algumas atividades que internamente, também, foram desenvolvidas em paralelo. Maiores detalhes podem ser visualizados no texto.



Fonte: Autores.

Com isso, durante o processo de desenvolvimento do *workflow*, foram gerados dois modelos que acabaram por gerar uma ferramenta, a ProtCool, que hoje é dividida em dois módulos: ProtCool_Dynamic (preparação de dinâmica molecular de complexos) e ProtCool_Docking (realização de *docking* de múltiplos ligantes).

Para cada um dos *workflows*, foi gerado um arquivo de controle e gerenciamento do *workflow*:

i. ProtCool_Dynamic – `ProtCool_Dinamica.py`;
ii. ProtCool_Docking – `ProtCool_Docking.py`.

Esses *scripts* especiais possibilitam a execução de atividades em paralelo, bem como a execução de tarefas que já foram programadas para serem realizadas em paralelo. Esses *scripts* foram preparados para que, no caso de necessidade de geração de apenas parte do *workflow*, basta acertar o arquivo de configuração e disponibilizar os arquivos necessários para cada uma das tarefas na pasta determinada. Isso facilitará a execução das atividades do *workflow*.

Existem alguns campos no arquivo de configuração que controlam a execução do `ProtCool_Dinamica.py`, indicando se todas as atividades deverão ser executadas, são eles:

i. [*Prepare*] – indica que a primeira etapa do *workflow*, configuração inicial das moléculas será realizada;
ii. [*Docking*] – indica que o *docking* será realizado;
iii. [*MolecularModel*] – indica que a modelagem molecular será realizada;
iv. [*Protonation*] – indica que a protonação será realizada;

v. [*ExecutionForceField*] – indica que a etapa de definição de campo de força, solvatação e ionização serão realizadas;
vi. [*PrepareNamd*] – Indica que a etapa de preparação final dos arquivos de configuração será realizada.

Todos os campos acima conterão como opção apenas "Yes" ou "No". Essa possibilidade não funcionará se não existirem os arquivos conforme descrito no processo na pasta (mesmos nomes e todos os arquivos presentes). Ou seja, isso permitirá apenas que o pesquisador possa optar por exemplo, em rodar até a parte de *docking*, verificar a escolha do *workflow* e prosseguir com o processo desse ponto em diante, sem a necessidade de executar todo o *workflow* novamente.

Isso também permitirá que o pesquisador possa optar por outras ferramentas que ainda não estão contempladas no *workflow* e realizem a inserção dos arquivos necessários para que o *workflow* continue do passo que parou. Mas o pesquisador deve ter o cuidado de no momento de iniciar a próxima etapa, os arquivos estejam disponibilizados da forma correta e com os mesmos nomes com os quais são tratados no *workflow* para que não ocorram falhas.

Outra opção de uso desses elementos, é quando por algum motivo ocorrer uma falha durante a execução do sistema, pode ser por exemplo uma queda de energia, ou outro motivo qualquer, e o pesquisador quiser continuar o processo do ponto em que parou.

Outro cuidado que deve ser tomado é que existem ao longo do processo três arquivos `confdinamica.txt` (no caso do ProtCool_Dynamic), conforme as alterações necessárias nele para o andamento do processo. Assim, para voltar a executar o *workflow*, deve ser alterado o arquivo de configuração conforme a necessidade do próximo *script* a ser executado, além do arquivo de configuração inicial do sistema.

Para o ProtCool_Docking, por sua característica de ser executado normalmente em servidores devido a carga de processamento e arquivos, foi gerado o *script* de controle do processo `ProtCool_Docking.py` que é responsável por todo o processo de gestão do processamento. Além disso, todos os *scripts*, incluindo o `ProtCool_Docking.py`, o arquivo de configuração inicial, o arquivo do receptor, todos os arquivos de ligantes que serão atracados ao receptor devem estar disponíveis em uma mesma pasta. Outro arquivo importante é o `smina.static`, que deve ser disponibilizado nessa mesma pasta de execução.

Uma observação importante é que o pesquisador só precisa ter em seu computador os *scripts* referentes ao *workflow* que deseja utilizar, uma vez que os *scripts* são independentes entre si.

Para o desenvolvimento e testes da ProtCool foi utilizado um computador *MACBook Air*, *Intel Core* i7 *QuadCore*, 16GB de RAM com todos os sistemas funcionando na máquina virtual *Parallels* (com dois processadores e 4GB de RAM). A versão do sistema operacional instalada nessa máquina foi o Ubuntu 18.04.5 LTS – 64 bits e os programas instalados foram: BioNimbuZ 2[19] (MENDES, 2018); Namd 2.11 for Linux-x86_64-multicore-CUDA (PHILLIPS, 2005); VMD[20] (HUMPHREY *et al.*, 1996); PyMol[21] (SCHRÖDINGER, 2015); WORDOM[22] (wordom_0.22-rc3.x86-64) (SEEBER, 2011); AmberTools[23]; Open-Babel[24]; RDKit[25]; Pacote *Requests* do python[26]; *Vina*; *Smina*;

[19] https://github.com/bionimbuz/Bionimbuz
[20] https://www.ks.uiuc.edu/Research/vmd/)
[21] https://pymol.org/2/
[22] http://wordom.sourceforge.net
[23] https://ambermd.org
[24] http://openbabel.org/wiki/Main_Page
[25] https://www.rdkit.org
[26] https://requests.readthedocs.io/en/master/

MGLTools[27] ; Modeller; Sellenium[28]; Geckodriver[29]; Pacote pyPDB[30]; Cuda; R; Python (2.7 e 3); GGNU (gcc, g++, gfortran); entre outros.

Devido à quantidade de sistemas necessários para a utilização da ProtCool e, devido à estrutura de pastas e arquivos, foi desenvolvido ao final do projeto um instalador que é responsável por fazer todas as instalações para o usuário, facilitando a utilização do sistema. Também foram desenvolvidos vídeos e tutoriais para facilitar tanto a instalação quanto a utilização do sistema. Isso pode ser recuperado no drive, a partir do link: https://drive.google.com/drive/folders/1X5Qk8IAJPT552o6fRrkhIE0PwBSf6u2S?usp=sharing.

Para fins de comparação com sistemas similares, a Tabela 1 contém uma comparação entre a ProtCool e duas outras ferramentas semelhantes encontradas a QwikMD (RIBEIRO *et al.*, 2016) e a *Kepler workflow* (PURAWAT *et al.*, 2017)

Tabela 1 – Comparação de ferramentas. Tabela comparativa das principais características das ferramentas.

Caratcterísticas	ProtCool	*QwikMD*	*Kepler Workflow*
Realização de preparação de Dinâmica Molecular	Complexos	Proteína	Não
Alteração de Estrutural	Eliminar Ocupâncias, Modelagem Molecular e Protonação	Pontos de Mutação e Protonação	Não
Realização de *Docking*	Vina e Smina	Não	Não
Preparação do Sistema de Simulação	Campo de Força, Solvatação, Ionização, definição de Restrições, Seleção de Protocolos, Preparação de arquivos de Simulação	Campo de Força, Ionização, Seleção de Protocolos, Preparação de arquivos de Simulação	Não
Executar Simulação	Não é automático	Automático	Realiza – minimização, Equilibração, Produção
Análises	Não	Análise de trajetórias	Não

Fonte: Autores.

Pelas informações da Tabela 1, pode-se perceber que a ProtCool se assemelha mais com a *QwikMD* que com a *Kepler Workflow*. A *Kepler Workflow* na verdade executa e acompanha a execução de dinâmica molecular, ao invés de realizar a preparação de dinâmica molecular. Logo, a ProtCool não realiza o mesmo tipo de atividade que esse *workflow*, porém, é uma característica interessante de posteriormente passar a ser considerada na ProtCool.

[27] http://mgltools.scripps.edu
[28] https://www.selenium.dev/documentation/en/webdriver/
[29] https://github.com/mozilla/geckodriver
[30] https://github.com/williamgilpin/pypdb

Quanto ao *QwikMD*, as principais diferenças se encontram nas ferramentas utilizadas, que se diferenciam um pouco. Além disso, a ProtCool trata o complexo (proteína e ligante), enquanto a *QwikMD* realiza a preparação de dinâmicas de proteínas apenas, sem que elas estejam complexadas, ou seja, sem estarem atracadas a ligantes. Ou seja, a *QwikMD* não realiza a etapa de *docking*. Outra característica é que a *QwikMD* não realiza a etapa de modelagem molecular. Quanto à etapa de execução da dinâmica, a *QwikMD* realiza a execução e acompanhamento das simulações e além disso, realiza algumas análises das trajetórias geradas pela dinâmica. A ProtCool não realiza estas duas etapas, uma vez que ela está preparada para realizar a preparação de simulação de dinâmicas moleculares e não das demais fases do processo.

3.2.ProtCool_Dynamic

Aqui está descrita a ferramenta gerada ProtCool_Dynamic, mostrando detalhes do seu desenvolvimento. As tarefas têm seu encadeamento e visualização definido no modelo da Figura 2.

A primeira etapa da realização do *workflow* é a de "Configuração inicial das Moléculas". Essa etapa, como descrito na Seção 2.4.1.1 é responsável por fazer todos os ajustes iniciais dos arquivos a serem trabalhados, de acordo com as configurações iniciais do pesquisador. Para a realização dessa etapa são executadas as tarefas: Buscar PDB; Preparar PDB; Remover HETATM e Eliminar Ocupância.

O primeiro passo é buscar o PDB, para isso, o *script* "`1.buscar_pdb.py`" permite a inserção dos arquivos PDB. Existem duas formas de recuperar o arquivo PDB da molécula, que podem ser pesquisados diretamente no *site* do PDB[31], bem como com os arquivos fornecidos pelo usuário.

Para conseguir realizar a busca direta no *site* RCSB está sendo utilizada a biblioteca pypdb. Esta biblioteca desenvolvida em *Python* realiza uma série de funcionalidades que permitem ações em arquivos do tipo PDB, bem como a recuperação de dados no *site* RCSB. Nesse caso, é necessário o PDBID do arquivo cristalográfico que deseja recuperar e que o computador que está sendo utilizado tenha acesso à Internet. No caso de o pesquisador querer trabalhar com arquivo do tipo PDB já recuperado do RCSB ou de outra fonte, basta manter o arquivo na pasta do sistema de forma que o *script* busca o arquivo.

O passo preparar PDB, realizado pelo *script* "`2.prepara.py`" é responsável por fazer as configurações e limpezas iniciais do arquivo PDB. Todo arquivo PDB, além das informações referentes à estrutura 3D da molécula, disponibiliza algumas informações a respeito da estrutura que podem ser importantes durante a preparação de moléculas para a dinâmica. Uma dessas informações são as pontes dissulfeto existentes na estrutura. As pontes dissulfeto são estruturas importantes das moléculas, pois uma vez presentes na molécula precisam de tratamento especial quando da definição do campo de força utilizado, pois essas estruturas seguem parâmetros diferentes e, por isso, devem ser devidamente destacadas quando da sua definição.

Com isso, um passo dessa tarefa recupera as pontes dissulfeto e grava em um arquivo separado para que possa ser utilizada posteriormente. Essa recuperação pode ser realizada de duas formas. A primeira é recuperando as pontes determinadas no arquivo PDB com os valores SSBOND e a outra forma é por meio de informações disponibilizadas pelo pesquisador no arquivo de configuração. Assim, caso o pesquisador não concorde com as informações disponíveis no próprio arquivo PDB, ele poderá fornecer os dados a respeito das pontes dissulfeto. No caso do arquivo PDB não possuir pontes dissulfeto, um arquivo pontes é gravado com a informação "VAZIO".

Outro ponto importante desse *script* é que existem diversos métodos experimentais que podem ser utilizados para gerar os arquivos PDBs que são armazenados no *site* RCSB, tais como: difração de raios-X; difração de nêutrons; método híbrido de raios-X/nêutrons, RMN entre outras. De acordo com o tipo de método utilizado, algumas informações são disponibilizadas de forma diferente nos arquivos

[31] https://www.rcsb.org - *Protein Data Bank*

PDB, gerando PDBs levemente diferentes. As informações principais do arquivo tais como, ATOM, HETATM, TER, CONECT entre outras não são alteradas, porém, existem algumas diferenças estruturais no arquivo que devem ser levadas em consideração.

Um exemplo disso, é a existência de PDBs que possuem mais de uma estrutura representada pelo MODEL, como é comum em estruturas resolvidas por RMN. Cada MODEL possui as informações referentes à estrutura da molécula em 3D com pequenas diferenças, representando diferentes soluções possíveis para os dados gerados por RMN. Nesse caso o utilizador do *workflow* deverá informar qual dos modelos é o que lhe interessa para que a preparação o considere.

A Figura 4 possui uma imagem de trecho de um arquivo PDB que possui MODEL. O PDB apresentado é o 6UJV[32] (*Model of the HIV-1 gp41 membrane-proximal external region, transmembrane domain and cytoplasmic tail*) – Método usado: RMN. Nesse caso, o arquivo possui ao todo 15 MODEL e cada um deles com 3 cadeias (A, B e C). O pesquisador nesse caso deverá escolher com qual MODEL deseja trabalhar e qual ou quais cadeias serão consideradas.

Figura 4 – PDB 6UJV – Exemplo PDB com MODEL. Método usado: NMR. Nesse caso o arquivo possui ao todo 15 MODEL e cada um deles com 3 cadeias (A, B e C). Trecho que mostra um pedaço do PDB onde é possível observar o MODEL 1.

```
SCALE2      0.000000  1.000000  0.000000        0.00000
SCALE3      0.000000  0.000000  1.000000        0.00000
MODEL        1
ATOM      1  N   LEU A 660     617.133-917.575  37.122  1.00  0.00           N
ATOM      2  CA  LEU A 660     617.130-917.560  35.629  1.00  0.00           C
ATOM      3  C   LEU A 660     615.681-917.495  35.126  1.00  0.00           C
ATOM      4  O   LEU A 660     614.842-918.329  35.473  1.00  0.00           O
ATOM      5  CB  LEU A 660     617.801-918.853  35.097  1.00  0.00           C
ATOM      6  CG  LEU A 660     619.148-918.549  34.417  1.00  0.00           C
ATOM      7  CD1 LEU A 660     619.907-919.861  34.189  1.00  0.00           C
ATOM      8  CD2 LEU A 660     618.901-917.867  33.066  1.00  0.00           C
ATOM      9  H1  LEU A 660     617.892-918.198  37.464  1.00  0.00           H
ATOM     10  H2  LEU A 660     616.215-917.924  37.467  1.00  0.00           H
```

Fonte: PDB do *site* RCSB.

A mesma coisa acontece no caso das cadeias, especialmente quando a técnica de resolução é por cristalografia. Nesse processo, a *asymmetric unit* (menor porção do cristal capaz de compor uma célula unitária por operações de simetria) pode conter parte de uma cadeia ou várias cadeias, gerando cópias poliméricas que não necessariamente são aquelas existentes e funcionais num ambiente celular ou fisiológico (*biological assembly*). Assim, por exemplo, um arquivo PDB pode conter duas cadeias idênticas em sequência (indicando um homodímero) como sua *asymmetric unit*, mas no ambiente celular, seu estado funcional, o seu *biological assembly,* poderia envolver uma cadeia só (monômero). Sendo assim, o pesquisador deverá informar qual a cadeia ou cadeias com as quais deseja trabalhar.

A seleção de MODEL e CHAIN será realizada de forma que o arquivo PDB a ser preparado leve em consideração apenas o que o pesquisador desejar. O tratamento de PDBs que possuem MODEL permitirá que uma maior quantidade de PDBs registrados no *site* RCSB sejam possíveis de serem trabalhados no ProtCool_Dynamic.

Após realizar essas seleções, o *script* realiza uma limpeza dos arquivos PDB, eliminando informações do cabeçalho, além de informações de CONECT. O cabeçalho contém metadados envolvendo a biomolécula, que não são importantes durante o processo de preparação da dinâmica molecular. Já as *tags* CONECT, contêm informações sobre as conexões entre heteroátomos. Sendo

[32] https://www.rcsb.org/structure/6UJV

assim, ao final desse *script* tem-se um arquivo PDB mais limpo e as informações importantes guardadas para posterior uso.

O passo "remover HETATM" é responsável por finalizar a remoção de itens no PDB. A ideia aqui é que ao final desse *script* fiquem apenas os itens referentes ao ATOM, ou seja, que apenas fiquem os itens que estejam relacionados à estrutura da proteína em si com a qual se está trabalhando. Nesse cenário, aqueles possíveis ligantes cristalografados junto à molécula, bem como outros possíveis heteroátomos seriam retirados. Aqui também pode-se retirar ou não as águas cristalográficas e os hidrogênios presentes na molécula, de acordo com o desejo do pesquisador, bastando apenas deixar no arquivo de configuração essa necessidade assinalada.

Mesmo existindo os *workflows* pré-cadastrados e que facilitarão o trabalho do pesquisador, é ele, que durante a definição da pesquisa especificará quais atividades serão de fato utilizadas para cada problema a ser tratado. Fato é que os *workflows* trazem rapidez, facilidade, mas principalmente flexibilidade para o pesquisador, de forma a que ele determina o que deseja com relação à molécula. Assim sendo, o arquivo de configuração dará a diretriz do que deve ou não ser executado no sistema.

Figura 5 – PDB 4EY6 – Trecho com Ocupância. Na imagem é destacada, em vermelho, a marcação de existência de ocupância, bem como, o valor dessa ocupância. A ocupância deve ser acertada no arquivo, uma vez que ela representa mais de uma conformação para o mesmo elemento.

```
ATOM   2333  OG1 THR A 311       1.363 -29.411  14.384  1.00 40.59           O
ATOM   2334  CG2 THR A 311      -0.727 -28.199  14.436  1.00 31.11           C
ATOM   2335  N   PRO A 312       1.522 -30.133  17.986  1.00 38.57           N
ATOM   2336  CA  PRO A 312       2.566 -30.948  18.605  1.00 36.07           C
ATOM   2337  C   PRO A 312       3.418 -31.667  17.564  1.00 37.14           C
ATOM   2338  O   PRO A 312       4.627 -31.758  17.740  1.00 43.79           O
ATOM   2339  CB  PRO A 312       1.770 -31.945  19.455  1.00 35.94           C
ATOM   2340  CG  PRO A 312       0.523 -31.216  19.795  1.00 31.64           C
ATOM   2341  CD  PRO A 312       0.196 -30.416  18.561  1.00 42.91           C
ATOM   2342  N  AGLU A 313       2.785 -32.165  16.506  0.49 41.48           N
ATOM   2343  N  BGLU A 313       2.802 -32.158  16.493  0.51 41.49           N
ATOM   2344  CA AGLU A 313       3.502 -32.792  15.397  0.49 46.09           C
ATOM   2345  CA BGLU A 313       3.564 -32.807  15.427  0.51 46.09           C
ATOM   2346  C  AGLU A 313       4.610 -31.884  14.867  0.49 43.78           C
ATOM   2347  C  BGLU A 313       4.640 -31.875  14.877  0.51 43.78           C
ATOM   2348  O  AGLU A 313       5.751 -32.313  14.701  0.49 44.92           O
ATOM   2349  O  BGLU A 313       5.788 -32.279  14.703  0.51 44.90           O
ATOM   2350  CB AGLU A 313       2.539 -33.158  14.260  0.49 44.33           C
ATOM   2351  CB BGLU A 313       2.658 -33.281  14.288  0.51 44.42           C
ATOM   2352  CG AGLU A 313       1.705 -34.413  14.502  0.49 44.80           C
ATOM   2353  CG BGLU A 313       3.376 -34.198  13.307  0.51 52.13           C
ATOM   2354  CD AGLU A 313       0.334 -34.121  15.100  0.49 47.39           C
ATOM   2355  CD BGLU A 313       2.769 -34.184  11.918  0.51 56.50           C
ATOM   2356  OE1AGLU A 313      -0.618 -34.861  14.763  0.49 47.36           O
ATOM   2357  OE1BGLU A 313       1.571 -33.848  11.787  0.51 61.39           O
ATOM   2358  OE2AGLU A 313       0.207 -33.168  15.904  0.49 32.18           O
ATOM   2359  OE2BGLU A 313       3.500 -34.505  10.956  0.51 45.33           O
ATOM   2360  N   ALA A 314       4.269 -30.625  14.615  1.00 38.63           N
ATOM   2361  CA  ALA A 314       5.234 -29.656  14.107  1.00 39.14           C
ATOM   2362  C   ALA A 314       6.346 -29.419  15.118  1.00 43.90           C
ATOM   2363  O   ALA A 314       7.526 -29.439  14.765  1.00 45.92           O
ATOM   2364  CB  ALA A 314       4.551 -28.347  13.752  1.00 35.28           C
```

Fonte: Autores.

O passo eliminar ocupância é responsável por resolver a questão de regiões com mais de uma conformação possível para o mesmo grupo de átomos. A Figura 5 apresenta um trecho do arquivo

PDB 4EY6[33] com Ocupância. Na Figura 5 existem duas marcações em vermelho no resíduo 313. A primeira marcação mostra um conjunto de letras, no caso A e B antes do nome do resíduo. Essa letra indica que existe o registro de ocupância.

A segunda marcação da Figura 5 apresenta o valor da porcentagem. No caso dos átomos 2342 e 2343, observa-se que eles representam o átomo de nitrogênio da amida de uma glutamina. A coluna assinalada mostra que 51% (quase meio a meio) dos átomos de nitrogênio desse resíduo foram encontrados na posição do átomo com id 2343. No caso do *script*, sempre são escolhidos no *script* aqueles com valor maior de ocupância ou, no caso de valores iguais, fica o primeiro átomo.

A Figura 6 apresenta o trecho com o resíduo 313, já sem a ocupância. Pode-se observar que todos os átomos que permaneceram possuem o valor de ocupância 0.51, indicando que ficaram os que possuíam maior ocupância. Observa-se, também, que os átomos ficaram com valores fora de sequência, de 2345 segue para o 2347. Esses valores da numeração dos átomos serão corrigidos posteriormente, quando outros acertos nos números tanto de átomos quanto de resíduos serão realizados.

Figura 6 – PDB 4EY6 – Trecho com ocupância corrigida. Mesmo trecho da Figura 5, agora com a ocupância corrigida, permanecendo os átomos com maior ocupância.

```
ATOM   2341  CD  PRO A 312       0.196 -30.416  18.561  1.00 42.91           C
ATOM   2343  N   GLU A 313       2.802 -32.158  16.493  0.51 41.49           N
ATOM   2345  CA  GLU A 313       3.564 -32.807  15.427  0.51 46.09           C
ATOM   2347  C   GLU A 313       4.640 -31.875  14.877  0.51 43.78           C
ATOM   2349  O   GLU A 313       5.788 -32.279  14.703  0.51 44.90           O
ATOM   2351  CB  GLU A 313       2.658 -33.281  14.288  0.51 44.42           C
ATOM   2353  CG  GLU A 313       3.376 -34.198  13.307  0.51 52.13           C
ATOM   2355  CD  GLU A 313       2.769 -34.184  11.918  0.51 56.50           C
ATOM   2357  OE1 GLU A 313       1.571 -33.848  11.787  0.51 61.39           O
ATOM   2359  OE2 GLU A 313       3.500 -34.505  10.956  0.51 45.33           O
ATOM   2360  N   ALA A 314       4.269 -30.625  14.615  1.00 38.63           N
ATOM   2361  CA  ALA A 314       5.234 -29.656  14.107  1.00 39.14           C
ATOM   2362  C   ALA A 314       6.346 -29.419  15.118  1.00 43.90           C
ATOM   2363  O   ALA A 314       7.526 -29.439  14.765  1.00 45.92           O
```

Fonte: Autores.

Com a execução desse último *script*, a primeira etapa de configuração inicial dos arquivos PDB foram realizados. A segunda etapa é a responsável pela realização do *Docking* Molecular. O *Docking* precisa que uma série de tarefas sejam realizadas para que o processo seja concluído.

Até o momento as atividades consideravam apenas uma única molécula receptora, e as atividades anteriores dependiam de arquivos que eram gerados nas atividades antecessoras. Aqui tem-se um novo contexto. Tem-se primeiro atividades que podem ser consideradas isoladas, que não dependem de atividades anteriores, logo podem ser executadas em paralelo. Isso foi considerado no ProtCool_Dynamic, sendo executadas em paralelo todas as atividades que não possuem interferência. O paralelismo dessas atividades permite que a execução seja realizada de forma mais ágil, aumentando a velocidade de execução do *workflow*. Mas além disso, o sistema possibilita que múltiplos ligantes sejam tratados de uma única vez. Assim, caso o usuário deseje fazer a simulação de uma série de ligantes para uma mesma molécula receptora, isso é possível. Ou seja, o ProtCool_Dynamic possibilita que diversas preparações sejam realizadas ao mesmo tempo. Com isso, além do paralelismo de tarefas, usou-se o paralelismo em algumas tarefas específicas. Na descrição de cada atividade isso será destacado.

[33] https://www.rcsb.org/structure/4EY6

A primeira atividade necessária para se realizar o *docking* é a busca por ligantes. O *workflow* aceita arquivos de ligante no formato mol2, que podem ser recuperados no *site* do ZINC15 ou por meio de um arquivo fornecido pelo pesquisador. Para a recuperação no ZINC, o pesquisador deve fornecer o código completo do ligante no ZINC, por exemplo (ZINC00491073) que é a Galantamina (GNT). Quando o usuário quiser fornecer um arquivo próprio, deve colocar o arquivo no formato mol2 na pasta do usuário. Para realizar a busca do ligante direto no *site* do ZINC o *workflow* utiliza o módulo *requests* do *Python*. Nesse *script* é necessário que o computador tenha acesso à Internet para buscar os arquivos do tipo ZINC.

Quando o *workflow* recupera um arquivo do tipo mol2 do ZINC15, é disponibilizado um arquivo que contém as informações de todas as representações 3D existentes no banco. Existem 4 possíveis representações que são geradas de acordo com o *range* do pH (STERLING e IRWIN, 2015). O primeiro é o pH de referência que é 7.4, o segundo é o pH médio, entre 6.4 a 8.4, o terceiro é com baixo pH de 5.4 a 6.4 e a última é com pH alto, de 8.4 a 9.4. Essas representações nem sempre estão presentes em todas as estruturas. Assim, a tarefa "separar ligantes" criará um arquivo que contém um único mol2, que é o primeiro listado (referência). Pode ocorrer de nem todos os ligantes possuírem todos os formatos, mas nos casos de existir o arquivo mol2 no ZINC normalmente o de referência é listado.

Conforme dito antes, ProtCool hoje é preparado para realizar *docking* a partir de dois sistemas, o *Autodock Vina* e o *Smina*. O *Smina* é um sistema que é baseado no *Vina*, realizando alguns tipos de cálculo de forma diferenciada (KOES; BAUMGARTNER; CAMACHO, 2013). A seleção da pose é realizada a partir de uma função de pontuação (*score*), e é esta função que foi alterada no desenvolvimento do Smina (KOES; BAUMGARTNER; CAMACHO, 2013).

O pesquisador pode então fazer a escolha entre os dois sistemas, ou até mesmo nos dois ao mesmo tempo. Se o pesquisador optar por realizar o *docking* nos dois sistemas, todos os arquivos gerados a partir desse ponto no *workflow* serão separados em pastas diferenciadas *Vina* e *Smina*, sendo fornecidos todos os arquivos das duas opções. Assim, ao final de todo o processo o pesquisador poderá avaliar quais das simulações serão de fato executadas.

Para a execução correta do *docking* nos dois sistemas, é necessário que o pesquisador informe alguns dados. Como os dois sistemas são programados na mesma base, essas informações são as mesmas e devem ser passadas em um arquivo de configuração. Nesses arquivos são informados os nomes do receptor e do ligante e no caso de se realizar o *docking* flexível também é informado o nome do receptor flexível, a posição e o tamanho do *pocket* onde se pretende inserir o ligante e, além disso, o dado de *num_modes* (define o número máximo de conformações que devem ser geradas na execução do *docking*) e quantidade de CPU que deseja utilizar para fazer o *docking*. Além desses dados pode-se também determinar o *energy_rage* (define a máxima diferença de energia em kcal/mol aceita entre a melhor e a pior conformação gerada) e o *exhaustiveness* (determina o esforço do algoritmo que será realizado até que se consiga todas as conformações solicitadas).

Um arquivo desse deve ser gerado para cada um dos ligantes com o qual se está trabalhando. A tarefa preparar arquivos de configuração realiza a geração de todos os arquivos conforme os dados disponibilizados pelo pesquisador no `confdinamica.txt`.

Para a realização do *docking* é preciso transformar o arquivo PDB da proteína em um arquivo no formato pdbqt. Isso é realizado pela tarefa "preparar proteína". Essa tarefa é composta por dois *scripts*. Isso é necessário pois a geração de *log* da conversão de arquivos PDB em pdbqt e esse é um *log* importante de ser armazenado na proveniência para posterior análise do pesquisador. Para realizar a conversão dos dados é utilizado o MGLTools-1.5.6 (DALLAKYAN, 2010), que contém diversas funcionalidades que auxiliam na preparação do *docking* e na geração dos arquivos finais de conformações geradas. No caso específico desse *script* é utilizada a funcionalidade `prepare_receptor4.py`. O pesquisador pode ainda especificar se deseja que os hidrogênios do ligante e do receptor sejam considerados no momento do *docking*.

A tarefa “preparar proteína flexível” possibilita que o *docking* seja realizado de forma flexível. Nos *dockings* tradicionais, a proteína fica rígida e apenas o ligante se movimenta de forma a se obter a melhor conformação. Isso garante um menor custo computacional, porém, em algumas situações não se consegue as conformações mais adequadas (DE PARIS, 2017). Os *dockings* flexíveis possibilitam que além do ligante a proteína, ou parte dela se movimente, o que pode gerar poses mais confiáveis, porém, o custo computacional disso pode ser elevado (DE PARIS, 2017). O *Vina* e *Smina* possibilitam que o pesquisador faça a escolha de resíduos que deseja que sejam flexíveis, assim, ao invés de apenas o ligante se movimentar, é possível que partes da proteína também se movimentem aumentando a qualidade do *docking* efetuado. Normalmente os pesquisadores optam por permitirem que o ligante e os resíduos que fazem parte do sítio ativo do receptor sejam flexíveis, e o restante do receptor seja rígido, o que garante um bom resultado, porém, sem que aumente muito o custo computacional necessário.

No caso do *script* preparar proteína flexível, o pesquisador pode especificar todos os resíduos da proteína que deseja tornar flexível no arquivo de configuração, de forma a possibilitar que o *docking* flexível seja realizado. Para a realização do *docking* flexível o pesquisador deverá colocar no arquivo de configuração algumas *tags* (campos). A *tag* `[FlexDocking]` é opcional e só deve aparecer no arquivo `confdinamica.txt` no caso de realização do *docking* flexível, caso contrário essa *tag* é omitida do arquivo. Deve ser apresentado um resíduo por linha sem a separação entre o nome do resíduo e o seu número e com a cadeia a que pertence no início separado por dois pontos (“A:TRP86”). Assim, mesmo que a molécula tenha mais de uma cadeia a ser estudada e os resíduos flexíveis pertençam a cadeias diferentes, pode-se estabelecer a flexibilidade do receptor. Também é utilizado o MGLTools-1.5.6 (funcionalidade `prepare_flexreceptor4.py`) para criar o arquivo de proteína flexível. O *script* gera novo arquivo pdbqt da proteína rígida que será utilizado no caso de *docking* flexível.

O passo “preparar ligante” será responsável pela geração dos arquivos pdbqt de cada um dos ligantes. Apesar da atividade de preparação do ligante não ser demorada em termos computacionais, caso a quantidade de ligantes seja grande, existe a possibilidade de se demandar um grande tempo para a execução desse *script*. Sendo assim, essa tarefa foi desenvolvida de forma a que as atividades sejam realizadas em paralelo. Para que isso possa ocorrer, essa tarefa necessita de dois *scripts* para que possa ser realizada.

O primeiro *script* é responsável por organizar todos os dados e criar um *script* no momento da sua execução (dinâmico) que realiza as chamadas do segundo *script*, que é quem de fato realiza a preparação do ligante. Na prática existem 3 *scripts* envolvidos no processo, de forma a possibilitar que a atividade seja realizada adequadamente. A tarefa já recupera o número de processadores existentes na máquina de forma a garantir o aproveitamento de todos os recursos computacionais disponíveis.

Para a realização da preparação do ligante, inicialmente é necessário realizar uma conversão do arquivo mol2 existente para um que possua cargas, assim, foi utilizado o *antechamber* do AMBER para conseguir gerar o novo arquivo mol2 a partir do arquivo de entrada. Após isso, foi utilizada a funcionalidade `prepare_ligand4.py` do MGLTools-1.5.6.

Com todos os arquivos preparados (receptor e ligantes) e os arquivos de configuração gerados, é possível a realização do *docking*. O pesquisador poderá escolher se deseja fazer o *docking* apenas com *Vina*, com *Smina* ou com os dois. Além disso, para cada ligante o pesquisador poderá escolher quantas rodadas deseja fazer. Sabe-se que quanto mais rodadas forem realizadas melhor o resultado, porém, existe um custo computacional associado a essa escolha. Assim, o pesquisador pode optar pelo número de vezes que deseja e no momento de seleção de poses, o *workflow* escolhe aquela que dentre todas as rodadas realizadas a que tiver o menor *score*.

Essa tarefa funciona da mesma forma que a tarefa de preparar ligante, ou seja, também é realizada em paralelo. Só que nesse caso específico do *docking*, como o pesquisador pode escolher em rodar o número de vezes que desejar para cada um dos ligantes, a chamada do processo de realizar *docking* passa a ser não apenas considerando o ligante, mas a rodada daquele ligante.

Como o ProtCool_Dynamic permite dois tipos de *docking* (*Vina* e *Smina*), essa tarefa para ser executado terá ao todo 6 *scripts* envolvidos na sua execução, sendo 3 para cada tipo de *docking* e um dos *scripts* gerado de forma dinâmica pelo sistema (`paralelo.py`). A opção de fazer a separação entre arquivos de *Vina* e *Smina* ao invés de se criar um único que controla o desejado pelo pesquisador foi devido aos fatores:

i. Facilidade de manutenção – quando ocorrer algum problema ou mudança de versão apenas um local será afetado pela modificação;
ii. Reutilização – como está se trabalhando com tarefas de *workflow*, ter as atividades separadas (e resolvendo um único problema) facilita na reutilização do mesmo *script* em outro *workflow*, conforme verificado no ProtCool_Docking (Seção 3.3);
iii. Expansão do *workflow* – quando novos sistemas de *docking* forem acrescentados ao *workflow*, será mais fácil a sua inserção no sistema;
iv. Gestão do *workflow* – no caso de o pesquisador escolher fazer os dois *dockings*, arquivos individuais de cada uma das estratégias serão gerados, assim, fica mais fácil para o controle do pesquisador dos resultados e *logs* recebidos pelo processo.

Obviamente que, para a execução dos *scripts* são necessários os seguintes sistemas instalados: *Vina* e *Smina*. No caso do *Smina* ele deve estar presente na pasta "`/ProtCool/path`.

A *Vina* e o *Smina* entregam o resultado com um arquivo único no formato pdbqt e que contém todas as poses listadas separadas por *MODEL*. A tarefa separar poses é responsável por separar cada modelo gerado em um arquivo pdbqt único. O *script* utiliza a função *vina_split* para realizar a separação dos arquivos. Nessa tarefa também existem dois *scripts Vina* e *Smina*, conforme já discutido anteriormente. Apesar de poderem existir diversos ligantes, essa funcionalidade não foi implementada em paralelo. Isso porque essa funcionalidade realiza funções rápidas de arquivos que demandam tempo de execução muito curto. Nos testes realizados, a parte de realizar a formatação do paralelismo; e o tempo gasto pelo Python para gerenciar o multiprocessamento acabam por ter alto custo computacional, fazendo que nos testes essa atividade em paralelo fosse mais custosa computacionalmente do que sendo realizada sequencialmente. Só haverá ganho com um número muito alto de ligantes, logo, decidiu-se por deixar essa atividade sem paralelismo.

Quando o *vina_split* separa as poses em arquivos distintos ela cria o seu próprio padrão de nomes, o que dificulta o tratamento ao longo do ProtCool. A tarefa padronização de nomes coloca o nome de cada pose gerada em um formato mais fácil de ser tratado pelo ProtCool. Essa atividade também foi realizada sem usar processamento paralelo.

O *docking* irá gerar até o valor máximo de conformações definidas pelo parâmetro `num_modes`, porém, nem sempre ele consegue gerar todas as poses solicitadas. Assim, caso não consiga gerar alguma pose em específico, os *scripts* escreverão um *warning* no arquivo `Errorlog.log` informando quando alguma conformação não for encontrada. Isso só será considerado um problema se nenhuma pose conseguir ser gerada.

A tarefa "converter pdbqt em PDB" realiza a conversão de formato das conformações. A conversão é realizada pela funcionalidade `pdbqt_to_pdb.py` do MGLTools-1.5.6. Também se optou por não se fazer o paralelismo pela rapidez de processamento. Essa tarefa acaba por possuir 4 *scripts* associados a ela, dois para cada tipo de *docking* realizado para poder armazenar o *log* do processo de conversão.

A tarefa "selecionar pose" finaliza o *docking*. Ela é responsável por indicar a melhor pose a ser utilizada no ProtCool. Para avaliar a melhor pose, o *script* realiza uma varredura de todos os *scores* alcançados em cada uma das conformações geradas, considerando todas as rodadas para aquele ligante que está sendo avaliado. A conformação com o melhor *score* (menor energia) é selecionada para seguir na preparação. Existem dois *scripts* nesse caso, um *Vina* e outro *Smina*.

A modelagem molecular é uma etapa importante do processo de preparação da dinâmica molecular. Ela é que garante que o receptor não terá falhas (omissões) em sua estrutura. A modelagem é realizada, conforme descrito na Seção 2.4.1.3 por homologia com estruturas similares. No caso do ProtCool_Dynamic utiliza-se o Modeller para realizar a modelagem. O Modeller é uma ferramenta bastante completa, mas que é toda executada em linha de comandos, sendo muita vezes necessária a programação em python para a realização da modelagem. Este é um conhecimento que diversos pesquisadores não possuem, o que dificulta o processo de modelagem. No caso do *workflow*, o pesquisador não terá que se preocupar com isso, uma vez que o sistema já executa todos os passos automaticamente.

Figura 7 – Exemplo Arquivo SEQ gerado pelo Modeller. Molécula 4EY6. No arquivo é apresentada a sequência de aminoácidos da molécula a ser modelada. No caso, só foi considerada a cadeia A, uma vez que o arquivo já passou pela primeira etapa do ProtCool, que seleciona apenas a cadeia com a qual a proteína será trabalhada ao longo do *workflow*.

```
>P1;4EY6
structureX:4EY6:   4 :A:+530 :A:::-1.00:-1.00
EDAELLVTVRGGRLRGIRLKTPGGPVSAFLGIPFAEPPMGPRRFLPPEPKQPWSGVVDATTFQSVCYQYVDTLYP
GFEGTEMWNPNRELSEDCLYLNVWTPYPRPTSPTPVLVWIYGGGFYSGASSLDVYDGRFLVQAERTVLVSMNYRV
GAFGFLALPGSREAPGNVGLLDQRLALQWVQENVAAFGGDPTSVTLFGESAGAASVGMHLLSPPSRGLFHRAVLQ
SGAPNGPWATVGMGEARRRATQLAHLVGCPNDTELVACLRTRPAQVLVNHEWHVLPQESVFRFSFVPVVDGDFLS
DTPEALINAGDFHGLQVLVGVVKDEGSYFLVYGAPGFSKDNESLISRAEFLAGVRVGVPQVSDLAAEAVVLHYTD
WLHPEDPARLREALSDVVGDHNVVCPVAQLAGRLAAQGARVYAYVFEHRASTLSWPLWMGVPHGYEIEFIFGIPL
DPSRNYTAEEKIFAQRLMRYWANFARTGDPNEPRDPQWPPYTAGAQQYVSLDLRPLEVRRGLRAQACAFWNRFLP
KLLSA*
```

Fonte: Autores.

