PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

UM MODELO DE *WORKFLOW* CIENTÍFICO PARA O REFINAMENTO DA ESTRUTURA 3D APROXIMADA DE PROTEÍNAS

LEONARDO VERONESE SOLETTI

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Ciência da Computação na Pontifícia Universidade Católica do Rio Grande do Sul.

Orientador: Prof. Osmar Norberto de Souza

Porto Alegre
Dezembro, 2015

Ficha Catalográfica

S685m Soletti, Leonardo Veronese

Um Modelo de Workflow Científico Para o Refinamento da Estrutura 3D Aproximada de Proteínas / Leonardo Veronese Soletti . – 2016.
142 f.
Dissertação (Mestrado) – Programa de Pós-Graduação em Ciência da Computação, PUCRS.

Orientador: Prof. Dr. Osmar Norberto de Souza.

1. Bioinformática. 2. Bioinformática estrutural. 3. Workflow científico. 4. Predição da estrutura 3D de proteinas. 5. Dinâmica Molecular. I. Souza, Osmar Norberto de. II. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da PUCRS com os dados fornecidos pelo(a) autor(a).

Pontifícia Universidade Católica do Rio Grande do Sul
FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

TERMO DE APRESENTAÇÃO DE DISSERTAÇÃO DE MESTRADO

Dissertação intitulada "Um Modelo de Workflow Científico para o Refinamento da Estrutura 3D Aproximada de Proteínas" apresentada por Leonardo Veronese Soletti como parte dos requisitos para obtenção do grau de Mestre em Ciência da Computação, aprovada em 30 de março de 2016 pela Comissão Examinadora:

Prof. Dr. Osmar Norberto de Souza– Orientador — PPGCC/PUCRS

Prof. Dr. Duncan Dubugras Alcoba Ruiz – — PPGCC/PUCRS

Dr. Rafael Andrade Caceres - — UFCSPA

Homologada em 06/04/17, conforme Ata No. 05 pela Comissão Coordenadora.

Prof. Dr. Luiz Gustavo Leão Fernandes
Coordenador.

PUCRS PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Campus Central
Av. Ipiranga, 6681 – P. 32 – sala 507 – CEP: 90619-900
Fone: (51) 3320-3611 – Fax (51) 3320-3621
E-mail: ppgcc@pucrs.br
www.pucrs.br/facin/pos

DEDICATÓRIA

A minha mãe e ao meu pai pelo apoio e incentivo.

AGRADECIMENTOS

Aos meus pais, pelo suporte e incentivo constante. Agradeço a minha mãe, por me incentivar, inspirar e ajudar a vencer mais este desafio, sem os quais nada disso teria sido possível.

Ao colega Mirocem Fernandes de Oliveira, pelas horas dos finais de semana que tomei do seu tempo e, principalmente, pela atenção e paciência por seus ensinamentos em esclarecer tantas dúvidas em Biologia e pela parceria imprescindível para a realização desta dissertação.

Gostaria de agradecer ao meu orientador, Prof. Dr. Osmar Norberto de Souza, pela confiança e apoio na execução deste projeto e pela credibilidade depositada em mim, especialmente por permitir o desenvolvimento do mestrado mesmo sem que eu estivesse disponível em tempo integral durante o primeiro ano.

Fico grato, igualmente, ao professor Dr. Rafael Caceres, por aceitar participar na minha banca examinadora.

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo auxílio que me foi concedido.

Enfim, agradeço também a minha namorada Carla, pela compreensão nas ausências nos finais de semana e a todos os amigos ouvintes que me apoiaram com palavras de incentivo e motivação.

Staudinger salientou que "as macromoléculas possuem propriedades que não podem ser previstas a partir das propriedades das suas unidades constituintes"... Aparentemente o único obstáculo para compreender a natureza da vida é a sua fantástica complexidade. A Informática dá-nos a esperança que um dia possamos vencer também esta dificuldade.

(H.A. Krebs, Persp. Biol. Med., 1971).

RESUMO

Com o advento da era pós-genômica surge, como consequência, uma explosão de informações onde inúmeras descobertas geram grande quantidade de dados biológicos. Mesmo com o avanço da tecnologia nas técnicas de predição de estruturas de proteínas, não é possível ainda se encontrar uma ferramenta capaz de predizer com precisão exata a estrutura 3D de proteínas. Em decorrência disso, surgem novos desafios para entender e organizar esses recursos nas pesquisas, o compartilhamento e reuso de experimentos bem-sucedidos, assim como prover interoperabilidade entre dados e ferramentas de diferentes locais e utilizados por usuários com perfis distintos. As atividades de estudos do fluxo destes dados, inicialmente, baseiam-se em *scripts* que auxiliam na entrada, processamento e resultado final da análise, normalmente executados por linha de comando, o que obriga seus usuários a terem domínio de algoritmos e lógica de programação. Tais *scripts* apresentam problemas em interferir, coletar e armazenar dados ao longo de sua execução, e podem ser muito complexos, ocasionando a dificuldades de implementação, manutenção e reuso. Outro problema é quando um conjunto de tarefas a serem realizadas através de *scripts*, podem ter o risco de faltar algum passo no processo ou não ser executado na ordem certa, obtendo-se com isso resultados não satisfatórios. Torna-se necessário técnicas e ferramentas que facilitem esse processo, de maneira organizada como uma sequência de etapas caracterizados por um fluxo de execução, automatizando-se assim este processo. Neste contexto, buscou-se desenvolver um modelo de *workflow* científico utilizando-se ferramentas de bioinformática e de conhecimentos da biologia para automatizar o processo de refinamento de proteínas, do polipeptídio predito pelo método CReF. Os *scripts* do processo de refinamento foram automatizados, com isso foi possível aumentar a quantidade de experimentos, mantendo um critério de qualidade aceitável. Para o resultado final do processo, desenvolveu-se uma interface *web* que facilita a visualização dos resultados de uma forma organizada.

Palavras chaves: Bioinformática, bioinformática estrutural, *workflows* científicos, predição da estrutura 3D de proteínas, dinâmica molecular.

ABSTRACT

As a consequence of the post-genomic era an explosion of information and numerous discoveries made available large amounts of biological data. Even with the technology enhancements regarding protein structure prediction techniques, it is still not possible to find a tool to predict with precision the exact the three-dimensional structure of a given protein. This brings new challenges, starting from how to understand and organize these resources until sharing and reuse of successful experiments, as well as how to provide interoperability between data from different sources, without mentioning the diversity between tools and different user profiles. This kind of data flow is regularly addressed as command line scripts which require users to have programming skills. Such scripts have problems interfering, collecting and storing data while executing. Furthermore, these scripts and can be very complex leading to difficulties of implementation, maintenance and reuse. Another problem that arises when a set of tasks are proposed to be conducted through scripts is the possibility of missing any step in the process or running at incorrect order, leading to inconsistent results. It becomes necessary techniques and tools to ease this process in an organized way as a sequence of steps characterized by a workflow, thus automating this process. In this context, we sought to develop a scientific workflow model using bioinformatics tools and biology expertise to automate the process of protein refinement of polypeptides predicted by CReF method once the refinement process scripts were automated, it was possible to increase the amount of experiments while maintaining an acceptable quality criteria. Finally, was developed a web interface that facilitates the visualization of the results in an organized way.

Keywords: Bioinformatics, structural bioinformatics, scientific workflows, three-dimensional protein 3D structure, molecular dynamics.

LISTA DE FIGURAS

Figura 1 – O gráfico mostra o número de estruturas pesquisáveis por ano. 26
Figura 2 – A sequência linear de aminoácidos de uma proteína define sua estrutura primária possibilitando a sua formação tridimensional. 27
Figura 3 – Exemplo de sequência de aminoácidos em formato FASTA, com identificação (cabeçalho) destacado em vermelho. A sequência FASTA identifica a proteína. 27
Figura 4 – A ligação peptídica ocorre entre o grupo a-carboxila de um aminoácido (1) e o grupo a-amino de outro aminoácido (2). 29
Figura 5 – Representação dos tipos de estruturas das proteínas. 31
Figura 6 – Exemplo de sequência de resíduos de aminoácidos na proteína cuja estrutura tridimensional foi resolvida e depositada sob o código " 1BBD" 32
Figura 7 – Estrutura secundária de uma proteína. 33
Figura 8 – Representação gráfica da estrutura tridimensional da hélice α, está representado apenas os átomos que participam na formação das pontes de hidrogénio. 33
Figura 9 - Nesta figura as cadeias laterais são representadas apenas por um átomo. 34
Figura 10 – Estrutura terciária de uma proteína. 35
Figura 11 – Estrutura quaternária de uma proteína (proteína de código PDB 1BVR). 36
Figura 12 – A modelagem por homologia apresenta quatro etapas principais. 38
Figura 13 – Representação esquemática do processo de modelagem *threading* 42
Figura 14 – Gráfico de Ramachandran gerado pelo Procheck 45
Figura 15 – A primeira figura (1) mostra trecho de um arquivo de saída da simulação por dinâmica molecular; a segunda (2) mostra trecho do arquivo de topologia utilizado pelo Ptraj e a última (3), arquivo PDB gerado após a execução do Ptraj 51
Figura 16 – Exemplo de execução comandos de execução com *shell script* do Ptraj 51
Figura 17 – Representação do ciclo de vida de um *workflow* 54
Figura 18 – Exemplo de workflow modelado com o Kepler 59
Figura 19 – Tela principal do Taverna 61
Figura 20 – Processo de CADD com flexibilidade explícita do receptor 63
Figura 21 – Etapas do processo de refinamento. 68
Figura 22 - Diagrama de todas as etapas, suas composições de arquivos que servem de entrada e arquivos gerados como saída. 70
Figura 23 – Estrutura do processo da etapa 1 do refinamento. 72
Figura 24 – Comandos para iniciar o estudo de caso do *workflow* 75
Figura 25 – Trecho do arquivo PDB da proteína 1GAB. 76
Figura 26 – Modelo do arquivo utilizado para informar as restrições. 77
Figura 27 – Modelo de arquivo para facilitar a interação do usuário com o *workflow* 79

Figura 28 – Estrutura do processo da etapa 2 do refinamento. 80
Figura 29 – Pâmetros de entrada para a minimização de energia. 81
Figura 30 – Arquivo da fase de aquecimento gerado para o refinamento da 1GAB. 83
Figura 31 – Arquivo da fase de produção gerado para o refinamento da 1GAB. 85
Figura 32 – Processo da etapa da criação dos arquivos que correspondem às restrições. 87
Figura 33 – Arquivo modelo utilizado para informar as restrições (secondary_structs_rst.in"), transcrito do Capítulo 5.4.1. 87
Figura 34 – Trecho extraído do arquivo de restrições. 88
Figura 35 – Trecho extraído do arquivo de restrições. 88
Figura 36 – Trecho extraído do arquivo gerado da minimização de energia. 89
Figura 37 – Exemplifica a flexibilidade do ângulo diedro 90
Figura 38 – Processo de execução do refinamento da etapa 4. 91
Figura 39 – Linha de comando em *shell script* 91
Figura 40 – Processo de execução do refinamento da etapa 5. 92
Figura 41 – Tela de *Login*, onde o usuário informa nome e senha para ter acesso a tela inicial. 96
Figura 42 – Tela inicial com os menus 97
Figura 43 – Tela com a lista de todos os refinamentos realizados 98
Figura 44 – Tela de edição para alterar o nome do refinamento. 99
Figura 45 – Tela inicial do detalhamento do refinamento selecionado. 100
Figura 46 – A sequência da Figura 45 após rolagem da barra 101
Figura 47 – A sequência da Figura 45 após rolagem da barra. 102
Figura 48 – A sequência da Figura 45 após rolagem da barra. 102
Figura 49 – A sequência da Figura 45 após a última rolagem da barra 103
Figura 50 – A sequência da Figura 45 após a última rolagem da barra 103
Figura 51 –Tela de Cadastro de Usuários. 104
Figura 52 – A sequência de imagens (letra A) representa três perspectivas da estrutura da proteína 1GAB 108
Figura 53 – Mapa de Ramachandran da estrutura experimental 109
Figura 54 – Mapa de Ramachandran da estrutura predita pelo CReF 109
Figura 55 – Mapa de Ramachandran da estrutura refinada 110
Figura 56 – A sequência de imagens (letra A) representa três perspectivas da estrutura da proteína 1YWJ 112
Figura 57 – Mapa de Ramachandran da estrutura experimental 113
Figura 58 – Mapa de Ramachandran da estrutura predita pelo CReF 113
Figura 59 – Mapa de Ramachandran da estrutura refinada 114
Figura 60 – A sequência de imagens (letra A) representa três perspectivas da estrutura da proteína 1GAB 116
Figura 61 – Mapa de Ramachandran da estrutura experimental 117

Figura 62 – Mapa de Ramachandran da estrutura predita pelo CReF..... 117
Figura 63 – Mapa de Ramachandran da estrutura refinada..... 118

LISTA DE QUADROS

Quadro 1 – Descrição dos parâmetros e sua respectiva funcionalidade..... 72
Quadro 2 – Descrição dos parâmetros, seu valor e sua respectiva funcionalidade. 77
Quadro 3 – Quadro comparativo do processo manual com o *workflow* desenvolvido 120

LISTA DE ABREVIATURAS E SIGLAS

3D - Tridimensional

AMBER - *Assisted Model Building with Energy Refinement*

BE - Bioinformática Estrutural

BLAST - *Basic local Alignment and Search Tool*

CReF - *Central Residue Fragment-based method*

CSS - *Cascading Style Sheets*

DM - Dinâmica Molecular

DNA - *Deoxyribonucleic Acid*

EM - *Expectation Maximization*

EP - Energia Potencial

GB - *Generalized Born*

HTML - *Hyper Text Markup Language*

LABIO - Laboratório de Bioinformática, Modelagem e Simulação de Biossistemas

LEAP - *Long-range Energy Alternatives Planning System*

MM - Mecânica Molecular

MoML - *Modeling Markup Language*

MWC - Modelo de Workflow Científico

NMR - *Derived Energy Restraints*

PDB - *Protein Data Bank*

PEP - Predição de Estruturas Tridimensionais de Proteínas

PIW - *Promoter Identification Workflow*

PMEMD - *Particle Mesh Ewald Molecular Dynamics*

PS - psicossegundo

PSP – *Protein Structure Predicition*

RMN - Ressonância Magnética Nuclear

RMSD - *Root Mean Square Deviation*

Sander - *Simulated Annealing with NMR-Derived Energy Restraints*

SGBD - *Data Base Management System*

SGWfC – Sistemas Gerenciadores de Workflows Científicos

SQL - *Structured Query Language*

SWfMS - *Scientific Workflow Management System*

UML - *Unified Modeling Language*

VMD - *Visual Molecular Dynamics*

WfMS- *Workflow Management Systems*

XML - *Extensible Markup Language*

SUMÁRIO

1 INTRODUÇÃO 17
1.1 Motivação 19
1.2 Objetivo Geral 21
1.2.1 Objetivos Específicos 21
1.3 Metodologia 21
1.4 Organização da Dissertação 23
2 REVISÃO DE LITERATURA 25
2.1 Bioinformática Estrutural 25
2.2 Proteínas 28
2.3 Dobramento de Proteínas 30
2.4 Estruturas de Proteínas 30
2.5 Organização Estrutural 31
2.5.1 Estrutura Primária 32
2.5.2 Estrutura Secundária 32
2.5.3 Estrutura Terciária 34
2.5.4 Estrutura Quartenária 36
2.6 Proteínas Homólogas 36
2.6.1 Modelagem de Proteínas Homolólogas 37
2.7 Etapas da modelagem por Homologia 38
2.7.1 Alinhamento Alvo / Template 39
2.7.2 Construção do Modelo 39
2.7.3 Avaliação dos Modelos 40
2.8 Modelagem Baseada em Conhecimento: *Threading* 41
2.9 Predição *Ab initio* 42
2.10 Ângulos de Torção 43
2.11 O Método CReF – Predição *Ab initio* da Estrutura 3D de Proteínas 45
2.11.1 Etapas do CReF 46
2.12 Dinâmica Molecular 48
2.12.1 O Pacote de Programa AMBER 14 49
3 *WORKFLOW* CIENTÍFICO 52
3.1 Importância do *Workflow* para a Bioinformática 55
3.2 Experimentos científicos 57
3.3 Reutilização de experimentos científicos 57
4 SISTEMA GERENCIADOR DE WORKFLOW CIENTÍFICO (SGWfC) 58
4.1 Kepler 58
4.2 Taverna 61