O primeiro passo necessário para que o Modeller realize a modelagem é a de gerar o arquivo SEQ. Esse arquivo contém a sequência de aminoácidos que a molécula que será modelada possui. Essa sequência é importante porque é a partir dela que serão detectadas as eventuais falhas existentes na estrutura e que o Modeller irá corrigir. O Modeller possui uma funcionalidade que dada a estrutura ele recupera a sua sequência. O *script* utiliza essa funcionalidade para que possa buscar a sequência de aminoácidos. Essa tarefa possui dois *scripts*, para que o *log* do Modeller seja adequadamente armazenado. No caso de o pesquisador ter selecionado que deseja considerar as águas cristalográficas e os hidrogênios da molécula original, o *script* irá passar essa informação ao Modeller para que o tratamento seja efetuado. A Figura 7 apresenta um exemplo do arquivo SEQ gerado pelo Modeller da molécula 4EY6.

A sequência de aminoácidos está no formato FASTA. Essa sequência possui todos os aminoácidos que deveriam estar na estrutura, como se fosse um modelo. A sequência FASTA é disponibilizada pelo *site* RCSB. A tarefa "buscar arquivo FASTA" será responsável por recuperar o arquivo direto do *site* RCSB. O *script* utiliza o módulo `requests` para fazer a recuperação do arquivo. O computador deverá possuir no momento acesso à Internet. Caso o acesso a Internet não possa ser estabelecido, será gravada mensagem de erro informando no `Errorlog.log`. O arquivo FASTA é recuperado completo, ou seja, da mesma forma que o PDB original vem com todas as cadeias com as quais foi cristalografado. Assim, após a recuperação do arquivo FASTA do *site*, é necessário fazer os ajustes, permanecendo apenas com as cadeias com as quais está se trabalhando. A

Figura **8** possui o arquivo FASTA final da molécula 4EY6 apenas considerando a cadeia A.

O Modeller precisa de um arquivo que ele chama de arquivo de alinhamento, em que ele consegue identificar onde estão os *gaps* para que a molécula possa ser modelada. Esse arquivo é gerado pela tarefa gerar arquivo ALI e ele usa para isso os arquivos SEQ e FASTA. Conforme relatado o

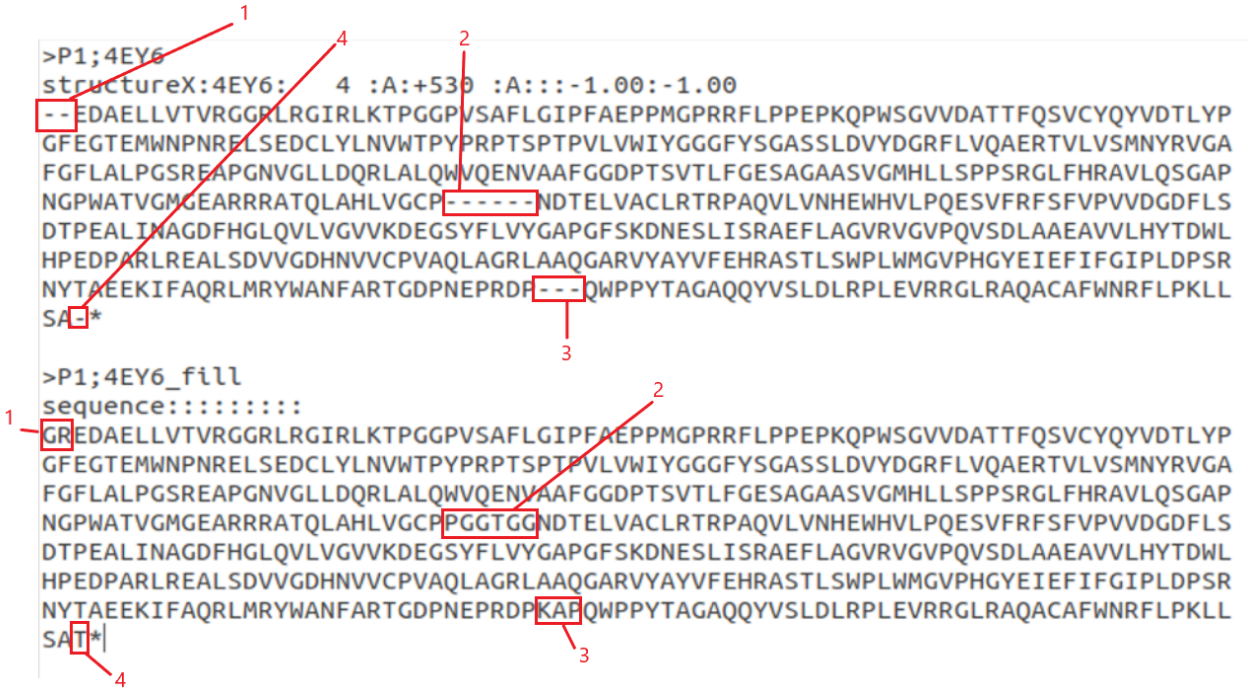
arquivo FASTA possui a sequência completa e o arquivo SEQ possui a sequência "real". O *script* dessa tarefa é responsável por realizar um *parse* de forma a identificar onde estão esses *gaps* comparando os dois arquivos. A comparação entre esses dois arquivos, ocorre com a realização de alinhamento da sequência, logo, essa tarefa é responsável por esse alinhamento, identificando esses pontos que estão vagos na molécula. A Figura 9 possui um exemplo de arquivo ALI da molécula 4EY6.

Figura 8 – Exemplo Arquivo FASTA final. Na imagem é apresentado o arquivo FASTA, com a sequência de aminoácidos, após ter realizado a filtragem por cadeia.

```
>4EY6_1|A|Acetylcholinesterase|Homo sapiens (9606)
GREDAELLVTVRGGRLRGIRLKTPGGPVSAFLGIPFAEPPMGPRRFLPPEPKQPWSGVVDATTFQSVCYQYVDTLYP
GFEGTEMWNPNRELSEDCLYLNVWTPYPRPTSPTPVLVWIYGGGFYSGASSLDVYDGRFLVQAERTVLVSMNYRVGA
FGFLALPGSREAPGNVGLLDQRLALQWVQENVAAFGGDPTSVTLFGESAGAASVGMHLLSPPSRGLFHRAVLQSGAP
NGPWATVGMGEARRRATQLAHLVGCPPGGTGGNDTELVACLRTRPAQVLVNHEWHVLPQESVFRFSFVPVVDGDFLS
DTPEALINAGDFHGLQVLVGVVKDEGSYFLVYGAPGFSKDNESLISRAEFLAGVRVGVPQVSDLAAEAVVLHYTDWL
HPEDPARLREALSDVVGDHNVVCPVAQLAGRLAAQGARVYAYVFEHRASTLSWPLWMGVPHGYEIEFIFGIPLDPSR
NYTAEEKIFAQRLMRYWANFARTGDPNEPRDPKAPQWPPYTAGAQQYVSLDLRPLEVRRGLRAQACAFWNRFLPKLL
SAT
```

Fonte: Autores.

Figura 9 – Exemplo Arquivo ALI – 4EY6 com GAP. Na imagem são marcados em vermelho onde estão os pontos com falhas no arquivo. Essas marcações identificam onde serão realizadas as modelagens (na parte de cima) e quais resíduos serão modelados em cada ponto (na parte de baixo).



Fonte: Autores.

Em vermelho na Figura 9 estão 4 marcações na sequência de cima e mais 4 marcações na sequência de baixo. Essas marcações identificam onde serão realizadas as modelagens (na parte de cima) e quais resíduos serão modelados em cada ponto (na parte de baixo). A marcação de número 1 mostra que estão faltando 2 resíduos no início da sequência que deveria ser o GR (GLY, ARG). Na marcação 2 tem-se 6 resíduos faltantes PGGTGG (PRO, GLY, GLY, THR, GLY, GLY). Na marcação

3 são 3 resíduos KAP (LYS, ALA, PRO) e na marcação 4, 1 resíduo está faltando, o T (THR). Ao todo tem-se 12 resíduos faltando. Outra observação importante é que faltam resíduos nos três pontos da proteína (início, meio e fim). Isso é importante, pois o tratamento dado pelo Modeller é diferente dependendo do ponto da molécula em que faltam elementos.

Já a Figura 10 apresenta um exemplo de PDB 1PPF[34] (Elastase de Leucócito Humano) que não possui *gap*, logo o Modeller não será processado.

No caso de o pesquisador ter escolhido trabalhar com as moléculas de água, ao final da listagem dos resíduos são disponibilizados ".", que representam as moléculas de água. Dessa forma durante a modelagem o Modeller irá considerar essas moléculas. A Figura 11 apresenta essa situação.

Figura 10 – Exemplo Arquivo ALI – 1PPF sem *gap*. Nesse arquivo não foram identificados pontos de modelagem necessários na proteína. Assim, quando passar pela tarefa de modelagem, o *script* não executará o Modeller.

```
>P1;1PPF
structureX:1PPF:  16 :E:+218 :E:::-1.00:-1.00
IVGGRRARPHAWPFMVSLQLRGGHFCGATLIAPNFVMSAAHCVANVNVRAVRVVLGAHNLSRREPTRQVFAVQRIFENGYDPVNLL
NDIVILQLNGSATINANVQVAQLPAQGRRLGNGVQCLAMGWGLLGRNRGIASVLQELNVTVVTSLCRRSNVCTLVRGRQAGVCFGD
SGSPLVCNGLIHGIASFVRGGCASGLYPDAFAPVAQFVNWIDSIIQ*

>P1;1PPF_fill
sequence:::::::::
IVGGRRARPHAWPFMVSLQLRGGHFCGATLIAPNFVMSAAHCVANVNVRAVRVVLGAHNLSRREPTRQVFAVQRIFENGYDPVNLL
NDIVILQLNGSATINANVQVAQLPAQGRRLGNGVQCLAMGWGLLGRNRGIASVLQELNVTVVTSLCRRSNVCTLVRGRQAGVCFGD
SGSPLVCNGLIHGIASFVRGGCASGLYPDAFAPVAQFVNWIDSIIQ*
```

Fonte: Autores.

Figura 11 – Exemplo Arquivo ALI com Água. Nesse caso a 1PPF foi selecionada para trabalhar com as águas cristalográficas, assim, no arquivo ALI são destacadas todas as moléculas de água, ao final da sequência de aminoácidos. As águas são identificadas com ".".

```
>P1;1PPF
structureX:1PPF:  16 :E:+428 :E:::-1.00:-1.00
IVGGRRARPHAWPFMVSLQLRGGHFCGATLIAPNFVMSAAHCVANVNVRAVRVVLGAHNLSRREPTRQVFAVQRIFENGYDPVNLLNDIVILQ
LNGSATINANVQVAQLPAQGRRLGNGVQCLAMGWGLLGRNRGIASVLQELNVTVVTSLCRRSNVCTLVRGRQAGVCFGDSGSPLVCNGLIHGI
ASFVRGGCASGLYPDAFAPVAQFVNWIDSIIQ.....................................................................
.....................................................................................................
.....................................................*

>P1;1PPF_fill
sequence:::::::::
IVGGRRARPHAWPFMVSLQLRGGHFCGATLIAPNFVMSAAHCVANVNVRAVRVVLGAHNLSRREPTRQVFAVQRIFENGYDPVNLLNDIVILQ
LNGSATINANVQVAQLPAQGRRLGNGVQCLAMGWGLLGRNRGIASVLQELNVTVVTSLCRRSNVCTLVRGRQAGVCFGDSGSPLVCNGLIHGI
ASFVRGGCASGLYPDAFAPVAQFVNWIDSIIQ.....................................................................
.....................................................................................................
.....................................................*
```

Fonte: Autores.

A partir do momento que se tem o arquivo de alinhamento (ALI), o ProtCool_Dynamic passará para a tarefa de realizar modelagem. Essa tarefa é a responsável por, a partir dos dados de cada *gap*, confrontar com os dados existentes de moléculas similares e realizar a modelagem do sistema. Conforme descrito acima, alguns pontos devem ser levados em consideração quando se está realizando a modelagem do sistema, pois a forma de chamada do comando Modeller vai variar em cada um dos casos, exemplos: se levará em consideração os hidrogênios, as águas, onde se encontra o *gap* (início, meio ou fim) da estrutura. Depois de definidos todos esses valores, é realizada então a chamada do comando Modeller que será executado.

[34] https://www.rcsb.org/structure/1PPF

Ao final de sua execução o Modeller fornece uma série de modelos propostos para a escolha do melhor. Então, nova funcionalidade do Modeller é executada para que seja informada a energia de cada um dos modelos gerados para que se possa realizar a comparação de cada uma delas e fazer a escolha entre os modelos. O *script* então seleciona aquele que teve a menor energia, finalizando o processo. A tarefa realizar modelagem possui dois *scripts* para que o *log* do Modeller seja armazenado adequadamente.

O Modeller além de realizar a modelagem da molécula também realiza alguns acertos no PDB, caso não tenham sido realizados anteriormente. Um desses casos é a ocupância que é resolvida quando se modela o receptor. No ProtCool_Dynamic, realiza-se a escolha da ocupância antes de fazer o *docking*. Nos casos em que não será necessário realizar o *docking*, pode-se deixar para o Modeller resolver a ocupância, retirando esse passo do *workflow*.

Ao final do processo, caso tenha escolhido por apenas uma cadeia no PDB, o Modeller realiza a retirada da identificação da cadeia, ou seja, ele entrega um PDB que não possui o registro da cadeia à qual cada resíduo pertence. Assim, a tarefa "acertar arquivo Modeller" é responsável por realizar o acerto da cadeia, voltando com a identificação de cadeia no PDB.

A última tarefa que deve ser realizada na modelagem de moléculas é a acertar numeração. Essa tarefa na verdade faz alguns passos importantes para que se possa seguir o processamento da preparação da dinâmica. A primeira atividade dessa tarefa é de fato realizar os últimos acertos necessários no arquivo. O Modeller realiza uma série de acertos no sistema, tais como, a ocupância e renumeração de resíduos e átomos de forma a garantir a integridade do sistema. Porém, tem que se pensar nos casos em que um PDF possa não ter passado pelo Modeller, ou algum caso que o Modeller possa deixar de realizar. Assim, esse *script* inicia realizando uma correção de numeração de resíduos e átomos dos arquivos PDB. A Figura 12 apresenta um trecho do PDB da molécula 1PPF antes do acerto. Nesse trecho pode-se perceber que o 1PPF inicia a numeração de resíduos pela numeração 16. Conforme verificado anteriormente, o 1PPF não possui *gaps* relacionados à falta de resíduos. Todos os resíduos estão presentes, conforme a análise do arquivo ALI (Figura 10), ou seja, é apenas uma falha na numeração dos registros e não a necessidade de se adicionar resíduos.

Figura 12 – Exemplo PDB 1PPF – antes do Acerto – Numeração Resíduos – Caso 1. Aqui apresenta-se o caso de um PDB que não inicia a numeração de resíduos com o valor 1. Assim, deve-se acertar o PDB para que ele passe a ser numerado a partir do valor 1.

```
ATOM      1  N   ILE E  16      37.590  58.140   6.539  1.00  2.00           N
ATOM      2  CA  ILE E  16      38.087  56.865   5.985  1.00  3.00           C
ATOM      3  C   ILE E  16      39.411  56.489   6.623  1.00  2.00           C
ATOM      4  O   ILE E  16      40.329  57.323   6.672  1.00  4.31           O
ATOM      5  CB  ILE E  16      38.419  57.046   4.471  1.00  3.43           C
ATOM      6  CG1 ILE E  16      37.181  57.472   3.653  1.00  6.30           C
ATOM      7  CG2 ILE E  16      39.058  55.804   3.856  1.00  3.09           C
ATOM      8  CD1 ILE E  16      36.055  56.430   3.536  1.00  2.00           C
ATOM      9  N   VAL E  17      39.430  55.332   7.250  1.00  2.00           N
ATOM     10  CA  VAL E  17      40.600  54.931   7.964  1.00  2.00           C
ATOM     11  C   VAL E  17      41.409  54.034   7.049  1.00  8.13           C
ATOM     12  O   VAL E  17      40.786  53.135   6.491  1.00  7.95           O
ATOM     13  CB  VAL E  17      40.191  54.143   9.219  1.00  3.03           C
ATOM     14  CG1 VAL E  17      41.418  53.597   9.952  1.00 13.55           C
ATOM     15  CG2 VAL E  17      39.403  54.990  10.268  1.00 10.41           C
```

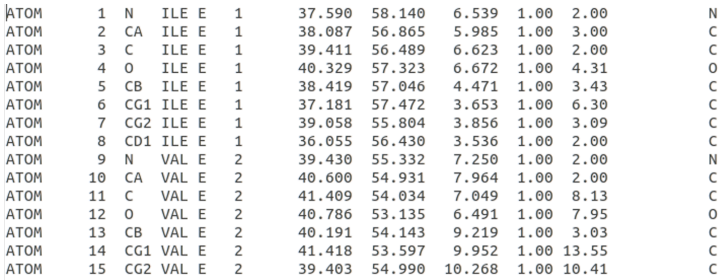
Fonte: Autores.

Assim, deve-se fazer esse ajuste na molécula de forma a garantir a sequência correta dos resíduos. A Figura 13 apresenta o mesmo trecho após passar pela etapa de "acertar numeração do *script"*.

Outro caso que se pode observar pela Figura 14 é a presença na 1PPF de resíduos numerados com a mesma numeração, porém, com uma letra à frente do número do resíduo. Marcado em vermelho está o resíduo 220 e 220A na molécula da 1PPF. A Figura 15, apresenta o mesmo trecho após o acerto. O resíduo 220 passou para a numeração 194 e o resíduo 220A passou a ter a numeração 195.

Esse acerto também resolve casos como o da molécula 1BSZ[35] (*Peptide Deformylase*) que possui 3 cadeias em sua sequência que são identificadas como A (1-168), B (501-668) e C (1001-1168). Passando pelo *script* de acerto de numeração as três cadeias ficam com seus resíduos numerados sequencialmente A (1-168), B (169-336) e C (337-504).

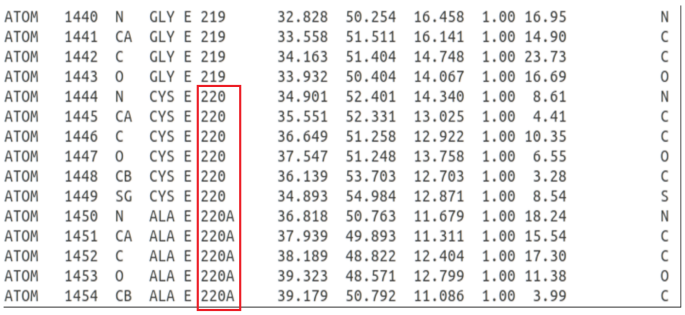
Figura 13 – Exemplo PDB 1PPF – depois do Acerto – Numeração Resíduos – Caso 1. Nesse caso, os resíduos do PDB foram renumerados, fazendo com que o PDB inicie com o valor 1.



```
ATOM      1  N   ILE E   1      37.590  58.140   6.539  1.00  2.00           N
ATOM      2  CA  ILE E   1      38.087  56.865   5.985  1.00  3.00           C
ATOM      3  C   ILE E   1      39.411  56.489   6.623  1.00  2.00           C
ATOM      4  O   ILE E   1      40.329  57.323   6.672  1.00  4.31           O
ATOM      5  CB  ILE E   1      38.419  57.046   4.471  1.00  3.43           C
ATOM      6  CG1 ILE E   1      37.181  57.472   3.653  1.00  6.30           C
ATOM      7  CG2 ILE E   1      39.058  55.804   3.856  1.00  3.09           C
ATOM      8  CD1 ILE E   1      36.055  56.430   3.536  1.00  2.00           C
ATOM      9  N   VAL E   2      39.430  55.332   7.250  1.00  2.00           N
ATOM     10  CA  VAL E   2      40.600  54.931   7.964  1.00  2.00           C
ATOM     11  C   VAL E   2      41.409  54.034   7.049  1.00  8.13           C
ATOM     12  O   VAL E   2      40.786  53.135   6.491  1.00  7.95           O
ATOM     13  CB  VAL E   2      40.191  54.143   9.219  1.00  3.03           C
ATOM     14  CG1 VAL E   2      41.418  53.597   9.952  1.00 13.55           C
ATOM     15  CG2 VAL E   2      39.403  54.990  10.268  1.00 10.41           C
```

Fonte: Autores.

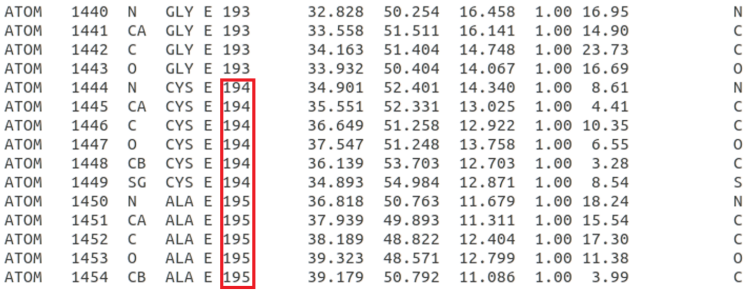
Figura 14 – Exemplo PDB 1PPF – antes do Acerto – Numeração Resíduos – Caso 2. Resíduos com mesma numeração, diferenciados com uma letra à frente do número do resíduo.



```
ATOM   1440  N   GLY E 219      32.828  50.254  16.458  1.00 16.95           N
ATOM   1441  CA  GLY E 219      33.558  51.511  16.141  1.00 14.90           C
ATOM   1442  C   GLY E 219      34.163  51.404  14.748  1.00 23.73           C
ATOM   1443  O   GLY E 219      33.932  50.404  14.067  1.00 16.69           O
ATOM   1444  N   CYS E 220      34.901  52.401  14.340  1.00  8.61           N
ATOM   1445  CA  CYS E 220      35.551  52.331  13.025  1.00  4.41           C
ATOM   1446  C   CYS E 220      36.649  51.258  12.922  1.00 10.35           C
ATOM   1447  O   CYS E 220      37.547  51.248  13.758  1.00  6.55           O
ATOM   1448  CB  CYS E 220      36.139  53.703  12.703  1.00  3.28           C
ATOM   1449  SG  CYS E 220      34.893  54.984  12.871  1.00  8.54           S
ATOM   1450  N   ALA E 220A     36.818  50.763  11.679  1.00 18.24           N
ATOM   1451  CA  ALA E 220A     37.939  49.893  11.311  1.00 15.54           C
ATOM   1452  C   ALA E 220A     38.189  48.822  12.404  1.00 17.30           C
ATOM   1453  O   ALA E 220A     39.323  48.571  12.799  1.00 11.38           O
ATOM   1454  CB  ALA E 220A     39.179  50.792  11.086  1.00  3.99           C
```

Fonte: Autores.

Figura 15 – Exemplo PDB 1PPF – depois do Acerto – Numeração Resíduos – Caso 2. Resíduos com mesma numeração e letra os diferenciando foram renumerados, retirando esta anomalia do PDB.



```
ATOM   1440  N   GLY E 193      32.828  50.254  16.458  1.00 16.95           N
ATOM   1441  CA  GLY E 193      33.558  51.511  16.141  1.00 14.90           C
ATOM   1442  C   GLY E 193      34.163  51.404  14.748  1.00 23.73           C
ATOM   1443  O   GLY E 193      33.932  50.404  14.067  1.00 16.69           O
ATOM   1444  N   CYS E 194      34.901  52.401  14.340  1.00  8.61           N
ATOM   1445  CA  CYS E 194      35.551  52.331  13.025  1.00  4.41           C
ATOM   1446  C   CYS E 194      36.649  51.258  12.922  1.00 10.35           C
ATOM   1447  O   CYS E 194      37.547  51.248  13.758  1.00  6.55           O
ATOM   1448  CB  CYS E 194      36.139  53.703  12.703  1.00  3.28           C
ATOM   1449  SG  CYS E 194      34.893  54.984  12.871  1.00  8.54           S
ATOM   1450  N   ALA E 195      36.818  50.763  11.679  1.00 18.24           N
ATOM   1451  CA  ALA E 195      37.939  49.893  11.311  1.00 15.54           C
ATOM   1452  C   ALA E 195      38.189  48.822  12.404  1.00 17.30           C
ATOM   1453  O   ALA E 195      39.323  48.571  12.799  1.00 11.38           O
ATOM   1454  CB  ALA E 195      39.179  50.792  11.086  1.00  3.99           C
```

Fonte: Autores.

[35] https://www.rcsb.org/structure/1BSZ

Outra situação que esse *script* acerta é quando o PDB inicia com valores 0, como o caso da 1BXW[36] (*Outer Membrane Protein* A - OMPA). PDBs que iniciam com valores negativos também são reorganizados, conforme 1D5T[37] (*Guanine Nucleotide Dissociation Inhibitor*).

Após realizar o acerto da sequência de numeração da molécula, esse *script* vai fazer uma análise de todos os valores que foram alterados até esse momento no sistema. Existem diversos acertos que foram realizados, tais como: numeração de átomos faltantes devido à ocupância; numeração de resíduos devido a modelagem ou os acertos descritos acima. Alguns desses acertos foram realizados pelo próprio ProtCool_Dynamic e outros foram realizados por ferramentas como o Modeller. O problema é que se precisa saber quais são as novas numerações de cada um dos resíduos. Saber disso, implica em saber como analisar os dados depois da dinâmica pronta, uma vez que com o tratamento, a molécula ficará com numeração diversa da que normalmente se encontra na literatura e, diferente do PDB original que foi recuperado no RCSB.

Além disso, tem que se considerar que durante a geração do arquivo de configuração para que se possa executar o *workflow*, devem ser fornecidos alguns resíduos para que a configuração possa ser realizada. Exemplos disso, são os resíduos fornecidos para se realizar o *docking* flexível, bem como os resíduos que ficarão livres na restrição harmônica que será preparada posteriormente. Outro arquivo que lista os resíduos são os arquivos que armazenam as pontes dissulfeto que serão utilizadas para a definição do campo de força.

Sendo assim, é necessário realizar tanto o mapeamento desses resíduos (antes e depois) para acertar os arquivos `confdinamica.txt` e `pontesmolecula.txt`, quanto para que o pesquisador tenha um relatório que lhe apresente o mapeamento de todos os dados modificados para que ele consiga realizar uma melhor análise dos seus resultados.

Figura 16 – Parte do Arquivo `4EY6_Acertos.txt`. Esse arquivo possui o mapeamento das informações do PDB. ALI – parte 1, Ali – parte 2, PROT1 – antes da modelagem, PROT2 – depois da modelagem e ajustes. A partir da análise do arquivo é possível observar todas as mudanças de numeração de resíduos que foram geradas durante a modelagem do sistema.

ALI1		ALI2		PROT1			PROT2		
1	-	1	GLY			-	GLY	A	1
2	-	2	ARG			-	ARG	A	2
3	GLU	3	GLU	GLU	A	4	GLU	A	3
4	ASP	4	ASP	ASP	A	5	ASP	A	4
5	ALA	5	ALA	ALA	A	6	ALA	A	5
6	GLU	6	GLU	GLU	A	7	GLU	A	6
7	LEU	7	LEU	LEU	A	8	LEU	A	7
8	LEU	8	LEU	LEU	A	9	LEU	A	8
9	VAL	9	VAL	VAL	A	10	VAL	A	9
10	THR	10	THR	THR	A	11	THR	A	10
11	VAL	11	VAL	VAL	A	12	VAL	A	11
12	ARG	12	ARG	ARG	A	13	ARG	A	12
13	GLY	13	GLY	GLY	A	14	GLY	A	13
14	GLY	14	GLY	GLY	A	15	GLY	A	14
15	ARG	15	ARG	ARG	A	16	ARG	A	15

Fonte: Autores.

Para realizar isso, o *script* realiza a avaliação de 3 arquivos: ALI, molécula antes da modelagem e molécula após todas as modificações. O arquivo ALI, conforme apresentado anteriormente, possui duas partes. A primeira contém a estrutura da molécula seguindo a sequência do arquivo antes da modelagem, mostrando com "-" os locais onde ocorrerão a modelagem. A segunda parte do arquivo

[36] https://www.rcsb.org/structure/1BXW
[37] https://www.rcsb.org/structure/1D5T

apresenta a sequência do arquivo FASTA completo. Os dois arquivos de molécula devem ser comparados para que se possa mapear os erros. Logo, pega-se os dois arquivos e se compara com a segunda parte da molécula de forma a se verificar em quais locais ocorreram as modificações e, também, para verificar se a molécula final de fato está completa. Com esses dados é montado um arquivo com as informações referentes a cada um desses 4 elementos (ALI – parte 1, Ali – parte 2, PROT1 – antes da modelagem, PROT2 – depois da modelagem e ajustes). Parte desse arquivo pode ser visualizado na Figura 16. Com a análise desse arquivo é possível avaliar se a molécula foi adequadamente modelada e quais as alterações nas numerações dos resíduos.

Além de fornecer esse arquivo, esse *script* ainda realiza a alteração dos arquivos de configuração e de pontes, criando arquivos que contém os valores corretos dos resíduos, considerando a nova numeração da molécula. Com essa tarefa a modelagem da molécula finaliza e é possível passar para a próxima etapa que será responsável pela protonação.

A etapa de protonação apresenta três tarefas. A primeira tarefa realiza a execução propriamente da protonação no *site* H++ e a segunda serve para acertar o PDB com os dados da protonação definida pelo H++. Já a terceira tarefa será responsável no final da protonação por agrupar os ligantes escolhidos no *docking* com a proteína toda preparada.

A atividade de protonação, conforme discutido na Seção 2.4.1.4 é importante na determinação da estrutura da molécula e na afinidade de proteínas com seus ligantes, sendo bastante importante para a preparação de dinâmicas moleculares. A protonação é importante uma vez que os arquivos PDB nem sempre possuem os hidrogênios. É importante que seja realizada a inserção dos hidrogênios, porém, existem resíduos, como as histidinas, por exemplo, que possuem números diferenciados de hidrogênios, de acordo com o ambiente ao qual estão inseridos e de acordo com o pH. Com isso, é importante fazer o estudo de protonação que levará em consideração diversos fatores do ambiente para determinar como os hidrogênios devem ser inseridos na molécula.

Para realização da protonação foi escolhido o *site* H++. Porém, quando da criação dos *scripts*, descobriu-se que, conforme descrito pelo próprio *site*, é possível pegar os códigos fonte do sistema e fazer a sua instalação, mas essa instalação não é trivial. Os desenvolvedores não disponibilizam uma API (*Application Programming Interface*) para que a funcionalidade seja utilizada de forma automática. Eles ainda relatam no *site*, que nunca testaram o sistema em outro ambiente computacional que não o que está disponibilizado e não sabem como o sistema irá se comportar. E informam que o recurso computacional utilizado é grande (ONUFRIEV *et al.*, 2005).

Um dos objetivos do sistema aqui proposto é que diversos pesquisadores tenham à sua disposição, apenas instalando os softwares de dinâmica (normalmente já possuem instalados), possam, com rapidez e facilidade utilizar a ferramenta. Disponibilizar uma ferramenta de difícil instalação e, além disso, a cada nova versão ter que gerar todo o transtorno de nova instalação não era interessante para a proposta deste trabalho. Como o *site* também não disponibiliza uma API para o acesso às funcionalidades, a única forma de acesso é por meio do *site*.

Para se ter essa parte automatizada, foram utilizados o *Selenium*[38] e o *Gheckodriver*[39]. Esses softwares são normalmente utilizados para realizar teste automático de sistemas em ambientes web. O que eles normalmente fazem é, a partir de parâmetros fornecidos e de dados maciços de entrada, acessam remotamente a *interface* e sem a necessidade de intervenção humana, conseguem entrar com esses dados nos campos encontrados na tela. Após a execução, eles retornam com os dados salvos, de acordo com o estabelecido no sistema.

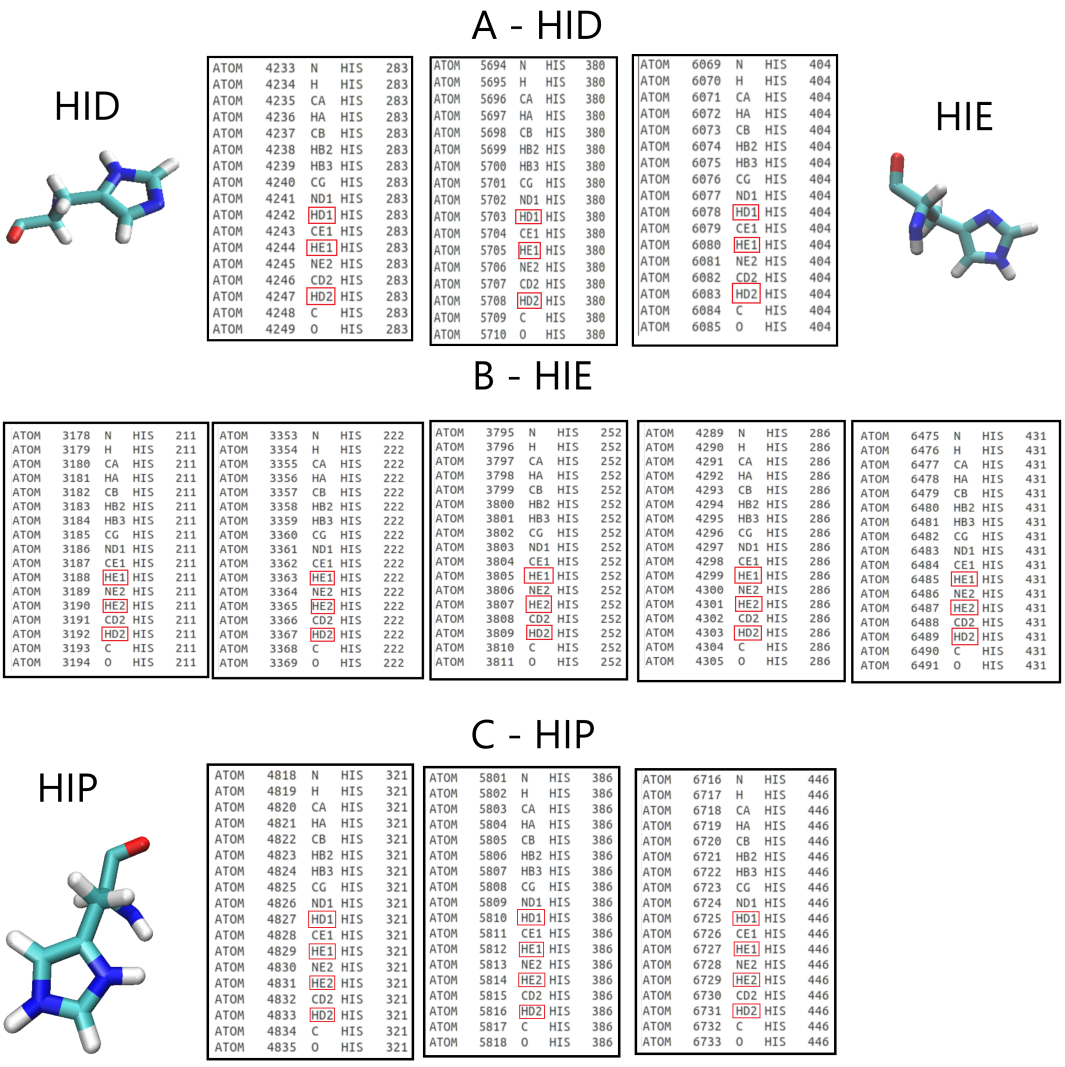
Para então acessar e recuperar os dados do *site* H++, estudou-se a interface e forma de requisição do sistema e programou-se o "robô" de acesso. Assim, o *script* de posse de todos os parâmetros fornecidos pelo usuário, acessa o H++, preenche todos os dados, espera o seu processamento e ao final retorna com o PDB gerado no processo. Essa solução mostrou-se viável, porém, deve-se observar

[38] https://www.selenium.dev/documentation/en/webdriver/
[39] https://github.com/mozilla/geckodriver

alguns fatores: primeiro é necessária a conexão com a Internet para que o sistema consiga fazer o acesso à plataforma; a conexão de Internet deve ser rápida e estável, percebeu-se que com Internet instável e que dê picos de acesso no momento da geração dos resultados faz com que o *Selenium* perca o acesso com o *site*, não conseguindo recuperar os resultados; hoje é possível entrar com qualquer um dos dados disponíveis na tela, mas ainda não são recuperados os *logs* do sistema, devido a alguns deles serem disponibilizados na tela, e outros serem grandes e diversos arquivos, o que poderia comprometer ainda mais a questão de erro devido a problemas de Internet; hoje o *script* faz uso do *Firefox Web Browser*, outras plataformas serão disponibilizadas posteriormente.

O H++ gera uma série de arquivos, inclusive arquivos de campo de força e outros já preparados para se fazer a simulação de dinâmica molecular, porém, no *workflow* apenas o arquivo PDB com os hidrogênios interessam. O campo de força e demais arquivos serão preparados por tarefas do ProtCool_Dynamic. O arquivo PDB recuperado do H++ servirá para acertar a protonação das Histidinas, os demais hidrogênios serão inseridos pelo tLeap durante a especificação do campo de força.

Figura 17 – 4EY6 – PDB H++ - *HID* – *HIE* - *HIP*. Na imagem é possível visualizar cada uma das histidinas do 4EY6 em sua devida classificação. Na imagem estão marcados de vermelho os hidrogênios que são levados em consideração para realizar a classificação.



Fonte: Autores.

As histidinas (HIS) são aminoácidos especiais em termos de protonação, dado seu pka próximo de 6.0 em solução. Assim, em pH abaixo de 6.0, há tendência do seu anel imidazol estar todo protonado, tornando-se catiônico. Mas, acima de 6.0, especialmente próximo do pH fisiológico (entre 7.0 e 7.5), um dos dois prótons do imidazol pode ser perdido, fazendo com que algumas HIS possam se apresentar também na forma de 2 tautômeros neutros (SHENHUI; HONG, 2011). Deve-se levar em conta ainda que, no contexto de proteínas, seu pka é dependente do ambiente químico e estrutural em que se encontra, podendo ter variações tão grandes como entre 3 e 9 (EDGCOMB; MURPHY, 2002). O programa H++ tenta lidar com todas essas complexidades em seu *framework* teórico e heurístico. Esses resíduos podem ter os seguintes nomes, segundo o AMBER[40]:

i. HIS – em PDB original;
ii. HID – quando o hidrogênio está presente no nitrogênio delta;
iii. HIE – quando o hidrogênio está presente no nitrogênio épsilon;
iv. HIP – quando hidrogênio em ambos os nitrogênios, isto é, positivamente carregados.

A tarefa acertar protonação será então responsável por recuperar no arquivo fornecido pelo H++ os dados referentes a cada uma das histidinas existentes no arquivo e realizar a verificação dos hidrogênios para se descobrir em qual das situações elas se encontram, realizando a correta denominação de cada uma delas.

A molécula 4EY6 possui ao todo 11 histidinas. Elas foram classificadas como sendo 3 HID (283 – 386 – 404), 4 HIE (211 – 222 – 252 – 286 – 431) e 3 HIP (321 – 386 – 446). A Figura 17 apresenta um compilado de informações. Primeiro existem as imagens de cada uma das moléculas de histidina, onde é possível ver as ligações dos hidrogênios nos anéis, sendo possível perceber a diferença entre cada um dos três tipos nas estruturas dos resíduos.

Também é possível ver na figura, trechos do PDB que veio do H++, separados pelas categorias. Na Figura 17 – A, encontram-se as 3 histidinas que foram identificadas como HID, na Figura 17- B as histidinas identificadas como HIE e na Figura 17 – C as histidinas identificadas como HIP. Em cada um dos trechos foram marcados de vermelho os hidrogênios que devem ser observados para a tomada de decisão de qual será a sua classificação. Sendo assim, as histidinas HID tiveram marcados os hidrogênios HD1, HD2 e HE1. Nas HIE tiveram marcados HE1, HE2 e HD2. E nas HIP possuem os hidrogênios HD1, HD2, HE1 e HE2. Com essas informações é possível realizar então a identificação em cada um dos tipos determinados, acertando o arquivo com essa informação. Posteriormente, na definição do campo de força, o antechamber e o tLeap do AMBER, farão a inserção dos hidrogênios conforme especificado.

Hoje o ProtCool_Dynamic está preparado para tratar a protonação das histidinas. Demais resíduos serão posteriormente implementados na ferramenta.

Outra informação de troca de nomes é trocar as pontes dissulfeto para a nomenclatura do resíduo para “CYX” ao invés de “CYS”, conforme o arquivo original. Sendo assim, nesse *script* são alteradas essas informações no PDB. Para que os arquivos fiquem com a nomenclatura correta, tanto o arquivo de configuração quanto o arquivo de acertos são corrigidos com os nomes corretos das histidinas e da cisteína.

A última tarefa dessa etapa é responsável por agrupar a proteína com cada um dos ligantes, formando um arquivo de complexo para cada um dos ligantes. No caso, como é uma atividade que requer resultado do *docking*, precisam de dois *scripts*, uma para a saída do *docking Vina* e a outra para a saída do *docking Smina*. Uma observação importante é que como o *docking* pode colocar nome padrão no PDB da conformação gerada, os *scripts* já realizam o acerto do nome do ligante, colocando o nome passado pelo pesquisador no arquivo de configuração. Além disso, um arquivo foi gerado para armazenar os dados referentes ao resíduo de início e fim de cada uma das cadeias presentes no PDB e,

[40] http://ambermd.org/Questions/HIS.html

também, do ligante com o qual está se trabalhando. Nesse arquivo são apresentadas as informações de cada um dos arquivos gerados dos ligantes, apresentando os valores dos resíduos das cadeias e do ligante. Esse arquivo será importante para que o pesquisador consiga identificar esses dados nos arquivos gerados posteriormente.

A etapa de campo de força, solvatação e ionização é realizada utilizando-se o AMBER e por isso todas as atividades ficaram agrupadas. É importante ressaltar que nessa etapa está sendo realizada a definição da topologia a partir de um campo de força escolhido pelo pesquisador.

A primeira tarefa é responsável por gerar o campo de força do ligante (topologia do ligante). Nessa etapa com o arquivo mol2 do ligante são criados dois arquivos de cada ligante, o *frcmod* e o arquivo *lib*. Esses dois arquivos são necessários para a entrada de dados do campo de força do complexo. Para realizar essa etapa, o pesquisador fornecerá o campo de força com o qual deseja trabalhar. Por enquanto só foram realizados testes mais apurados para o campo de força *ff99SB*, apesar de todos os campos de força AMBER poderem ser utilizados. Para gerar esses arquivos inicialmente se utiliza o *antechamber* para pegar o arquivo mol2 e converter para o mol2 no formato necessário (com cargas). Após esse passo, é utilizado o *parmchk2* que é responsável por gerar o arquivo *frcmod* e finalmente é utilizado o *tleap* para fazer a geração do arquivo *lib*.

Para executar o *antechamber* (com -c bcc) e o *parmchk2* (com -s gaff) são utilizados os parâmetros padrão do *antechamber* e para a utilização do *tleap*, além de usar o campo de força *ff99SB* também é utilizado o *gaff*. Todos os arquivos de *log* gerados pelo sistema AMBER são armazenados na pasta *log* do pesquisador. Para ter todos os *logs*, inclusive o do passo a passo de todas as execuções realizadas pelo AMBER, dois arquivos de *script* foram gerados, com um apenas para realizar a geração do *log*.

As duas próximas tarefas dessa etapa são interligadas (preparar campo de força complexo e gerar campo de força complexo). Um dos objetivos do ProtCool_Dynamic é que o pesquisador tenha condições de gerar os arquivos de preparação de forma automática, sem intervenção ao longo do processo. Nesse ponto seriam necessárias algumas informações para que se conseguisse gerar os campos de força de forma adequada. Para gerar o campo de força do complexo, sua solvatação e ionização, é necessário saber o campo de força utilizado, a caixa de solvatação (seu tamanho e tipo), as pontes dissulfeto existentes e, também, informar os íons a serem utilizados e a quantidade de íons que deve ser inserida no sistema.

As informações de campo de força e caixa de solvatação são fornecidas pelo pesquisador. No caso da solvatação já é fornecido o nome completo a ser utilizado, com seu tamanho sendo especificado (exemplo, TIP3PBOX 12). As pontes dissulfeto são recuperadas ao longo do processo. Quanto aos íons, é considerado no ProtCool_Dynamic os íons Na+ e Cl- e para saber a quantidade de íons esses valores são calculados de acordo com o volume e a carga do sistema. Assim, para realizar o cálculo da ionização era necessário recuperar a carga do sistema e o seu volume.

Para isso, a tarefa "preparar campo de força" é responsável por gerar as informações de volume e carga do complexo. Isso é conseguido por meio do *tleap*, que consegue gerar os dados necessários. Na verdade, é rodado o processo até a etapa de solvatação, porém apenas é armazenado o *log* do *tleap* que contém essas informações. Assim, o "gerar campo de força do complexo" recupera esses valores do *log* do *tleap* e fazem os cálculos para fazer todo o processo.

A tarefa "preparar campo de força complexo" realiza então a geração do *script* do *tleap* e faz a sua execução para gerar o *log*. A saída principal é esse *log* que será analisado na próxima tarefa para que os dados possam ser calculados. Como essa atividade pode ter uma grande quantidade de ligantes e a execução do *tleap* pode demorar um pouco, o *script* foi realizado em paralelo, conforme a gente já utilizou anteriormente. Sendo assim, são necessários 4 *scripts* para a geração dos dados, dois para o *Vina* e dois para o *Smina*. Um *script* para preparar a chamada paralela e o outro para efetivamente executar a funcionalidade.

A Figura 18 possui um trecho do arquivo de *log* gerado pela tarefa "preparar campo de força proteína". Está marcado de vermelho as informações de volume e carga que são utilizadas na próxima

tarefa “gerar campo de força proteína”. O *script* “gerar campo de força da proteína” recupera esse valor e realiza os cálculos da quantidade de íons Na+ e Cl- que devem ser inseridos. Para realizar esse cálculo, o pesquisador informou a concentração iônica desejada (no arquivo de configuração) e é com relação a ela que os cálculos são efetuados.

A partir do volume recuperado do arquivo de log, ProtCool realiza o cálculo de litros do sistema. Sabendo-se qual é a quantidade de íons e a força iônica desejada pelo usuário, é calculada a quantidade de mol de íons que o sistema deverá inserir e, finalmente, considerando o número de Avogrado, é calculada a quantidade de íons que deve ser inserida no sistema. Para a inserção destes íons, o primeiro passo é verificar a carga do sistema. O sistema deve estar neutro para que a inserção seja realizada de forma correta. Assim, verifica-se a carga do sistema e faz a neutralização, caso seja necessário. Após esta neutralização são inseridos a quantidade de íons de Na+ ou Cl- até a quantidade de íons calculados e posteriormente faz-se a inserção do outro íon (Na+ ou Cl-) até a neutralidade do sistema.

Figura 18 – Arquivo *log* complexo 4EY6GNT - Preparação. Na imagem estão marcados de vermelho os dados que são importantes e que serão avaliados para o cálculo dos valores de ionização do sistema.

```
AMBER General Force Field for organic molecules (Version 1.81, May 2017)
Loading parameters: ./gnt.frcmod
Reading force field modification type file (frcmod)
Reading title:
Remark line goes here
Loading library: ./gnt.lib
Loading PDB file: ./4EY6gnt.pdb
  total atoms in file: 4227
  Leap added 4110 missing atoms according to residue templates:
       4110 H / lone pairs
  Solute vdw bounding box:              63.724 63.120 80.328
  Total bounding box for atom centers:  87.724 87.120 104.328
  Solvent unit box:                     18.774 18.774 18.774
  Total vdw box size:                   90.530 90.423 106.952 angstroms.
  Volume: 875510.116 A^3
  Total mass 450525.286 amu,  Density 0.855 g/cc
  Added 21695 residues.
Total unperturbed charge:  -3.999999
Total perturbed charge:    -3.999999
        Quit

Exiting LEaP: Errors = 0; Warnings = 0; Notes = 0.
```

Fonte: Autores.

A Figura 19 apresenta o *script* usado no *tleap* para de fato gerar o campo de força, solvatação e ionização. Nesse *script*, além da entrada dos dados do complexo, são carregados os dados apenas da proteína e do ligante. Isso é realizado para facilitar a análise do pesquisador. Em algumas análises, por exemplo MMPBSA - *Molecular Mechanics Poisson-Boltzmann Surface Area* (WANG *et al.*, 2018), usada para estimar a energia livre de complexos moleculares, são necessários os arquivos prmtop não apenas do complexo solvatado, mas da proteína e do ligante antes da solvatação. Assim, o *workflow* já faz a geração desses arquivos, evitando que o pesquisador tenha que gerar dados a partir dos arquivos já existentes.

Após executar esse *script*, tem-se o complexo com a topologia definida, sua solvatação e ionização. Conforme ocorre com o *script* de preparação, a geração também é realizada em paralelo, usando, portanto, 4 *scripts*, conforme mencionado anteriormente.

Quando o AMBER (*tleap*) realiza o salvamento do arquivo PDB com os íons e as moléculas de água, ele coloca um nome padrão para o ligante, perdendo a configuração do nome correto. Para prosseguir no ProtCool_Dynamic, a tarefa organizar PDB é responsável por acertar o nome do ligante de acordo com o nome fornecido pelo pesquisador no arquivo de configuração. Apesar do NAMD salvar o arquivo PDB sem a identificação das cadeias, decidiu-se permanecer com o arquivo assim, uma vez que ele possui a identificação de TER (*tag* para terminador de cadeia), além de o arquivo já

está com os resíduos numerados sequencialmente, não comprometendo a identificação das cadeias, mesmo que existam mais de uma cadeia no arquivo. Porém, posteriormente, na análise isso deve ser observado para conseguir fazer as análises adequadamente. Ao finalizar essa tarefa, será executada a última etapa do *workflow*. São dois *scripts*, um para *Vina* e outro para *Smina*.

Figura 19 – Arquivo *script* complexo 4EY6GNT - Geração. *Script* completo do *tleap* para geração da solvatação e ionização do sistema. São salvos arquivos adicionais que poderão auxiliar o pesquisador em análises futuras das dinâmicas.

```
source ~/amber18/dat/leap/cmd/oldff/leaprc.ff99SB
source leaprc.gaff
loadamberparams gnt.frcmod
loadoff gnt.lib
complex = loadpdb 4EY6gnt.pdb
ligand = loadmol2 gnt.mol2
protein = loadpdb 4EY6_7.pdb
bond complex.68.SG complex.95.SG
bond protein.68.SG protein.95.SG
bond complex.256.SG complex.271.SG
bond protein.256.SG protein.271.SG
bond complex.408.SG complex.528.SG
bond protein.408.SG protein.528.SG
saveamberparm ligand gnt_Vina.prmtop gnt_Vina.inpcrd
saveamberparm protein 4EY6_Vina.prmtop 4EY6_Vina.inpcrd
saveamberparm complex 4EY6gntnoSol_Vina.prmtop 4EY6gntnoSol_Vina.inpcrd
solvatebox complex TIP3PBOX 12
saveamberparm complex 4EY6gntSol_Vina.prmtop 4EY6gntSol_Vina.inpcrd
charge complex
addIonsRand complex Na+ 3.999999
addIonsRand complex Na+ 79.0
addIonsRand complex Cl- 0
saveamberparm complex 4EY6gnt_Vina.prmtop 4EY6gnt_Vina.inpcrd
savepdb complex 4EY6gnt_Vina.pdb
quit
```

Fonte: Autores.

Na última etapa do ProtCool_Dynamic existem 5 tarefas que devem ser realizadas. As 3 primeiras tarefas são responsáveis por definir os dados referentes às configurações necessárias da dinâmica NAMD, a sua metodologia, informando como será realizada a minimização e relaxamento do sistema para que ele possa ser colocado em produção. As outras duas tarefas são para organização final, sendo responsáveis por disponibilizar um arquivo TAR com todos os arquivos necessários para executar a dinâmica e o outro para envio de *email* para o pesquisador informando que o processo está finalizado e enviando os *logs* do processo para que ele possa analisar como foi a preparação.

Na Seção 2.4.1.6 foi especificado que para se definir o relaxamento do sistema (complexo-ligante), é importante se definir os passos desse relaxamento, bem como, se definir as restrições harmônicas que serão impostas ao sistema. A primeira tarefa é responsável por essa atividade.

Para se entender melhor como isso é realizado no sistema, tem que se analisar o arquivo de configuração (`confdinamica.txt`) que o pesquisador fornece para o ProtCool_Dynamic. Existe uma parte desse arquivo de configuração que é extremamente importante nessa parte do *workflow*. Essa parte se inicia com `[NumberSimulations]`, esse campo apresenta a quantidade de simulações que serão executadas. Nesse valor estão inclusas todas as etapas do experimento, ou seja, desde a minimização, todas as etapas de relaxamento e a etapa final de produção. Com isso, o ProtCool_Dynamic já realiza a geração de todos os arquivos necessários e gera o *script* final para ser executado na máquina escolhida pelo pesquisador.

Dentro dessa seção do arquivo de configuração existem as N subseções delimitadas por `[Number]` e `[EndNumber]`. Dentro desses dois delimitadores existem os campos necessários para que se gerem tanto os arquivos PDB de restrição do sistema, quanto os arquivos de configuração NAMD gerados. Ou seja, para cada uma das execuções o pesquisador deve informar o valor dos campos para geração dos arquivos de configuração.

Para a tarefa de definir restrições, o campo importante das subseções é o campo [FREE]. Esse campo indica todos os resíduos que serão especificados como livres durante a simulação. Especifica-se os campos livres ao invés dos campos restritos para diminuir a quantidade de informações a serem digitadas. Existem as seguintes possibilidades para definição do campo [FREE]:

i. Água livre – coloca todas as moléculas de água livres na simulação. Usa o valor – Water;
ii. Íons livres – os íons Na+ e Cl- ficam livres durante a simulação. Usa o valor – Ions;
iii. Ligante livre – o ligante será colocado como livre. Usa o valor – Ligand;
iv. Liberar apenas hidrogênios nos ligantes – apenas os hidrogênios do ligante ficam livres e os demais átomos continuam restritos. Usa o valor: LigandH;
v. Cadeia lateral do ligante – toda a cadeia lateral do ligante ficará livre. Usa o valor – SCLigand;
vi. Cadeia lateral da proteína – toda a cadeia lateral da proteína ficará livre. Usa o valor – SCProtein;
vii. Proteína livre – a proteína inteira fica livre. Usa o valor: Protein;
viii. Cadeia da proteína – uma cadeia específica da proteína fica livre. Usa o valor: ProteinChain A;
ix. Cadeia lateral de uma cadeia da proteína – todos os átomos da cadeia lateral de uma das cadeias da proteína ficarão livres. Usa o valor: SCProteinChain A;
x. Cadeia lateral de um dado resíduo – a cadeia lateral daquele resíduo especificado fica livre. Usa o valor - SC TRP A 86;
xi. Resíduo livre – o resíduo especificado fica livre. Usa o valor - TYR A 72;
xii. Liberar N resíduos no início de uma cadeia da proteína – a quantidade de resíduos desejados pelo pesquisador no início da cadeia da proteína estará livre. Usa o valor: ProteinBegin A 5;
xiii. Liberar N resíduos no fim de uma cadeia da proteína – a quantidade de resíduos desejados pelo pesquisador no fim da cadeia da proteína estará livre. Usa o valor: ProteinEnd A 10;
xiv. Tudo livre – todo o PDB fica livre (proteína, ligante, íons e água). Usa o valor: ALL.

Estas definições especificam o que será considerado livre ou o que ficará com restrições harmônicas no sistema, durante a etapa de simulação de dinâmica molecular. Conforme descrito na seção 2.4.1.6, as restrições harmônicas diminuem a mobilidade dos átomos. Assim, uma boa estratégia de relaxamento é aquela em que o complexo inicia com alta restrição harmônica, e que com o passar do tempo, vá diminuindo as restrições, ou seja, libera os átomos para movimentos livres na dinâmica. O campo [Free] determina, assim, os átomos (ou grupo de átomos) que estarão livres durante aquele tempo de simulação.

Existem ao todo 14 possibilidades de seleção para o campo [Free]. Estas seleções foram construídas de forma a possibilitar que grupos diversos no complexo possam ser atendidos para liberação das restrições harmônicas. Foram identificados possíveis grupos que podem ser observados no complexo e que fazem sentido quanto a liberação das restrições, tais como: proteína, ligantes, resíduos específicos, cadeias laterais de resíduos específicos, cadeia da proteína. É comum que pesquisadores também façam a distinção entre liberar todo o resíduo ou apenas a cadeia lateral de determinados resíduos, pois as cadeias laterais normalmente possuem maior mobilidade do que o *backbone* do resíduo. Outra estratégia é a liberação de resíduos no início e final da proteína, também devido à mobilidade destes resíduos.

Essa tarefa de posse desses dados do arquivo de configuração realiza a atualização do arquivo PDB para que ele apresente a última coluna numérica (coluna B – *Beta – Temp Factor*) com o valor 1.00 para átomos restritos e 0.00 para átomos livres. Essa coluna no arquivo de configuração do NAMD deve ser especificada com o valor B para [*Conskcol*].

Uma observação importante a ser considerada é que o pesquisador deve ficar atento no momento de definir os tipos de restrições, pois uma acaba por sobrescrever a outra. Exemplo disso é que se o pesquisador colocar LigandH e Ligand, em uma simulação, o ligante ficará todo livre e não apenas os hidrogênios.

Assim, segue uma hierarquia de acordo com os tipos:

i. Water e Ions – só tem a possibilidade de deixar livre ou não. Não são sobrescritos por nenhum outro a não ser quando coloca a opção ALL.
ii. Ligante – As opções são LigandH, SCLigand e Ligand. Nessa ordem de sobrescrição, lembrando que o SCLigand só ocorrerá no caso de ligantes do tipo peptídico. Sobrescrito quando seleciona ALL.
iii. Resíduo – SC TRP A 86, TRP A 72. Nessa ordem de sobrescrição. Lembrando que se colocar proteína toda livre sobrescreve os dois, porém, se colocar proteína SC TRP A 72 e TRP A 72. A proteína ficará com a cadeia lateral livre mais o resíduo desejado completamente livre. Sobrescrito quando seleciona ALL. Ficar atento ao selecionar junto qualquer opção de cadeia e proteína, pois pode afetar o resultado.
iv. Cadeia da Proteína – SCProteinChain A, ProteinBegin A 5, ProteinEnd A 10, ProteinChain A. Considerando os itens que levam em consideração as cadeias das proteínas esta é a ordem se sobrescição dentro da cadeia. Porém, como são parte da proteína, no caso de selecionar a restrição da proteína inteira, ela é sobrescrita, ou seja, mesmo falando que queria a restrição apenas da cadeia lateral da cadeia A da proteína, se colocar também a opção Protein, toda a proteína ficará livre. Sobrescrito quando seleciona ALL. Ficar atento ao selecionar junto qualquer opção de proteína e resíduo, pois pode afetar o resultado.
v. Proteína – SCProtein, Protein. Nessa ordem de sobrescrição. Sobrescrito quando seleciona ALL. Ficar atento ao selecionar junto qualquer opção de cadeia e resíduo, pois pode afetar o resultado.

Para cada complexo (proteína-ligante) é gerada uma pasta e dentro da pasta os arquivos PDB com as restrições definidas. Os arquivos possuem o nome moleculaligante_N.pdb, onde N é o número da simulação. Ao final do processo, quando o arquivo for selecionado para a pasta simulação, seu nome será alterado, ficando restrict_N.pdb, com N indicando o número da simulação. No arquivo de configuração será considerado o nome restrict_N.pdf. Nesse caso, dois *scripts* foram criados, um para o *Vina* e outro pro *Smina*.

A segunda tarefa dessa etapa é a "buscar centro geométrico". Como o *workflow* tem por princípio, tanto fazer toda a preparação de forma automática, quanto realizar todo o processo de forma a que o pesquisador tenha menos trabalho, os cálculos de PME (*Particle Mesh Ewald*) são realizados pelo sistema. O PME é um método eletrostático usado em condições periódicas de contorno. Para realizar esse cálculo é necessário buscar o Centro Geométrico da molécula. O *workflow* utiliza o VMD para fazer esta busca, gerando o *script* VMD. Na tarefa de geração do arquivo de configuração são calculados os valores de acordo com os dados desse arquivo. O *script* fornece os valores do centro geométrico e de valores de *min* e *max* de x, y e z. Assim, o valor de x, y e z são calculados a partir dos valores de *min* e *max* de x, y e z.

A terceira tarefa é responsável por montar o arquivo de configuração. Conforme descrito anteriormente, a parte do arquivo descrita por [*NumberSimulations*], [*Number*] e [*EndNumber*] apresenta a quantidade de simulações a serem realizadas e as suas características. São desses campos que o *workflow* busca as informações para gerar o arquivo de configuração de cada simulação NAMD. Alguns campos, tais como os campos referentes a nomes de arquivos são preenchidos automaticamente pelo *script*, com os nomes utilizados no *workflow*. Além disso, os dados de PME são calculados pelo *script* para facilitar a tarefa do pesquisador.

Os dados utilizados pelo *script* são os referentes ao centro (*center x*, *center y* e *center z*) e os dados referentes aos valores de x, y e z. Em *cellBasisVector1*, *cellBasisVector2* e *cellBasisVector3* são utilizados os valores de x, y e z e em *cellOrigin* são utilizados os valores do centro. Para os valores de *PMEGridSizeX*, *PMEGridSizeY* e *PMEGridSizeZ* calcula-se um valor elevado ao cubo em que os valores de x, y e z são inferiores ao valor calculado.

Os demais dados do arquivo de configuração são fornecidos pelo pesquisador por meio do arquivo de configuração. Nessa tarefa existem dois *scripts*, um *Vina* e outro *Smina*.

A próxima tarefa "gerar *script* NAMD", é responsável por criar uma pasta com todos os arquivos necessários para que a simulação NAMD seja realizada. Nela estão os seguintes arquivos: PDB, *prmtop*, *inpcrd*, todos os arquivos de restrição, todos os arquivos de configuração e um arquivo que é um *script* que deverá ser executado no servidor em que o pesquisador fará a pesquisa. Sendo assim, basta que o pesquisador copie a pasta para o local desejado e execute o *script* gerado para que a dinâmica seja realizada. Para cada ligante existente no arquivo de configuração, uma pasta separada é disponibilizada.

O arquivo `dinamica.py` gerado é preparado para que seja realizada a dinâmica em triplicata, sendo a preparação (minimização e relaxamento) executada apenas uma única vez, copiada para as demais pastas e então, a simulação em produção (a última simulação programada) é realizada três vezes, cada uma separada devidamente em pastas que são nomeadas como 1Rodada, 2Rodada e 3Rodada. Internamente a cada pasta haverá todos os arquivos de dinâmica e cada arquivo gerado pela dinâmica terá o formato MoleculaLigante_N, onde "Molecula" é o nome da molécula, "Ligante" é o nome do ligante e N é o número da simulação realizada. Exemplo, 4EY6gnt_9.dcd é o arquivo DCD da proteína 4EY6, ligante gnt e é a 9 simulação. Nesse caso, esse era o arquivo em produção, uma vez que essa simulação foi programada para ter 9 passos (1 de minimização, 7 de relaxamento e 1 em produção).

A última tarefa do *workflow* é a de envio de *email*. Essa tarefa é responsável por enviar um *email* ao final do processo para o pesquisador, informando que a execução de todo o processo finalizou. No *email* é informada a pasta onde os arquivos podem ser recuperados e são anexados dois arquivos de *log*. O *log* da execução de todo o *workflow* (`logWorkflow.log`), que mostra passo a passo a execução de todo o sistema. E o *log* de execução do `ProtCool_Dinamica.py` que indica como foi a execução do sistema de chamadas realizadas. Com esses arquivos é possível verificar a situação da execução. Se ocorreram erros e se o *workflow* finalizou gerando todos os arquivos adequadamente.

A preparação dos arquivos de entrada consta apenas da geração do arquivo `confdinamica.txt` e da disponibilização dos arquivos PDB e mol2 apenas nos casos em que o pesquisador quiser usar arquivos próprios. Todos os demais arquivos são gerados em tarefas do ProtCool_Dynamic e um arquivo de saída serve como entrada para o próximo arquivo do *workflow*. O processo ficou completamente automatizado de forma a que o pesquisador tenha que apenas montar o seu arquivo `confdinâmica.txt` e executar o *script* `ProtCool_Dinamica.py`.

Todos os *scripts* foram desenvolvidos checando se os arquivos iniciais necessários estavam presentes, gerando entradas do arquivo `Errorlog.txt` em caso de alguma falha. Além disso, todos os *scripts* possuem tratamento de exceção, o que garante mais uma gestão de erros no sistema.

3.3.*ProtCool_Docking*

O ProtCool_Docking foi preparado para realizar a etapa de *docking* de múltiplos ligantes do *virtual screening*. Para a realização desse *workflow* foram utilizados os *scripts* da parte de *docking* já existente na ProtCool_Dynamic. Porém, alguns ajustes foram necessários principalmente devido ao volume de dados que são processados quando se realiza um *virtual screening*, bem como por, na maioria das vezes, o processo ter que ser realizado em servidores, devido ao grande volume de informações processadas.

Como em *screening* normalmente se trabalha com grandes volumes de dados, torna-se inviável a criação manual de um arquivo de configuração com todos os ligantes. Assim, a primeira atividade é gerar arquivo de configuração geral, que é responsável por, a partir de um arquivo de configuração inicial (`confdinamica.txt`), um novo arquivo de configuração `confGeral.txt` é gerado automaticamente pelo sistema. Para realizar isso, o *script* busca todos os arquivos do tipo pdbqt existentes na pasta de execução do *script* e, separa entre os arquivos lidos aquele que é o pdbqt do receptor dos pdbqt do ligante.

O `confdinamica.txt` é bem mais simples no ProtCool_Docking, uma vez que só precisa de dados para o *docking* e, que os dados dos ligantes serão recuperados automaticamente. O `confdinamica.txt` possui as informações do *pocket* e das configurações definidas para uso do *docking*. Além disso, deve-se também informar se serão realizados *docking Vina* e *Smina* (para o caso de o pesquisador desejar utilizar apenas um deles). E, são informados também dados da linguagem de *log* (*Language*) e dados do receptor (*Protein*). No caso do receptor, não é necessário colocar o nome completo do receptor, mas um nome que o diferencia do nome dos arquivos pdbqt do ligante. Além disso, pode-se fazer tanto o *docking* rígido quanto o *docking* flexível, e se definir a quantidade de rodadas de *docking* que serão realizadas. A quantidade de *dockings* nesse caso será global, ou seja, valerá para todos os ligantes.

A segunda tarefa é a de gerar arquivo de configuração *docking*. Essa atividade não sofreu muita modificação com relação à tarefa existente no ProtCool_Dynamic. As mudanças foram mais em relação ao arquivo de configuração usado que passou a ser o `confGeral.txt` e em relação à pasta de acesso do arquivo. Uma pasta *ligandconf* é criada para armazenar todos os arquivos de configuração de *docking* criados nesse *script*.

A terceira atividade é a geração de proteínas flex. Ela permite que o *docking* de múltiplos ligantes seja realizado em receptores flexíveis, o que amplia muito o poder da ProtCool_Docking.

A quarta atividade é a realização do *docking* propriamente dito. Só foram realizadas as alterações no *script* para que pudesse considerar o `confGeral.txt`. Além disso, a *tag CPUCluster* foi utilizada para que o pesquisador possa definir a quantidade de processadores a serem utilizados ao invés do sistema pegar todos os núcleos da máquina, o que é importante em relação a servidores. Isso garante que um mesmo servidor possa ser utilizado para processar diversas tarefas ao mesmo tempo. Dependendo do número de núcleos da máquina e da forma como a máquina é compartilhada, servidores podem rodar mais de um processo ao mesmo tempo. Poder selecionar para o ProtCool quantos servidores estão disponíveis é uma configuração interessante. Outra modificação é que o número de rodadas de *docking* é apenas uma entrada no arquivo de configuração e não mais uma entrada para cada ligante. A quinta atividade de separar poses, também só teve as alterações para considerar o `confGeral.txt` e o `DocNum`.

A quinta atividade gerar mol2 é responsável por gerar o arquivo mol2 das poses obtidas. Essa funcionalidade não existia e foi desenvolvida. Existem dois *scripts*, um para o *Vina* e outro para o *Smina*. Foi utilizado a API *open babel* para realizar a conversão.

Finalmente a última atividade é a que recupera os *scores* dos *dockings* realizados, gerando um único arquivo TXT com todos os ligantes especificados. Existe um *script* para cada *docking* (*Vina* e *Smina*). Os dados são fornecidos conforme exemplo da Figura 20. Nela é possível verificar alguns resultados da base DB, tanto no *Smina* quanto no *Vina*.

Ao final de todo o processo, o pesquisador terá disponível todos os *logs* gerados pelo sistema, bem como todos os arquivos gerados ao longo do processo, tendo todas as poses tanto em pdbqt quanto em mol2 e o arquivo com os dados de *scores* de cada conformação alcançada. Em processos de *screening* esse arquivo com os *scores* é importante, pois ele será uma das medidas a serem consideradas para avaliação dos melhores ligantes do conjunto. Muitas vezes na verificação de melhores poses e na geração de consenso em VS, o *score* alcançado por cada pose é um dos parâmetros utilizados. A ProtCool_Docking não realiza a análise dos *scores*, apenas gera os arquivos para que etapas posteriores de *VS* possam realizar essa tarefa.

Figura 20 – Exemplo arquivo *scores* - ProtCool_Docking. Na imagem estão os exemplos dos dados gerados para uma base DB. Primeira tabela são os resultados do *Smina* e na Segunda são os resultados do *Vina*.

poses	scoreSmina
DB07978_5	-7.61454916
DB03024_6	-3.90049601
DB14013_9	-5.5461545
DB09189_8	-4.77050734
DB03869_5	-7.1114459
DB11807_8	-3.52930427
DB02650_7	-2.62530255
DB13212_6	-5.87798119
DB07028_9	-5.19035721
DB02420_1	-5.40125179
DB06261_3	-3.66488194

poses	scoreVina
DB07978_5	-6.9
DB03024_6	-3.8
DB14013_9	-5.6
DB09189_8	-5.0
DB03869_5	-6.4
DB11807_8	-3.6
DB02650_7	-2.5
DB13212_6	-5.5
DB07028_9	-5.2
DB02420_1	-5.2
DB06261_3	-3.8

Fonte: Autores.

4. ACETILCOLINESTERASE HUMANA

4.1.Doença de Alzheimer

Descoberta em 1906 pelo psiquiatra alemão Alois Alzheimer (DE MEDEIROS FILHO *et al.*, 2020), a doença de *Alzheimer* (DA) é uma doença degenerativa que traz grande impacto tanto para o paciente quanto para seus familiares e amigos. Isso ocorre devido à característica da DA que modifica a personalidade do paciente (ROCHA, 2017). A DA possui como características a perda progressiva neural (INESTROSA *et al.*, 1996).

O paciente com DA passa por alguns estágios, sendo a doença progressiva. Nos primeiros estágios, o paciente apresenta cansaço, depressão, mudança de humor. Já em estágios mais avançados se verifica a presença de alucinações, dificuldade para falar, dentre outros sintomas (ROCHA, 2017; SERENIKI, VITAL, 2008; BORGES *et al.*, 2018). O primeiro aspecto clínico observado é a perda de memória recente, sendo que as memórias antigas continuam preservadas (SERENIKI; VITAL, 2008; BORGES *et al.*, 2018). Existe posteriormente a dificuldade de atenção, problemas na fala, deterioração das funções cognitivas, da capacidade de realização de cálculos e deterioração das habilidades visuais espaciais e de se usar objetos comuns e ferramentas do dia a dia (SERENIKI; VITAL, 2008; BORGES *et al.*, 2018).

Uma característica que se observa em autópsias realizadas na região cerebral de pacientes com a DA, é que existe uma grande quantidade de placas *senis* (amiloides) emaranhadas com o tecido neural (neurofibrilas). Um acúmulo de filamentos anormais da proteína tau com a formação de novelos neurofibrilares (NTF) (ROCHA, 2017; SERENIKI; VITAL, 2008; LONG; HOLTZMAN, 2019; INESTROSA, 1996). São a presença dessas placas amiloides e da proteína tau que são classificadas como a principal causa da progressão da DA e do aparecimento dos sintomas em pacientes (PICANÇO, 2018). O acúmulo de placas amiloides parece ser a chave para o início da DA, porém, eventos como a neuroinflamação e o acúmulo da proteína tau acabam por serem identificados como os principais responsáveis pela neurodegeneração (LONG; HOLTZMAN, 2019).

Existe, além da deposição das placas *senis* e das neurofibrilas, uma atrofia cerebral, principalmente na região do hipocampo (HUANG; CHAO; HU, 2020). Uma informação importante é que a deposição cerebral das fibrilas Aβ pode ocorrer décadas antes de o paciente apresentar os sintomas clínicos da DA (HUANG; CHAO; HU, 2020; LONG; HOLTZMAN, 2019).

A hipótese colinérgica é um dos mecanismos bioquímicos associado à neurodegeneração. Nessa hipótese, o sistema colinérgico de neurotransmissão possui um lapso que ocorre devido à senescência e deposição de placas amiloides nos neurônios. A acetilcolina (Ach, *acetylcholine*) é um neurotransmissor, que é sintetizado no citoplasma dos neurônios pela ação da colina-O-acetilcolinesterase, utilizando o substrato acetil-coenzima A (Acetil-CoA) e a colina que estão disponíveis no sangue (ROCHA, 2017). Com a produção da Ach, ela é excretada na fenda sináptica e interage com os receptores colinérgicos, o que gera um sinal colinérgico entre um par de neurônios. Com a ação da acetilcolinesterase (AchE, *acetylcholinesterase*) a Ach é hidrolisada em colina e Acetil-CoA (ROCHA, 2017). O que se observa é que uma baixa concentração de Ach é encontrada em pacientes que possuem quadros da DA (ROCHA, 2017; BORGES *et al.*, 2018). A AchE é responsável por regular, dessa forma, a concentração da acetilcolina quando ocorre uma transmissão do sinal nervoso (WIESNER *et al.*, 2007; BORGES *et al.*, 2018).

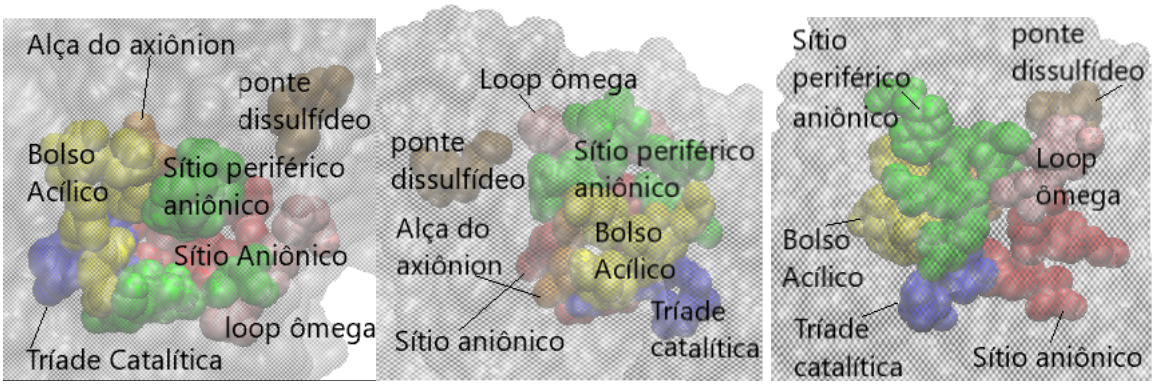
Com isso, existem ações que buscam aumentar a concentração de Ach na fenda sináptica e uma forma de se fazer isso é inibir a AchE, diminuindo assim, a velocidade de degradação da Ach. Essa estratégia busca diminuir os sintomas da DA (ROCHA, 2017).

A AchE é da família enzimática serino-hidrolase, pertencente às colinesterases. Encontrada no tecido neural, periférico, nervos e músculos, é importante para o funcionamento do sistema nervoso, com maior atividade em neurônios motores (ROCHA, 2017). Possui estrutura globular dimérica, com 60 kDa, tamanho de 60x70x45 Å por monômero. Seu sítio ativo é estreito e possui profundidade de

20 Å, com grande quantidade de resíduos aromáticos (ROCHA, 2017; BORGES, 2017). Os resíduos TYR337, TRP286 e TRP86 são os três resíduos que mais interagem com os ligantes (BORGES, 2017). Segundo Wiesner *et al*. (2007) a AchE tem um sítio ativo bem descrito na literatura, mas ainda não se sabe todas as funções desempenhadas por todos os resíduos no sítio ativo. É delimitada por diversos subsítios (WIESNER *et al*., 2007), que podem ser visualizadas pela Figura 21:

i. Tríade catalítica (sítio esterástico) especificadas pelos resíduos SER203, HIS447 e GLU334;
ii. Sítio aniônico delimitado pelos resíduos TRP86, TYR133, GLU202, GLY448 e ILE451;
iii. Alça do oxiânion (orifício do oxiânion) formado pelos resíduos GLY121, GLY122 e ALA 204;
iv. Bolso acílico delimitado pelos resíduos PHE295, PHE297, TRP236 e PHE338;
v. Sítio periférico aniônico delimitado pelos resíduos ASP74, TYR124, SER125, TRP286, TYR337, TYR341;
vi. Loop ômega delimitado pelos resíduos THR83, ASN87 e PRO88 que é ligada por uma ponte dissulfeto formada por CYS69 e CYS96.

Figura 21 – 4EY6 – Destaque para o sítio ativo. Apresentação da tríade catalítica (azul); sítio aniônico (vermelho); alça do axiônion (laranja); bolso acílico em amarelo; sítio periférico aniônico em verde; *loop* ômega em rosa; e a ponte dissulfídeo em marrom.



Fonte: Autores.