5 TRABALHOS RELACIONADOS 63
6 RESULTADOS: DESENVOLVIMENTO DO MODELO DE *WORKFLOW* 66
6.1 Implementação do Modelo de *Workflow* 66
6.2 Elicitação de Requisitos 69
6.3 Arquitetura do Modelo de *Workflow* Desenvolvido 69
6.4 Detalhes da Implementação: Estudo de Caso 71
6.4.1 Etapa 1: Entrada de Dados 71
6.4.2 Etapa 2: Leitura dos arquivos de entrada e preparação do ambiente 80
6.4.2.1 Criação dos arquivos de minimização de energia 81
6.4.2.2 Criação dos arquivos da fase de aquecimento 82
6.4.2.3 Criação dos arquivos da fase de produção 84
6.4.3 Etapa 3: Criação do arquivo de restrições 86
6.4.4 Etapa 4: Execução do refinamento 90
6.4.5 Etapa 5: Arquivos de análise de resultados 92
6.4.5.1 Gráfico de Ramachandran 92
6.4.5.2 Visualização da trajetória da Dinamica Molecular 93
6.4.5.3 Visualização tridimensional das estruturas refinadas 94
6.4.5.4 Arquivos de *Log's* 94
7 RESULTADOS: AMBIENTE WEB 96
7.1 Interface de *Login* 96
7.2 Tela inicial do sistema 97
7.3 Modulo de Resultados dos Refinamentos 98
7.4 Ação: Edição do Refinamento (Lápis) 99
7.5 Ação: Detalhes do Refinamento (Lupa) 100
7.6 Módulo de Controle de Usuários 104
8 RESULTADOS: EXPERIMENTOS UTILIZANDO O MODELO DE *WORKFLOW* 106
8.1.1 Definição do conjunto de proteínas utilizadas 106
8.1.2 Experimento 1: Refinamento da Proteína 1GAB 107
8.1.3 Experimento 2: Refinamento da Proteína 1YWJ 110
8.1.4 Experimento 3: Refinamento da Proteína 1GPT 114
9 CONSIDERAÇÕES FINAIS E DISCUSSÕES 119
9.1 Contribuições 123
9.2 Trabalhos Futuros 123
REFERÊNCIAS 125

1 INTRODUÇÃO

Predizer a estrutura tridimensional (3D) de proteínas a partir do conhecimento de sua sequência linear de aminoácidos, sem o auxílio de estruturas de referência pré-determinadas, tem sido um dos maiores desafios da biologia e bioinformática estrutural. Apesar do avanço computacional nas técnicas de predição de estruturas de proteínas, ainda não existe uma ferramenta capaz de predizer com alta precisão, comparável aos métodos experimentais, a estrutura 3D de proteínas.

O problema de predição de estruturas 3D de proteínas (PSP, *Protein Structure Predicition*) através de métodos computacionais pode ser visto como uma das questões centrais da Biologia Molecular, ainda sem uma solução completa (Pedersen e Moult, 1996; Zhang, 2008). O conhecimento de estruturas tridimensionais é de vital importância para o entendimento das funções das diferentes proteínas, as quais são fundamentais para a maioria dos processos biológicos e tem sido alvo de crescente interesse por parte de pesquisadores.

A disponibilidade de estruturas no banco de dados público de proteínas PDB (*Protein Data Bank*), repositório de estruturas de proteínas já resolvidas, tem favorecido a utilização de técnicas baseadas em conhecimento, as quais têm obtido sucesso para várias proteínas. O problema para esse tipo de abordagem é que nem todas as proteínas possuem similaridades no PDB.

Existem métodos de modelagem por similaridade que realizam predições bem-sucedidas, mas esses métodos só são possíveis para proteínas que possuem estruturas similares já conhecidas. Por essa razão, a predição de estruturas tridimensionais de proteínas permanece um desafio, uma vez que diversas estruturas não podem ser preditas e/ou determinadas eficientemente com os métodos existentes.

As pesquisas em bioinformática são realizadas efetuando-se experimentos científicos totalmente executados e analisados através de computadores. Grande parte destes experimentos, chamados *in silico* [1], corresponde à composição de vários programas em sequência, onde a saída de um deles é utilizada como entrada de dados do próximo, com a

1 *In silico* é uma expressão comumente usada na simulação computacional e áreas correlatas para designar algo que ocorreu em ou através de uma simulação computacional.

utilização de um grande número de bancos de dados. *Workflows* científicos são projetados para realizar experimentos *in silico* com o intuito de processar e analisar uma grande quantidade de dados usando simulação computacional. Os experimentos são organizados como uma sequência de etapas que caracterizam um fluxo de execução, onde em cada etapa utilizam-se diferentes *softwares*.

Estes *softwares* não possuem um modelo de representação de dados comum. Por isto, quando um *workflow* científico é construído com o objetivo de integrar e processar dados, cada etapa precisa analisar a estrutura dos dados, processá-los e preparar os mesmos de acordo com a estrutura necessária para a execução da próxima etapa do *workflow*. Desta maneira, os *workflows* científicos usam diferentes algoritmos provenientes das áreas da matemática e da ciência da computação, os quais são capazes de processar, armazenar e transformar os dados utilizados em informação útil para diferentes análises de pesquisadores de biologia.

A comunidade científica vem, cada vez mais, utilizando-se de computadores para execução e análise de seus experimentos. Geralmente esses experimentos, chamados *in sílico*, correspondem à composição de vários programas em sequência, onde sua execução é realizada manualmente com o uso de *shell scripts*. Ainda existe uma grande deficiência devido a heterogeneidade e natureza distribuída dos dados, por conta dos diversos formatos específicos de entrada e saída das ferramentas disponíveis. A execução manual de programas sequenciais ou o uso de *shell scripts* também apresentam problemas relacionados com a clareza, flexibilidade, fluxo dos dados e a própria manutenção do processo. De acordo com Wainer (1997), o uso de *workflows* científicos oferece o apoio necessário ao ciclo de execução e análise, tornando possível a criação de um ambiente com independência entre as diversas aplicações científicas.

Este projeto, tem como objetivo desenvolver um modelo de *workflow* científico, capaz de automatizar o processo de refinamento da estrutura tridimensional de proteínas preditas pelo método CReF, com o objetivo de se obter resultados confiáveis e menor tempo de execução do processo. O modelo está organizado em etapas, e com as definições de acordo com suas responsabilidades, que atende aos requisitos levantados.

Este documento também define uma solução que possibilita a melhoria da execução de experimentos em Bioinformática de forma automatizada e com um ganho de tempo, tendo com isso o aumento do número de experimentos realizados, assim como a redução do tempo total dos experimentos. Isso é possível devido a automatização de execução das etapas do processo

de refinamento. Descreve-se os resultados realizados com três proteínas, com características diferentes e com refinamentos já conhecidos.

EsPara implementar o modelo do *workflow* proposto, foi utilizado um estudo de caso usando-se a proteína cujo o código PDB é 1GAB, cadeia A da proteína PAB (*Escherichia coli*), determinada por NMR, considerada proteína pequena (53 aminoácidos) classe SCOP α, (Johansson *et al*.,1997), juntamente com as proteínas 1YWJ, da estrutura do domínio FBPWW1 (*Homo sapiens*), determinada por NMR, considerada proteína pequena (28 aminoácidos), classe SCOP β, (Pires *et al*., 2005) e a 1GPT, da estrutura em solução das tioninas gama 1-H e gama 1-P de cevada (*Hordeum vulgare*) e endosperma de trigo determinada por 1H-NMR determinada por NMR, considerada proteína pequena (47 aminoácidos), classe SCOP pequenas proteínas (αβ), (Bruix *et al*., 1993) que serviram para completar os testes, validando assim a execução *do workflow.*

1.1 Motivação

As motivações que norteiam este trabalho no âmbito da bioinformática, deve-se à possissibilidade de se integrar a área computacional e biológica para o desenvolvimento de programas computacionais, com intuito de ajudar na resolução de problemas enfrentados pelos pesquisadores.

A Bioinformática é uma área que vem crescendo desde a década de 90, com o avanço das pesquisas de sequenciamento do Genoma Humano. Com isso, surgiu a necessidade de automatização dos processos de pesquisa genética de forma otimizada com a utilização de ferramentas da computação para entender e resolver os problemas de biologia. O resultado desta composição foi um grande avanço nas pesquisas e um aumento relevante no volume de informações extraídas e armazenadas em bancos de dados públicos e privados disponíveis na internet, fatores decisivos para consolidar a informática como importante área do conhecimento científico.

De acordo com Westhead *et al.* (2002), os computadores são importantes na Bioinformática porque muitos problemas nesta área requerem que a mesma tarefa seja realizada várias vezes, por exemplo, comparar uma nova sequência genômica com outras sequências de mesmo tipo armazenadas em banco de dados a fim de descobrir similaridades.

As pesquisas em bioinformática são realizadas efetuando-se experimentos científicos

totalmente executados e analisados por meio de computadores. Grande parte destes experimentos, denominada na comunidade científica por experimentos *in silico,* que correspondem à composição de vários programas em sequência, onde a saída de um deles é utilizada como entrada de dados do próximo, com a utilização de um grande número de bancos de dados. Normalmente, esses diversos programas são executados com controle manual pelos cientistas através do uso de linguagens de *shell scripts*, que sendo executado dessa forma, ocasiona problemas para se definir a ordem correta em que as etapas serão executadas, monitorar sua execução, entre outros.

Uma preocupação para a realização de experimentos *in silico* se refere à utilização dos SGWfC. Esses sistemas geralmente usam linguagens específicas, obrigando os pesquisadores a descrever o *workflow* científico em baixo nível de abstração exclusivamente para este sistema. Por conseguinte, os pesquisadores se concentram mais nas questões de implementação e menos na definição de requisitos essenciais do experimento *in silico*, o que torna toda a tarefa de concepção mais complexa.

O desenvolvimento de métodos computacionais para prever estruturas tridimensionais a partir de sequências é provavelmente o único caminho para preencher a lacuna entre a quantidade de sequências e resolução das estruturas tridimensionais (Lee *et al*., 2004). Dorn e Norberto de Souza (2008) propuseram um novo método para predição aproximada de estrutura tridimensional de proteínas, O método CReF (*Central Residue Fragment-based method*), que realiza a predição da estrutura 3D aproximada de proteínas ou polipeptídios, já demonstrou bons resultados, consolidando o potencial científico para mais estudos e aplicações (Dorn e Norberto de Souza, 2010, 2008; Dorn *et al*., 2008).

A ideia do método CReF é que as estruturas aproximadas preditas sejam boas o suficiente para serem submetidas a protocolos de refinamento pelas técnicas de simulação pela dinâmica molecular (DM) (Dorn e Norberto de Souza, 2008). Existem vários métodos de Mecânica Molecular, os quais diferem-se pela natureza das equações, assim como detalhes das suas parametrizações que opera a partir das coordenadas cartesianas dos átomos contidos em um arquivo no formato PDB. Entre eles está o AMBER 14, um pacote de programas utilizado para conduzir cálculos de Mecânica e Dinâmica Molecular de biomoléculas em determinados campos de força.

Na revisão da literatura não foi possível encontrar tecnologias maduras para lidar com esses experimentos, desde a concepção de *workflow* em níveis mais abstratos, até sua

implementação em um SGWfC, Sistemas Gerenciadores de *Workflows* Científicos.

Considerando este cenário, esta dissertação propõe uma abordagem de apoio para experimentos *in silico.* Para isso desenvolveu-se um modelo de *workflow* capaz de automatizar o processo de refinamento da estrutura 3D aproximada de proteínas, buscando manter suas principais características, automatizando sua execução com intuito de minimizar os riscos envolvidos com a utilização dessa tecnologia de resultados incorretos ou inconsistentes.

1.2 Objetivo Geral

O principal objetivo desse trabalho consiste no desenvolvimento de um *workflow* científico que auxilie os pesquisadores (ou usuários) em tarefas de experimentos *in silico*, com intuito de automatizar ao máximo a configuração de todos os requisitos necessários do AMBER 14 a fim de realizar o processo de dinâmica molecular, reduzindo-se o tempo necessário de execução e, assim, melhorar o desempenho de suas pesquisas.

1.2.1 Objetivos Específicos

O principal objetivo desta pesquisa consiste em apoiar os pesquisadores nos experimentos *in silic*o que utilizam a tecnologia de *workflow* científico. Esse objetivo pode ser decomposto em:

- Identificar, através de revisão da literatura, as características do *workflow* científico aplicáveis ao projeto;
- Definir as atividades de apoio à identificação dos requisitos;
- Testar com os resultados extraídos do CReF submetendo ao *workflow* desenvolvido os arquivos gerados pelo CReF para validar o resultado do refinamento.
- Definir a representação textual a partir das características definidas.

1.3 Metodologia

Para o desenvolvimento desse trabalho, segue cada um dos passos executados:

- Realizou-se revisão da literatura sendo que nesta atividade realizou-se uma revisão

sobre a técnica de composição de *workflow* científico para experimentos *in silico*. O foco desta revisão inicial foi entender os conceitos básicos desta área de pesquisa, os principais termos e identificar os elementos do domínio de *workflow* científico e suas características. Precisou-se compreender o funcionamento da dinâmica molecular, predição de proteínas assim como o método CReF.

- Estudou-se alguns sistemas de automação do fluxo de trabalho que utilizam *workflows* científicos e pode-se perceber que a composição de um *workflow* requer o encadeamento de tarefas, a definição de parâmetros destas, a definição dos dados de entrada, a inserção de fluxos de controle se necessário. A maioria dos SGWfC dispõe de uma interface gráfica para construir o *workflow*. Cabe salientar, que as ferramentas estudadas apresentam muitos recursos que não seriam utilizados no *workflow* de refinamento do processo de predição de estrutura pela ferramenta AMBER 14.
- Levantamento de requisitos: nessa atividade identifica-se os requisitos necessários para o desenvolvimento bem-sucedido deste trabalho;
- Implementação: nessa atividade foi realizada a implementação do modelo de *workflow* e interface web de resultados do refinamento;
- Análise dos resultados: nessa atividade foi realizada a validação dos resultados obtidos junto com o especialista da área.

Para a criação do *workflow* de automatização do processo de dinâmica molecular serão utilizados os seguintes recursos:

- **Linguagem de programação** o *workflow* será desenvolvido utilizando a linguagem *Python*, linguagem computacional orientada a objetos de alto nível, completa e adequada ao desenvolvimento de aplicações baseadas na *web*, redes fechadas ou programas *standalone*, podendo ou não depender do uso de recursos de conectividade. A interface web para visualização dos resultados foi desenvolvida em HTML (*HyperText Markup Language*), que significa Linguagem de Marcação de Hipertexto e para definir a formatação das páginas usou-se a linguagem de folhas de estilo CSS (*Cascading Style Sheets*).

  Sistema Operacional: É utilizado o Sistema Operacional GNU/Linux, devido o mesmo ser uma plataforma *open source* e também possuir uma grande quantidade de ferramentas de bioinformática para este sistema.

- **Aplicação de simulação por Dinâmica Molecular:** Para a modelagem e simulação simulação por dinâmica molecular será utilizado o AMBER 14, aplicação baseada nos termos de *software* livre. Recurso computacional de alto desempenho para investigação de sistemas biológicos. As informações sobres os ângulos do polipeptídio são processadas pelo módulo *teLeap* do pacote para modelagem molecular AMBER 14, gerando a conformação.
- **Bancos de dados:** Tudo que for submetido será armazenado em um banco de dados *MySQL* para a consulta, processamento e devolução dos resultados aos usuários do método. O *MySQL* é um sistema de gerenciamento de banco de dados (SGBD), que utiliza como interface a linguagem de consulta estruturada (SQL - *Structured Query Language*).

1.4 Organização da Dissertação

A dissertação está organizada em 8 capítulos da seguinte forma:

- No Capítulo 1, são apresentados o contexto biológico e computacional da pesquisa, necessários para este trabalho, os objetivos propostos e a metodologia.
- No Capítulo 2, contém a revisão de literatura definindo os conceitos diretamente relacionados com este trabalho, contextualizando sobre Bioinformática Estrutural, organização das proteínas em hierarquia e a classificação estrutural, resíduos de aminoácidos e sua divisão em grupos, formação da ligação peptídica, sobre os tipos de modelagem e métodos de predição de proteínas, o mapa de Ramachandran, o método CReF de predição de estrutura 3D aproximada de proteínas desenvolvido por Dorn & Norberto de Souza (2008, 2010), apresentando as nove etapas do CReF, sobre uma das técnicas computacionais estudo de macromoléculas biológicas, a Dinâmica Molecular, finalizando com o pacote de programa AMBER 14, ferramenta utilizada para essa pesquisa.
- No Capítulo 3, são descritos conceitos de *workflows*, em especial *workflows* científicos, sua importância para a bioinformática e a descrição de duas ferramentas de para geração de *workflow*, o Kepler e o Taverna.