O sítio aniônico é importante para a atividade catalítica da enzima. O resíduo TRP86 é preservado em organismos diferentes e em outras proteínas da família de colinesterases, o que sugere a sua importância para reconhecimento de substratos e inibidores da AchE (Rocha, 2017). Na hidrólise da Ach, os resíduos aromáticos do sítio periférico aniônico reconhecem o neurotransmissor (Ach). A Ach vai para o fundo do sítio ativo, onde está a tríade catalítica e é onde ocorre a hidrólise da Ach pela AchE (BORGES, 2017). Os inibidores de AchE são, portanto, ligados no final do sítio ativo da AchE, impedindo que ocorra a hidrólise da Ach

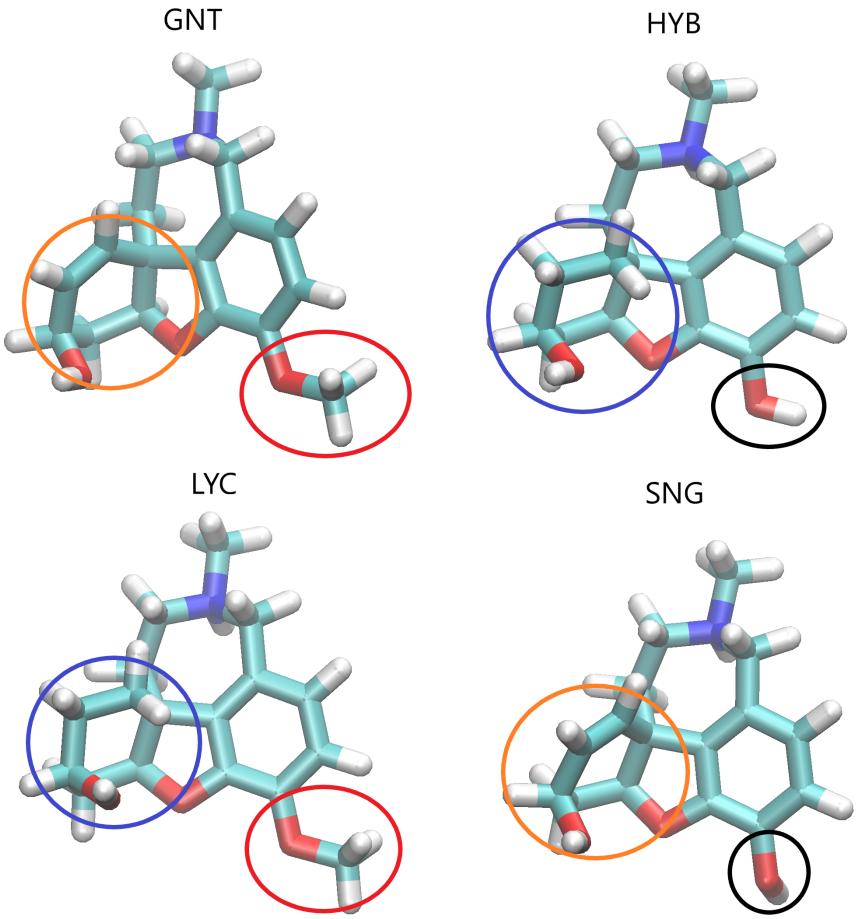
Devido à quantidade de fenilalanina, histidina e triptofano no sítio ativo, possibilita a utilização de compostos aromáticos para inibição da AchE, o que condiz com os compostos que são utilizados como tratamentos da DA (ROCHA, 2017). Exemplos de compostos utilizados como medicamentos da DA são: donepezil, rivastigmina e galantamina.

Esses compostos são inibidores seletivos da AchE, alterando a sua função colinérgica central. Isto é realizado inibindo as enzimas que degradam a acetilcolina, aumentando a capacidade da acetilcolina de estimular os receptores nicotínicos e muscarínicos cerebrais (SERENIKI; VITAL, 2008). Os inibidores de AchE acabam por ser o tratamento escolhido para os sintomas da DA.

Donepezil tem um efeito de redução de 38% no declínio funcional dos pacientes de DA quando utilizado como tratamento por 1 ano (SERENIKI; VITAL, 2008). Foi o segundo fármaco aprovado pela FDA, alcançando melhores resultados que a tacrina, possuindo baixa toxicidade e alta seletividade para a AchE (PICANÇO, 2018). Interage tanto com o sítio ativo quanto com o sítio aniônico periférico da AchE (PICANÇO, 2018).

Rivastigmina é um medicamento muito utilizado para o tratamento de DA. É um inibidor de acetilcolinesterase e butirilcolinesterase, sendo muito eficaz no aumento dos níveis de acetilcolina (PICANÇO, 2018). Porém, apresenta efeitos colinérgicos adversos, gastrointestinais e aumento de peso (SERENIKI; VITAL, 2008). É um composto sintético que foi derivado da fisostigmina, sendo mais seguro que o seu anterior (PICANÇO, 2018).

Figura 22 – Ligantes em formato PDB – saída final da etapa de Preparação – ProtCool_Ligand. O círculo preto identifica a ligação hidroxi (HYB e SNG), o círculo vermelho a ligação metoxi (GNT e LYC), o círculo azul, a ausência de ligação dupla no anel (HYB e LYC) e o círculo laranja destaca a presença de ligação dupla no anel (GNT e SNG).



Fonte: Autores.

A galantamina (GNT) é considerado um inibidor da AchE com IC50 de 1µM (ROCHA, 2017) (Figura 22– GNT). A galantamina possui duplo mecanismo de ação, pois inibe a acetilcolinesterase e modula alostericamente os receptores nicotínicos. A ligação da galantamina com os receptores nicotínicos mostrou melhora nas funções cognitivas e na memória de pacientes (SERENIKI; VITAL, 2008). É um inibidor reversível da AchE, sendo menos potente que a tacrina, porém, menos tóxico (PICANÇO, 2018). Diversos estudos são realizados de fármacos derivados deste composto

(PICANÇO, 2018). A mudança da ligação dupla para simples, acaba por modificar a geometria de planar para tetraédrica, o que promove uma projeção do C6 na direção da cadeia lateral do resíduo TRP86 (ROCHA, 2017).

Outros galantamínicos também possuem atividade inibitória, tais como sanguinina (SNG) com IC50 de 0,1 μM, 10 vezes maior que a GNT, devido à substituição do grupo metóxi por um grupo hidroxil (Figura 22 – SNG). Licoramina (LYC - Lycoramine) que possui capacidade inibitória de cerca de 20 vezes menor que a GNT devido à retirada da ligação π presente no anel não aromático de seis membros (ROCHA, 2017) (Figura 22 – LYC). Outro composto a ser considerado é um composto híbrido, chamado de HYB, modelado computacionalmente em Rocha (2017) por meio da junção de LYC com SNG (Figura 22 – HYB).

4.2.Metodologia

O objetivo dos estudos de caso é mostrar como a ferramenta poderia ser utilizada para gerar os arquivos do *workflow* escolhido, de forma a mostrar as possibilidades que ela fornece. Ou seja, mostrar como, a partir dos dados gerados pela ferramenta, o pesquisador pode avaliar se a preparação está adequada à pesquisa que se deseja realizar.

Assim, o primeiro estudo de caso partiu do trabalho de Rocha (2017) e mostra como ele poderia usar a ferramenta para gerar os arquivos da dinâmica e, como poderia trazer maior agilidade à sua pesquisa.

Esse estudo de caso envolve a proteína acetilcolinesterase humana com o objetivo de mostrar a utilização do *workflow* da dinâmica. Serão apresentados aqui como realizar a utilização do ProtCool_Dynamic, mostrando como definir o arquivo de configuração para a utilização do *wokflow*. Após isso, será realizada uma análise da preparação e posteriormente serão realizadas análises em cima das dinâmicas efetuadas, com o objetivo apenas de demonstrar que a preparação de fato permitiu a simulação de dinâmicas computacionais confiáveis.

Os ligantes utilizados na pesquisa são: Galantamina, Licoramina, Sanguinina e o híbrido (mistura de Sanguinina com Licoramina). A metodologia aqui utilizada foi baseada no trabalho de Rocha (2017):

i. A molécula de acetilcolinesterase foi recuperada do RSCB, tendo como PDBID o valor 4EY6;
ii. As moléculas foram modeladas usando o Modeller – versão 9, foi utilizado o protocolo padrão do sistema, com modelagem por homologia usando como modelos as estruturas com sequências semelhantes (KUNTAL *et al.*, 2010);
iii. Foi realizada a protonação com o software H++, assumindo condições fisiológicas de salinidade (0.15 M) e pH 7.0, com permissividade dielétrica relativa internas e externas de 10 e 80 respectivamente (ANANDAKRISHNAN; AGUILAR; ONUFRIEV, 2012);
iv. Os estados de protonação de cada histidina foram verificados e ajsutados;
v. A carga dos átomos dos ligantes foi calculada por AM1-BCC;
vi. Campo de força AMBER 99 (ff99sb) e GAFF (*General Amber Force Field)*, usando o ANTECHAMBER e Tleap do AMBERTools (SALOMON-FERRER, 2013);
vii. O *docking* foi realizado em triplicata usando o AutodocVina e Smina, com os resíduos TRP86, TYR124, SER203, TYR337 e HIS447 como flexíveis e o *pocket* utilizado abrangia toda a cavidade catalítica e foram solicitadas 10 conformações de cada um dos ligantes. Demais parâmetros foram definidos com o padrão do software;
viii. Na solvatação foi usado o modelo TIP3, em caixa cúbica de 12 Å a partir dos átomos mais externos da proteína usando para isso o *AMBER*;
ix. A força iônica foi ajustada para 0.15 M de NaCl após neutralização;

x. Para a simulação foi utilizado o campo de força AMBER 99, usando o software NAMD, com passo de integração de 2 fs (PHILLIPS *et al*., 2005; PHILLIPS *et al*., 2020);
xi. Foi realizada a transformação termodinâmica sob condições NPT, temperatura e pressão sendo controladas segundo algoritmo de *Langevin* em 300 K e 1atm.
xii. Foram utilizadas condições periódicas de contorno e corte de interações eletrostáticas de 12 Å para interações não ligadas;
xiii. O protocolo de relaxamento do complexo seguiu os seguintes passos:
    a. 1000 passos de minimização;
    b. 200 ps com restrição harmônica para todos os átomos da proteína e ligante;
    c. 200 ps com restrição harmônica para todos os átomos da proteína;
    d. 300 ps com restrição harmônica para os átomos das proteínas, exceto para as cadeias laterais dos resíduos próximos a 5Å do ligante;
    e. 300 ps com restrição harmônica para todos os átomos da proteína, exceto para as cadeias laterais dos resíduos próximos a 10 Å do ligante;
    f. 300 ps com restrição harmônica de todos os átomos da proteína, exceto próximos a 10 Å do ligante (incluindo o *backbone*);
    g. 300 ps com restrição harmônica apenas para o *backbone* dos resíduos a uma distância maior que 10 Å do ligante;
    h. 4000 ps de pré-produção sem restrições;
    i. 20 ns sobre as coordenadas finais do relaxamento sob condições NPT.

Importante destacar que o protocolo acima foi executado com o sistema ProtCool, definindo para isso os parâmetros acima no arquivo de configuração. Nas próximas seções será descrita a preparação do arquivo de configuração e depois uma análise dos resultados da dinâmica molecular executada.

Abaixo são apresentados os passos necessários para a geração do arquivo de configuração e, também, da execução do ProtCool_Dynamic. A primeira etapa do processo é a de preparação. Nessa etapa algumas informações importantes relacionadas à proteína com a qual se está trabalhando são importantes. Mesmo que o *workflow* já realize a busca pela proteína no *site* RCSB, é importante que o pesquisador conheça a proteína com a qual está trabalhando para conseguir efetuar as escolhas adequadas no *workflow*.

Pelo RCSB, a 4EY6 é uma acetilcolinesterase humana com a presença do ligante cristalográfico galantamina. Foi submetida em maio de 2012 e possui uma atualização realizada em outubro de 2012. Foi recuperada pelo método de difração de raio X, o que indica que o seu PDB não possui as informações de MODEL. Tem uma resolução de 2.4 Å, o que indica uma boa qualidade. É formada por duas cadeias A e B. Pelo arquivo fasta observa-se que as duas cadeias são similares. Não possui mutações identificadas em sua estrutura. Além da GNT (presente nas duas cadeias), possui outros ligantes na sua estrutura, identificados por:

i. PE8 (3,6,9,12,15,18,21-HEPTAOXATRICOSANE-1,23-DIOL na cadeia B);
ii. NAG (3,6,9,12,15,18,21-HEPTAOXATRICOSANE-1,23-DIOL na cadeia B);
iii. EDO (1,2-ETHANEDIOL nas duas cadeias);
iv. NO3 (NITRATE ION na cadeia B).

Verificando o arquivo PDB, observa-se que existem 3 pontes dissulfeto. Neste trabalho será utilizada apenas a cadeia A e sem qualquer dos ligantes da sua estrutura cristalográfica, conforme se verá adiante.

De posse dessas informações é possível realizar a criação do arquivo de configuração que deve ser fornecido pelo pesquisador. Além do arquivo de configuração, é importante a inserção de demais arquivos iniciais. No caso deste trabalho, foi usado um arquivo mol2 já fornecido pelo pesquisador, que foi a molécula modelada do híbrido, conforme descrito em 4.2. Esse arquivo será disponibilizado

na pasta do usuário (pasta de armazenamento). Esses são os dados necessários para que o `ProtCool_Dinamica.py` seja executado.

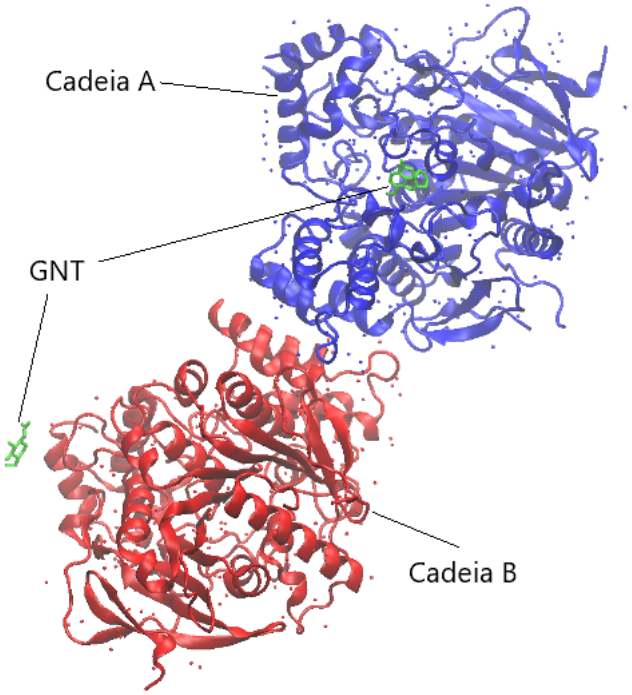
A primeira definição a ser realizada no arquivo de configuração é a que mostra os passos a serem seguidos pelo *workflow*. Como aqui o objetivo é o de preparar uma dinâmica inteira, os campos de passos da dinâmica estão todos definidos com "Yes". Nele são informados o nome e e-mail do pesquisador, a linguagem a ser utilizada e quais passos da dinâmica serão realizados. O sistema é preparado para fornecer todas as mensagens em português e inglês. Assim, o pesquisador pode realizar a escolha da linguagem desejada.

Para realização da primeira etapa os parâmetros importantes a serem observados são: PDBID, cadeia a ser trabalhada, se vai buscar direto o arquivo no *site* ou fornecer o PDB, se serão utilizados hidrogênios e água da estrutura cristalográfica, se serão utilizadas as pontes dissulfeto fornecidos pelo PDB ou se outras pontes serão apresentadas.

No caso deste trabalho, o PDBID será 4EY6. Como as duas cadeias são similares, conforme apresentado anteriormente, escolhe-se uma delas, no caso deste trabalho foi escolhida a cadeia A. Para evitar problemas em troca de arquivos, ele será recuperado direto do *site*. Não serão considerados os hidrogênios e águas cristalográficas para ficar condizente com o experimento realizado por Rocha (2017).

A Figura 23 possui a imagem da estrutura cristalográfica da proteína recuperada do RCSB. A imagem tem duas cadeias sendo apresentadas. A azul é a cadeia A e a vermelha é a cadeia B. Em verde está o GNT cristalográfico da estrutura. As moléculas de água cristalográficas também foram apresentadas. Pela imagem pode-se perceber que a GNT está no sítio ativo da cadeia A da proteína, o que indica que esta cadeia é uma melhor opção para que seja realizada a pesquisa, já que possui a molécula cristalográfica da GNT.

Figura 23 – 4EY6 – Molécula Cristalográfica - VMD. Cadeia A em azul, cadeia B em vermelho e em verde a galantamina cristalográfica.



Fonte: Autores.

Para decidir a respeito das pontes dissulfeto, observa-se o trecho do arquivo PDB da Figura 24. Nesse trecho está a relação das pontes dissulfeto existentes no arquivo. A Figura 25 possui uma

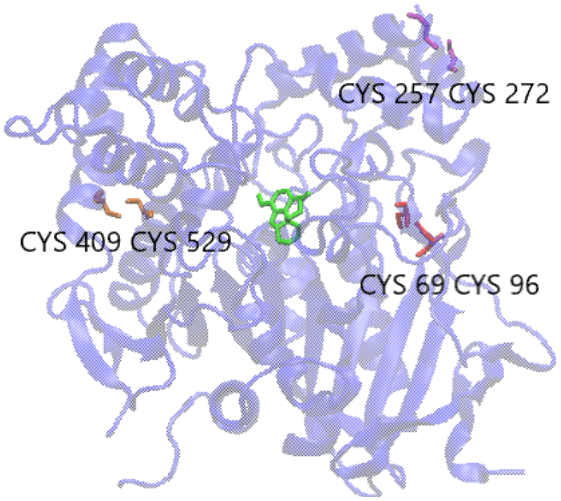
imagem extraída do VMD que apresenta a cadeia A da proteína com os resíduos CYS constantes na listagem selecionados.

Figura 24 – Trecho do arquivo PDB 4EY6 com as pontes dissulfeto. Nesse trecho está a parte de observações do arquivo PDB, onde são listadas as pontes dissulfeto existentes na estrutura cristalográfica.

```
SSBOND   1 CYS A   69    CYS A   96                          1555   1555  2.06
SSBOND   2 CYS A  257    CYS A  272                          1555   1555  2.06
SSBOND   3 CYS A  409    CYS A  529                          1555   1555  2.04
SSBOND   4 CYS B   69    CYS B   96                          1555   1555  2.07
SSBOND   5 CYS B  257    CYS B  272                          1555   1555  2.06
SSBOND   6 CYS B  409    CYS B  529                          1555   1555  2.04
```

Fonte: Autores.

Figura 25 – Pontes dissulfeto – 4EY6. Na imagem são apresentados em destaque os seis resíduos CYS existentes na cadeia A da proteína. Observa-se 3 possíveis pontes dissulfeto com esses resíduos.



Fonte: Autores.

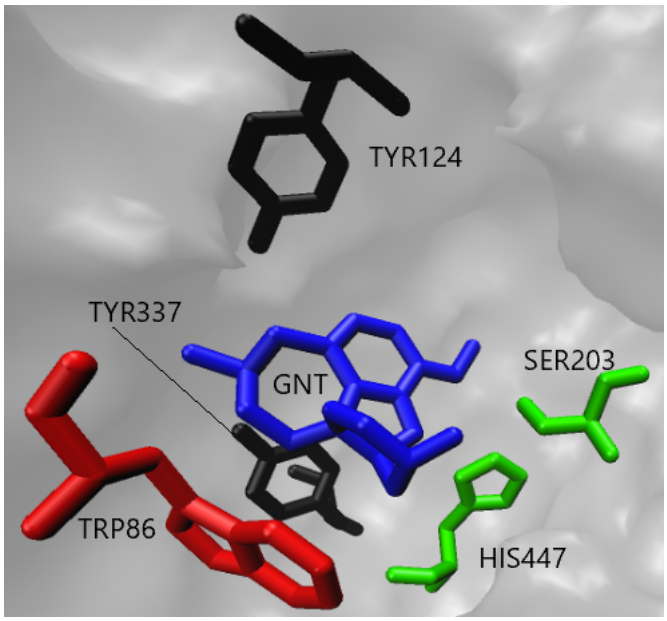
Os resíduos apresentados estão em cores diferentes, de acordo com os pares acima descritos para que possam ser analisados se de fato têm condições de representarem pontes dissulfeto na estrutura. Nenhum outro resíduo CYS foi encontrado na cadeia A. Pela imagem apresentada chega-se à conclusão que de fato a relação apresentada pelo PDB está condizente com três pontes dissulfeto, sendo assim, pode-se optar por realizar a entrada das pontes pelo próprio *workflow*, sem ser informado manualmente no arquivo de configuração.

Para a realização do *docking* é necessária a definição de diversos parâmetros. A primeira definição é em relação à ferramenta de *docking* que será utilizada no trabalho. O *workflow* apresenta duas ferramentas de *docking* a serem consideradas, o *Vina* e o *Smina*. Na pesquisa de Rocha (2017) foi realizado o atracamento com o *Vina*. Porém, como no ProtCool_Dynamic é possível gerar os dois tipos de *docking*, e, mesmo realizando os dois não é necessário muito tempo de máquina, neste trabalho foi realizado os dois tipos de atracamento para as moléculas.

Outra definição necessária é a respeito dos parâmetros do *docking*. A primeira definição é se será utilizado o *docking* rígido ou flexível. No caso da pesquisa de Rocha (2017), foi realizado o *docking* com os seguintes resíduos flexíveis: TRP86, TYR124, SER203, TYR337 e HIS447. Sendo assim, aqui também se optou por fazer o *docking* com os mesmos resíduos flexíveis.

Na Figura 26 estão destacados os 5 resíduos que Rocha (2017) colocou como flexíveis, juntamente com o ligante cristalográfico. O Ligante está em azul na imagem. Em vermelho tem-se o TRP86, conforme verificado na Seção 4.1, esse resíduo faz parte do sítio aniônico da proteína. Além disso, esse resíduo fica na parte de baixo do sítio de ligação e representa uma ligação importante com a GNT. Os anéis do triptofano formam uma ligação importante com um dos anéis da GNT, garantindo que a GNT permaneça ligada no sítio. Em verde estão a SER203 e HIS447, que fazem parte da tríade catalítica do sítio de ligação. E em preto estão o TYR124 e TYR337 que fazem parte do sítio periférico aniônico.

Figura 26 – 4EY6 com resíduos flexíveis do *docking* destacados. Em azul está o ligante cristalográfico galantamina, em vermelho o TRP86 (sítio aniônico), em verde SER203 e HIS447 (tríade catalítica), em preto o TYR124, e TYR337 (sítio periférico aniônico).



Fonte: Autores.

Todos os resíduos escolhidos para que sejam flexíveis fazem parte de pontos importantes do sítio de ligação e estão na parte em que a GNT cristalográfica se encontra. Vale ressaltar que o sítio ativo da 4EY6 é um sítio estreito e profundo, sendo que a galantamina, quando devidamente ligada ao sítio catalítico, fica posicionada na porção inferior da bolsa formada. Esses resíduos escolhidos, ficam nesse mesmo local, englobando o ligante. Sendo assim, de fato, colocar esses resíduos como flexíveis pode auxiliar no correto atracamento do ligante com a proteína. Portanto, serão utilizados os mesmos resíduos utilizados por Rocha (2017) como flexíveis no *docking*.

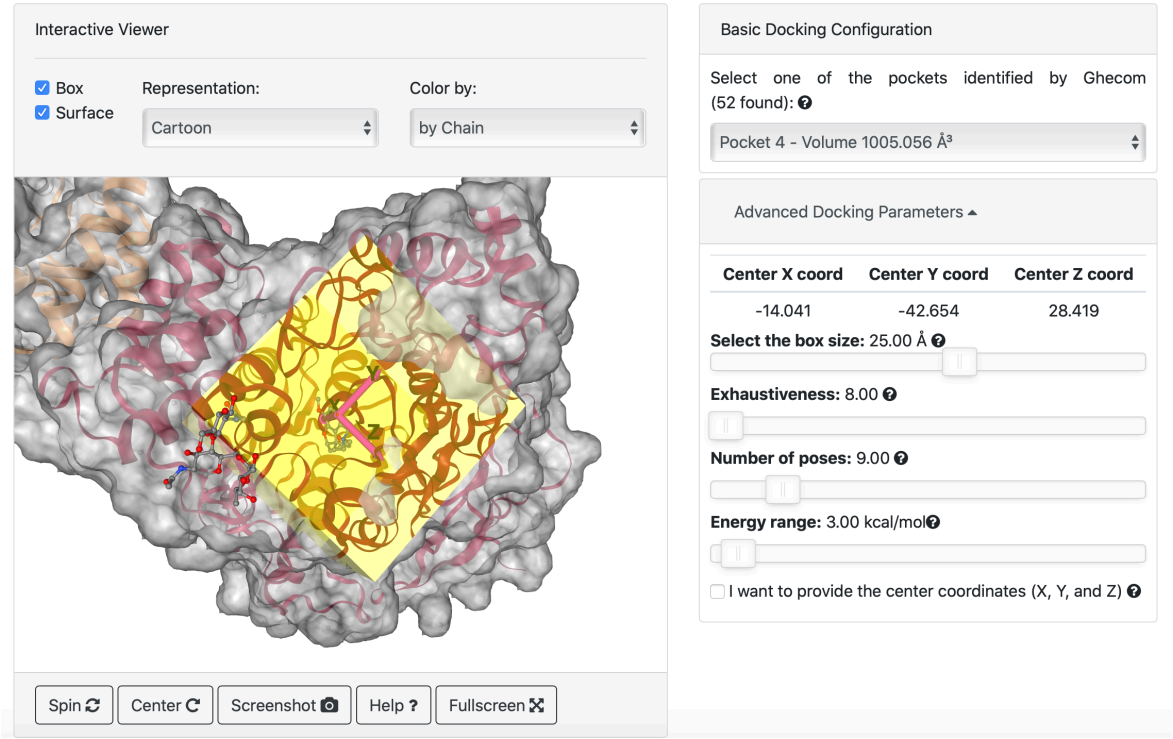
Outro ponto importante de escolha no momento de realizar um *docking*, consiste na definição da caixa de solvatação. É preciso definir o seu posicionamento, valores x, y e z, bem como a dimensão da caixa. Rocha (2017) não cita os parâmetros de caixa que foram utilizados na pesquisa, com isso, fez-se necessário que fosse definido o *pocket* com o qual a preparação será realizada.

Para se conseguir realizar essa análise, foi utilizado o software EasyVS, desenvolvido no grupo de pesquisa deste trabalho. O EasyVS é um software que realiza a triagem virtual de ligantes integrado ao *docking*. Porém, durante a sua preparação, existe uma *interface* em que é possível realizar a visualização dos possíveis *pockets* que a ferramenta sugere que sejam utilizados. A EasyVS[41], após a

[41] A tela de visualização e seleção do pocket pode ser visualizada no link: http://biosig.unimelb.edu.au/easyvs/step2/FC70C8AE250B466EB518673B53EBFB1B

inserção do alvo, apresentou 52 possíveis caixas de docagem. A Figura 27 apresenta a tela do EasyVs que permite a configuração do *pocket*. Nessa tela, pode-se verificar a seleção do quarto *pocket* encontrado. Foram analisadas as 52 possibilidades de *pockets* e verificou-se que 2 seriam mais promissoras para que a docagem fosse realizada de forma eficiente. As duas possibilidades foram o *pocket* 4 e o *pocket* 39. Ao lado da tela é possível fazer algumas configurações básicas para o sistema EasyVs, e que nos auxiliam nesse momento. A primeira informação da parte de parâmetros avançados estão os dados das coordenadas x, y e z que serão utilizadas. Outra informação importante é o tamanho da caixa que pode ser alterado no sistema e permite a visualização de como ficaria na molécula. Além disso, também é possível entrar com valores próprios das coordenadas x, y e z e o programa apresenta como ficariam essas informações.

Figura 27 – Tela de configuração do *pocket* de atracamento - EasyVs. É possível ver na imagem à esquerda a molécula com a caixa do *pocket* destacada e à direita o local onde é possível se configurar os itens selecionáveis. No caso, é apresentado o *pocket* de número 4.



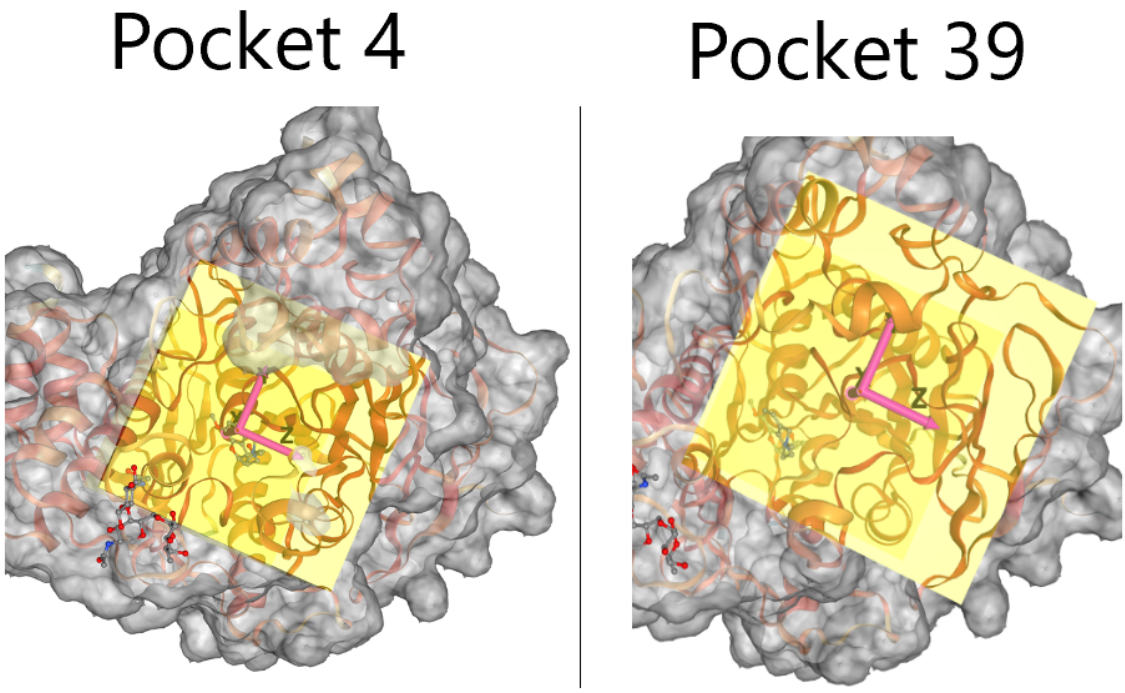
Fonte: Autores.

Na Figura 28 pode-se visualizar imagens dos dois *pockets* em maiores detalhes. O cubo em amarelo mostra o local em que será considerado o *pocket* de atracamento do *docking*. Observa-se que nos dois a estrutura cristalográfica da GNT está de fato dentro do *pocket* a ser considerado, demonstrando que o *docking* conseguirá identificar o local de docagem. Mas pode-se visualizar que no *pocket* 4 a GNT fica melhor posicionada que na *pocket* 39, ficando mais centralizada, o que indica que o *pocket* esta englobando melhor o sítio ativo da proteína.

Quando é carregado, o tamanho da caixa especificado pelo EasyVs foi de 20Å, porém foi setado o valor para 25Å, sendo esse o valor considerado. No caso do EasyVS, a dimensão nas três direções é a mesma, mas no ProtCool_Dynamic, pode-se entrar com dimensões diferentes para cada eixo. Como esse dimensionamento da caixa igual em todos os eixos foi condizente, foi utilizado o valor de dimensão igual para as três posições. Assim, ficou determinado os parâmetros para a caixa de *docking*

como sendo: *center_x* = -14.041, *center_y* = -42.654, *center_z* = 28.419, *size_x* = 25, *size_y* = 25 e *size_z* = 25.

Figura 28 – Visualização *pocket* 4 e 39 - EasyVs. É possível verificar que o *pocket* 4 possui melhor posicionamento que o *pocket* 39.



Fonte: Autores.

Outros parâmetros a serem considerados a respeito do *docking* é com relação ao número de conformações que serão apresentadas, o padrão são 9 conformações e aqui optou-se por fazer 10 conformações para cada ligante. Rocha (2017) usou o valor padrão. O *energy range* é responsável por identificar a diferença máxima de energia aceita entre as conformações e o *exhaustiveness* é responsável por determinar o esforço computacional que será empregado. Neste trabalho definiu-se que serão utilizados os valores padrão para os dois dados. Outro dado importante a ser considerado é a quantidade de CPU que será utilizada para a realização do *docking*. Colocou-se o valor 1 e aconselha-se que esse seja sempre o valor utilizado. Esse aconselhamento se baseia em que, o *script* de *docking* foi escrito para ser realizado em paralelo, assim sendo, fazer o *docking* também em paralelo pode tornar o sistema mais lento, devido à gestão que o SO terá que executar. Para os parâmetros *ProteinHydrogen* e *LigandHydrogen*, eles são responsáveis por fazer o *docking* considerando os hidrogênios na proteína ou ligante. Assim como Rocha (2017) esses valores serão considerados com os valores padrão do *Vina* e *Smina*.

A última decisão a ser realizada é a quantidade de rodadas que serão realizadas de *docking*. No *workflow*, é possível definir o valor desejado para cada um dos ligantes separadamente. Assim, o pesquisador pode decidir a quantidade de rodadas que deseja realizar. Neste trabalho, assim como em Rocha (2017) foi realizado o *docking* em triplicata para cada um dos ligantes desejados.

A respeito dos ligantes a serem estudados, optou-se por utilizar os mesmos ligantes que os trabalhados por (Rocha, 2017), conforme apresentado na Seção 4.2. A Galantamina, Licoramina e Sanguinina foram recuperadas pelo ZINC15, passando apenas a numeração do ZINC para o sistema.

Os códigos ZINC utilizados são: ZINC00491073 (GNT), ZINC04102421(LYC), ZINC4102420 (SNG). Os arquivos ZINC e a molécula HYB foram as mesmas utilizadas na pesquisa de Rocha (2017).

Para a entrada de dados do ligante no arquivo de configuração são necessários preencher os seguintes campos: *Ligand*, *LigandCode*, *NameLigand* e *ForceField*. O primeiro campo possui o código ZINC de cada ligante e para o caso de moléculas mol2 fornecidas pelo pesquisador deve-se colocar o nome HYB. O código do ligante possui o nome com 3 caracteres que o ligante é normalmente reconhecido. Em nome ligante é o nome que será de fato utilizado no PDB e demais arquivos gerados. Assim, obrigatoriamente esse campo deve ser de 3 caracteres, uma vez que ele será utilizado nos arquivos no local do nome do resíduo. No caso do ligante HYB, é passado o nome do arquivo com a extensão, mas o nome deve possuir as 3 letras apenas para garantir que não atrapalhe os PDBs gerados. Em *ForceField* é fornecido o campo de força de cada ligante. Aqui, poderia ser utilizado um campo de força diferente para cada ligante. Neste trabalho usa-se o *ff99SB*, da mesma forma que Rocha (2017).

Para a etapa de modelagem molecular nenhum dado específico será necessário para que seja realizada. Da mesma forma que Rocha (2017) também será utilizada a modelagem por homologia e todos os parâmetros necessários são recuperados pelo próprio sistema dos arquivos já existentes.

Para a protonação utiliza-se o *site* H++ e todos os parâmetros do sistema podem ser alterados por meio dos campos do arquivo de configuração. No caso deste trabalho foi realizada a protonação com força iônica de 0.15, pH de 7.0, constante dielétrica interna de 10 e externa de 80, da mesma forma que Rocha (2017).

Para realizar a solvatação e ionização é necessário fornecer no arquivo de configuração o solvente a ser utilizado e o tamanho da caixa. Esses valores são fornecidos já no padrão do AMBER. Para a ionização, basta fornecer a concentração iônica desejada e o próprio sistema já realiza o cálculo para inserir a quantidade de Na+ e Cl- que serão utilizados. Hoje o sistema só insere o NaCl, mas posteriormente pretende-se passar a usar mais tipos de íons.

A última parte do arquivo de configuração é a que trata das configurações da dinâmica propriamente dita. Nesse momento são definidos os protocolos de relaxamento e produção do sistema. Rocha (2017) utilizou ao todo 1 passo de minimização, 7 passos de relaxamento e 1 passo de simulação em produção. No caso foram realizadas 9 simulações. Além disso, quando for rodar a simulação será utilizado um servidor que possui 24 núcleos e CUDA instalado.

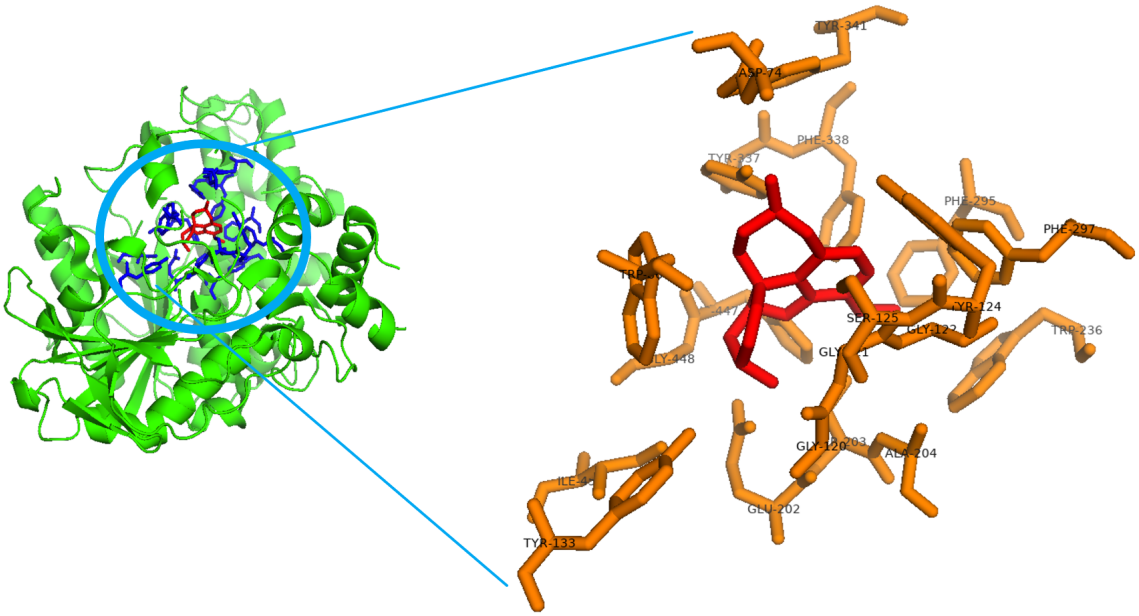
Figura 29 – Estrutura cristalográfica da 4EY6 – cadeia A, com destaque para o GNT. O ligante cristalográfico GNT está destacado em vermelho.

Fonte: Autores.

Depois de se definir esses itens, é importante a definição dos valores referente às restrições harmônicas de cada passo de relaxamento. Pela metodologia detalhada na Seção 4.2 foram definidas as restrições de acordo com o distanciamento que o ligante está de determinados resíduos. O primeiro passo é o de buscar esses resíduos e para isso utilizou-se a estrutura cristalográfica. A Figura 29 mostra a estrutura cristalográfica da 4EY6, apenas com a cadeia A e o ligante destacado. Essa imagem foi gerada a partir do Pymol.

A Figura 30 apresenta a 4EY6 com destaque para os resíduos que se encontram a 5Å do ligante GNT cristalográfico. Foram identificados 20 resíduos a 5Å do GNT, são eles: ASP A 74, TRP A 86, GLY A 120, GLY A 121, GLY A 122, TYR A 124, SER A 125, TYR A 133, GLU A 202, SER A 203, ALA A 204, TRP A 236, PHE A 295, PHE A 297, TYR A 337, PHE A 338, TYR A 341, HIS A 447, GLY A 448, ILE A 451.

Figura 30 – Estrutura cristalográfica da 4EY6 – resíduos a 5Å da GNT. No destaque estão os resíduos a 5Å do sítio ativo. Esses ligantes serão utilizados na restrição harmônica conforme descrito da Seção 4.2 . Na figura da cadeia A da Proteína à esquerda está destacado em vermelho o ligante GNT e em azul os resíduos a 5Å. No destaque à direita, em vermelho está o ligante e em alaranjado estão os resíduos a 5Å.



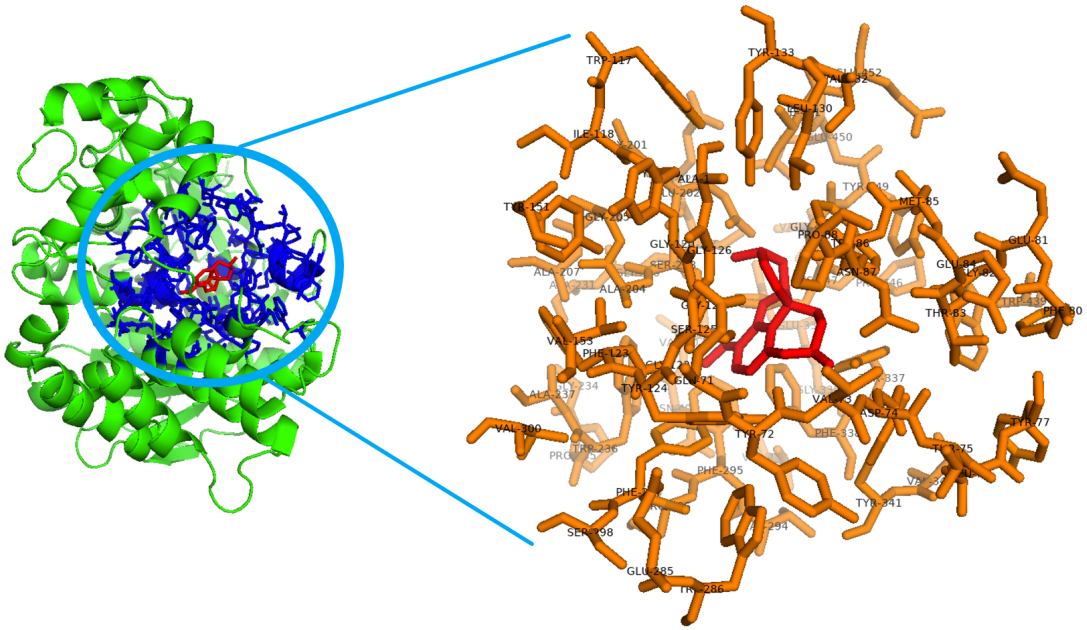
Fonte: Autores.

A Figura 31 apresenta os resíduos presentes a 10Å do ligante cristalográfico GNT. Foram identificados 73 resíduos a 10Å, são eles: GLN A 71, TYR A 72, VAL A 73, ASP A 74, THR A 75, LEU A 76, TYR A 77, PHE A 80, GLU A 81, GLY A 82, THR A 83, GLU A 84, MET A 85, TRP A 86, ASN A 87, PRO A 88, TRP A 117, ILE A 118, TYR A 119, GLY A 120, GLY A 121, GLY A 122, PHE A 123, TYR A 124, SER A 125, GLY A 126, ALA A 127, LEU A 130, VAL A 132, TYR A 133, TYR A 151, VAL A 153, GLY A 201, GLU A 202, SER A 203, ALA A 204, GLY A 205, ALA A 206, ALA A 207, GLN A 228, SER A 229, GLY A 230, ALA A 231, GLY A 234, PRO A 235, TRP A 236, ALA A 237, GLU A 285, TRP A 286, VAL A 294, PHE A 295, ARG A 296, PHE A 297, SER A 298, VAL A 300, VAL A 331, GLU A 334, GLY A 335, TYR A 337, PHE A 338, VAL A 340, TYR A 341, VAL A 402, ASN A 406, VAL A 407, TRP A 439, PRO A 446, HIS A 447, GLY A 448, TYR A 449, GLU A 450, ILE A 451, GLU A 452.

Para cada parte *Number* do arquivo de configuração, existe uma *tag Free*. Para cada etapa foram cadastrados os seguintes itens no campo *Free* (lembrando que no arquivo de configuração é um item em cada linha:

i. *Number* 1: *Water* e *Ions*;
ii. *Number* 2: *Water* e *Ions*;
iii. Number 3: Ligand, Water e Ions;
iv. *Number* 4: SC ASP A 74, SC TRP A 86, SC GLY A 120, SC GLY A 121, SC GLY A 122, SC TYR A 124, SC SER A 125, SC TYR A 133, SC GLU A 202, SC SER A 203, SC ALA A 204, SC TRP A 236, SC PHE A 295, SC PHE A 297, SC TYR A 337, SC PHE A 338, SC TYR A 341, SC HIS A 447, SC GLY A 448, SC ILE A 451, *Ligand*, *Water*, *Ions*;
v. *Number* 5: SC GLN A 71, SC TYR A 72, SC VAL A 73, SC ASP A 74, SC THR A 75, SC LEU A 76, SC TYR A 77, SC PHE A 80, SC GLU A 81, SC GLY A 82, SC THR A 83, SC GLU A 84, SC MET A 85, SC TRP A 86, SC ASN A 87, SC PRO A 88, SC TRP A 117, SC ILE A 118, SC TYR A 119, SC GLY A 120, SC GLY A 121, SC GLY A 122, SC PHE A 123, SC TYR A 124, SC SER A 125, SC GLY A 126, SC ALA A 127, SC LEU A 130, SC VAL A 132, SC TYR A 133, SC TYR A 151, SC VAL A 153, SC GLY A 201, SC GLU A 202, SC SER A 203, SC ALA A 204, SC GLY A 205, SC ALA A 206, SC ALA A 207, SC GLN A 228, SC SER A 229, SC GLY A 230, SC ALA A 231, SC GLY A 234, SC PRO A 235, SC TRP A 236, SC ALA A 237, SC GLU A 285, SC TRP A 286, SC VAL A 294, SC PHE A 295, SC ARG A 296, SC PHE A 297, SC SER A 298, SC VAL A 300, SC VAL A 331, SC GLU A 334, SC GLY A 335, SC TYR A 337, SC PHE A 338, SC VAL A 340, SC TYR A 341, SC VAL A 402, SC ASN A 406, SC VAL A 407, SC TRP A 439, SC PRO A 446, SC HIS A 447, SC GLY A 448, SC TYR A 449, SC GLU A 450, SC ILE A 451, SC GLU A 452, *Ligand*, *Water*, *Ions*;
vi. *Number* 6: GLN A 71, TYR A 72, VAL A 73, ASP A 74, THR A 75, LEU A 76, TYR A 77, PHE A 80, GLU A 81, GLY A 82, THR A 83, GLU A 84, MET A 85, TRP A 86, ASN A 87, PRO A 88, TRP A 117, ILE A 118, TYR A 119, GLY A 120, GLY A 121, GLY A 122, PHE A 123, TYR A 124, SER A 125, GLY A 126, ALA A 127, LEU A 130, VAL A 132, TYR A 133, TYR A 151, VAL A 153, GLY A 201, GLU A 202, SER A 203, ALA A 204, GLY A 205, ALA A 206, ALA A 207, GLN A 228, SER A 229, GLY A 230, ALA A 231, GLY A 234, PRO A 235, TRP A 236, ALA A 237, GLU A 285, TRP A 286, VAL A 294, PHE A 295, ARG A 296, PHE A 297, SER A 298, VAL A 300, VAL A 331, GLU A 334, GLY A 335, TYR A 337, PHE A 338, VAL A 340, TYR A 341, VAL A 402, ASN A 406, VAL A 407, TRP A 439, PRO A 446, HIS A 447, GLY A 448, TYR A 449, GLU A 450, ILE A 451, GLU A 452, *Ligand*, *Water*, *Ions*;
vii. *Number* 7: SCProtein, GLN A 71, TYR A 72, VAL A 73, ASP A 74, THR A 75, LEU A 76, TYR A 77, PHE A 80, GLU A 81, GLY A 82, THR A 83, GLU A 84, MET A 85, TRP A 86, ASN A 87, PRO A 88, TRP A 117, ILE A 118, TYR A 119, GLY A 120, GLY A 121, GLY A 122, PHE A 123, TYR A 124, SER A 125, GLY A 126, ALA A 127, LEU A 130, VAL A 132, TYR A 133, TYR A 151, VAL A 153, GLY A 201, GLU A 202, SER A 203, ALA A 204, GLY A 205, ALA A 206, ALA A 207, GLN A 228, SER A 229, GLY A 230, ALA A 231, GLY A 234, PRO A 235, TRP A 236, ALA A 237, GLU A 285, TRP A 286, VAL A 294, PHE A 295, ARG A 296, PHE A 297, SER A 298, VAL A 300, VAL A 331, GLU A 334, GLY A 335, TYR A 337, PHE A 338, VAL A 340, TYR A 341, VAL A 402, ASN A 406, VAL A 407, TRP A 439, PRO A 446, HIS A 447, GLY A 448, TYR A 449, GLU A 450, ILE A 451, GLU A 452, *Ligand*, *Water*, *Ions*;
viii. *Number* 8: ALL;
ix. *Number* 9: ALL.

Figura 31 – Estrutura cristalográfica da 4EY6 – resíduos a 10Å da GNT. No destaque estão os resíduos a 10Å do sítio ativo. Esses ligantes serão utilizados na restrição harmônica conforme realizado por Rocha (2017) e descrito da Seção 4.2 . Na figura da cadeia A da Proteína à esquerda está destacado em vermelho o ligante GNT e em azul os resíduos a 10Å. No destaque à direita, em vermelho está o ligante e em alaranjado estão os resíduos a 10Å.



Fonte: Autores.

O campo *Restart* foi setado "Yes" para todas as rodadas, exceto para a primeira que recebeu o valor "No". A partir desse dado, o *workflow* já deixa os parâmetros *bincoordinates*, *binvelocities* e *extendedSystem*, preenchidos pelo próprio *workflow*. *Structure* foi colocado "No" para todos, uma vez que não foi utilizado arquivo psf. *Temperature* foi 300 para todos, conforme a temperatura especificada por Rocha (2017). *FirstTimeStep* foi colocado com 0 para todos. E os campos *coordinates* e set *output* foram preenchidos automaticamente pelo *workflow*.

Para os parâmetros AMBER, foi fornecido o parâmetro AMBER com o valor "Yes", uma vez que foi utilizado o ff99sb AMBER como campo de força da simulação. E *Ambercoord* como "No", pois foi considerado o PDB inicial no *coordinates*. O valor de *ReadExclusions* foi "yes" e de *Scnb* de 2.0 (padrão do NAMD). *ParaTypeCharmm* e *Parameters* foram definidos para "no" em todos os arquivos de configuração. O *SimulationParameterTemperature*, foi definido com "Yes" na primeira simulação e não para as demais. Isso ocorre, pois como a primeira não tem *restart*, ela precisa do parâmetro *temperature* para conseguir realizar a simulação e, para os demais casos esse dado deve estar comentado no arquivo, pois serão usados os dados de *bincoordinates*, *binvelocities* e *extendedSyste.*

Para a definição das restrições harmônicas, todos os arquivos PDB com as devidas restrições foram gerados pelo sistema, e o nome será preenchido automaticamente em *consref* e *conskfile*. Os valores de *Constraints* foi "on", indicando que será realizada a restrição harmônica na simulação, *Consexp* "2" definindo como valor *default* para a função de restrição de energia harmônica, e *Conskcol* "B", indicando que os PDBs estão com as restrições definidas no campo Beta (*temperature-coupling*).

O corte de interações eletrostáticas foi realizado com um corte de 12Å para interações não ligadas. Para isto foi definido: *exclude* "scaled1-4" (define os pares de átomos que devem ser excluído das interações não ligadas), 1-4scaling "1.0" (uso do valor default), *cutoff* "12." (distância das interações eletrostáticas e de *van der Waals*), *switching* "on" (funções de suavização serão aplicadas para o corte), *switchdist* "10." (distância a ser considerada para usar a função de suavização),

pairlistdist “13.5”. Além desses valores também são definidos os seguintes itens: *timestep* “2.0, *rigidBonds* “all”, *nonbondedFreq* “1”, *fullElectFrequency* “2”, *stepspercycle* “10”.

Conforme Rocha (2017), também está sendo realizada a transformação termodinâmica sob condições de NPT, com a temperatura e pressão controladas pelo algoritmo de *Langevin* em 300K e 1atm. Para isso, foram definidos os parâmetros em todas as simulações: *langevin* “on”, *langevinDamping* “2” (coeficiente aplicado em cada átomo), *langevinTemp* “$temperature”, *langevinHydrogen* “off” (sem hidrogênios), *useGroupPressure* “yes”, *useFlexibleCell* “no”, *useConstantArea* “no”, *langevinPiston* “on” (realiza o controle da pressão), *langevinPistonTarget* “1.01325” (1 atm), *langevinPistonPeriod* “100.”, *langevinPistonDecay* “50.”, *langevinPistonTemp* “$temperature”.

O PME, conforme descrito na Seção 3.2, os valores referentes ao PME são preenchidos pelo próprio ProtCool_Dynamic, com valores calculados a partir dos arquivos do sistema. Assim, nenhuma informação é necessária, a não ser o campo PME do arquivo de configuração, que foi colocado com “yes”, indicando que PME será realizado durante a simulação.

A definição dos valores para gravação da saída dos arquivos *Namd* foram realizados considerando: *outputName* “$outputname”, *restartfreq* “1000” (tempo entre a geração de cada arquivo de reinício do Namd), *dcdfreq* “2000” (tempo entre cada geração de cada arquivo DCD), *outputEnergies* “1000” (tempo entre cada saída de energia do NAMD), *outputPressure* “1000” (tempo entre cada saída de pressão do NAMD).

Por fim, é necessário definir os parâmetros para execução. O Primeiro é o *minimize*, nesse caso, na primeira simulação foi utilizado o valor “1000”. Nas demais simulações o *minimize* foi definido com o valor “No”. Apenas a última simulação foi definida com o valor *reinitvels* “Yes”, todos os demais foram com o valor “No”. Apenas a última simulação teve os valores de temperatura conservados. E finalmente, o parâmetro *run* é definido com o tempo da simulação.

Com essas definições, todos os dados necessários para que o *workflow* seja devidamente executado foram cadastrados e a metodologia da pesquisa foi toda definida e pode-se considerar que foi possível realizar a mesma metodologia proposta por Rocha (2017). Após essas definições, foi executado o *script* `ProtCool_Dinamica.py`.

4.3.Resultados e discussões

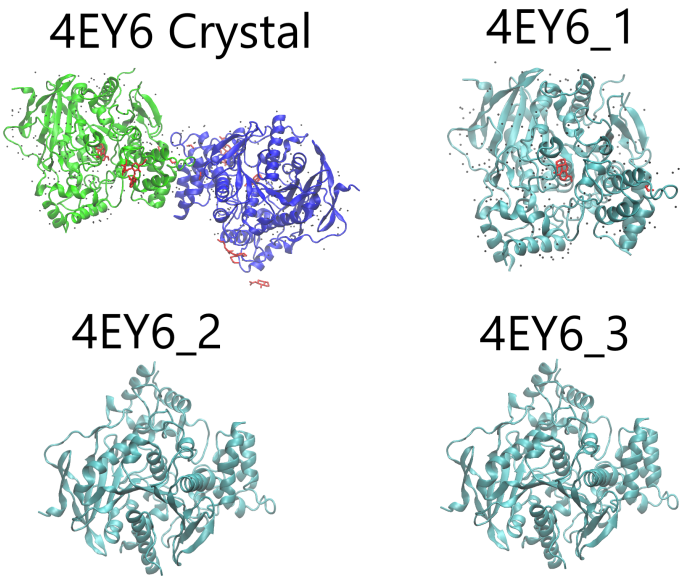
Nessa seção serão apresentados os resultados encontrados na preparação das dinâmicas, sendo discutida cada uma das etapas.

A primeira etapa de preparação é a responsável por buscar o arquivo PDB e fazer as modificações iniciais desses arquivos. A Figura 32 apresenta as 4 imagens, recuperadas pelo VMD destacando a evolução na etapa de preparação. Essa etapa é responsável por três pontos principais, o primeiro é o de fazer uma limpeza geral no arquivo PDB da 4EY6. O segundo ponto é a gravação do arquivo com as informações de pontes dissulfeto que auxiliarão no processo de campo de força do complexo. E o terceiro passo é a verificação de ocupâncias no arquivo PDB.

A 4EY6 Crystal é a proteína recuperada do *site* RSCB. Nela pode-se observar a presença das duas cadeias principais (A e B), a presença dos heteroátomos (GNT, EDO, NO3, PE8, NAG - cadeia C e D, FUC - cadeia C e D) e as águas. A 4EY6_1 é a molécula após realizar a separação de cadeias. Nota-se a presença da cadeia A (escolhida), dos heteroátomos da cadeia A (GNT e EDO) e das moléculas de água. Já a 4EY6_2 e a 4EY6_3, apresentam a proteína apenas com a cadeia escolhida, já tendo sido retirados todos os heteroátomos e águas presentes na proteína. Quanto às estruturas 4EY6_2 e 4EY6_3, a diferença se encontra na não existência de ocupâncias na molécula 4EY6_3.

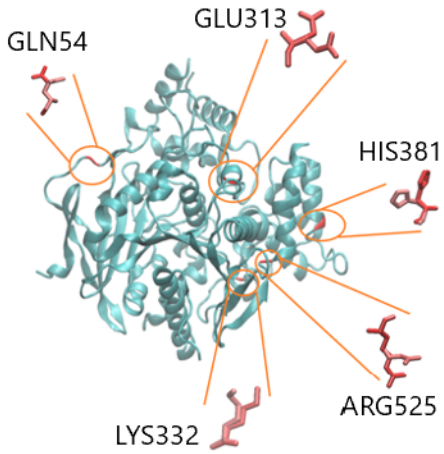
A Figura 33 apresenta a molécula 4EY6_2, com as marcações em vermelho dos resíduos que apresentam ocupância na estrutura. Foram encontrados 5 resíduos nessa situação: GLN54, GLU313, LYS332, HIS381 e ARG525. Os resíduos foram destacados na estrutura.

Figura 32 – Proteína 4EY6 e sua evolução na primeira etapa do *Workflow* Dinâmica. Apresenta as 4 estruturas da preparação, a primeira é a estrutura cristalográfica (4EY6 Crystal), a segunda é a cadeia A selecionada (4EY4_1), a terceira é a estrutura já sem as águas e os demais heteroátomos (4EY6_2) e a quarta é a estrutura com as ocupâncias acertadas (4EY6_3).



Fonte: Autores.

Figura 33 – 4EY6_2 - Ocupância. Imagem da estrutura da 4EY6 cadeia A com os resíduos que possuem ocupâncias destacados.

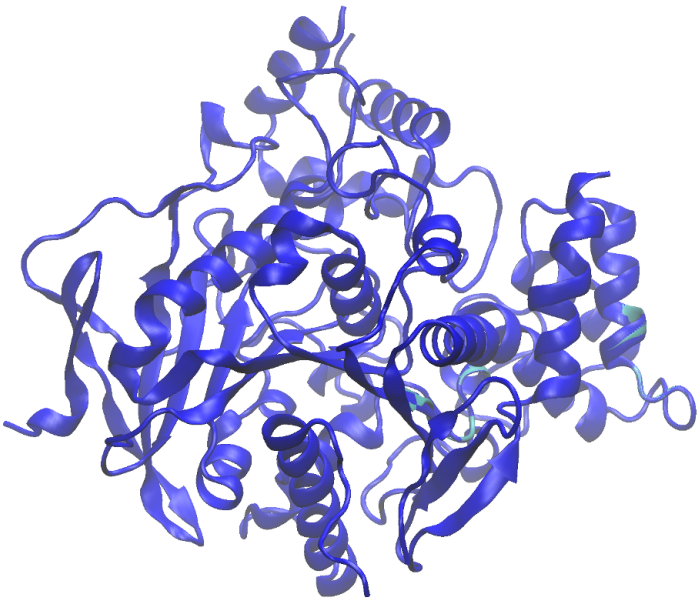


Fonte: Autores.

A Figura 34 apresenta a sobreposição entre os dois PDB. Em ciano está a 4EY6_2 e em azul a 4EY6_3. Apesar de existirem ocupâncias em 5 resíduos das estruturas, percebe-se uma diferença maior próximo aos resíduos LYS332, HIS381 e ARG525.

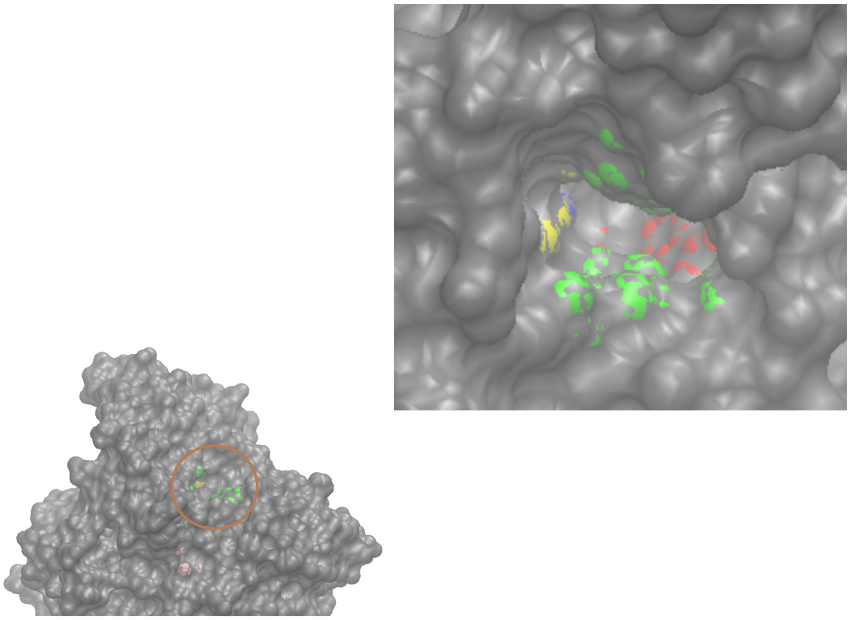
O segundo passo da preparação é o *docking*. Durante essa etapa, o objetivo principal é o de possibilitar que o ligante seja posicionado no sítio ativo da proteína. Conforme descrito na Seção 4.1 sabe-se que o sítio de ligação da acetilcolinesterase é estreito e profundo e que o ligante ficará na parte de baixo da molécula. A Figura 35 mostra o sítio ativo da 4EY6. No canto inferior esquerdo tem uma parte da molécula, delimitado pelo círculo pode-se observar a entrada do sítio ativo. Coloridos estão os subsítios que a 4EY6 possui. No detalhe é possível visualizar o sítio ativo.

Figura 34 – Sobreposição e 4EY6_2 e 4EY6_3 - Ocupância. Em ciano está a 4EY6_2 que possui ocupâncias e em azul a 4EY6_3 que não possui as ocupâncias. É possível observar que existe uma diferença nas estruturas sobrepostas.



Fonte: Autores.

Figura 35 – 4EY6 – Destaque para o sítio ativo. Na imagem à esquerda, circulado em vermelho esta a entrada do sítio ativo de ligação e à direita é apresentado o sítio ativo em maiores detalhes.



Fonte: Autores.

Cada subsítio teve uma coloração diferente: tríade catalítica em azul; sítio aniônico em vermelho, alça do oxiônico em laranja; bolso acílico em amarelo; sítio periférico aniônico em verde; *loop* ômega em rosa; e a ponte dissulfídeo em marrom. Na imagem inferior esquerda, pode-se observar em rosa, abaixo do círculo delimitado na figura, o *loop* ômega.

Observando-se a imagem do detalhe, vê-se o sítio internamente e nele pode-se observar o sítio periférico aniônico posicionado no início do sítio ativo, que junto ao bolso acílico delimitam a entrada

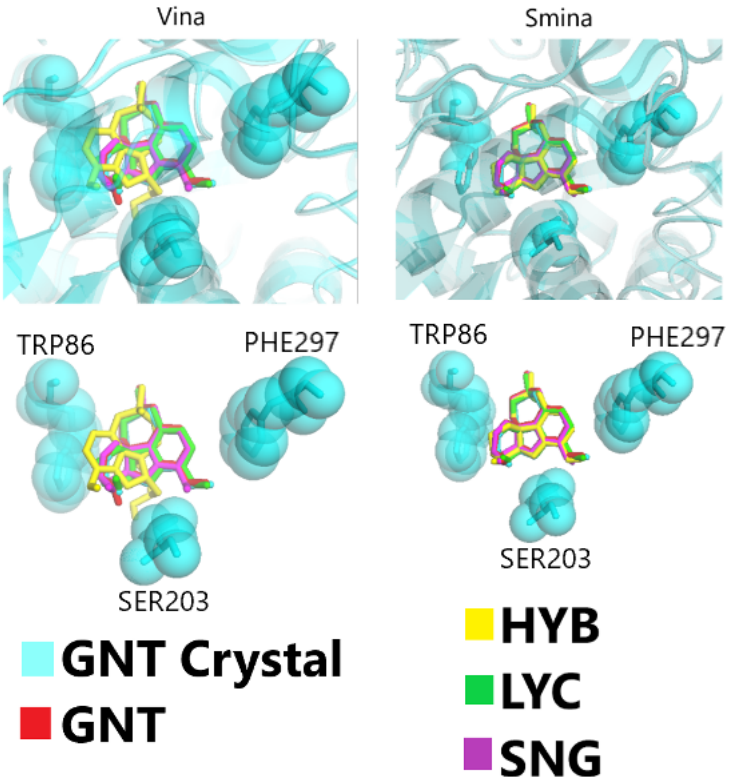
do sítio. Ao fundo, vê-se uma pequena área da tríade catalítica (azul), da alça do axiônion (laranja) e o sítio aniônico (vermelho), que fica bem ao fundo marcando o final do sítio ativo. É nesse trecho final que os ligantes deverão ficar posicionados após a finalização do *docking*. Na imagem não é possível visualizar a ponte dissulfeto e o *loop* ômega só é apresentado na parede da proteína e não na imagem do interior do sítio ativo. A Figura 21 apresenta três imagens mostrando os subsítios marcados em diferentes posicionamentos.

Todos os subsítios ativos irão interagir em algum momento com o ligante, seja na entrada/saída do sítio, ou durante a sua permanência dentro do sítio ativo.

De posse dessas informações, é possível realizar uma análise do resultado dos *dockings* efetuados pela ferramenta. Conforme relatado, realizou-se o *docking* com duas ferramentas de atracamento, o *Vina* e o *Smina*. Nos dois casos foram utilizados os mesmos parâmetros de configuração do sistema, e foram utilizados resíduos flexíveis (TRP86, TYR124, SER203, TYR337 e HIS447). Além disso, o *Smina* é uma ferramenta que foi desenvolvida levando em consideração a base *Vina*. Sendo assim, toda a base de parametrização dos dois sistemas é a mesma, desde o tratamento dado ao ligante quanto o tratamento dado à proteína. O que diferencia os dois é de fato as funções internas de minimização de energia que permitem a especificação e escolha das melhores poses.

Os dois tipos de *docking* foram realizados em triplicata. Os resultados de *Vina* foram analisados separadamente dos resultados de *Smina*. Nos dois casos, se avaliou todas as poses alcançadas durante o processo e se escolheu aquela com a menor energia considerando todas as poses que foram geradas, garantindo que a pose com o melhor *score* seja a escolhida. Além disso, durante o processo foi solicitado que cada rodada do *docking* gerasse 10 poses, assim, seriam geradas até no máximo de 30 conformações possíveis.

Figura 36 – *Docking* 4EY6 – *Vina* e *Smina* – Conformações identificadas. Imagem apresentando as estruturas sobrepostas, mostrando todos os ligantes docados sobrepostos ao ligante cristalográfico GNT. Do lado esquerdo estão as poses *Vina* e do lado direito as poses *Smina*.

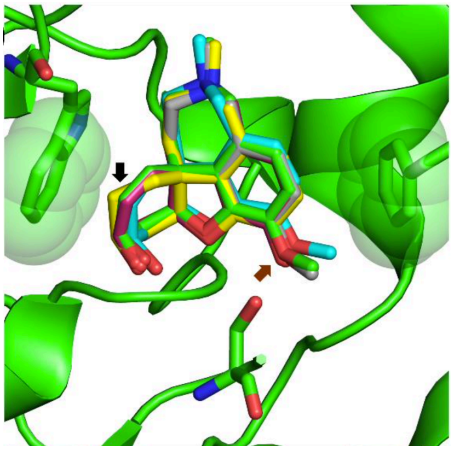


Fonte: Autores.

A Figura 36 apresenta o resultado do atracamento pelas ferramentas *Vina* e *Smina*. Foi utilizado o software *Pymol* para verificar se as poses foram adequadamente atracadas ao ligante. Esperava-se que os ligantes conseguissem reproduzir a pose cristalográfica da GNT, uma vez que todos os ligantes são galantamínicos, com poucas variações na estrutura dos compostos. Além disso, Rocha (2017) usando o software Vina, conseguiu reproduzir a pose cristalográfica da GNT para todos os ligantes, o que reforça que a pose cristalográfica da galantamina pode ser alcançada por todos os ligantes escolhidos. Possuir todos os ligantes com poses similares sugere que os mesmos resíduos da 4EY6 são utilizados nas interações com os ligantes.

Percebe-se que os 4 ligantes conseguiram uma boa reprodutibilidade no *docking Smina* tendo um desvio mínimo (Tabela 2) em relação ao ligante cristalográfico GNT. Os resíduos TRP86 (sítio aniônico), SER203 (Tríade catalítica) e PHE297 (bolso acílico) parecem realizar interações com os ligantes nesse sítio de ligação. Esses resíduos são destacados como farmacóforos importantes para os inibidores de AchE (Rocha, 2017). Isso indica que quando o ligante está devidamente atracado no sítio ativo, esses três subsítios e resíduos estão envolvidos na interação com o ligante. Isso também está condizente com o que foi encontrado por Rocha (2017) em seus experimentos e com o descrito na literatura a respeito dos resíduos indicados como farmacóforos da AchE. Segundo Rocha (2017), o TRP86 mostra-se como o responsável por fazer interações de empilhamento de anéis não aromáticos. A SER203 consegue estabelecer ligações de hidrogênio com o grupo oxigenado que aparece no anel aromático e o resíduo PHE297 realiza interações hidrofóbicas entre o grupo metoxi dos ligantes hidroxi não modificados, o que também pode ser observado nos atracamentos realizados. Na Figura 37 é possível visualizar a imagem retirada de Rocha (2017), em que é possível ver o atracamento alcançado na sua pesquisa. A imagem, conforme pode ser observado, é semelhante com o que se conseguiu com o *Smina* e com o *Vina* com a exceção do HYB.

Figura 37 – *Docking* 4EY6 – *Vina* Rocha (2017). A imagem retirada de Rocha (2017) apresenta as posições mais favoráveis dos ligantes GNT, HYB, LYC e SNG comparadas ao GNT cristalográfico. A seta preta indica o local de substituição no anel e a seta marrom a modificação metoxi-hidroxi que foram realizadas no HYB.



Fonte: Rocha (p. 53, 2017).

Com relação ao *docking Vina*, a GNT, LYC e SNG tiveram o mesmo comportamento apresentado pelo *Smina* (Tabela 2) e pelo trabalho de Rocha (2017) (Figura 37). Porém, com a HYB o *Vina* não conseguiu reproduzir o atracamento similar ao que havia sido encontrado no Smina e em Rocha (2017) tendo um RMSD de 2.75 Å. A Figura 38 apresenta as conformações da SNG, LYC e GNT, sem a presença do HYB, sendo mais facilmente verificável que esses três ligantes se comportaram exatamente da mesma forma que os compostos de Rocha (2017) (Figura 37) e com o

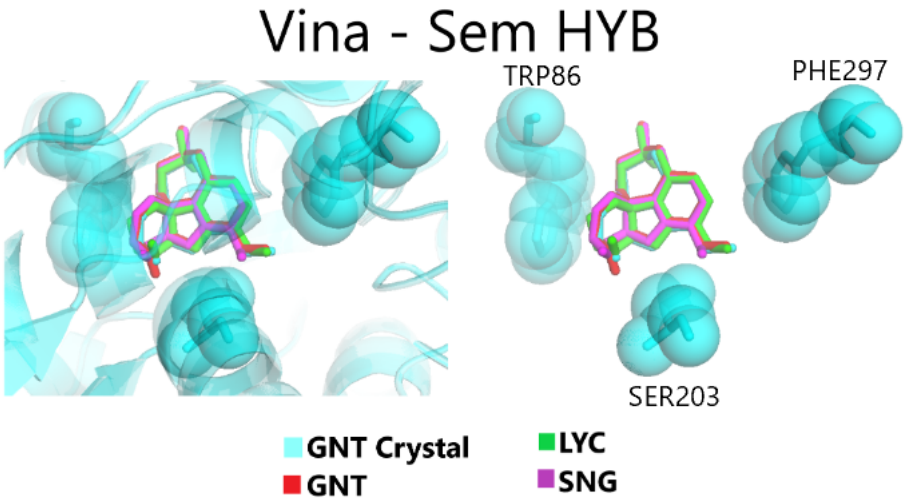
atracamento *Smina* (Figura 36). As interações com os resíduos TRP86, SER203 e PHE297 se preservaram, quanto a questão da projeção dos ligantes também se repete. Além disso, os dados da Tabela 2 que possuem os RMSD dos ligantes no *Vina* e *Smina*, percebe-se que todos os ligantes, exceto o HYB tiveram os mesmos valores de RMSD no *Vina* e *Smina*.

Tabela 2 – RMSD *Vina* e *Smina*. A tabela apresenta os resultados de RMSD dos ligantes GNT, HYB, LYC e SNG no *Vina* e no *Smina* em relação ao ligante cristalográfico.

	Vina	***Smina***
GNT	0.13 Å	0.13 Å
HYB	2.75 Å	0.12 Å
LYC	0.11 Å	0.11 Å
SNG	0.12 Å	0.12 Å

Fonte: Autores

Figura 38 – *Docking* 4EY6 – *Vina* – Sem o HYB. São apresentadas as conformações alcançadas pelo GNT, LYC e SNG em comparação do GNT cristalográfico. Os resultados foram similares aos do *Smina* e de Rocha (2017).



Fonte: Autores.

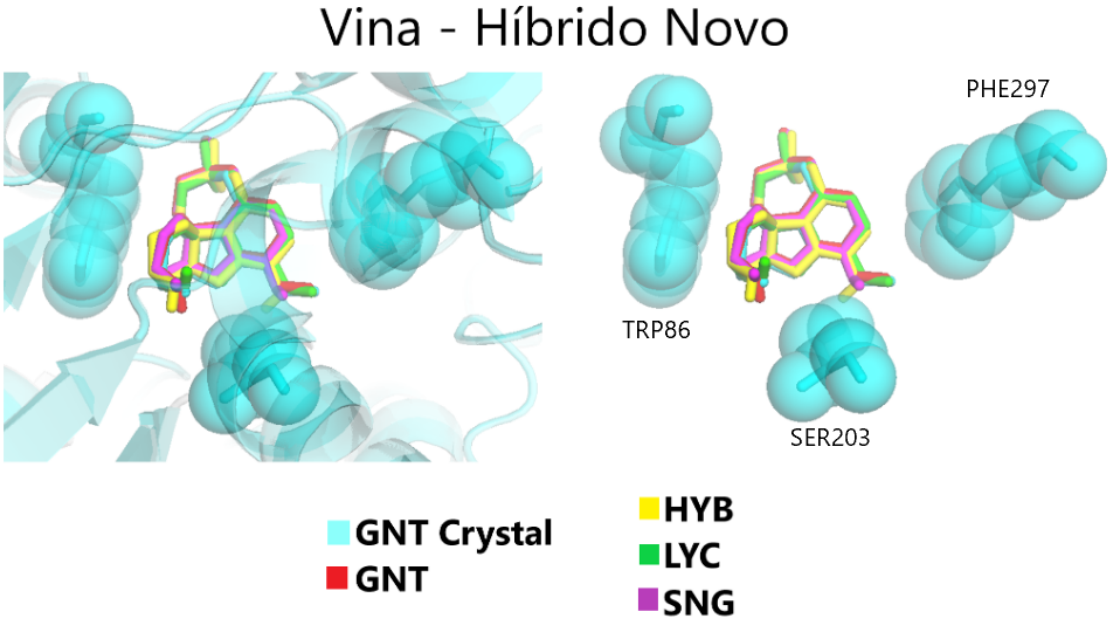
Analisando esses dados, apesar do *Smina* ter conseguido reproduzir a pose da galantamina cristalográfica com pequeno desvio (Tabela 2), o sistema não gerou todas as poses desejadas. Sabe-se que o *Smina* é uma evolução do *Vina*, pode ser que conseguiu reproduzir melhor as poses, devido a essa característica. Uma vez que o atracamento ocorreu de forma satisfatória com o *Smina* para todas as poses, poder-se-ia decidir em seguir o processo com as moléculas preparadas com o *Smina*, ou, seguir o processo da GNT, LYC e SNG com as poses identificadas pelo *Vina* e com a HYB identificada pelo *Smina*.

Porém, considerando-se que o *Smina* tem por base o *Vina* e sabendo-se que o atracamento do *Vina* não é determinístico, ou seja, podem ocorrer pequenos desvios devido aos cálculos efetuados durante a execução do sistema. E que os valores de RMSD do *Vina* e *Smina* foram os mesmos em todas as poses do GNT, LYC e SNG, e apenas os valores de HYB tiveram valores divergentes, optou-se por realizar nova execução do sistema, mas apenas considerando a HYB. Ou seja, executou-se todo o sistema novamente, mas ao invés de refazer para todos os ligantes, foi realizada nova execução apenas com o HYB no arquivo de configuração.

O novo atracamento realizado com o ligante HYB pode ser visualizado na Figura 39. Pode-se perceber que na nova rodada do *docking*, a nova pose selecionada conseguiu reproduzir o posicionamento do *Crystal*, tendo um desvio mínimo da pose do *Crystal* (0.12 Å). Esse novo resultado foi condizente com o esperado. O novo posicionamento faz com que o *Vina* passe a ter resultado semelhante ao *Smina*. Assim, segue-se com o uso das poses selecionadas pelo *Vina*, porém, com a troca da pose pela nova pose alcançada pelo *Vina*. Vale ressaltar, que nessa nova execução o *Vina* conseguiu gerar as 10 poses por triplicata, sendo geradas com isso 30 novas poses.

A Tabela 3 apresenta uma comparação entre os 3 grupos de *scores*. Apesar de possuir diferenças nos valores, o que era de se esperar uma vez que o processo é não determinístico, os dados de *Vina* e Rocha (2017) estão fornecendo o mesmo ranqueamento de *score*.

Figura 39 – *Docking* 4EY6 – *Vina* – HYB Novo. Apresentação do resultado da nova rodada do ProtCool_Dynamic apenas com o HYB. Na imagem são apresentados todos os ligantes novamente e pode-se observar que nessa nova execução, o HYB teve um resultado condizente com os demais ligantes.



Fonte: Autores.

Tabela 3 – *Scores Vina*, *Smina* e Rocha (2017). *Scores* fornecidos em kcal/mol. Dados do *Vina*, *Smina* e Rocha (2017).

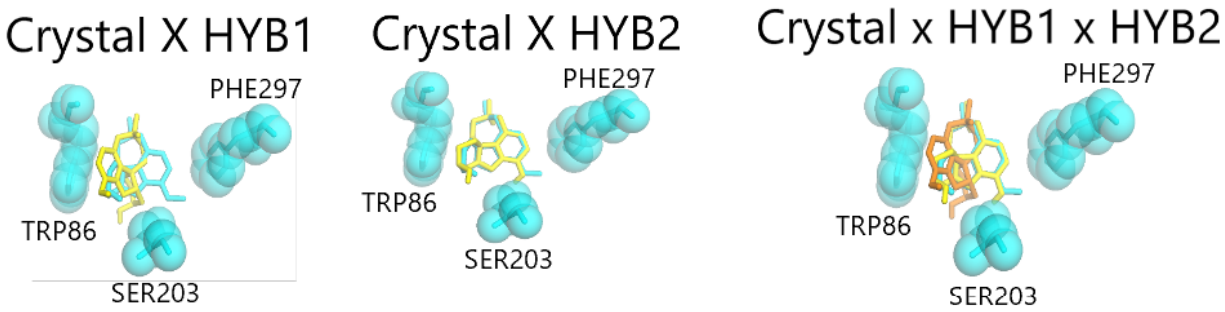
	Vina	*Smina*	Rocha (2017)[42]
GNT	-9.52	-9.88	-9.25
HYB	-9.50	-10.26	-8.95
LYC	-9.34	-9.74	-8.85
SNG	-9.78	-10.30	-9.30

Fonte: Autores.

[42] Os dados apresentados nesta coluna são aproximados, foram retirados de um gráfico.