- No Capítulo 4, são apresentados alguns trabalhos relacionados ao uso de *workflows* científicos em Bioinformática.
- No Capítulo 5, são descritas de forma detalhada todas as etapas envolvidas para execução do modelo de *workflow* desenvolvido, que inclui a implementação, a elicitação dos requisitos, a arquitetura do modelo sendo utilizada a proteína 1GBA como Estudo de Caso para ilustrar o uso do *workflow* desenvolvido.
- No Capítulo 6, é apresentada a interface *web* para disponibilizar os resultados dos refinamentos de forma organizada, de modo a ajudar o usuário para visualizar os arquivos de resultados armazenados na pasta gerada pelo refinamento. Na sequência descreve detalhadamente cada tela da interface.
- No Capítulo 7, são apresentados os experimentos com as proteínas 1GPT e 1YWJ, com o objetivo descrever os resultados realizados com o modelo de *workflow* desenvolvido, que juntamente com a proteína utilizada como estudo de caso 1GAB, serviram para desta forma validar a eficiência do processo, demonstrando que os resultados foram muito satisfatórios.
- Por fim, o Capítulo 8 apresenta as considerações finais, concluindo que o *workflow* executa corretamente o refinamento dos diferentes experimentos, as principais contribuições, seguido dos trabalhos futuros. No final dessa dissertação encontra-se as Referências utilizadas para escrita deste trabalho e o material suplementar (Apêndices).

2 REVISÃO DE LITERATURA

Este capítulo tem por objetivo definir conceitos diretamente relacionados com este trabalho, a fim de contextualizá-lo.

2.1 Bioinformática Estrutural

Bioinformática Estrutural (BE) é a conceituação da biologia em termos de moléculas, no sentido físico-químico, e a aplicação de técnicas computacionais (derivadas de disciplinas como: matemática, ciência da computação e estatística) para entender, organizar a informação estrutural associada a essas moléculas (Luscombe *et al*., 2001). Consolida-se como uma nova área do conhecimento, graças à crescente necessidade de se desenvolver programas computacionais que permitam reconhecer sequências de genes; predizer a estrutura tridimensional de proteínas; identificar inibidores de enzimas; organizar e relacionar informação biológica; agrupar proteínas homólogas; estabelecer árvores filogenéticas; analisar experimentos de expressão gênica, entre outros (Alberts *et al*., 2008; Lesk, 2008).

Um dos principais desafios da bioinformática estrutural começa a aparecer na era pós-genômica. Isso compreende entender como a informação decodificada em uma sequência linear de aminoácidos, ou estrutura primária de uma proteína e possibilita a formação de sua estrutura tridimensional gerando conhecimento na qual a proteína se torna um dos principais alvos de estudo juntamente com o entendimento estrutural e funcional de proteínas (da Silveira, 2005).

Para poder armazenar e disponibilizar a quantidade de dados dentro da BE, criou-se o PDB (*Protein Data Bank)*, um repositório público de modelos tridimensionais de macromoléculas biológicas, sendo o mais volumoso banco de dados com informações de proteínas (Berman *et al*., 2000).

Estudos permitiram a separação, identificação e caracterização das proteínas (Dorn e Norberto de Souza, 2010; Lesk, 2008) sendo o próximo desafio da bioinformática a predição da sua estrutura tridimensional (da Silveira, 2005). O aumento do volume de informações sobre estruturas tridimensionais (3D) obtidas por meio de experimentos estimulou a criação de uma subdisciplina na Bioinformática: a Bioinformática Estrutural. Seu principal foco é em representação, armazenamento, recuperação, análise e visualização da informação estrutural das proteínas (da Silveira, 2005). Pode-se verificar este crescente aumento de dados referente

às proteínas, observando-se o gráfico de estruturas depositadas no PDB nos últimos 19 anos. A Figura 1 mostra o crescimento anual do total de estruturas, dados extraídos em 15 de setembro de 2015.

O conhecimento de estruturas tridimensionais é de vital importância para o entendimento das funções das diferentes proteínas, as quais são fundamentais para a maioria dos processos biológicos e tem sido alvo de crescente interesse por parte de pesquisadores.

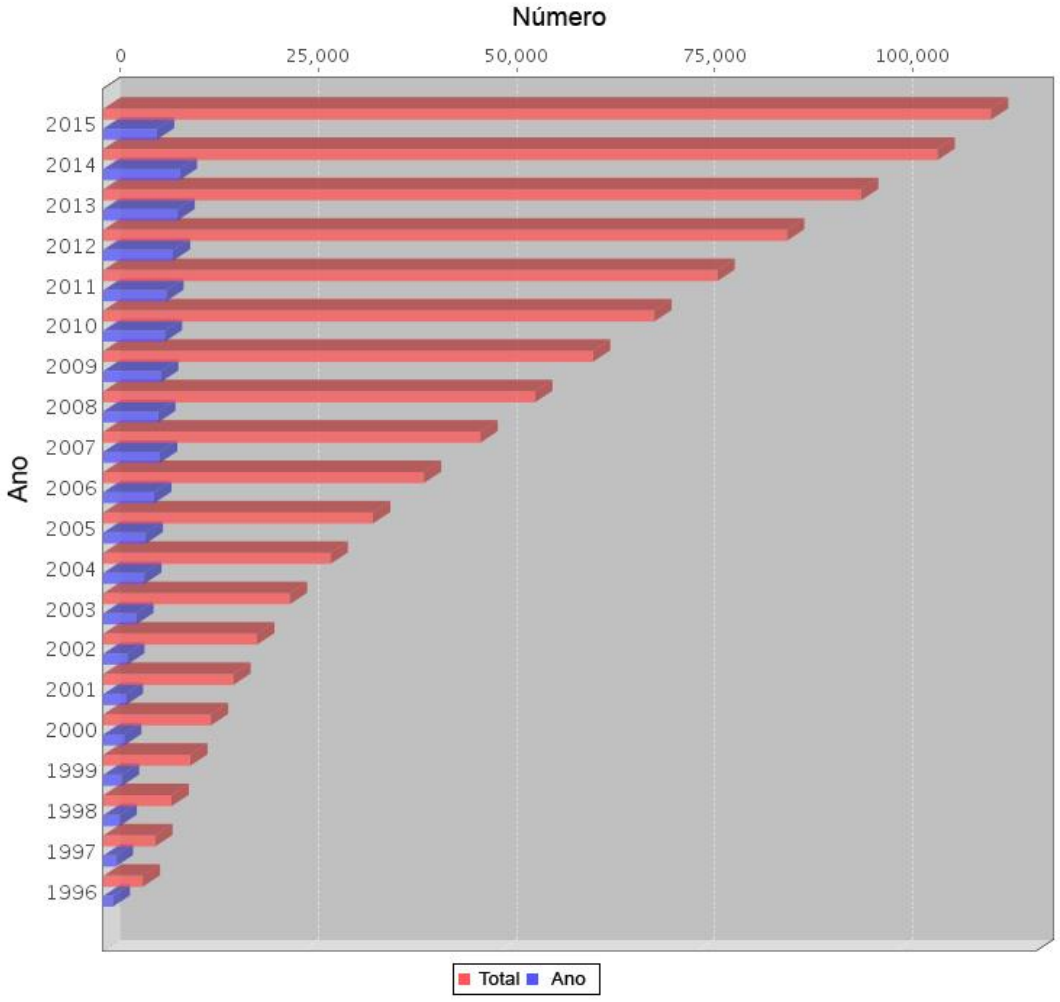

Figura 1 – O gráfico mostra o número de estruturas pesquisáveis por ano. A cor vermelha define o total de dados de estruturas no PDB e a azul o número de estruturas depositadas por ano. Em 1996 o PDB possuía 4.985 estruturas e atualmente, 19 anos após, no final de 2015 somam-se 112.131. Figura extraída do PDB. Fonte: (http://www.rcsb.org/pdb/statistics/contentGrowthChart.do?content=total&seqid=100Protein Data Bank).

Um dos principais desafios da Bioinformática Estrutural é entender como a informação decodificada em uma sequência linear de aminoácidos, ou estrutura primária de uma proteína, possibilita a formação de sua estrutura tridimensional (Figura 2).

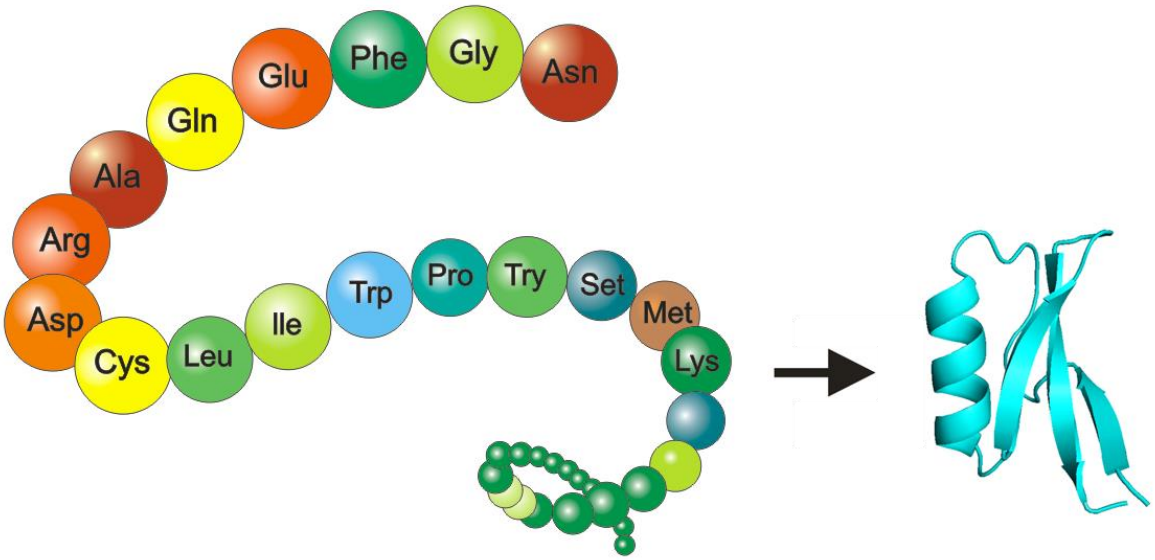

Figura 2 – A sequência linear de aminoácidos de uma proteína define sua estrutura primária possibilitando a sua formação tridimensional.

Uma sequência no formato FASTA é baseada em texto para representar as sequências dos aminoácidos, no qual, para cada sequência existe uma linha de identificação começando com o símbolo “>” e que descreve a sequência com informações variadas, sendo seguida por outras linhas contendo a sequência propriamente dita em um total de 60 a 80 caracteres por linha, como exemplificado na Figura 3:

```
>gi|37221963|gb|AAN78258.1| resistance protein [Arachis simpsonii]
LAKALYNSICDRFECACFLFNVRTISDQEEGLVRLQQTLLSKLLGEWEIKVRSVEEGISMIKEKLSKKRA
LIVLDDVNKIEQLKALAGECDWFSYGTRIVITRDKYLLTAHKVEKIYKMKLLSDPESLELFCWNAFKISR
PKENYEDLSNQAIHYAQ
```

Figura 3 – Exemplo de sequência de aminoácidos em formato FASTA, com identificação (cabeçalho) destacado em vermelho. A sequência FASTA identifica a proteína.

Proteínas pertencentes a uma mesma família conservam a mesma estrutura 3D, mesmo não apresentando grande similaridade entre suas sequências, (Lesk, 2001). Mais importante que

o percentual de identidade entre duas sequências é a identidade de resíduos-chave, sendo estes os verdadeiros responsáveis pela função da proteína.

2.2 Proteínas

As proteínas são moléculas orgânicas, responsáveis pela maioria das atividades fundamentais dos organismos, constituindo-se em polímeros formados por uma sequência de aminoácidos diferentes, onde a atividade biológica depende da sua estrutura (Oliva, 2008). A palavra proteína deriva da palavra grega *protos*, significando "primeiro" ou "mais importante" (Gonçalves, 1994). Apesar de mais tarde se descobrir que a hipótese de Mulder não era correta na sua abrangência, o nome persistiu, e é de certa forma apropriado, pois apesar de constituírem apenas cerca de 20% da massa orgânica nos seres vivos (Gonçalves, 1994), são a mais versátil classe de compostos orgânicos.

Em condições fisiológicas, adota uma estrutura 3D funcional estável e única. São vitais para o funcionamento da maioria dos processos biológicos. Conhecer sua estrutura 3D implica em também conhecer a sua função.

A diferença entre as proteínas está na sequência de aminoácidos que constitui cada uma (Branden e Tooze, 1998). Cada tipo de proteína possui uma sequência única de aminoácidos, correspondendo a uma organização molecular única. Podem ser constituídas por milhares de aminoácidos que se adicionam uns aos outros através da formação sucessiva de ligações deste tipo.

As proteínas são formadas por uma ou mais cadeias polipeptídicas (Figura 4), sequências de aminoácidos ligados através de ligações peptídicas, que em condições fisiológicas adotam estruturas funcionais estáveis, únicas e invariáveis (Dorn, 2008) através de um processo chamado enevolamento, ou dobramento (Branden; Tooze, 1998).

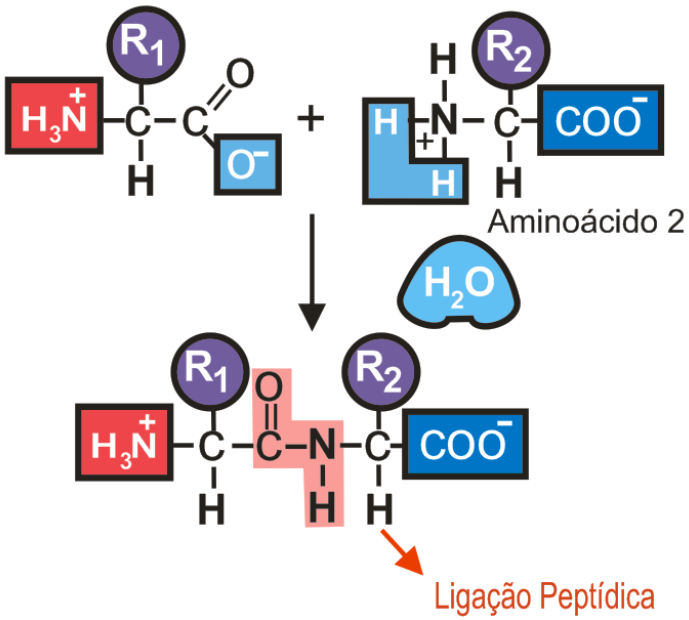

Figura 4 – A ligação peptídica ocorre entre o grupo a-carboxila de um aminoácido (1) e o grupo a-amino de outro aminoácido (2).

As diferentes propriedades das cadeias laterais dos aminoácidos são responsáveis por produzir uma diversa variedade de proteínas funcionalmente estrutural e distintas (Leach, 2001).

Para que duas proteínas sejam consideradas iguais é necessário que a sequência de aminoácidos seja a mesma, sabendo-se que cada sequência de aminoácidos corresponde a uma organização molecular única. A organização molecular é a maneira em que os aminoácidos interagem entre si e/ou com o meio. Existem níveis de organização estrutural de proteínas, que são chamadas de estruturas que podem ser classificadas em quatro tipos (Branden; Tooze, 1998): primária, secundária, terciária e quaternária.

A sequência de aminoácidos representa a estrutura primária das proteínas. Os aminoácidos são formados por um carbono central que se ligam a um hidrogénio, um grupo carboxílico (COOH) e um grupo amina (NH_2). Estes dois grupos estão ligados a um átomo de carbono que é designado como Cα (Carbono alfa). Esta região formada pelo grupo amina, o grupo carboxílico e Carbono alfa é comum a todos os aminoácidos que participam na formação de proteínas e é denominada esqueleto da cadeia principal de uma proteína.

Em sistemas biológicos já foram encontrados mais de 300 aminoácidos, mas de regra geral as proteínas contêm apenas 20 aminoácidos diferentes (Gonçalves, 1994).

2.3 Dobramento de Proteínas

O dobramento de proteínas (*Protein Folding*) é um processo químico através do qual a estrutura de uma proteína assume a sua configuração funcional. Todas as moléculas de proteínas são cadeias heterogêneas não-ramificadas de aminoácidos. Ao dobrar e enrolar-se para tomar uma forma tridimensional específica.

As proteínas enovelam-se para alcançar sua conformação nativa por rotas direcionadas, nas quais pequenos elementos da estrutura se fundem em estruturas maiores (Voet e Voet, 2006). A estrutura de diversas centenas de proteínas já foi determinada, e em todos os casos, a cadeia principal polipeptídica se enovela para adotar a conformação específica, processo do enovelamento de proteínas. Estudos mostram que a sequência de aminoácidos determina o padrão de enovelamento de uma proteína.

Segundo Levinthal (1969), revela que o enovelamento de proteínas está longe de ser um processo aleatório, devendo seguir um processo ordenado que seja capaz de evitar a maior parte das conformações intermediárias possíveis.

Estudos mostram que a sequência de aminoácidos determina o padrão de enovelamento de uma proteína. A sequência primária contém informações suficientes para o enovelamento de uma proteína, entretanto os pesquisadores não são capazes de predizer a estrutura terciária de uma proteína a partir apenas da sua sequência, devido ao fato de o "código" de enovelamento de proteínas ser muito complexo (Lehninger; Nelson; Cox, 2005).