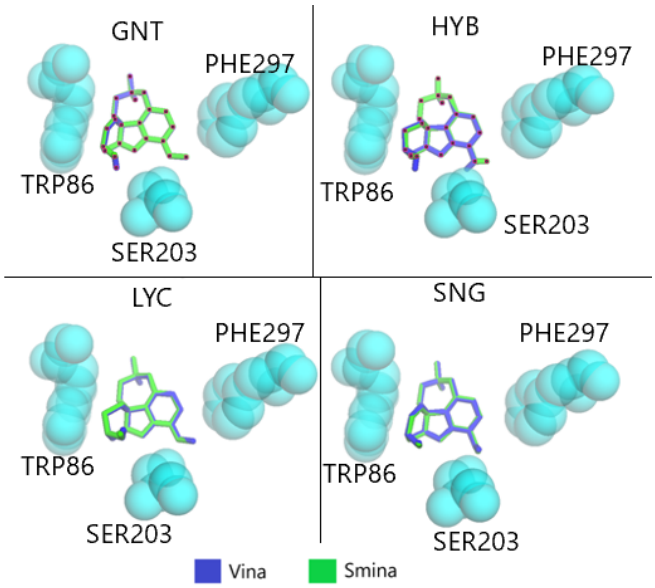
A Figura 40 mostra as conformações do HYB1 e HYB2. Na *Crystal* X HYB1, o ligante cristalográfico está em ciano e a HYB1 está em amarelo. É possível perceber que o *Vina* não conseguiu encontrar um bom posicionamento para a HYB1. Na *Crystal* x HYB2, já é possível perceber que o *Vina* conseguiu uma conformação adequada para a HYB2 (amarela). E na terceira mostra-se as três conformações juntas.

Figura 40 – Apresentação das conformações do HYB1 e HYB2. Visualização da pose *Crystal* (ciano) em relação ao HYB1 (amarelo). Crystal (ciano) com relação ao HYB2 (amarelo). Na terceira imagem são apresentados os três em conjunto *Crystal* (ciano), HYB1 (laranja) e HYB2 (amarelo).



Fonte: Autores.

Figura 41 – Conformações *Vina* e *Smina*. Poses *Vina* e *Smina* sobrepostas de cada um dos ligantes. Em azul a pose *Vina* e em verde a pose *Smina*.



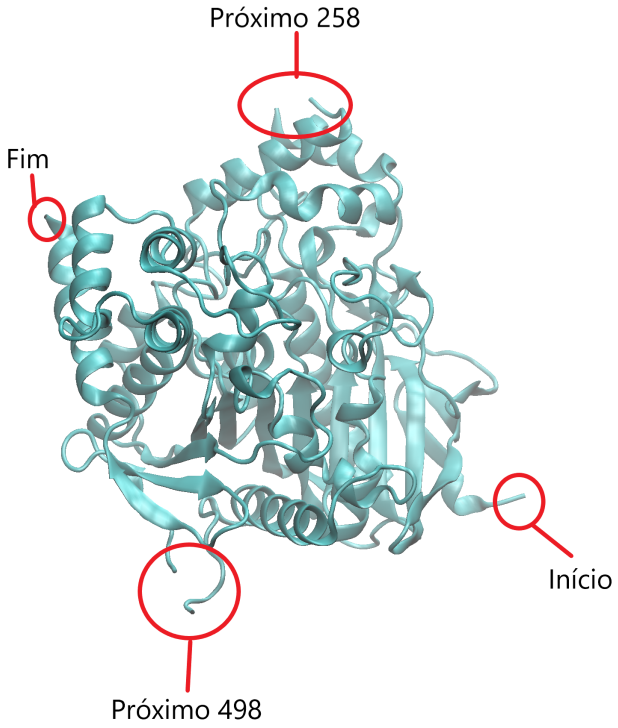
Fonte: Autores.

A Figura 41 apresenta as conformações de *Vina* e *Smina* para cada um dos Ligantes. Observando-se as imagens é possível perceber que nos quatro casos os dois sistemas encontraram conformações semelhantes para cada um dos ligantes, o que de acordo com os valores de RMSD também é verdade.

A próxima etapa do processo é a modelagem molecular. Nessa etapa, conforme verificado na Seção 3.2 o objetivo é que os *gaps* existentes na proteína sejam corrigidos. Na Figura 42 é apresentada a 4EY6, destacando-se os *gaps* existentes na proteína. Foram identificadas 4 áreas de *gaps*, conforme Figura 9. Os *gaps* identificados, um está no início da proteína, outro no final. Além desses dois, existem

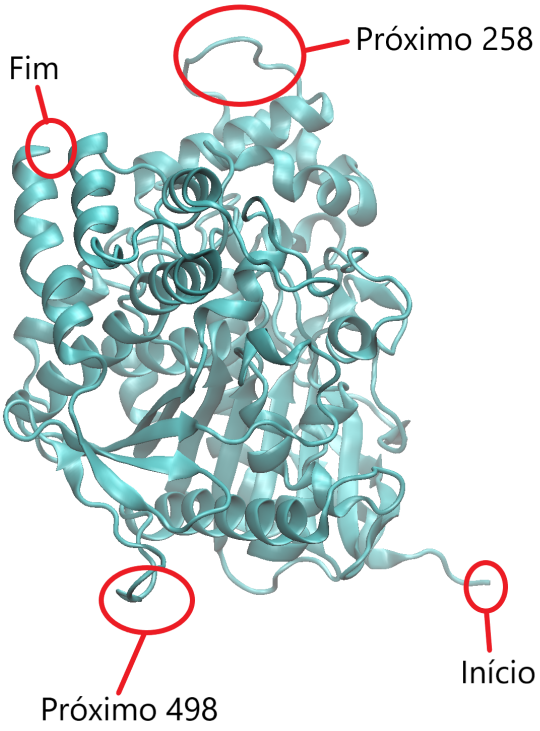
dois no meio da proteína, um próximo ao resíduo 258 e outro próximo aos resíduos 498, conforme destacado na imagem.

Figura 42 – 4EY6 – Identificação dos *Gaps*. Na imagem é possível ver os 4 *gaps* da molécula, marcados em vermelho.



Fonte: Autores.

Figura 43 – 4EY6 – Acerto dos *Gaps*. Acertos realizados na molécula, apresentados em vermelho.

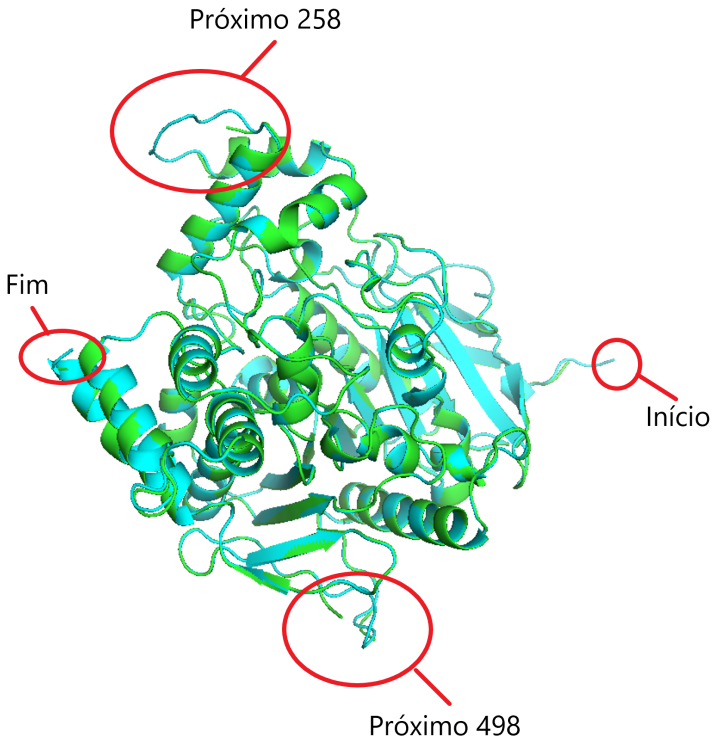


Fonte: Autores.

A Figura 43 apresenta a molécula 4EY6 após o acerto dos *gaps*. Pela imagem é possível verificar que os 4 *gaps* identificados foram devidamente modelados pelo *workflow*. A Figura 44 apresenta as duas estruturas sobrepostas no Pymol. A 4EY6_3, antes de se realizar a modelagem e a 4EY6_4, após a modelagem. Na imagem estão destacados os pontos em que o Modeller realizou as modelagens, inserindo os resíduos necessários.

Após esse passo, a estrutura está com todas as numerações modificadas, uma vez que foram inseridos os resíduos na molécula. Assim, o arquivo `4EY6_Acertos.txt` que se encontra na pasta do usuário, dentro de *log*, mostra o mapeamento da nova estrutura. A Figura 45 apresenta trechos do arquivo `4EY6_Acertos.txt` com os principais pontos modificados. Lembrando que como ocorrerá uma modificação de toda a estrutura e numeração dos resíduos, não apenas esses valores serão modificados, logo o pesquisador deve ficar atento a partir desse momento para que ele possa fazer as análises do seu experimento.

Figura 44 – 4EY6_3 e 4EY6_4 – Estruturas Sobrepostas. São apresentadas as duas imagens sobrepostas, uma antes da modelagem e a outra após a realização da modelagem molecular.

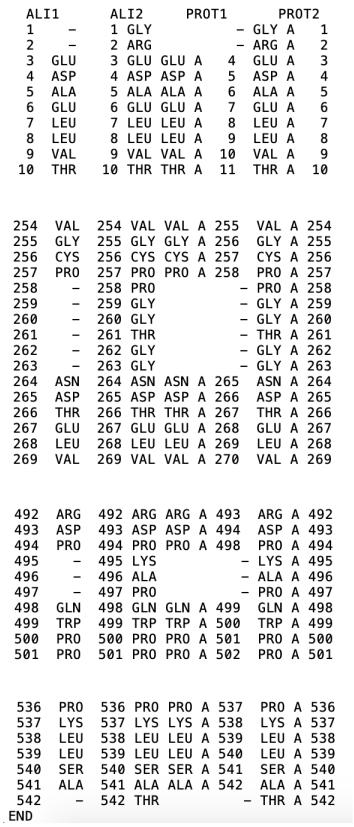


Fonte: Autores.

Após a realização da modelagem, o ProtCool_Dynamic realizou a protonação do sistema. Na protonação é utilizado o *site* H++ e por meio dos dados recuperados do *site* são acertadas as nomenclaturas do arquivo com relação ao HIS. A 4EY6 possui ao todo 11 resíduos HIS na sua cadeia A. Na Figura 17 pode-se observar que esses resíduos receberam a seguinte configuração após a protonação: Resíduo 211 – HIE; Resíduo 222 – HIE; Resíduo 252 – HIE; Resíduo 283 - HID; Resíduo 286 – HIE; Resíduo 321 – HIP; Resíduo 380 – HID; Resíduo 386 – HIP; Resíduo 404 – HID; Resíduo 431 – HIE; Resíduo 446 – HIP.

Além dessa modificação do arquivo PDB, também são modificados os nomes das CYS que participam de pontes dissulfeto para a nomenclatura CYX. Todos esses dados são disponibilizados no arquivo `4EY6_Acertos_1.txt`, conforme verificado na Figura 46.

Figura 45 – Trecho do arquivo `4EY6_Acertos.txt` com os principais pontos modificados. No arquivo é possível verificar que os trechos que antes possuíam o valor "-", passaram a ter os resíduos identificados.



```
 ALI1      ALI2     PROT1      PROT2
 1    -    1 GLY          - GLY A   1
 2    -    2 ARG          - ARG A   2
 3  GLU    3 GLU GLU A   4  GLU A   3
 4  ASP    4 ASP ASP A   5  ASP A   4
 5  ALA    5 ALA ALA A   6  ALA A   5
 6  GLU    6 GLU GLU A   7  GLU A   6
 7  LEU    7 LEU LEU A   8  LEU A   7
 8  LEU    8 LEU LEU A   9  LEU A   8
 9  VAL    9 VAL VAL A  10  VAL A   9
10  THR   10 THR THR A  11  THR A  10


254  VAL  254 VAL VAL A 255  VAL A 254
255  GLY  255 GLY GLY A 256  GLY A 255
256  CYS  256 CYS CYS A 257  CYS A 256
257  PRO  257 PRO PRO A 258  PRO A 257
258    -  258 PRO          - PRO A 258
259    -  259 GLY          - GLY A 259
260    -  260 GLY          - GLY A 260
261    -  261 THR          - THR A 261
262    -  262 GLY          - GLY A 262
263    -  263 GLY          - GLY A 263
264  ASN  264 ASN ASN A 265  ASN A 264
265  ASP  265 ASP ASP A 266  ASP A 265
266  THR  266 THR THR A 267  THR A 266
267  GLU  267 GLU GLU A 268  GLU A 267
268  LEU  268 LEU LEU A 269  LEU A 268
269  VAL  269 VAL VAL A 270  VAL A 269


492  ARG  492 ARG ARG A 493  ARG A 492
493  ASP  493 ASP ASP A 494  ASP A 493
494  PRO  494 PRO PRO A 498  PRO A 494
495    -  495 LYS          - LYS A 495
496    -  496 ALA          - ALA A 496
497    -  497 PRO          - PRO A 497
498  GLN  498 GLN GLN A 499  GLN A 498
499  TRP  499 TRP TRP A 500  TRP A 499
500  PRO  500 PRO PRO A 501  PRO A 500
501  PRO  501 PRO PRO A 502  PRO A 501


536  PRO  536 PRO PRO A 537  PRO A 536
537  LYS  537 LYS LYS A 538  LYS A 537
538  LEU  538 LEU LEU A 539  LEU A 538
539  LEU  539 LEU LEU A 540  LEU A 539
540  SER  540 SER SER A 541  SER A 540
541  ALA  541 ALA ALA A 542  ALA A 541
542    -  542 THR          - THR A 542
END
```

Fonte: Autores.

A etapa de campo de força tem 3 atividades principais. A primeira delas é a responsável por gerar o campo de força que será utilizado no NAMD. Além dessa etapa existe a solvatação e a ionização. A ionização é calculada pelo sistema e para realização do cálculo foi utilizada a força iônica em 0.15M. Após a realização da solvatação e ionização, os seguintes números foram identificados para cada ligante tanto para os arquivos *Vina* quanto para os arquivos *Smina*:

i. GNT:
   a. Número de átomos da proteína: 8294;
   b. Número de átomos do ligante: 43;
   c. Número total de átomos: 73102;
   d. Número de moléculas de água: 21535 (64605 átomos);
   e. Carga do complexo: -4.0
   f. Número de Na+: 82;
   g. Número de Cl-: 78.

ii. HYB:
   a. Número de átomos da proteína: 8294;
   b. Número de átomos do ligante: 42;
   c. Número total de átomos: 73101;
   d. Número de moléculas de água: 21535 (64605 átomos);
   e. Carga do complexo: -4.0
   f. Número de Na+: 82;

g. Número de Cl-: 78.

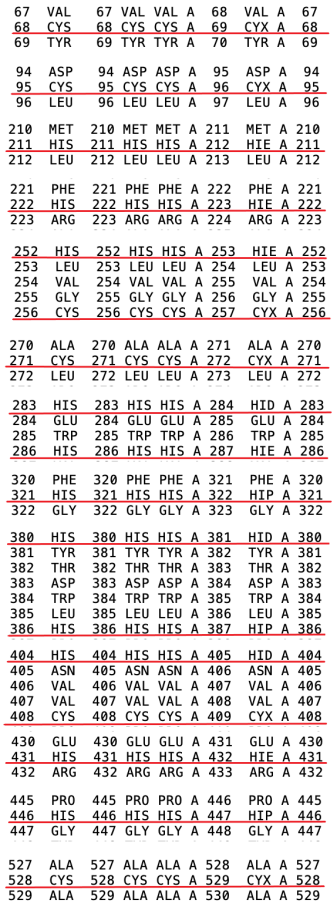
iii. LYC:

a. Número de átomos da proteína: 8294;
b. Número de átomos do ligante: 45;
c. Número total de átomos: 73104;
d. Número de moléculas de água: 21535 (64605 átomos);
e. Carga do complexo: -4.0
f. Número de Na+: 82;
g. Número de Cl-: 78.

iv. SNG:

a. Número de átomos da proteína: 8294;
b. Número de átomos do ligante: 40;
c. Número total de átomos: 73099;
d. Número de moléculas de água: 21535 (64.605 átomos);
e. Carga do complexo: -4.0
f. Número de Na+: 82;
g. Número de Cl-: 78.

Figura 46 – Trecho do arquivo `4EY6_Acertos.txt` com pontos modificados HIS e CYS. Marcados em vermelho existem cada uma das modificações do arquivo de acertos.

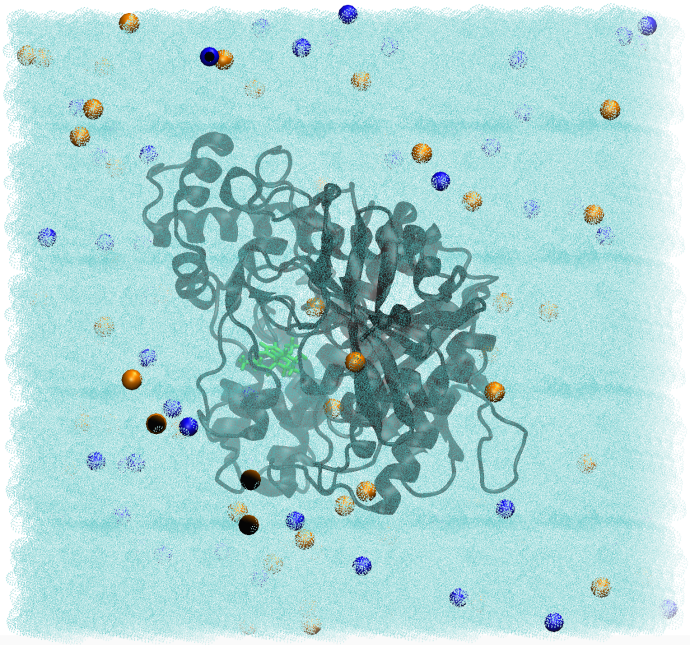


```
 67  VAL   67 VAL VAL A  68  VAL A  67
 68  CYS   68 CYS CYS A  69  CYX A  68
 69  TYR   69 TYR TYR A  70  TYR A  69

 94  ASP   94 ASP ASP A  95  ASP A  94
 95  CYS   95 CYS CYS A  96  CYX A  95
 96  LEU   96 LEU LEU A  97  LEU A  96

210  MET  210 MET MET A 211  MET A 210
211  HIS  211 HIS HIS A 212  HIE A 211
212  LEU  212 LEU LEU A 213  LEU A 212

221  PHE  221 PHE PHE A 222  PHE A 221
222  HIS  222 HIS HIS A 223  HIE A 222
223  ARG  223 ARG ARG A 224  ARG A 223

252  HIS  252 HIS HIS A 253  HIE A 252
253  LEU  253 LEU LEU A 254  LEU A 253
254  VAL  254 VAL VAL A 255  VAL A 254
255  GLY  255 GLY GLY A 256  GLY A 255
256  CYS  256 CYS CYS A 257  CYX A 256

270  ALA  270 ALA ALA A 271  ALA A 270
271  CYS  271 CYS CYS A 272  CYX A 271
272  LEU  272 LEU LEU A 273  LEU A 272

283  HIS  283 HIS HIS A 284  HID A 283
284  GLU  284 GLU GLU A 285  GLU A 284
285  TRP  285 TRP TRP A 286  TRP A 285
286  HIS  286 HIS HIS A 287  HIE A 286

320  PHE  320 PHE PHE A 321  PHE A 320
321  HIS  321 HIS HIS A 322  HIP A 321
322  GLY  322 GLY GLY A 323  GLY A 322

380  HIS  380 HIS HIS A 381  HID A 380
381  TYR  381 TYR TYR A 382  TYR A 381
382  THR  382 THR THR A 383  THR A 382
383  ASP  383 ASP ASP A 384  ASP A 383
384  TRP  384 TRP TRP A 385  TRP A 384
385  LEU  385 LEU LEU A 386  LEU A 385
386  HIS  386 HIS HIS A 387  HIP A 386

404  HIS  404 HIS HIS A 405  HID A 404
405  ASN  405 ASN ASN A 406  ASN A 405
406  VAL  406 VAL VAL A 407  VAL A 406
407  VAL  407 VAL VAL A 408  VAL A 407
408  CYS  408 CYS CYS A 409  CYX A 408

430  GLU  430 GLU GLU A 431  GLU A 430
431  HIS  431 HIS HIS A 432  HIE A 431
432  ARG  432 ARG ARG A 433  ARG A 432

445  PRO  445 PRO PRO A 446  PRO A 445
446  HIS  446 HIS HIS A 447  HIP A 446
447  GLY  447 GLY GLY A 448  GLY A 447

527  ALA  527 ALA ALA A 528  ALA A 527
528  CYS  528 CYS CYS A 529  CYX A 528
529  ALA  529 ALA ALA A 530  ALA A 529
```

Fonte: Autores.

A Figura 47 apresenta a 4EY6gnt solvatada e ionizada. Foi apresentada apenas uma delas, pois os resultados foram similares em todos os complexos em termos de numerações de átomos.

Figura 47 – 4EY6gnt – Solvatada e Ionizada. Na imagem são apresentadas a proteína, o ligante, as moléculas de água e os íons NaCl.



Fonte: Autores.

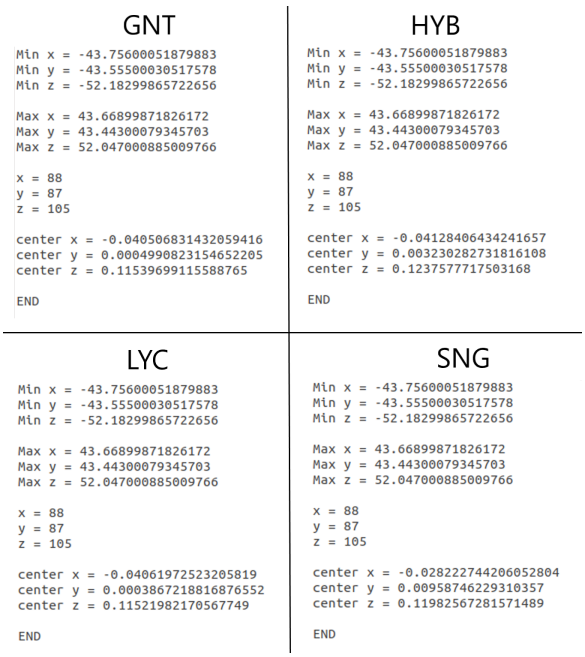
A última etapa é a geração dos arquivos finais para a realização da dinâmica. Nesse ponto, todos os arquivos necessários já foram preparados (PDB, *prmtop*, *inpcrd*). Esses arquivos já possibilitam que uma dinâmica seja realizada. Agora é momento de realizar a preparação dos arquivos de configuração e de restrições. O restante já está todo disponível. Para isso foi utilizado o arquivo PDB e os dados das *tags Free* disponíveis no arquivo de configuração. No caso desse trabalho foram gerados 9 arquivos de restrição para cada um dos ligantes, um para cada simulação que será realizada (1 minimização, 7 de relaxamento e a de produção final).

Outros arquivos gerados nessa etapa são os arquivos de configuração. Esses arquivos possuem a parte de PME gerada automaticamente pelo sistema. Para gerar esses dados é necessário obter algumas informações dos arquivos relacionadas ao posicionamento da sua estrutura. Assim, o centro geométrico da molécula e os pontos mínimo e máximo da molécula são calculados pelo VMD. Os resultados são depois utilizados nos cálculos de PME.

Os dados de centro geométrico e valores de mínimo, máximo e coordenadas x, y e z dos pontos *minmax* são calculados a partir das coordenadas x, y, z de cada um dos átomos existentes no complexo. Apesar dos valores da proteína serem todos iguais, uma vez que é utilizada a mesma proteína modificada para geração dos arquivos PDB, os ligantes, moléculas de água e íons possuem coordenadas diferentes, logo os dados serão diferentes de um ligante para o outro. Na Figura 48 estão os dados referentes aos arquivos *Smina*. Da mesma forma possuem poucas diferenças entre os arquivos.

De posse desses arquivos são gerados os dados de PME dos arquivos de configuração. Os dados serão os mesmos para todos os arquivos de configuração de todas as simulações de cada ligante, variando apenas de ligante para ligante. A Figura 49 apresenta o trecho de cada uma das moléculas do *Vina*. Não serão apresentados os dados do *Smina*, uma vez que as alterações neles são pequenas e irão seguir o que se obteve no arquivo do centro geométrico *Smina*.

Figura 48 – Arquivos Centro Geométrico - *Smina*. Apresentação dos valores de cada um dos ligantes.



GNT

```
Min x = -43.75600051879883
Min y = -43.55500030517578
Min z = -52.18299865722656

Max x = 43.66899871826172
Max y = 43.44300079345703
Max z = 52.047000885009766

x = 88
y = 87
z = 105

center x = -0.040506831432059416
center y = 0.0004990823154652205
center z = 0.11539699115588765

END
```

HYB

```
Min x = -43.75600051879883
Min y = -43.55500030517578
Min z = -52.18299865722656

Max x = 43.66899871826172
Max y = 43.44300079345703
Max z = 52.047000885009766

x = 88
y = 87
z = 105

center x = -0.04128406434241657
center y = 0.003230282731816108
center z = 0.1237577717503168

END
```

LYC

```
Min x = -43.75600051879883
Min y = -43.55500030517578
Min z = -52.18299865722656

Max x = 43.66899871826172
Max y = 43.44300079345703
Max z = 52.047000885009766

x = 88
y = 87
z = 105

center x = -0.04061972523205819
center y = 0.0003867218816876552
center z = 0.11521982170567749

END
```

SNG

```
Min x = -43.75600051879883
Min y = -43.55500030517578
Min z = -52.18299865722656

Max x = 43.66899871826172
Max y = 43.44300079345703
Max z = 52.047000885009766

x = 88
y = 87
z = 105

center x = -0.028222744206052804
center y = 0.00958746229310357
center z = 0.11982567281571489

END
```

Fonte: Autores.

Figura 49 – Trecho PME Arquivos de configuração - *Vina*. Trecho dos arquivos de configuração que apresentam os dados do PME que utilizaram os valores calculados no Centro Geométrico.

GNT

```
# Periodic Boundary Conditions
cellBasisVector1   88.    0.    0.
cellBasisVector2    0.   87.    0.
cellBasisVector3    0.    0.  105.
cellOrigin         -0.017462824187473381 0.010280743628876544 0.12522638049073292
wrapAll             on

# PME (for full-system periodic electrostatics)
PME                 yes
PMEGridSizeX        125
PMEGridSizeY        125
PMEGridSizeZ        125
```

HYB

```
# Periodic Boundary Conditions
cellBasisVector1   88.    0.    0.
cellBasisVector2    0.   87.    0.
cellBasisVector3    0.    0.  105.
cellOrigin         -0.0339521402167113 0.011409090470222458 0.1273113618060112
wrapAll             on

# PME (for full-system periodic electrostatics)
PME                 yes
PMEGridSizeX        125
PMEGridSizeY        125
PMEGridSizeZ        125
```

LYC

```
# Periodic Boundary Conditions
cellBasisVector1   88.    0.    0.
cellBasisVector2    0.   87.    0.
cellBasisVector3    0.    0.  105.
cellOrigin         -0.016482838220581121 0.006322388841823038 0.12836401393596816
wrapAll             on

# PME (for full-system periodic electrostatics)
PME                 yes
PMEGridSizeX        125
PMEGridSizeY        125
PMEGridSizeZ        125
```

SNG

```
# Periodic Boundary Conditions
cellBasisVector1   88.    0.    0.
cellBasisVector2    0.   87.    0.
cellBasisVector3    0.    0.  105.
cellOrigin         -0.038697949184671336 0.009293888024035407 0.11124415962368384
wrapAll             on

# PME (for full-system periodic electrostatics)
PME                 yes
PMEGridSizeX        125
PMEGridSizeY        125
PMEGridSizeZ        125
```

Fonte: Autores.

Nesse ponto todos os arquivos de preparação da dinâmica foram gerados e todos foram verificados para saber se a preparação está adequada e bem feita para cada um dos ligantes no complexo. É importante esse passo de verificação dos arquivos pelo pesquisador. Apesar de o ProtCool já realizar a geração de todos os arquivos, existem observações e análises que só o pesquisador pode realizar. Um exemplo foi a questão do *docking* neste estudo de caso, que não conseguiu encontrar a melhor conformação na primeira execução, só sendo possível esta avaliação após a observação de todos os dados. O conhecimento da proteína e dos ligantes com os quais se está trabalhando é crucial nesse processo. A tomada de decisão do pesquisador foi importante uma vez que ela foi realizada antes de se iniciar a dinâmica molecular e se gastar tempo de máquina desnecessário. Além do mais, analisar passo a passo é muito mais fácil do que ao final tentar descobrir onde pode estar o erro.

Assim, uma ferramenta extremamente importante nesse processo é a possibilidade dada pela ProtCool de se fazer o processo de preparação por partes. Isso permite que o pesquisador coloque pontos de checagem e vá avaliando cada uma das informações que lhe são fornecidas antes de se passar para a próxima fase. Isso poderá garantir melhor qualidade para a pesquisa. Outro fator que deve ser considerado nesse momento é que as proteínas e ligantes, cada qual tem a sua particularidade e ter esses pontos de checagem acabam sendo essenciais no processo como um todo.

Para se realizar toda a parte de preparação dos arquivos para a dinâmica molecular, foi gasto um tempo total de 1 hora e 20 minutos, lembrando que foram preparados 4 ligantes, realizando dois tipos de *docking* (*Vina* e *Smina*) em triplicata e flexível para cada um dos ligantes. Além disso, após passar pelo processo de *docking*, todos os arquivos posteriores foram gerados para as duas moléculas selecionadas. Todo o processo foi executado em uma máquina virtual MAC.

O tempo gasto em cada uma das etapas do *workflow* foram:

i. Configuração Inicial das Moléculas – menos de um minuto para fazer as 4 atividades, contando com a tarefa de buscar arquivo no *site* RSCB;
ii. *Docking* Molecular – cerca de uma hora. Essa foi a etapa que mais demorou para gerar os resultados. A parte inicial de buscar os arquivos de ligantes e preparar os dados para que fosse realizado o *docking* gastou 16 segundos. Cada um dos *docking Vina* e *Smina* gastou cerca de 30 minutos para serem executados. E para realizar a parte final do *docking* de verificar as poses até realizar a escolha da pose gastou 13 segundos;
iii. Modelagem Molecular – gastou 1 minuto;
iv. Protonação – gastou cerca de 14 minutos;
v. Campo de Força, Solvatação e Ionização – gastou cerca de 4 minutos;
vi. Preparar Arquivos da Dinâmica – gastou cerca de 5 minutos.

O Processo de preparação de dinâmicas moleculares, quando realizado pelo pesquisador, sem o auxílio do ProtCool, demanda a utilização e conhecimento de diversas ferramentas. Além disso, é gasto um tempo do pesquisador para que as moléculas sejam preparadas. Este tempo do pesquisador pode ser de um dia inteiro, podendo se estender por mais tempo. Quando o processo é desenvolvido pelo ProtCool, para que fossem entregues todos os arquivos prontos para que a dinâmica possa ser realizada, o tempo gasto foi mínimo (1 hora e 20 minutos). Além disso, deve-se considerar, que enquanto o computador está gerando esses arquivos, o pesquisador não precisa ficar em frente à máquina, uma vez que ao final de todo o processo, será enviado o *e-mail* informando que os arquivos estão prontos e que podem ser avaliados pelo pesquisador. Com isso, o tempo que o pesquisador passa preparando a pesquisa torna-se muito menor utilizando o ProtCool_Dynamic, do que realizando tudo manualmente. O tempo gasto para o estudo das moléculas, da metodologia e dos parâmetros a serem considerados, também precisam ser realizados na preparação tradicional de dinâmica molecular. O tempo efetivo do pesquisador é o tempo de preparação do arquivo de configuração que conterá o protocolo que será seguido na pesquisa.

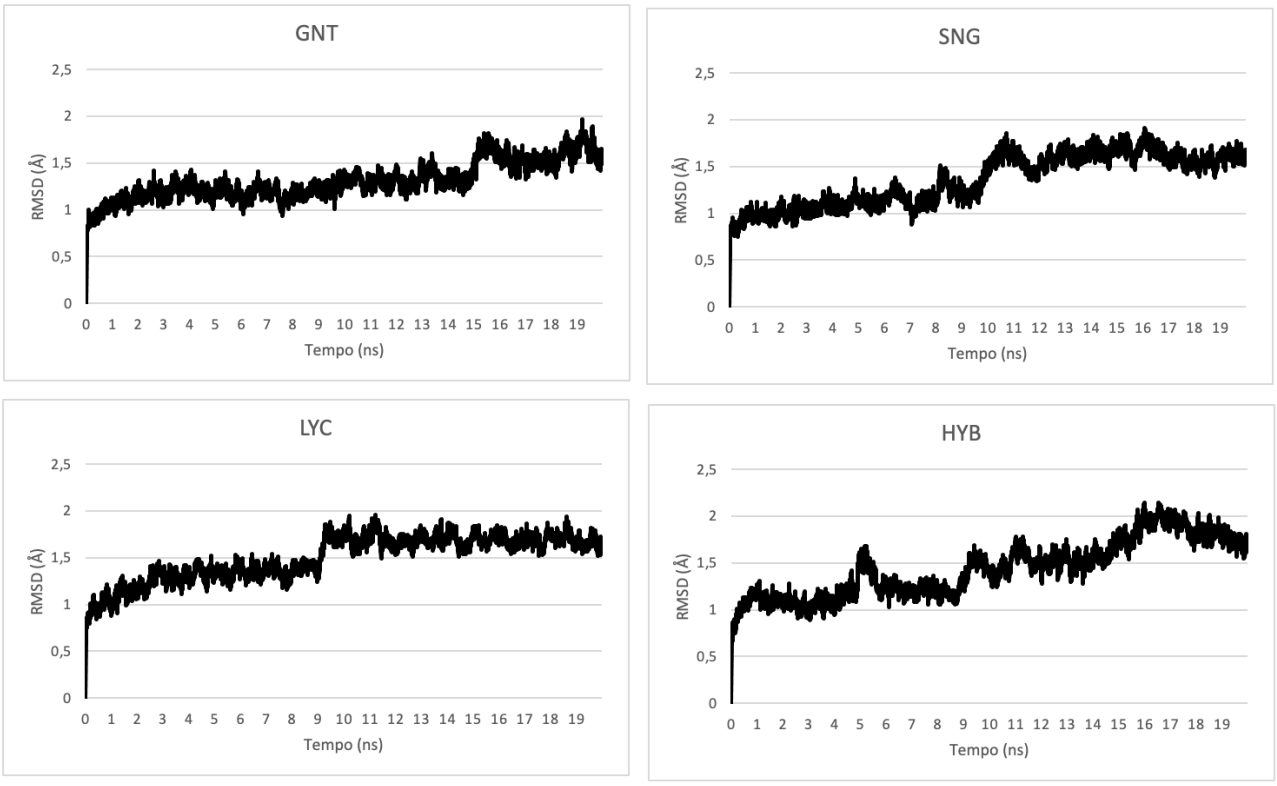
Apesar de se ter gerado todos os arquivos para *Vina* e *Smina*, apenas foi realizada a dinâmica dos arquivos *Vina*. Como os arquivos estão validados tanto pela resposta do *Smina* quanto pela comparação com os dados de Rocha (2017), resolveu-se seguir com a realização da dinâmica apenas do *Vina*. Foi gerada uma única rodada de simulação de 20 ns cada. Essa escolha foi feita uma vez que Rocha (2017) realizou 20 ns de simulação da 4EY6. Não se realizou em triplicata uma vez que o objetivo é apenas validar se os arquivos de preparação criados são suficientes para a realização da simulação de dinâmicas moleculares.

A simulação foi realizada segundo os parâmetros definidos na Seção 4.2. Ao final da simulação foram gerados 5000 *frames* de cada um dos ligantes no complexo. A análise de RMSD das trajetórias foi realizada para identificar se a simulação está adequada. Todos os RMSD foram gerados contra o primeiro *frame* e considerando-se o *backbone* da proteína

A Figura 50 apresenta os RMSD das trajetórias do complexo. Em todas elas, o RMSD do complexo com a 4EY6 está em preto. Na Figura 51 estão os RMSD calculados por Rocha (2017).

Na Figura 51 estão representadas não só o RMSD da 4EY6, mas também do 3LII, além de duas trajetórias de ligantes livres. Para a realização da análise neste trabalho, está se considerando na figura de Rocha (2017) apenas o RMSD da trajetória da 4EY6 que está em preto na imagem.

Figura 50 – RMSD de trajetórias 4EY6 – Complexo em preto. RMSD de cada trajetória de simulação de MD contra o primeiro *frame* da simulação. O RMSD do complexo foi calculado considerando-se o *backbone* da proteína.

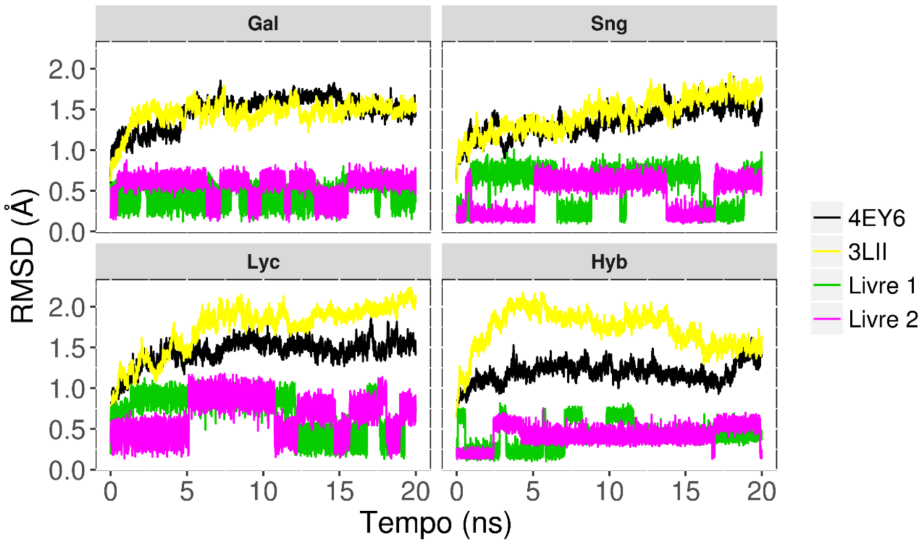


Fonte: Autores.

Comparando os gráficos de RMSD, os dois conjuntos de gráficos apresentam a mesma tendência qualitativa. Os RMSD sugerem que as simulações estão com tendência de equilíbrio, indicando que as simulações estão adequadas. Conforme destacado na Seção 4.1, a 4EY6 possui um sítio ativo profundo e estreito. Isso faz com que quando o ligante está dentro do complexo, exista pouca margem para que

ele possa se movimentar, assim, ligantes dentro do complexo acabam por possuírem um menor grau de liberdade.

Figura 51 – RMSD de trajetórias do complexo. Imagem retirada de Rocha (2017). Apresenta o RMSD de cada trajetória de simulação de MD relacionada ao *frame* inicial da simulação. Na imagem existem as simulações do 4EY6, 3LII e as duas simulações de cada ligante. Os valores de RMSD dos complexos foram calculados considerando o *backbone* da proteína.



Fonte: Rocha (p. 54, 2017).

Pela análise visual das dinâmicas, percebe-se que o ligante permanece estável no sítio ativo, demonstrando que uma preparação inicial foi bem feita, além disso, os dados selecionados para a dinâmica de equilíbrio também podem ser considerados como adequados ao problema. A Figura 52 mostra a 4EY6GNT no *frame* 1 e no *frame* 5000. Pela imagem é possível perceber que o ligante de fato, no primeiro e no último *frame* permanecem no sítio ativo, bem próximo dos resíduos de ligação no sítio ativo. Pela imagem do *frame* 1, é possível perceber que o ponto de partida do ligante pode ser considerado bom, uma vez que está bem localizado no seu sítio ativo, bem próximo do TRP85 (nova numeração depois da finalização da preparação). Isso sugere que tanto a pose do *docking*, quanto os parâmetros de restrição harmônica, quanto as dinâmicas iniciais (minimização e relaxamento) foram adequadas para o complexo.

As Figura 53, Figura 54 e Figura 55 apresentam as demais estruturas (HYB, LYC e SNG, respectivamente). Por elas também é possível verificar a mesma coisa que foi discutida na GNT, ou seja, a pose do *docking* foi adequada e assegura-se que a minimização e relaxamento tenham sido também adequados para o complexo.

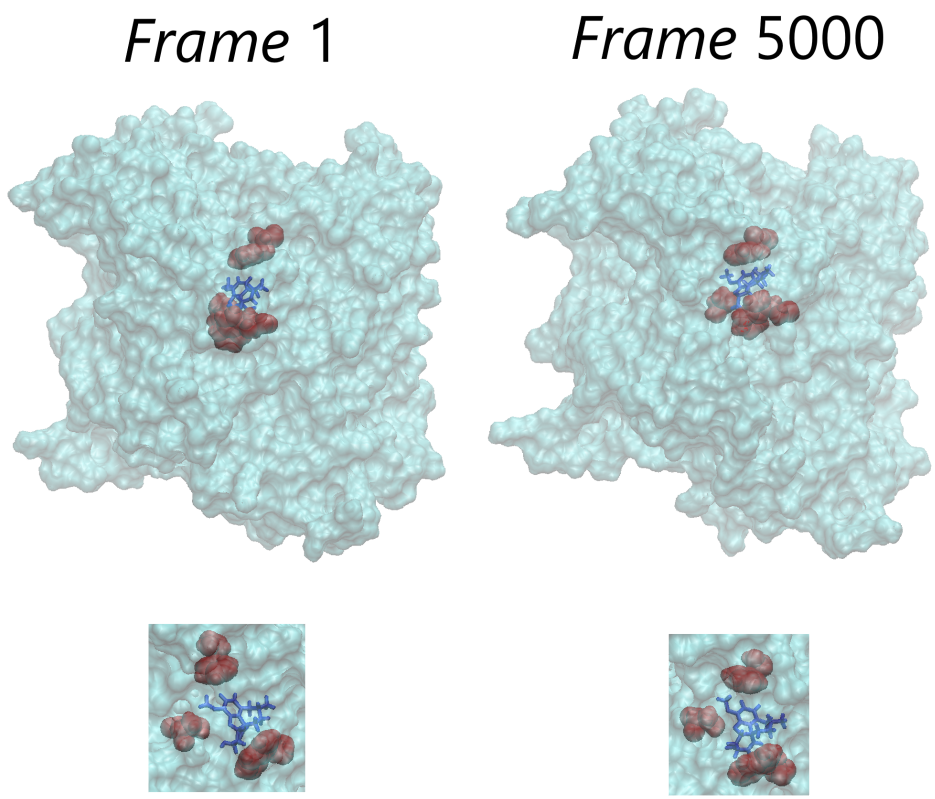
A Figura 56 apresenta o ligante no complexo. Cada imagem é a apresentação dos diversos quadros, com o ligante alinhado, possibilitando a visualização das inúmeras conformações que foram alcançadas pelos ligantes durante a simulação. Comparando os ligantes complexados GNT e SNG com o HYB e LYC, os dois primeiros têm menor grau de liberdade próximo ao anel com a dupla ligação do GNT e SNG, sugerindo que essa ligação dupla entre os C5 e C6 do anel acaba por tornar o ligante mais rígido em relação aos demais, fazendo com que o ligante fique com menor mobilidade.

Foi realizada a análise de *cluster* do complexo. Usando o *Wordon*[43], foram divididas as trajetórias do complexo em 3 trajetórias distintas, cada uma delas possuindo apenas os *frames* de cada um dos

[43] Comando: ~/wordom -F Cluster3.txt -itrj 4EY6gnt9.dcd -otrj 4EY6gntCluster3.dcd

clusters existentes. Para isso, é passado para o *Wordon* uma lista de todos os *frames* que pertencem a um determinado *cluster*, e o software cria uma trajetória apenas com estes *frames*. A partir dessas novas trajetórias foi utilizado o software VMD para que pudesse visualizar as múltiplas trajetórias em cada *cluster*. A Figura 57 mostra as conformações complexadas de cada ligante. São apresentados na primeira coluna os *clusters* sobrepostos (trajetória completa), na segunda coluna estão o *cluster* 0, na terceira o *cluster* 1 e na quarta coluna o *cluster* 2.

Figura 52 – 4EY6GNT – *Frame* 1 e 5000. Na imagem são apresentadas as imagens do *frame* 1 e do *frame* 5000 do 4EY6GNT. Na imagem são apresentadas duas visões de cada *frame*, uma mostrando a proteína inteira e outra mostrando mais o detalhe próximo ao ligante.

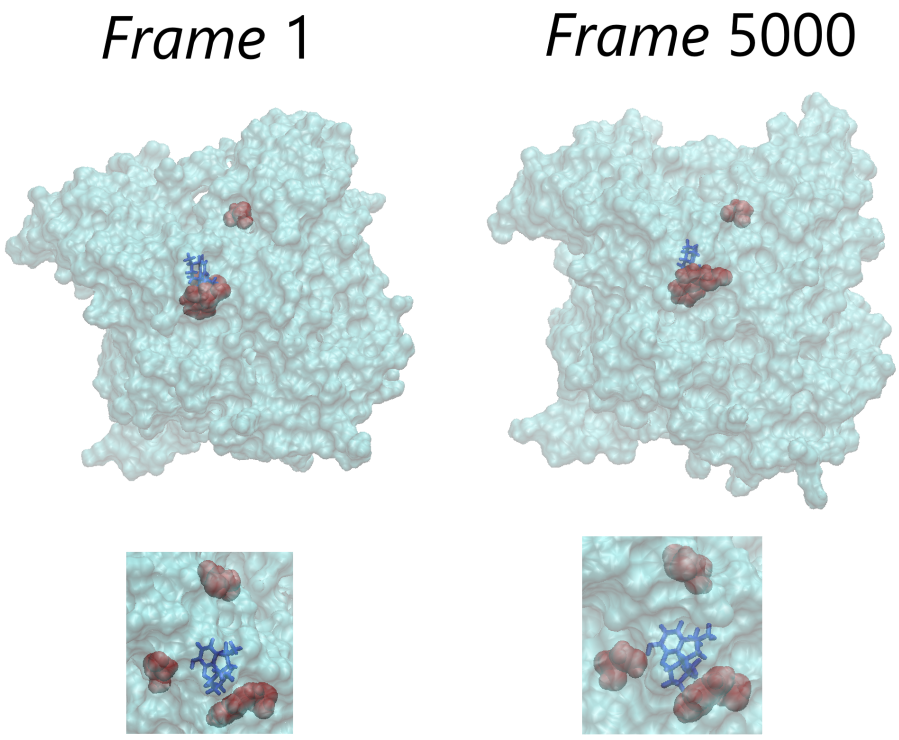


Fonte: Autores.

Observando-se os complexos, de fato a mobilidade dos ligantes complexados é restrita, indicando que apesar de todos os ligantes possuírem 3 *clusters*, apenas o *cluster* 0 é o que possui um maior povoamento, indicando que as conformações são bem próximas entre si. Além disso, observa-se que tanto no LYC, quanto no HYB existem maiores populações nos *clusters* 1 e 2 que em relação ao GNT e SNG, o que corrobora com as análises realizadas anteriormente.

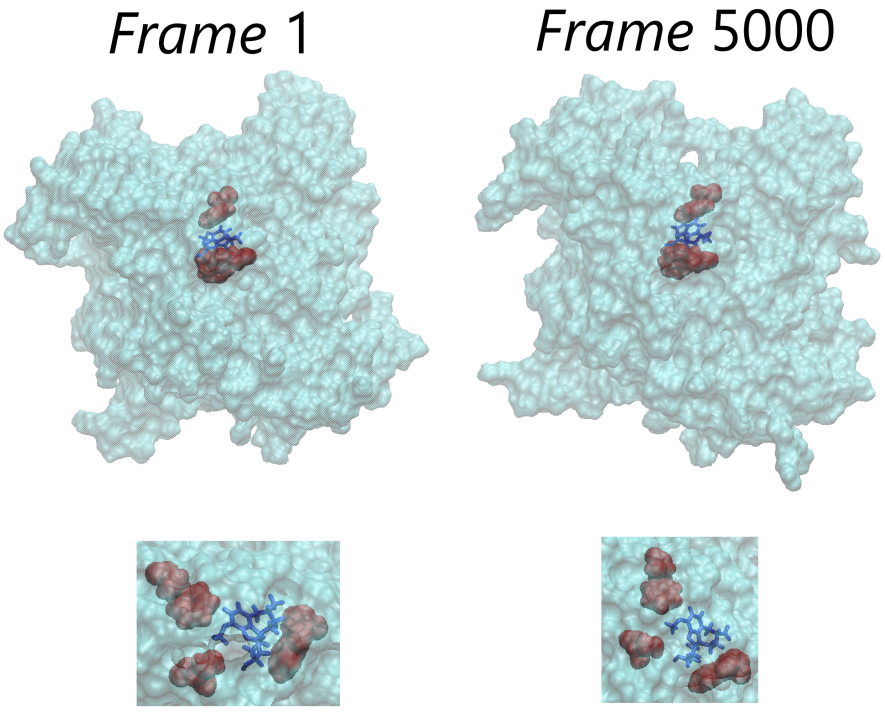
Essas análises das trajetórias obtidas sugerem que as simulações de dinâmica molecular estão condizentes com o que se espera na natureza, ou seja, os ligantes apresentam pouca mobilidade dentro do sítio ativo por ser um sítio estreito e que os ligantes com ligação dupla no anel possuem menos mobilidade comparado aos outros ligantes. Assim, pode-se classificar que a preparação da dinâmica molecular do complexo (ProtCool_Dynamic) fornece bons resultados, possibilitando que dinâmicas moleculares sejam devidamente realizadas.

Figura 53 – 4EY6HYB – *Frame* 1 e 5000. Na imagem são apresentadas as imagens do *frame* 1 e do *frame* 5000 do 4EY6HYB. Na imagem são apresentadas duas visões de cada *frame*, uma mostrando a proteína inteira e outra mostrando mais o detalhe próximo ao ligante.



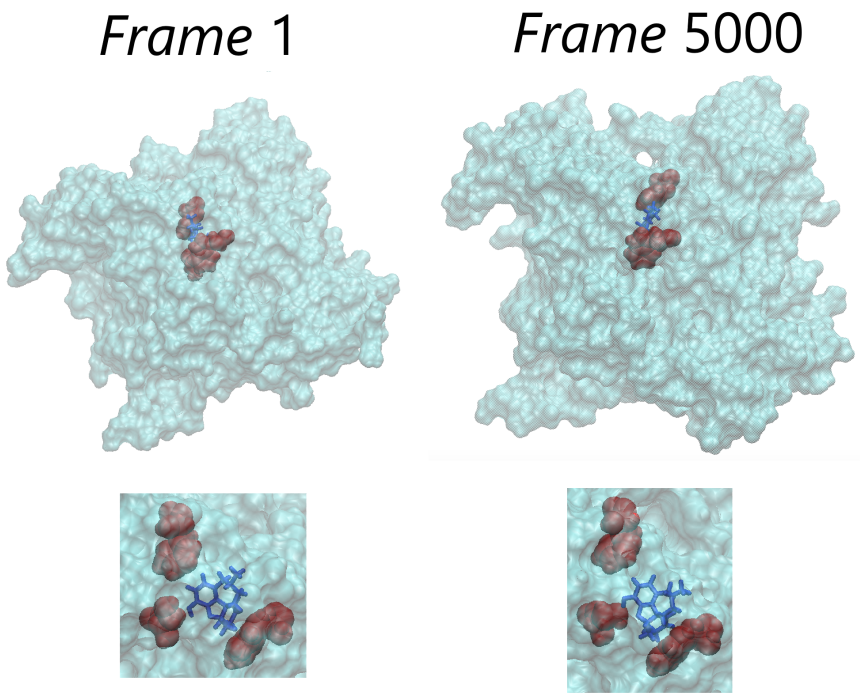
Fonte: Autores.

Figura 54 – 4EY6LYC – *Frame* 1 e 5000. Na imagem são apresentadas as imagens do *frame* 1 e do *frame* 5000 do 4EY6LYC. Na imagem são apresentadas duas visões de cada *frame*, uma mostrando a proteína inteira e outra mostrando mais o detalhe próximo ao ligante.



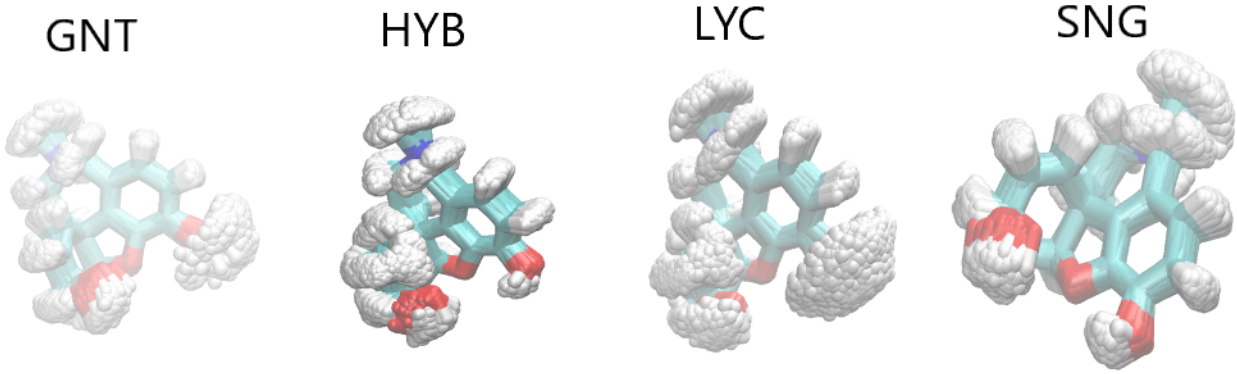
Fonte: Autores.

Figura 55 – 4EY6SNG – *Frame* 1 e 5000. Na imagem são apresentadas as imagens do *frame* 1 e do *frame* 5000 do 4EY6SNG. Na imagem são apresentadas duas visões de cada *frame*, uma mostrando a proteína inteira e outra mostrando mais o detalhe próximo ao ligante.



Fonte: Autores.

Figura 56 – Simulações – *Frame* 1 ao 5000. Apresentação de todos os quadros, com o ligante alinhado, possibilitando a visualização das conformações ao longo da trajetória.



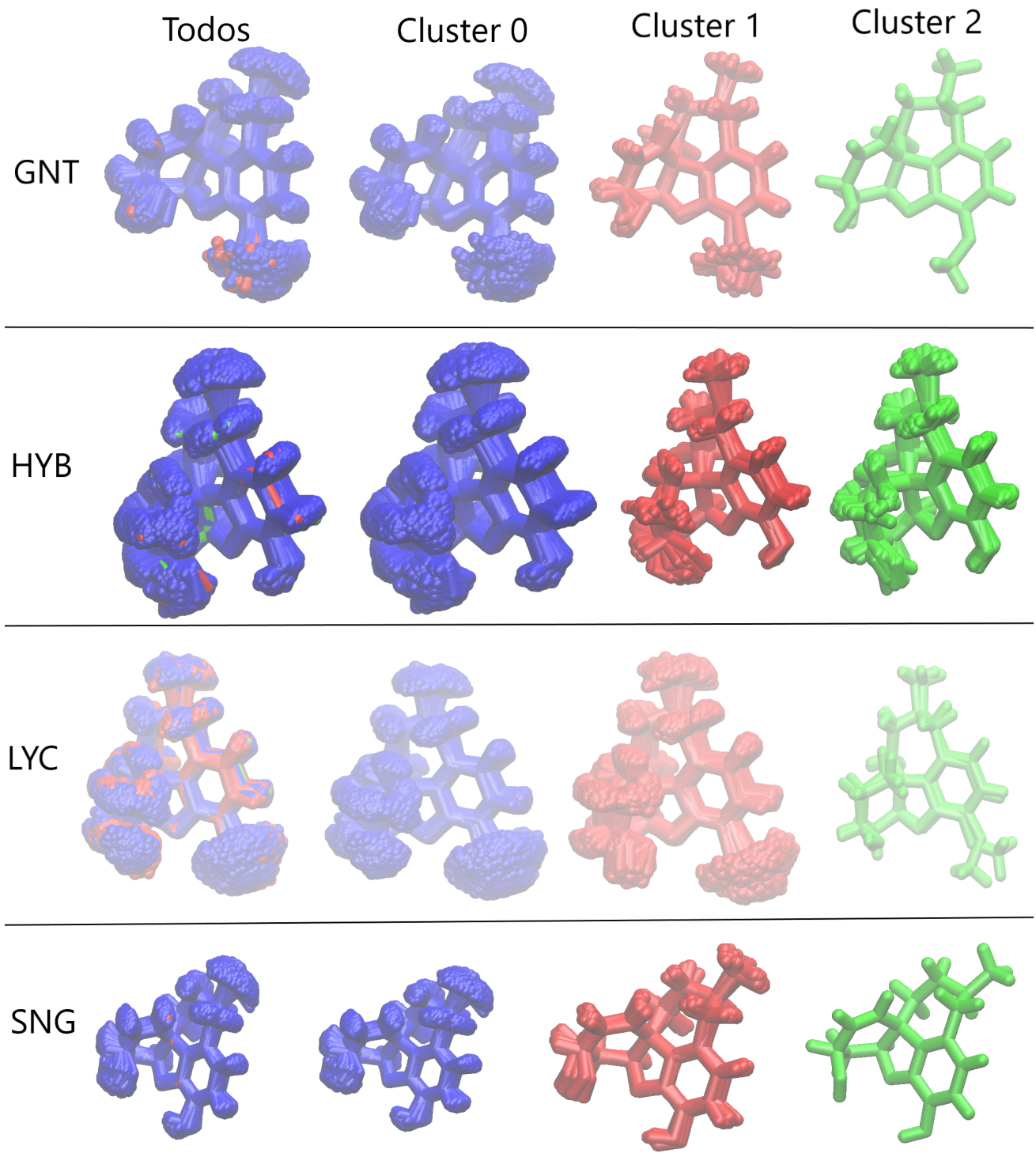
Fonte: Autores.

A utilização do ProtCool_Dynamic traz algumas vantagens em relação a fazer as preparações sem a utilização de uma ferramenta automatizada, como é o caso da aqui apresentada. Uma das vantagens é a facilidade para geração dos arquivos. O pesquisador, precisa apenas criar o arquivo de configuração, de forma a garantir que o protocolo desejado será realizado e, assim, todos os arquivos são gerados e armazenados de forma organizada em uma pasta especificada pelo pesquisador.

Uma vantagem da utilização da ferramenta, é que, uma vez que a máquina esteja toda configurada e o arquivo de configuração preparado, um único comando de execução faz com que a preparação seja realizada. Isso contribui para o dia a dia dos pesquisadores, que conseguem fazer preparações de dinâmica molecular rapidamente. Além disso, usuários com pouca experiência conseguem realizar a

preparação de dinâmicas moleculares. A ProtCool pode ser utilizada em treinamentos, facilitando que novos pesquisadores sejam inseridos nos laboratórios e realizem as pesquisas em simulação de dinâmica molecular.

Figura 57 – Análise de *Clusters* – Conformações – Ligantes Complexados. São apresentados na primeira coluna os *clusters* sobrepostos (trajetória completa), na segunda coluna estão o *cluster* 0, na terceira o *cluster* 1 e na quarta coluna o *cluster* 2.



Fonte: Autores.

Uma das maiores vantagens da ProtCool é a possibilidade de geração de dinâmicas moleculares para múltiplos ligantes, ou seja, dado um receptor e diversos ligantes, os arquivos de preparação de simulação de dinâmica molecular serão gerados para todos os ligantes. Essa facilidade de preparar inúmeros ligantes oferece uma melhor gestão dos experimentos e, ao mesmo tempo, amplia as possibilidades de pesquisa.

Outro fator a ser considerado é a reprodutibilidade de experimentos. A reprodutibilidade é tanto de protocolos quanto da preparação. É possível, passar para outros pesquisadores apenas o arquivo de configuração e, assim, pode-se gerar a mesma preparação já realizada. Da mesma forma, é possível definir um protocolo padrão e preparar diversos complexos a partir de um mesmo protocolo definido no arquivo de preparação.

Todos esses fatores fazem com que a utilização do ProtCool traga maior capacidade de gestão dos experimentos, o que acaba por gerar maior qualidade às pesquisas efetuadas.

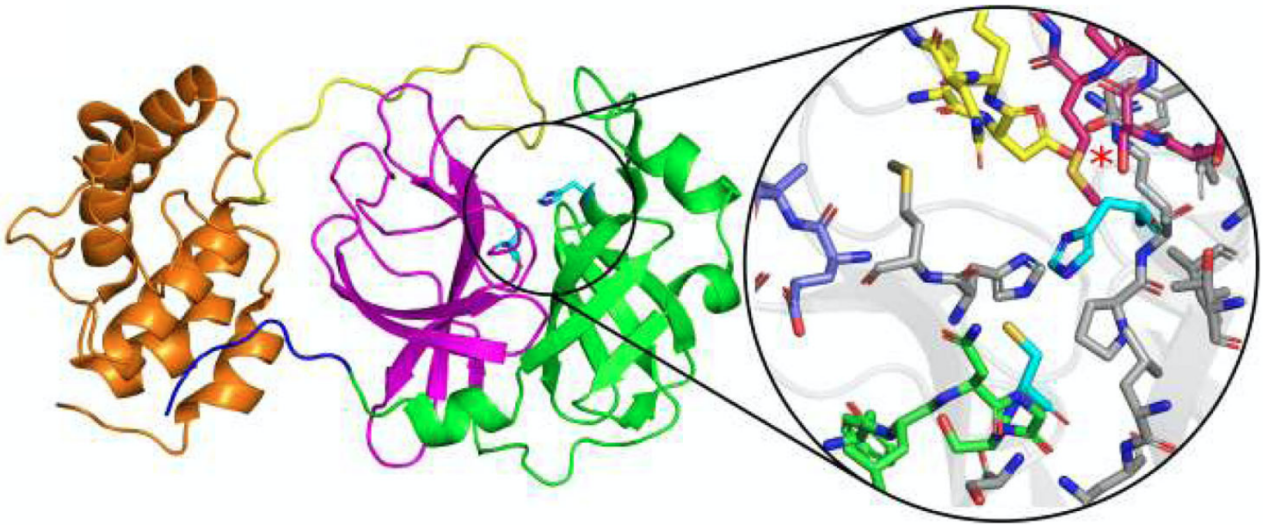
5. MPRO DA SARS-COV-2

Para utilização do *ProtCool_Docking* foi realizado um estudo a respeito da MPro (*Main-protease*) do SARS-Cov-2 (doravante MPro2).

A MPro2 é uma enzima que desenvolve um papel chave na replicação do SARS-Cov_2, especialmente na clivagem da poli-proteína viral. Devido a isso pode ser um alvo interessante para o desenvolvimento de fármacos (JIN *et al.*, 2020). A Mpro2 guarda homologia com a Mpro do SARS-CoV (doravante Mpro1), e diferem entre si por 12 pontos de mutação.

A Figura 58 apresenta a estrutura cristalográfica da MPro2 representada em cartoon, que foi retirada de Rocha *et al.* (2021). A MPro2 é formada por 3 domínios (I, II e III). Em verde está o domínio I, em rosa o domínio II e em laranja o domínio III. O sítio ativo fica na fenda que se apresenta entre os domínios I e II. No detalhe da imagem é possível visualizar o sítio ativo da estrutura. O domínio I é formado pelos resíduos de 8 a 101, o domínio II pelos resíduos de 102 a 184 e domínio III dos resíduos 201 a 303. Na fenda que se encontra entre os domínios I e II existe o sítio de ligação para os potenciais inibidores (JIN *et al.*, 2020).

Figura 58 – Estrutura cristalográfica da Mpro2 em cartoon – Sars-Cov-2.



Fonte: Rocha *et al.*, p.2, 2021.

5.1.Metodologia

Este estudo se insere dentro de um projeto maior que envolve pesquisadores da Universidade Federal de Minas Gerais - UFMG, University of Salzburg – Austria, Universidade Federal da Paraíba - UFPB, Universidade Federal de São João Del Rei - UFSJ, Universidade Federal de Itajubá - Unifei, Universidade Federal do Rio Grande – UFRG e do Laboratório Nacional de Computação Científica – Petrópolis e que gerou o artigo Rocha *et al.* (2021).

Dentro do trabalho proposto, a ProtCool_Docking foi utilizada para realizar o docking de múltiplos ligantes, usando o Vina e o Smina. O *docking* deveria ao final do processo entregar o conjunto de arquivos mol2 de todas as poses geradas, bem como, um arquivo final contendo o *score* de cada uma das conformações selecionadas.

Foi realizado o *docking* de múltiplos ligantes considerando:

i. 6 conformações diferentes do receptor. No caso foram fornecidos 6 arquivos pdbqt da MPro2 do Sars-Cov-2 das conformações desejadas no estudo;
ii. 19637 ligantes distintos da biblioteca ZINC;
iii. 8752 ligantes distintos da biblioteca *DrugBank*;
iv. 8520 ligantes distintos da biblioteca SistematX (base de produtos naturais e metabólitos secundários hospedada na Universidade Federal da Paraíba, Brasil).

Todos os arquivos de entrada do sistema, ligantes e proteína foram fornecidos no formato pdbqt. O conjunto de arquivos foi recebido em momentos diversos. Em um primeiro momento foram recebidas as bases *DrugBank* e SistematX, e, posteriormente, foi recebida a base ZINC. Para realização do estudo de caso foi utilizado um servidor que possui 64 núcleos.

Para que os núcleos fossem adequadamente utilizados, fazendo com que o servidor fornecesse sua capacidade máxima de processamento, foi estabelecida a seguinte estratégia de processamento dos dados:

i. Cada base de dados foi replicada 6 vezes, uma para cada conformação do receptor. Na primeira rodada, existiam 12 pastas ao todo e na segunda rodada existiam 6 pastas. O volume total de arquivos era similar, conforme descrito anteriormente;
ii. Em cada pasta foi executado o ProtCool_Docking, sendo especificado no arquivo de configuração a quantidade de processadores utilizada por cada um deles. Como ProtCool possui os *scripts* com processamento paralelo, os 64 núcleos da máquina foram utilizados ao mesmo tempo.

Os parâmetros utilizados na pesquisa estão apresentados na Figura 59. Além desses parâmetros, também se definiu que seria realizado *docking* rígido e apenas uma rodada seria realizada para cada ligante.

Figura 59 – Parâmetros *docking* definidos para a pesquisa. Parâmetros de *docking* a serem utilizados para a realização dos atracamentos.

```
center_x = 78.5
center_y = 65
center_z = 58
size_x = 38
size_y = 38
size_z = 38
num_modes = 10
energy_range = 4
exhaustiveness = 8
```

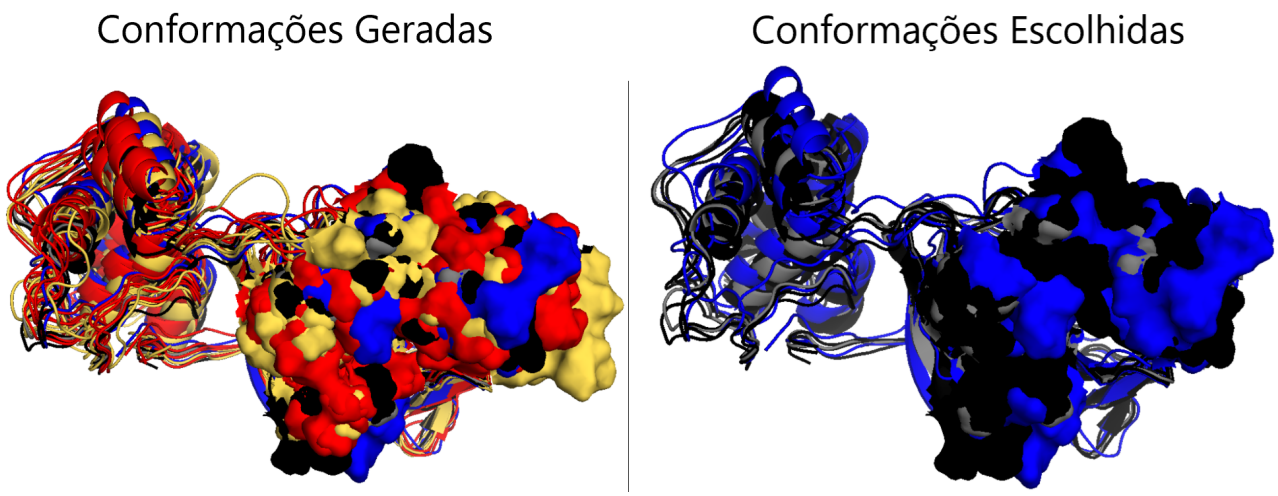
Fonte: Autores.

Após a geração de todos os dados, foi realizado um *redocking*. Para a realização do *redocking*, foram selecionados 84 complexos de Mpro2 cristalográficas encontradas nas bases. Para cada base foi preparado o seu ligante cristalográfico de duas formas distintas, com *am1bcc* e *gasteiger*. Para cada um desses elementos foram gerados os *docking Vina* e *Smina*. Assim sendo, ao todo foram gerados 168 arquivos *Vina* e 168 arquivos *Smina* com esses dados. Para cada um deles foram geradas ao todo 10 conformações possíveis.

5.2.Resultados e discussões

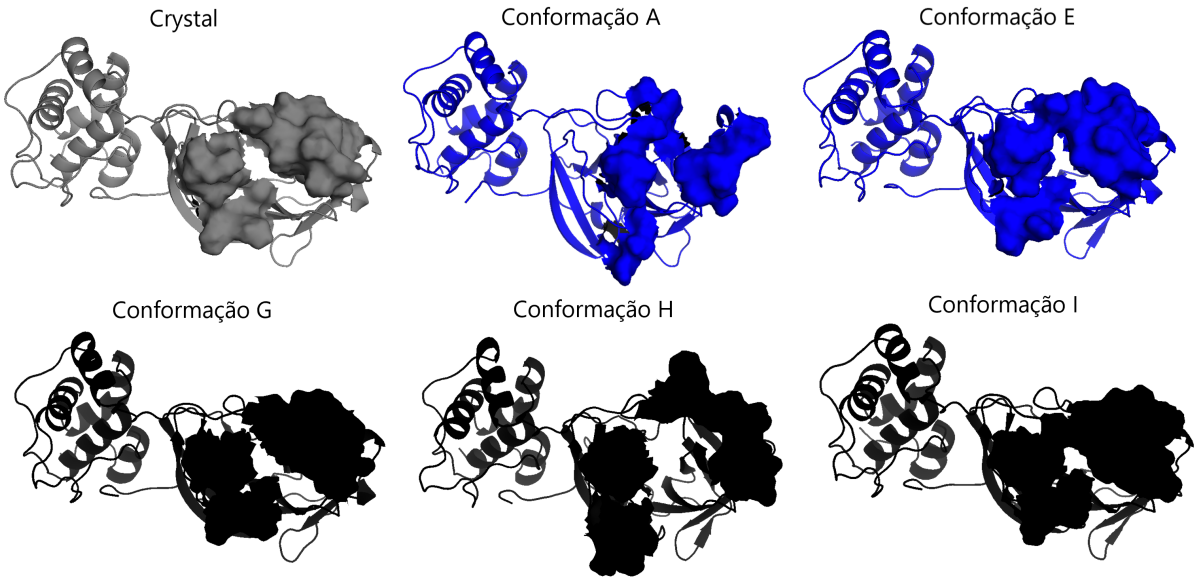
Conforme verificado na Seção 3.3 o primeiro passo é a definição do arquivo de configuração inicial. Todos os dados fornecidos foram disponibilizados no arquivo de configuração inicial, de acordo com o definido na Seção 5.1. Também foram criadas as pastas com a estrutura definida da Seção 5.1.

Figura 60 – Conformações MPro. Lado esquerdo estão 15 conformações mais a pose cristalográfica e à direita estão as 6 conformações escolhidas, incluindo a pose cristalográfica.



Fonte: Autores.

Figura 61 – Conformações MPro – Conformações trabalhadas. Apresentação de todas as 6 estruturas cristalográficas com as quais foram realizados os *dockings* com múltiplos ligantes.



Fonte: Autores.

Cada conjunto de seis pastas, contém os arquivos do receptor escolhido, de forma que, cada receptor foi atracado com todos os ligantes solicitados. A Figura 60 apresenta as conformações da MPro2 que foram geradas pelo processamento da metadinâmica. Do lado esquerdo estão todas as 15 conformações geradas, sobrepostas com a molécula cristalográfica da MPro2. À direita estão as 6 moléculas escolhidas, dentre as 6 está a molécula cristalográfica. A Figura 61 apresenta as seis conformações escolhidas para se fazer os *dockings*.

Após a criação do conjunto de pastas e do arquivo de configuração para a realização do *docking* de múltiplos ligante*s*, é possível iniciar o ProtCool_Docking. O primeiro passo foi então a criação do arquivo de configuração geral. Esse arquivo, conforme visualizado na Seção 3.3 servirá de dado de entrada de todo o processo de *docking* a ser realizado.

O segundo passo do processo é a criação dos arquivos de configuração dos *dockings* que serão realizados. É gerado um arquivo para cada *docking*, e todos os arquivos estão disponíveis na pasta *ligandconf*. Na Figura 62 existem 3 arquivos de configuração gerados, um de cada uma das bases de ligantes trabalhadas, todos com a proteína cristalográfica.

A Figura 63 apresenta os *pockets* definidos na pesquisa, desenhados via *EasyVs*, em cada um dos conformeros escolhidos.

Com esses dados, são realizados os *dockings Vina* e *Smina*. Para a sua execução foi definida a *tag CPUCluster* para que todas as execuções de todas as pastas fossem realizadas ao mesmo tempo no servidor. Foram geradas no máximo 10 poses para cada um dos ligantes existentes. Após gerar todas as poses, o ProtCool_Docking gera os arquivos mol2 e os arquivos com os resultados dos *scores* gerados.

Figura 62 – Exemplos Arquivos de Configuração do *Docking*. Na imagem são apresentados três exemplos de arquivos de configuração do *docking*, Cada um dos exemplos apresenta um ligante de uma das bases de dados, com a estrutura cristalográfica.

```
receptor = 00_crystal_covid-19_Mpro.pdbqt
ligand = ZINC000000148660_81.pdbqt

center_x = 78.5
center_y = 65
center_z = 58

size_x = 38
size_y = 38
size_z = 38

num_modes = 10
energy_range = 4
exhaustiveness = 8
cpu = 1
```

```
receptor = 00_crystal_covid-19_Mpro.pdbqt
ligand = DB00114.pdbqt

center_x = 78.5
center_y = 65
center_z = 58

size_x = 38
size_y = 38
size_z = 38

num_modes = 10
energy_range = 4
exhaustiveness = 8
cpu = 1
```

```
receptor = 00_crystal_covid-19_Mpro.pdbqt
ligand = sistx_13310.pdbqt

center_x = 78.5
center_y = 65
center_z = 58

size_x = 38
size_y = 38
size_z = 38

num_modes = 10
energy_range = 4
exhaustiveness = 8
cpu = 1
```

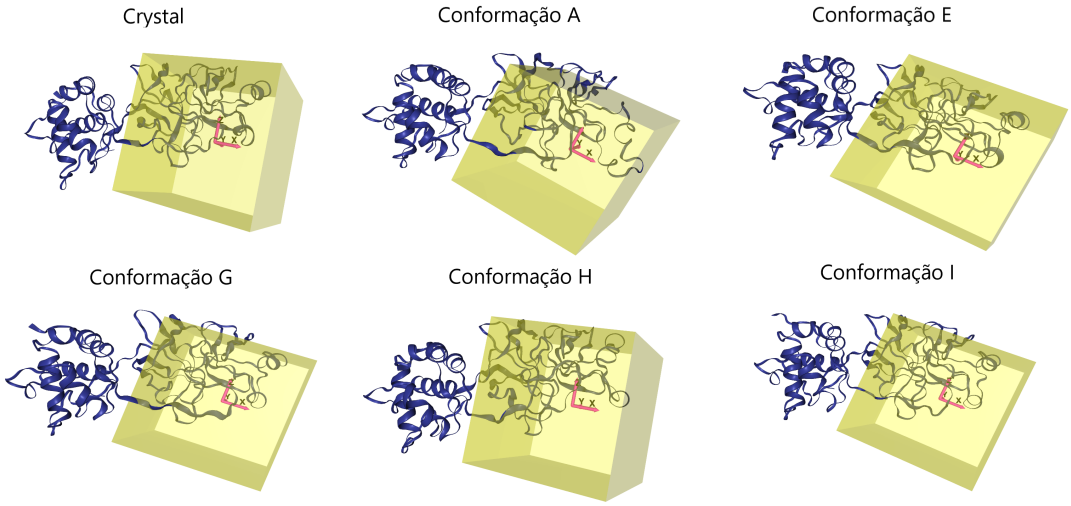
Fonte: Autores.

A Tabela 4 apresenta o número total de poses *Vina* e *Smina* esperados e o total de fato alcançados pelo ProtCool_Docking. O número de poses esperadas, considerando todas as bases e todas as 6 proteínas estudadas, seria um total de 4429080 poses (*Vina* e *Smina*). Foram geradas ao todo 4427839 poses (*Vina* e *Smina*). Ou seja, 1242 poses não foram geradas, o que representa 0.03% do total que deveria ter sido gerado. Ou seja, 99.97% da base teve suas poses geradas.

Marcado em verde na tabela estão todas as situações que conseguiram alcançar 100% de aproveitamento. Assim, pode-se observar que a melhor base foi a *Zinc* em que todas as poses foram adequadamente geradas, tanto no *Vina* quanto no *Smina*, tendo 100% de aproveitamento. A base SistematX conseguiu um melhor resultado no *Vina* do que no *Smina*. A base *DrugBank* foi a mais problemática das 3 bases de dados. Um total de nove ligantes não gerou poses nem no *Vina* e nem no *Smina,* em nenhuma das conformações apresentadas. Ao se verificar com maior atenção esses ligantes, percebeu-se que o *Vina* e o *Smina* geravam erro, acusando que o arquivo pdbqt fornecido possuía erros

em sua estrutura. Mesmo após nova geração dos arquivos, o erro persistiu. Assim, foi decidido que essas moléculas seriam consideradas nas próximas fases com valores de NA no *Vina* e no *Smina*.

Figura 63 – *Pockets* desenhados em cada conformação escolhida. Nas imagens é possível visualizar cada um dos conformeros selecionados com a caixa de docagem apresentada.



Fonte: Autores.

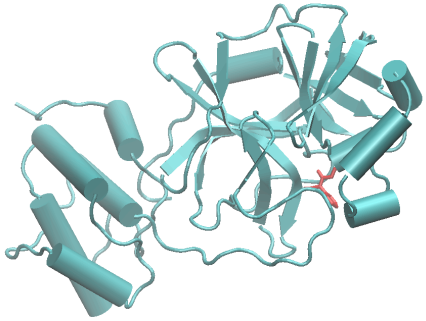
Tabela 4 – Número de Poses *Vina*, *Smina*. Na tabela é possível visualizar a quantidade de poses esperadas e geradas pelo processo (*Vina* e *Smina*). Em verde estão assinaladas todas as bases que obtiveram o número de poses esperado, para cada um dos conformeros.

Conformação	Base	Número Ligantes	Número poses esperadas	Poses *Vina*	Poses *Smina*
Crystal	**DrugBank**	8752	87520	87430	87430
	SistematX	8520	85200	85200	85200
	Zinc	19637	196370	196370	196370
A	**DrugBank**	8752	87520	87427	87427
	SistematX	8520	85200	85143	85128
	Zinc	19637	196370	196370	196370
E	**DrugBank**	8752	87520	87425	87430
	SistematX	8520	85200	85200	85200
	Zinc	19637	196370	196370	196370
G	**DrugBank**	8752	87520	87430	87430
	SistematX	8520	85200	85200	85197
	Zinc	19637	196370	196370	196370
H	**DrugBank**	8752	87520	87430	87427
	SistematX	8520	85200	85199	85197
	Zinc	19637	196370	196370	196370
I	**DrugBank**	8752	87520	87419	87430
	SistematX	8520	85200	85200	85199
	Zinc	19637	196370	196370	196370
Total		**221454**	**2214540**	**2213923**	**2213915**

Fonte: Autores.