Muito tem sido escrito sobre o “Paradoxo de Levinthal” (Levinthal, 1969), e suas várias resoluções. O paradoxo envolve a constatação de que existe tempo insuficiente para pesquisar aletoriamente todo o espaço conformacional, disponível para o *Folding* (*folding pathway*) ou dobra de um polipeptídio de cadeia como as proteínas.

2.4 Estruturas de Proteínas

A estrutura 3D de uma proteína pode ser obtida, experimentalmente, através de técnicas de cristalografia por difração de raios X ou por ressonância magnética nuclear NMR (*Derived Energy Restraints*). A cristalografia por difração de raios X é o mais antigo e mais preciso método para determinação da estrutura de uma proteína. A técnica permite determinar a

estrutura 3D de proteínas, não existindo limites para o tamanho das moléculas em estudo, porém, as amostras (cristais) sofrem com danos causados pela radiação aplicada, não podendo ser analisada a dinâmica das interações entre proteínas, substratos e solventes.

A ressonância magnética nuclear, por sua vez, é uma técnica mais nova, apresentando vantagens referentes a possibilidade de estudo da estrutura e da dinâmica da molécula em estado líquido ou em um ambiente fisiológico. A principal desvantagem dos métodos experimentais para a determinação da estrutura 3D de proteínas está relacionada ao alto custo dos experimentos e ao elevado grau de complexidade. Isto, motivou cientistas da computação, físicos, biólogos e matemáticos a trabalharem no desenvolvimento de novas metodologias que pudessem predizer de forma correta a estrutura terciária de uma proteína, dada unicamente a sua sequência de aminoácidos.

2.5 Organização Estrutural

Proteínas podem ser representadas em até quatro níveis distintos de organização estrutural (Lehninger; Nelson; Cox, 2005): primário, secundário, terciário e quaternário, conforme mostra a Figura 5.

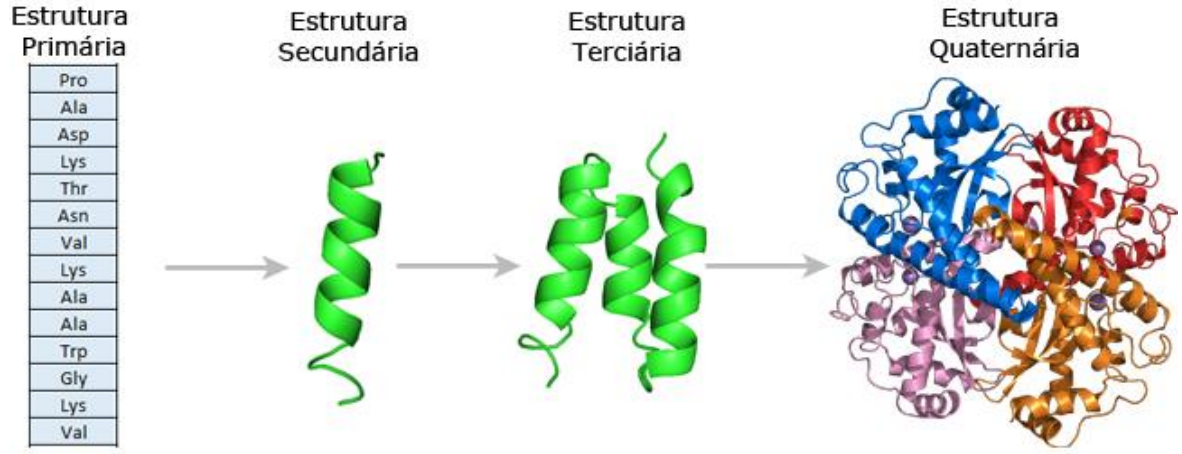

Figura 5 – Representação dos tipos de estruturas das proteínas. A estrutura primária consiste em uma sequência de aminoácidos unidos por ligações peptídicas. A estrutura secundária é o arranjo espacial dos átomos da cadeia principal em um determinado segmento da cadeia polipeptídica. A estrutura terciária é o arranjo tridimensional total de todos os átomos de uma proteína. A estrutura quaternária é o arranjo tridimensional de duas ou mais cadeias polipeptídicas. Fonte: Adaptada (Carneiro e Junqueira, 2005).

2.5.1 Estrutura Primária

A estrutura primária é o nível estrutural mais simples, representado pela sequência de resíduos de aminoácidos ao longo da cadeia polipeptídica em ordem linear (Lehninger; Nelson; Cox, 2005), sem preocupação com orientação espacial da molécula. É formada pela sequência de resíduos de aminoácidos constituintes em ordem de ligação, conforme mostrado na Figura 6. Essa sequência é determinada geneticamente.

```
>BBD:L|PDBID|CHAIN|SEQUENCE
DIVMTQSPSSLTVTTGEKVTMTCKSSQSLLNSRTQKNYLTWYQQKPGQSPKLLIYWASTRESGV
PDRFTGSGSGTDFTLSISGVQAEDLAVYYCQNNYNYPLTFGAGTKLELKRADAAPTVSIFPPSS
EQLTSGGASVVCFLNNFYPKDINVKWKIDGSERQNGVLNSWTDQDSKDSTYSMSSTLTLTKDEY
ERHNSYTCEATHKTSTSPIVKSFNRNEC
```

Figura 6 – Exemplo de sequência de resíduos de aminoácidos da proteína cuja estrutura tridimensional foi resolvida e depositada no *Protein Data Bank* (PDB).

Um polipeptídio, formado pelas ligações peptídicas entre aminoácidos da estrutura primária, se enovela formando a estrutura terciária resultado do enrolamento através de elementos de estrutura secundária, representado pelos arranjos estáveis de resíduos de aminoácidos que formam os padrões estruturais (Dorn, 2008). A estrutura primária é o nível estrutural mais simples e mais importante já que dele deriva todo o arranjo espacial da molécula.

2.5.2 Estrutura Secundária

O termo estrutura secundária refere-se ao conjunto de estruturas locais, estáveis, e que são elementos comuns à maioria das proteínas (Stryer, 1988). Por estas razões, é possível prever, com alguma confiança, estes elementos estruturais, pelo que serão potencialmente uma importante fonte de informação acerca da estrutura da proteína.

Arranjos estáveis de resíduos de aminoácidos formam padrões estruturais, ou regulares que representam o nível secundário de organização estrutural de uma proteína (Dorn, 2008). As cadeias da estrutura primária se dobram e se enrolam de modo complexo, para constituírem um arranjo espacial dos átomos de uma proteína conhecido como sua estrutura secundária, como mostra a Figura 7 (Carneiro e Junqueira, 2005).

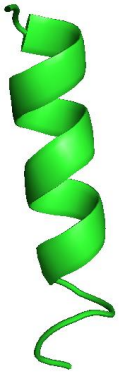

Figura 7 – Estrutura secundária de uma proteína.

A estrutura secundária é caracterizada por estabelecer ligações de hidrogénio entre aminoácidos de uma cadeia proteica, conforme mostra a Figura 8. Os elementos deste tipo de estrutura mais comuns em proteínas são as hélices-α e as folhas-β. As hélices-α são estruturas regulares cuja principal força de estabilização são as pontes de hidrogênio entre os grupos amino e carboxílico do mesmo segmento ocorrendo em escala espiral, Figura 7 (Lesk *et al.*, 2008). Essa é a mais típica dos agrupamentos de estrutura secundária. Essas estruturas podem girar em dois sentidos: para direita (dextrógira) e para a esquerda (levógiros), sendo que na natureza são conhecidas apenas em sua forma dextrógira.

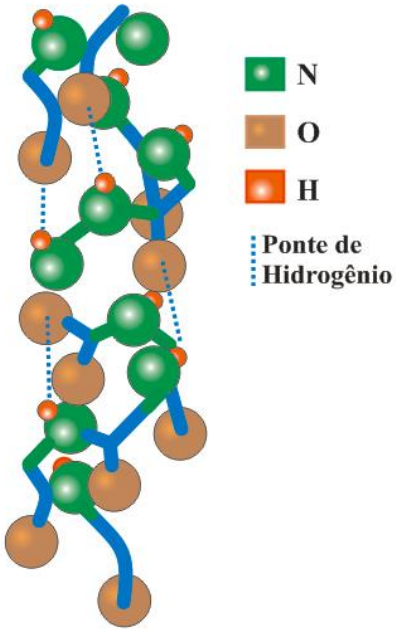

Figura 8 – Representação gráfica da estrutura tridimensional da hélice α, as linhas pontilhadas e as pontes de hidrogênio.