Apenas a título de exemplo, foi gerado o PDB de uma pose da DB00265 na conformação cristalográfica para se ter uma visualização de um *docking* realizado. Deve-se atentar que o ProtCool_Docking não gera o arquivo PDB com o ligante em conjunto da proteína. Ele só tem como saídas os arquivos mol2 e o arquivo txt com os dados dos *scores*, logo, a Figura 64 não serve para se realizar análise dos dados. Outros fatores foram considerados para se realizar as devidas análises e que fogem do escopo deste trabalho.

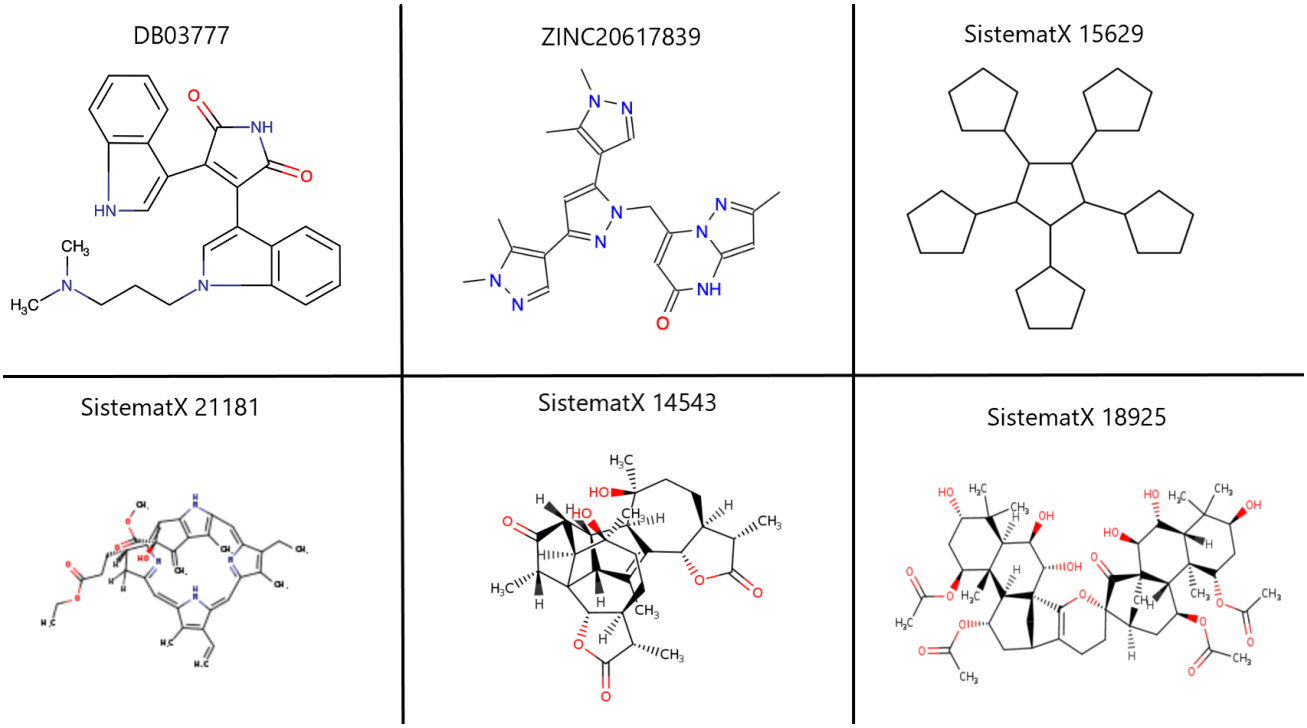
Figura 64 – Exemplo de ligante atracado. A pose gerada com a DB00265 e a Mpro2 cristalográfica.



Fonte: Autores.

O *redocking*, conforme descrito na Seção 5.1 foi realizado seguindo o mesmo padrão de processamento. No caso do *redocking* se esperava um total de 3360 poses geradas, o que foi alcançado pelo processo de *redocking*.

Figura 65 – Ligantes candidatos a fármaco selecionados pela pesquisa. Ao final de todas as análises realizadas pela pesquisa, foram identificados 6 ligantes candidatos a fármacos. Drugbank – DB03777, Zinc15 – 20617839, SistematX – 15629, 21181, 14543, 18925.



Fonte: Autores.

A análise geral dos dados tanto de *docking*, quanto de *redocking* não faziam parte do escopo do trabalho, uma vez que essa etapa serve como uma filtragem dos dados a serem de fato analisados. A análise dos resultados só foi realizada a partir da mineração dos dados, e, só considerou os ligantes que passassem dessa etapa. Assim sendo, não será realizada aqui uma análise em relação aos *scores* alcançados. Porém, após a análise realizada de todos os dados da pesquisa, foram indicados, no artigo de Rocha *et al.* (2021) desenvolvido na pesquisa, alguns ligantes como sendo promissores como candidatos a fármacos para a MPro2. São eles: Drugbank – DB03777, Zinc15 – 20617839, SistematX – 15629, 21181, 14543, 18925. A Figura 65 apresenta os ligantes. Os ligantes aqui apresentados foram identificados no artigo, porém, ainda é necessário um estudo aprofundado desses ligantes para de fato conseguir determinar se eles podem ser considerados candidatos a fármacos da MPro2. Observa-se que esses ligantes possuem uma grande diversidade estrutural e farmacofórica, e a indicação dessa diversidade foi o principal objetivo do artigo, que teve foco maior na estratégia de indicar novos ligantes que nos ligantes indicados em si.

O tempo de processamento foi de aproximadamente 76 horas (cerca de 3 dias) para cada um dos conjuntos de dados (em um servidor de 64 cores e 64 GB RAM). Totalizando para toda a pesquisa 152 horas (cerca de 6 dias) de processamento bruto. O *redocking* gastou menos de 1 dia para processamento. Aqui está se considerando apenas o tempo que o servidor ficou executando o ProtCool_Docking para geração dos dados.

Além desse tempo, foi gasto um tempo adicional para que todas as bases fossem validadas e checadas. Toda a parte de adequações do ProtCool, definição da pesquisa, recebimento de dados (pdbqt proteína e ligantes) processamento, validação, análises das inconsistências e análises dos resultados durou cerca de 1 mês, até a entrega final dos resultados.

Para a realização dessa pesquisa foi necessário a adequação do ProtCool, para passar a realizar o *docking* de múltiplos ligantes, conforme foi discutido na Seção 3.3. Esta adequação foi rápida, pois já existiam os scripts que foram adaptados para a realização de *docking* com múltiplos ligantes. Ter o ProtCool_Dynamic foi importante neste processo de desenvolvimento.

Vale lembrar que a forma de geração e armazenamento dos dados e a proveniência de dados realizada pela ProtCool foram responsáveis por facilitar a gestão de mais de 4 milhões de arquivos que foram gerados ao longo da execução do sistema. Como os arquivos já haviam sido separados em pastas distintas pelas diferentes conformações, a gestão e transferência dos arquivos para a continuidade da pesquisa foram importantes no processo.

Uma característica importante da ProtCool que foi utilizada e precisa ser destacada é a execução do sistema em paralelo. Isso contribuiu para que fosse utilizada toda a capacidade técnica disponível no servidor de escolha.

Uma outra vantagem da utilização da ProtCool é a facilidade de configuração do ambiente para que os *docking* de múltiplos ligantes possam ser executados. Com a cópia dos arquivos para a pasta e a preparação do arquivo de configuração, a execução do *script* de controle garante a execução dos múltiplos *docking*. Esta forma de trabalho, também garante que o processo seja realizado com uma baixa intervenção do pesquisador no processo.

A utilização do ProtCool_Docking possui diversas facilidades e controles que permitem a gestão eficiente de uma pesquisa, com um protocolo conforme o especificado nesse *workflow*. A ProtCool_Docking está preparada assim, para realizar *dockings* de múltiplos ligantes, possibilitando uma gestão eficiente de recursos computacionais, bem como, uma efetiva gestão de arquivos, facilitando a realização de *virtual screening* eficiente.

Uma grande vantagem identificada no ProtCool_Docking é a possibilidade de realização de *docking* de múltiplos ligantes utilizando-se duas ferramentas, com a gestão eficiente de arquivos. Além disso, tem que se destacar que é aceito tanto o *docking* rígido quanto o *docking* flexível, o que garante uma maior flexibilidade da ferramenta, possibilitando um maior poder de pesquisa para o usuário do ProtCool.

6. Conclusões

Neste trabalho foi desenvolvida uma ferramenta que busca a automação de protocolos de preparação de dinâmica molecular e o *docking* de múltiplos ligantes, chamada de ProtCool. A preparação de dinâmicas moleculares para complexos realizada pela ProtCool_Dynamic fornecem resultados eficientes para a preparação de dinâmica molecular, e o *workflow* de *docking* de múltiplos ligantes também alcançou excelentes resultados.

Durante o desenvolvimento do trabalho, diversos objetivos específicos foram alcançados. Os primeiros objetivos necessários para o desenvolvimento deste trabalho dizem respeito à modelagem e implementação dos dois *workflows* existentes neste trabalho. O primeiro *workflow* é o de preparação de dinâmicas moleculares e o segundo o de *docking* de múltiplos ligantes. Para o desenvolvimento desses *workflows* foi utilizada a metodologia descrita na Seção 3.1. Esta metodologia prevê uma descrição dos *workflows* a partir da análise de hipóteses e problemas de pesquisa. Essa metodologia é interessante pois garante que os problemas de pesquisa podem gerar novos *workflows* e protocolos de pesquisa, possibilitando a geração de *workflows* científicos eficientes para aquele conjunto de problemas e hipóteses.

Após a modelagem dos *workflows*, outro objetivo é o de desenvolvimento dos *scripts* para que o *workflow* pudesse ser executado. Os *scripts* devem possuir características importantes para que o sistema seja considerado um *workflow* e *estas* características foram alcançadas no desenvolvimento deste trabalho. Estas características são: utilização de processamento paralelo; possibilidade de execução de partes do *workflow*; capacidade de recomeçar o *workflow* de um determinado passo; armazenamento e controle dos arquivos gerados ao longo do processo (proveniência de dados); capacidade de automatização do processo; possibilidade de se utilizar outras ferramentas que não as descritas no protocolo sugerido; capacidade de configuração do ambiente de forma a garantir uma maior gama de possibilidades para o pesquisador; possibilidade de inserção de novas ferramentas no *workflow*.

Além destes pontos apresentados, outro objetivo da pesquisa que foi alcançado foi a reprodutibilidade do protocolo, no sentido de registrar e documentar todas as etapas do processo. Essa reprodutibilidade é alcançada com a utilização do arquivo de configuração. Além dessa reprodutibilidade, também foi alcançada a reprodutibilidade de código. Durante o experimento foi possível utilizar *scripts* da ProtCool_Dynamic para geração da ProtCool_Docking.

O ProtCool também é responsável por realizar o gerenciamento de informações e por manter o registro dos processos e protocolos utilizados durante a pesquisa. Com o ProtCool, o gerenciamento dos arquivos de pesquisa é realizado facilmente pelo pesquisador, sendo permitido o acesso a qualquer arquivo gerado, bastando acessar a pasta de armazenamento.

Outro fator importante da ProtCool é a possibilidade de preparação de dinâmicas moleculares por pesquisadores que não possuem uma facilidade de utilização de computadores. Como a ProtCool possui um instalador que configura todos os sistemas necessários, e, possui uma interface gráfica que possibilita a geração dos arquivos de configuração, isso facilita a geração dos arquivos de preparação da simulação de dinâmicas moleculares. A configuração de todos os sistemas, além da incompatibilidade de formatos de arquivos e a própria utilização de sistemas computacionais são obstáculos que muitos pesquisadores podem possuir e a ProtCool acaba por facilitar este trabalho.

A automatização do processo de preparação das simulações de dinâmicas moleculares garante que elementos importantes do ambiente químico/bioquímico (águas, pontes dissulfetos, força iônica, dentre outros) sejam levados em consideração sem a necessidade de cálculos e interferências do pesquisador.

Um ponto forte da ProtCool é a existência de duas ferramentas de *docking*. Isso possibilita que o pesquisador possa fazer a escolha entre uma delas, ou fazer um estudo mais abrangente, de forma a comparar o resultado fornecido pelas duas ferramentas. Além disso, a forma como o sistema foi

desenvolvido permite que novas ferramentas sejam implementadas, ampliando as possibilidades do pesquisador.

A capacidade que a ferramenta fornece de possibilitar que uma atividade seja refeita, sem a necessidade de nova execução de todos os *scripts* também é uma característica importante da ProtCool. Com este tipo de funcionalidade o pesquisador pode fazer a execução passo a passo, validando os resultados antes de dar prosseguimento à execução da preparação da dinâmica molecular. Isso também garante a possibilidade de usar ferramentas que não estão presentes no *workflow*.

O desenvolvimento dos *scripts* em paralelo garante que todos os recursos computacionais sejam utilizados, de forma a diminuir o tempo de processamento do sistema.

Com relação à metodologia utilizada pelo pesquisador quando estiver utilizando a ProtCool, é importante destacar que ela pode ser facilmente reproduzida e, além disso, pode ser facilmente descrita pelo pesquisador. Todas as informações são retiradas do arquivo de configuração gerado para execução da ProtCool e dos arquivos de log gerados ao longo do processo de preparação.

Também deve ser considerado que pesquisadores novatos conseguem fazer a preparação de dinâmicas moleculares com a utilização da ProtCool, uma vez que não é necessário a utilização de diversas ferramentas ao longo do processo, o que facilita a execução do sistema.

O maior benefício observado, no entanto, é a facilidade de trabalhar com múltiplos ligantes. Ou seja, a partir de um único protocolo definido, pode-se, em uma rodada única, preparar inúmeros ligantes. O sistema organiza estes ligantes separando todos os arquivos gerados em pastas, possibilitando um controle de tudo o que foi realizado. Com o avanço computacional, cada vez será exigido em pesquisas de dinâmicas moleculares sejam realizadas com um número maior de ligantes e com uma estatística de execução cada vez maior. A ProtCool_Dynamic está preparada para esse contexto que se apresenta. É importante destacar que a ProtCool foi desenvolvida considerando o processamento paralelo, ou seja, quanto maior for o número de ligantes, maior poderá ser a capacidade de processamento das máquinas, possibilitando que todo o processo seja realizado rapidamente.

Isso também foi verificado com o ProtCool_Docking. A diferença é que *docking* de múltiplos ligantes já é uma realidade que se apresenta. Assim, esta ferramenta já está preparada para a gestão eficiente de múltiplos arquivos.

O estudo de caso com a Acetilcolinesterase Humana, mostrou que a preparação das moléculas foi realizada conforme Rocha (2017) e que isso ocorreu considerando múltiplos ligantes (4 ligantes). Além disso, pelos resultados da dinâmica pode-se verificar que eles foram simulares aos alcançados por Rocha (2017). Tanto os *docking* como os RMSD de cada ligante foram condizentes visualmente e quantitativamente, apresentando as mesmas ordens de grandeza com o trabalho apresentado por Rocha (2017). Não foi objetivo deste trabalho fazer uma análise estatística pormenorizada desses resultados, uma vez que isso exigiria um maior número de amostragens, o que demandaria elevado custo computacional. Além do que, a intenção foi sempre de indicar a adequação dos protocolos, e não a validação estatísticos dos resultados.

Quanto ao *docking* de múltiplos ligantes, as docagens realizadas com a MPro do SARS-CoV-2 obtiveram 99% de todas as poses esperadas, mostrando que o processo de docking foi realizado conforme os protocolos definidos na ferramenta. As análises detalhadas das poses, ainda em andamento, farão parte de um segundo artigo em preparação.

Como trabalhos futuros se pretende:

i. Melhorar os *scripts* já existentes de forma a garantir que as muitas opções das diversas ferramentas utilizadas estejam disponíveis para o pesquisador;
ii. Aumentar a quantidade de ferramentas existentes no *workflow*, de forma a possibilitar que novas ferramentas estejam disponíveis ao pesquisador;
iii. Melhorar a *interface* gráfica de cadastro do arquivo de configuração, possibilitando que o pesquisador faça escolhas mais adequadas. Possibilitar que a partir da *interface* do sistema, o pesquisador possa avaliar as moléculas que estão sendo utilizadas, de forma a

trazer maior rapidez no preenchimento do arquivo de configuração. Exemplo disso, é disponibilizar na própria ferramenta a parte de escolha de *pocket* de docagem, conforme ocorre na EasyVS;

iv. Desenvolver novos *workflows* que possibilitem, a partir de dinâmicas moleculares prontas, os dados para análises possam ser recuperados. Assim, seria possível uma entrega para o pesquisador não só dos dados para simulação prontos, mas todo um processo automatizado das principais análises, possibilitando ainda maior rapidez no processo de geração de pesquisas.

Finalmente, é possível destacar que este trabalho conseguiu trazer uma melhora na gestão de experimentos e de laboratórios, o que configura uma das qualidades destacadas de *workflows* científicos, fazendo com que pesquisas com maior qualidade possam ser realizadas. Além disso, a possibilidade de reprodutibilidade dos protocolos de ancoragem e simulações de dinâmica molecular dos complexos, pelo registro detalhado de cada etapa do processo, constitui uma das principais contribuições desta tese para a química computacional e a quimioinformática.

REFERÊNCIAS

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

AGUILAR, Charles Martins. **Aplicação de metodologias teóricas para o estudo do processo de solvatação e espectroscopia eletrônica de íons de metais de transição em solução.** 2009. 93 f. Dissertação (Mestrado em Química do Departamento de Química). Universidade Federal de Minas Gerais. Belo Horizonte, MG. 2009.

ALMEIDA, Rodrigo *et al*. AProvBio: An architecture for data provenance in bioinformatics *workflows* using graph database. **In: Bioinformatics and Biomedicine (BIBM)**, 2017 IEEE International Conference on. IEEE, 2017. p. 2139-2144, 2017.

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

ALTINTAS, Ilkay *et al*. Kepler: an extensible system for design and execution of scientific *workflows*. In: Scientific and Statistical Database Management, 2004. **Proceedings 16th International Conference on IEEE**. p. 423-424. 2004.

ALVES, Tiago Henrique Costa Rodrigues. **Uma arquitetura baseada em containers para *workflows* de bioinformática em nuvens federadas**. 2017. 93 f. Dissertação (Mestrado em Informática). Universidade de Brasília. Brasília, DF. 2017.

ANANDAKRISHNAN, Ramu; AGUILAR, Boris; ONUFRIEV, Alexey V., "H++ 3.0: automating pK prediction and the preparation of biomolecular structures for atomistic molecular modeling and simulation", **Nucleic Acids Res.**, 40(W1):W537-541. 2012.

APPEL, Andre Luiz; MACIEL, Maria Lucia; ALBAGLI, Sarita. A e-Science e as novas práticas de produção colaborativa de conhecimento científico. **Revista Internacional de Ciencia y Sociedad**, v. 3, n. 1, p. 41-52, 2016.

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

BARGA, R.; GANNON, D. Scientific versus business *workflows*. In Taylor, I., Deelman, E., Gannon, D., and Shields, M., editors, Workflow*s* for e-Science, pages 9–16. **Springer London**. 2007.

BORGES, Nadia Melo *et al*. Similarity search combined with docking and molecular dynamics for novel hAChE inhibitor scaffolds. **Journal of molecular modeling**, v. 24, n. 1, p. 1-12, 2018.

BORGES, Nádia Melo. Similaridade, docking e dinâmica molecular: combinação de estratégias na busca de novos inibidores da hAChE. 2017. 67 f. Tese (Doutorado em Física do Departamento de Física da Universidade de Brasília). Universidade de Brasília. Brasília, DF. 2017.

BRAGHETTO, Kelly Rosa; CORDEIRO, Daniel. Introdução à modelagem e execução de *workflows* científicos. **Atualizações em Informática**. 1ed. Porto Alegre: SBC, p. 1-40, 2014.

BROOKS B.R., BRUCCOLERI R.E., OLAFSON B.D., STATES D.J., SWAMINATHAN S., KARPLUS M. (1983). CHARMM: A program for macromolecular energy, minimization, and dynamics calculations. **J Comp Chem**. 4 (2): 187–217. doi:10.1002/jcc.540040211, 1983.

CALIXTO, Paulo Henrique Matayoshi *et al*. DINÂMICA MOLECULAR DA ADENOSINA DEAMINASE DE Edwardsiella tarda EM ÁGUA-TUTORIAL EXPERIMENTAL. **Biota Amazônia (Biote Amazonie, Biota Amazonia, Amazonian Biota)**, v. 5, n. 4, p. 8-14, 2015.

CUEVAS-VICENTTÍN, Víctor *et al*. Scientific *workflows* and provenance: Introduction and research opportunities. **Datenbank-Spektrum**, v. 12, n. 3, p. 193-203, 2012.

DA CRUZ, Sérgio Manuel Serra *et al*. A provenance-based approach to resource discovery in distributed molecular dynamics *workflows*. In: **International Workshop on Resource Discovery**. Springer, Berlin, Heidelberg. p. 66-80. 2009.

DALLAKYAN, S. MGLTools. http://mgltools.scripps.edu/. 2010.

DE MEDEIROS FILHO, Francisco Carlos *et al*. Estudo da inibição da acetilcolinesterase por docking molecular: aplicação no tratamento da doença do alzheimer. **Educação, Ciência e Saúde**, v. 7, n. 2, p. 18, 2020.

DEELMAN, Ewa *et al*. *Workflows* and e-Science: An overview of *workflow* system features and capabilities. **Future generation computer systems**, v. 25, n. 5, p. 528-540, 2009.

DE PARIS, Renata. **An effective method to optimize docking-based virtual screening in a clustered fully-flexible receptor model deployed on cloud platforms.** 2017. 167 f. Tese (Doutorado em Ciência da Computação da Pontifícia Universidade Católica do Rio Grande do Sul). Porto Alegre, RS. 2017.

DE PINHO VELOSO, Wandré Nunes *et al*. **Easyvs: uma ferramenta para triagem virtual mista baseada em alvo e ligante.** 2019. 71 f. Tese (Doutorado em Bioinformática do Instituto de Ciências Biológicas). Universidade Federal de Minas Gerais. Belo Horizonte. MG. 2019.

EDGCOMB, Stephen P.; MURPHY, Kenneth P. Variability in the pKa of histidine side-chains correlates with burial within proteins. **Proteins: Structure, Function and Genetics**, [S. l.], v. 49, n. 1, p. 1–6, 2002. DOI: 10.1002/prot.10177. 2002.

FERNANDES, FMSS. Perspectivas da Química Computacional. **Boletim da Sociedade Portuguesa de Química**, v. 123, p. 47-53, 2011.

GARGANO, Furia. **Efeito da temperatura na enzima 2-trans-enoil-ACP (CoA) redutase (EC 1.3. 1.9) de Mycobacterium tuberculosis em complexo com o NADH: um estudo por simulação pela dinâmica molecular.** 2009. 149 f. Tese (Doutorado Programa de pós-Graduação em Biologia Celular e Molecular). Pontifícia Universidade Católica do Rio Grande do Sul. Porto Alegre, RS. 2009.

GEORG, H. C.; CANUTO, S. Métodos Híbridos para Modelagem do Ambiente Molecular. **Métodos de Química Teórica e Modelagem Molecular**, p. 453-488, 2007.

GOMES, Jefferson Chaves. Modelo Multi-estratégico de Tolerância a Falhas para Ambiente de Nuvem Federada. 2018. 97 f. Dissertação (Mestrado em Informática). Universidade de Brasília. Brasília, DF. 2018.

GU, Jenny; BOURNE, Philip E. **Structural bioinformatics**. John Wiley & Sons, ed. 2, 2009.

GUIMARÃES, Valeria *et al*. A study of genomic data provenance in NoSQL document-oriented database systems. In: **Bioinformatics and Biomedicine (BIBM)**, 2015 IEEE International Conference on. IEEE, 2015. p. 1525-1531. 2015.

GULER, Arzu Tugce; WAAIJER, Cathelijn JF; PALMBLAD, Magnus. Scientific *workflows* for bibliometrics. **Scientometrics**, v. 107, n. 2, p. 385-398, 2016.

GUNSTEREN, W. F. van; BILLETER, S. R.; EISING, A. A.; HÜNENBERGER, P. H.; KRÜGER, P.; MARK, A. E.; SCOTT, W. R. P.; TIRONI, I. G. Biomolecular Simulation: The GROMOS96 Manual and User Guide, **Vdf Hochschulverlag AG an der ETH Zürich**, Zürich, Switzerland, 1996, pp. 1-1042. 1996.

HIDEN, Hugo *et al*. Developing cloud applications using the e-science central platform. **Phil. Trans. R. Soc. A**, v. 371, n. 1983, p. 20120085, 2013.

HIDEN, Hugo *et al*. e-Science Central: Cloud-based e-Science and its application to chemical property modelling**. Relatório Técnico CS-TR-1227, School of Comp. Sci. Newcastle University**, 2011.

HONDO, Fernanda *et al*. Data provenance management for bioinformatics *workflows* using NoSQL database systems in a cloud computing environment. **In: Bioinformatics and Biomedicine (BIBM)**, 2017 IEEE International Conference on. IEEE, 2017. p. 1929-1934. 2017.

HUANG, Li-Kai; CHAO, Shu-Ping; HU, Chaur-Jong. Clinical trials of new drugs for Alzheimer disease. **Journal of biomedical science**, v. 27, n. 1, p. 1-13, 2020.

HULL, Duncan *et al*. Taverna: a tool for building and running *workflows* of services. **Nucleic Acids Res**, v. 34, n. suppl 2, p. W729–W732, 2006.

HUMPHREY, W., DALKE, A. and SCHULTEN, K., "VMD - Visual Molecular Dynamics", **J. Molec. Graphics**, vol. 14, pp. 33-38. 1996.

INESTROSA, Nibaldo C. *et al*. Acetylcholinesterase Accelerates Assembly ao Amiloyd-β- Peptides into Alzheimer's Fibrils: Possible Role of the Peripheral Site of Enzyme. **Neuron**, v. 16, p. 881-891., 1996.

JIN, Z. *et al*. Structure of Mpro from SARS-CoV-2 and discovery of its inhibitors. **Nature** 2020, 582, 289–293. 2020.

KOES, D. R.; BAUMGARTNER, M. P.; CAMACHO, C. J. 2013. Lessons Learned in Empirical Scoring with smina from the CSAR 2011 Benchmarking Exercise. **Journal of Chemical Information and Modeling**. 53(8): 1893-1904. 2013.

KUNTAL, Bhusan K.; APAROY, Polamarasetty; REDDANNA, Pallu. EasyModeller: A graphical interface to MODELLER. **BMC research notes**, v. 3, n. 1, p. 226, 2010.

LEAVER-FAY, Andrew *et al*. Scientific benchmarks for guiding macromolecular energy function improvement. **Methods in enzymology**, v. 523, p. 109-143, 2013.

LONG, Justin M.; HOLTZMAN, David M. Alzheimer disease: an update on pathobiology and treatment strategies. **Cell**, v. 179, n. 2, p. 312-339, 2019.

LUDÄSCHER, Bertram *et al*. Scientific process automation and *workflow* management. **Scientific Data Management: Challenges, Existing Technology, and Deployment, Computational Science Series**, v. 230, p. 476-508, 2009.

MARTÍNEZ, L.; BORIN, I. A.; SKAF, M. S. Fundamentos de simulação por dinâmica molecular. **Métodos de Química Teórica e Modelagem Molecular**, p. 413-452, 2007.

MCGIBBON, Robert T. *et al*. MDTraj: a modern open library for the analysis of molecular dynamics trajectories. **Biophysical journal**, v. 109, n. 8, p. 1528-1532, 2015.

MENDES, Felipe Lopes de Souza. **Uma Plataforma de Federereação de Nuvens em uma Arquitetura Orientada a Microsserviços**. 2018. 111 f. Dissertação (Mestrado em Informática). Universidade de Brasília. Brasília, DF. 2018.

MISSIER, Paolo *et al*. Golden trail: Retrieving the data history that matters from a comprehensive provenance repository**. International Journal of Digital Curation**, v. 7, n. 1, p. 139-150, 2012.

MOURA, Breno Rodrigues de. **Arquitetura de um controlador de SLA para ambiente de nuvens federadas**. 2017. 98 f. Dissertação (Mestrado em Informática). Universidade de Brasília. Brasília, DF. 2017.

NAMBA, A. M.; DA SILVA, V. B.; DA SILVA, C. H. T. P. Dinâmica molecular: teoria e aplicações em planejamento de fármacos. **Eclética Química**, v. 33, n. 4, 2008.

NUNES, Vinicius Schmitz Pereira *et al*. **Análise comparativa das Ecto-NTPDase 1 de Homo sapiens e Schistosoma mansoni por meio de modelagem tridimensional, dinâmica molecular e docking receptor-ligante**. 2015. 153 f. Tese (Doutorado em Modelagem Computacional). Universidade Federal de Juiz de Fora, MG. 2015.

ONUFRIEV, A. *et al*. H++. http://biophysics.cs.vt.edu. 2005.

PHILLIPS, James C.; HARDY, David J.; MAIA, Julio D. C.; STONE, John E.; RIBEIRO, Joao V.; BERNARDI, Rafael C.; BUCH, Ronak; FIORIN, Giacomo; HENIN, Jerome; JIANG, Wei; MCGREEVY, Ryan; MELO, Marcelo C. R.; RADAK, Brian K.; SKEEL, Robert D.; SINGHAROY, Abhishek; WANG, Yi; ROUX, Benoit; AKSIMENTIEV, Aleksei; LUTHEY-SCHULTEN, Zaida; KALE, Laxmikant V.; SCHULTEN, Klaus; CHIPOT Christophe; TAJKHORSHID, Emad. Scalable molecular dynamics on CPU and GPU architectures with NAMD. **Journal of Chemical Physics**, 153:044130, 2020. doi:10.1063/5.0014475. 2020.

PHILLIPS, James C. *et al*. Scalable molecular dynamics with NAMD. **Journal of Computational Chemistry**, 26:1781-1802, 2005.

PICANÇO, Leide Caroline dos Santos. **Planejamento de candidatos a fármacos multialvo inibidores de Acetilcolinesterase (AChE) e glicogênio sintase quinase-3β (GSK-3β) para tratamento da doença de Alzheimer**. 2018. 86 f. Dissertação (Mestrado em Ciências Farmacêuticas). Universidade Federal do Amapá, AP. 2018.

POLYAKOV, Stanislav P.; DEMICHEV, Andrey P.; KRYUKOV, Alexander P. Web toolkit for scientific research: state of the art and the prospect for development. **Procedia Computer Science**, v. 66, p. 429-438, 2015.

PRONK, Sander *et al*. GROMACS 4.5: a high-throughput and highly parallel open source molecular simulation toolkit. **Bioinformatics**, v. 29, n. 7, p. 845-854, 2013.

PURAWAT, Shweta *et al*. A Kepler *Workflow* Tool for Reproducible AMBER GPU Molecular Dynamics. **Biophysical Journal**, v. 112, n. 12, p. 2469-2474, 2017.

RACKERS, Joshua A.; WANG, Zhi; LU, Chao; LAURY, Marie L.; LAGARDÈRE, Louis; SCHNIEDERS, Michael J.; PIQUEMAL, Jean-Philip; REN, Pengyu; PONDER, Jay W. Tinker 8: Software Tools for Molecular Design. **Journal of Chemical Theory and Computation**. 2018 14 (10), 5273-5289. DOI: 10.1021/acs.jctc.8b00529. 2018.

RIBEIRO, J., BERNARDI, R., RUDACK, T. *et al*. QwikMD — Integrative Molecular Dynamics Toolkit for Novices and Experts. **Sci Rep 6**, 26536. https://doi.org/10.1038/srep26536. 2016.

ROCHA, RAFAEL E. O. ; CHAVES, ELTON J. F. ; FISCHER, PEDRO H. C. ; COSTA, LEON S. C. ; GRILLO, IGOR BARDEN ; DA CRUZ, LUIZ E. G. ; GUEDES, FABIANA C. ; da Silveira, Carlos H. ; SCOTTI, MARCUS T. ; CAMARGO, ALEX D. ; MACHADO, KARINA S. ; WERHLI, ADRIANO V. ; FERREIRA, RAFAELA S. ; ROCHA, GERD B. ; DE LIMA, LEONARDO H. F. . A higher flexibility at the SARS-CoV-2 main protease active site compared to SARS-CoV and its potentialities for new inhibitor virtual screening targeting multi-conformers. JOURNAL OF BIOMOLECULAR STRUCTURE & DYNAMICS, v. 1, p. 1-21, 2021.

ROCHA, Rafael Eduardo Oliveira. **Estudos de modelagem molecular dos mecanismos de afinidade relativa para quatro galantamínicos com potencial anti-Alzheimer.** 2017. 81 f. Dissertação (Mestrado em Bioinformática). Universidade Federal de Minas Gerais. Belo Horizonte, MG. 2017.

ROSA, Michel *et al*. BioNimbuZ: A federated cloud platform for bioinformatics applications. **In: Bioinformatics and Biomedicine (BIBM)**, 2016 IEEE International Conference on. IEEE, 2016. p. 548-555. 2016.

ROSA, Michel Junio Ferreira. **Predição de tempo e dimensionamento de recursos para *workflows* científicos em nuvens federadas.** 2017. 97 f. Dissertação (Mestrado em Informática). Universidade de Brasília. Brasília, DF. 2017.

ROWE, Anthony *et al*. The discovery net system for high throughput bioinformatics. **Bioinformatics**, v. 19, n. Suppl 1, p. i225–i231, 2003.

SALDANHA, Hugo Vasconcelos. **Bionimbus: uma arquitetura de federação de nuvens computacionais híbrida para a execução de *workflows* de bioinformática**. 2012. 82 f. Dissertação (Mestrado em Informática). Universidade de Brasília. Brasília, DF. 2012.

SALDANHA, Hugo *et al*. Towards a hybrid federated cloud platform to efficiently execute bioinformatics *workflows*. **In: BioInformatics**. InTech, 2012.

SALOMON-FERRER, Romelia; CASE, David A.; WALKER, Ross C. An overview of the Amber biomolecular simulation package. **Wiley Interdisciplinary Reviews**: Computational Molecular Science, v. 3, n. 2, p. 198-210, 2013.

SCHRÖDINGER, LLC. (2015). The PyMOL Molecular Graphics System, **Version 1.8**. 2015.

SEEBER, Michele *et al*. Wordom: a user-friendly program for the analysis of molecular structures, trajectories, and free energy surfaces. **Journal of computational chemistry**, v. 32, n. 6, p. 1183-1194, 2011.

SERENIKI, Adriana; VITAL, Maria Aparecida Barbato Frazão. A doença de Alzheimer: aspectos fisiopatológicos e farmacológicos. **Rev. psiquiatr. Rio Gd. Sul**, Porto Alegre , v. 30, n. 1, supl. 2008. Available from <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0101-81082008000200002&lng=en&nrm=iso>. access on 24 Jan. 2021. https://doi.org/10.1590/S0101-81082008000200002. 2008.

SHELLEY, John C.; CHOLLETI, A.; Frye, L; GREENWOOD, J.R.; TIMLIN, M.R.; UCHIMAYA, M. Epik: a software program for pK a prediction and protonation state generation for drug-like molecules. **Journal of computer-aided molecular design**, v. 21, n. 12, p. 681-691, 2007.

SHEN, Meng; KETEN, Sinan; LUEPTOW, Richard M. Rejection mechanisms for contaminants in polyamide reverse osmosis membranes. **Journal of Membrane Science**, v. 509, p. 36-47, 2016.

SILVA, E. *et al*. Especificação Formal e Verificação de *Workflows* Científicos. **IV e-Science**, 2010.

SITE H++. http://biophysics.cs.vt.edu, 2021. Acesso: junho, 2021.

SMITH, W.; YONG, C.W.; RODGER, P.M. **DL_POLY**: Application to molecular simulation, Molecular Simulation, 28:5, 385-471, DOI: 10.1080/08927020290018769, 2002.

SOARES, R.O.S. **Dinâmica Molecular de Proteínas: estabilidade e renaturação térmica**. 2009. 86 p. Dissertação (Mestrado) – Faculdade de Ciências Farmacêuticas de Ribeirão Preto, Universidade de São Paulo, Ribeirão Preto, 2009.

SOMOGYI, Endre; MANSOUR, Andrew Abi; ORTOLEVA, Peter J. ProtoMD: A prototyping toolkit for multiscale molecular dynamics. **Computer Physics Communications**, v. 202, p. 337-350, 2016.

SONDERGAARD, Chresten R.; OLSSON, Mats HM; ROSTKOWSKI, Michal; JENSEN, Jan H. "Improved Treatment of Ligands and Coupling Effects in Empirical Calculation and Rationalization of pKa Values." Journal of Chemical Theory and Computation 7, no. 7 (2011): 2284-2295. doi:10.1021/ct200133y. 2011.

SOUZA, Anivaldo Xavier de; SANT'ANNA, Carlos Mauricio R. UDP-N-acetilglicosamina enolpiruvil transferase: determinação dos estados de protonação de resíduos de aminoácidos do sítio ativo pelo método PM6. Química Nova, v. 35, n. 8, p. 1522-1586, 2012.

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

TAYLOR, Ian *et al*. Visual grid *workflow* in Triana. **Journal of Grid Computing**, v. 3, n. 3-4), p. 153–169, 2005.

THOMSEN, René; CHRISTENSEN, Mikael H. MolDock: A New Technique for High-Accuracy Molecular Docking. **Journal of Medicinal Chemistry**. 2006 49 (11), 3315-3321. DOI: 10.1021/jm051197e. 2006.

TROTT, O.; OLSON, A. J. AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization and multithreading, **Journal of Computational Chemistry** 31 (2010) 455-461. 2010.

VERGARA, Guilherme Fay. **Arquitetura de um controlador de elasticidade para nuvens federadas**. 2017. 89 f. Dissertação (Mestrado em Informática). Universidade de Brasília. Brasília, DF. 2017.

WANG, Changhao; GREENE, D'Artagnan; XIAO, Li; QI, Ruxi; LUO, Ray. Recent developments and applications of the MMPBSA method. **Frontiers in Molecular Biosciences**, [S. l.], v. 4, n. JAN, p. 1–18, 2018. DOI: 10.3389/fmolb.2017.00087. 2018.

WEBB, Benjamin; SALI, Andrej. Comparative protein structure modeling using MODELLER. **Current protocols in bioinformatics**, v. 54, n. 1, p. 5.6. 1-5.6. 37, 2016.

WIESNER, Jiří *et al*. Acetylcholinesterases–the structural similarities and differences. **Journal of enzyme inhibition and medicinal chemistry**, v. 22, n. 4, p. 417-424, 2007.

WISHART, D. S.; FEUNANG, Y. D.; GUO, A. C.; Lo, E. J.; MARCU, A.; GRANT, J. R.; Sajed, T.; JOHNSON, D.; Li, C.; SAYEEDA, Z.; ASSEMPOUR, N.; IYNKKARAN, I.; Liu, Y.; MACIEJEWSKI, A.; GALE, N.; WILSON, A.; CHIN, L.; CUMMINGS, R.; Le, D.; PON, A.; KNOX, C.; WILSON, M. (2018a). DrugBank 5.0: a major update to the DrugBank database for 2018. **Nucleic acids research**, 46(D1):D1074–D1082. ISSN 1362-4962. 2018.

YU, Jia; BUYYA, Rajkumar. A taxonomy of *workflow* management systems for grid computing. **Journal of Grid Computing**, v. 3, n. 3-4, p. 171-200, 2005.

ZOOKEEPER. https://zookeeper.apache.org/ . 2018.