As folhas-β são estruturas regulares formadas quando as estruturas polipeptídicas estão dispostas lado a lado (Pauling; Corey; Branson, 1951). A Figura 9 mostra a estrutura de uma folha-beta anti-paralela formada por duas cadeias. As cadeias de peptídeos se unem formando filas paralelas que se estabilizam de maneira intermolecular mediante pontes de hidrogênio (Pauling; Corey; Branson, 1951). O que diferencia a hélice-alfa da folha-beta é basicamente o fato de que aquelas são estabilizadas por pontes de hidrogênio dentro da cadeia, as últimas são estabilizadas por ligações de hidrogênio entre os filamentos dos peptídeos (Berg *et al*., 2008), isso faz com que a folha-beta possua geometria achatada e rígida.

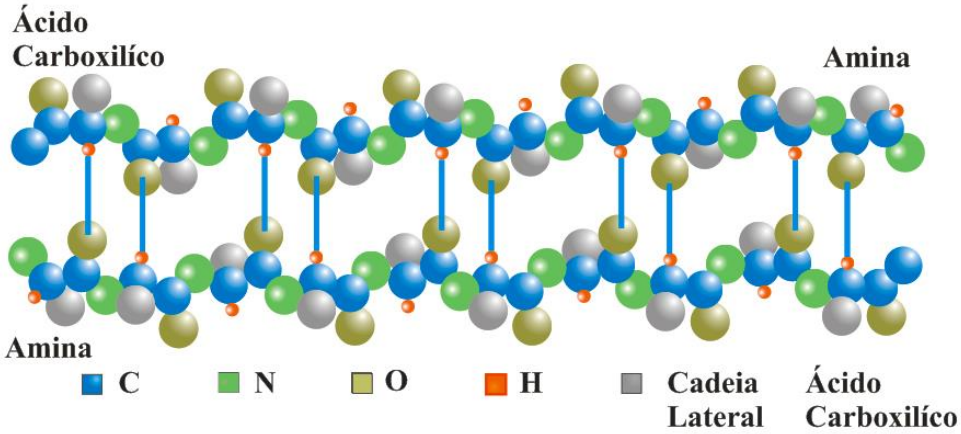

Figura 9 - Nesta figura as cadeias laterais são representadas apenas por um átomo. Os terminais amina e ácido carboxílico estão identificados, ilustrando a orientação antiparalela das cadeias. Fonte: Adaptado de Kerian, 2011.

Além de hélices e folhas, há as estruturas irregulares como as voltas e as alças, que são ditas espirais desorganizadas e conectam sucessivas estruturas secundárias regulares (hélice ou folha) (Voet e Voet, 2006). Uma proteína globular contém, aproximadamente, dois terços de resíduos em hélices e folhas e um terço em estruturas irregulares (Lesk, 2001). Voltas e alças tendem a ser mais flexíveis do que hélices e folhas nas mudanças conformacionais. As voltas acontecem onde o polipeptídeo muda de direção, isto é, acontecem após uma estrutura secundária regular (Voet e Voet, 2006). No mapa de Ramachandran, as conformações em estruturas irregulares podem ocupar qualquer região, inclusive regiões de hélices α e folhas β. Em razão disso, é difícil predizer estas estruturas por meio de métodos computacionais (Dorn, 2008).

O termo estrutura terciária refere-se à relação espacial entre aminoácidos distantes na sequência (Stryer, 1988). Na prática, a fronteira entre estrutura secundária e terciária não é muito nítida (por exemplo, uma folha β contém relações espaciais entre aminoácidos distantes

na sequência), mas podemos conceber a estrutura terciária como a forma tridimensional que uma cadeia de aminoácidos toma, incluindo normalmente vários elementos de estrutura secundária.

A estrutura terciária é resultante do enrolamento e distribuição no espaço 3D dos arranjos de estruturas secundárias de uma proteína, isto é, quando as estruturas secundárias das proteínas se dobram sobre si mesmas, elas originam uma disposição espacial denominada de estrutura terciária (Figura 10).

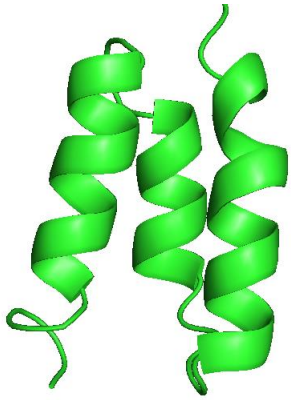

Figura 10 – Estrutura terciária de uma proteína.

A forma tridimensional assumida pela proteína é também chamada de estrutura nativa da proteína ou estrutura funcional (Dorn, 2008). A estrutura 3D assumida por uma proteína está diretamente relacionada à sua topologia (dobramento ou enovelamento), essa estrutura confere a atividade biológica às proteínas, sendo que, através do seu conhecimento, é possível analisar e predizer a função de determinada proteína em uma célula, assim é possível identificar-se o sítio ativo, ou de ligação de uma proteína (Lehninger; Nelson; Cox, 2005).

A forma como as estruturas secundárias se dispõe no espaço tridimensional e o tipo de sucessão como essas estruturas estão conectadas é que determinam uma topologia. A combinação da topologia e estrutura 3D são os fatores que tornam possível analisar a função da proteína no organismo (Dorn, 2008).

Algumas proteínas podem apresentar duas ou mais cadeias polipeptídicas, cada uma denominada de “subunidade”, exibindo um nível de organização estrutural a mais, são as estruturas quaternárias.

2.5.3 Estrutura Quartenária

O arranjo espacial dessas subunidades em suas formas terciárias e suas interações formam a estrutura quaternária (Figura 11). Esta estrutura é mantida pelas mesmas forças que determinam os níveis estruturais anteriores (Dorn, 2008).

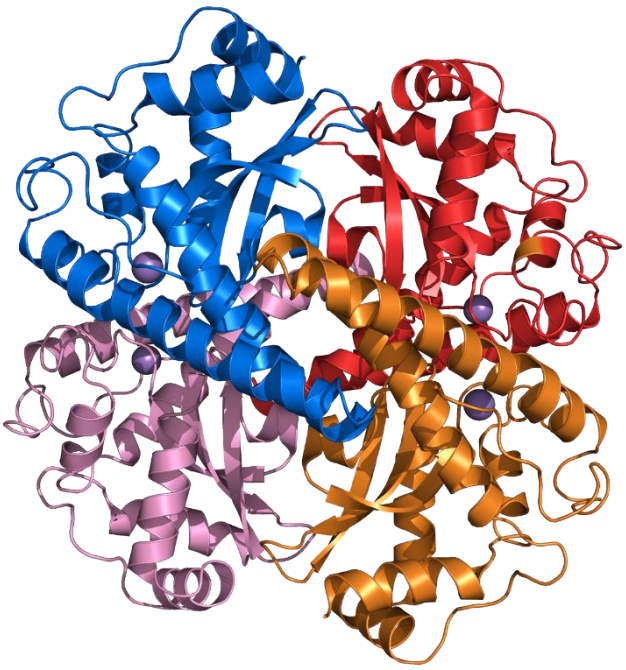

Figura 11 – Estrutura quaternária de uma proteína de código PDB 1VAR.

2.6 Proteínas Homólogas

Algumas sequências de aminoácidos apresentam grande similaridade, e sendo assim, desempenham a mesma função em organismos distintos, bem como nos genes que as codificam o que sugere que os organismos em questão possuem uma mesma origem evolucionária (Khouri, 2002). Devido a mutações relacionadas ao mecanismo de evolução genética, podem ocorrer divergências moleculares e, como consequência, podem-se formar famílias de proteínas com estruturas equivalentes (Santos, 2008).

Proteínas homólogas tendem a ser semelhantes em sua forma, mesmo apresentando sequências um pouco diferentes (Khouri, 2002). Para serem consideradas homólogas, as sequências semelhantes de proteínas precisam ser originadas de um ancestral comum, o que pode ser descoberto por meio de comparação das sequências de aminoácidos destas proteínas (Prosdocimi, 2002).

2.6.1 Modelagem de Proteínas Homolólogas

A modelagem de proteínas está baseada na semelhança entre estruturas primárias de uma proteína-molde de estrutura 3D conhecida e de proteínas alvos de estruturas desconhecidas, ou aquela para a qual se deseja obter a estrutura 3D, que trazem consigo uma similaridade na sua estrutura (Chothia; Lesk, 1986). Desta maneira, um modelo 3D de uma proteína alvo pode ser construído a partir de proteínas que possuem relação de similaridade estrutural (Ginalski, 2006). A modelagem por homologia é bastante satisfatória em prever estruturas 3D de proteínas, pois baseia-se em alguns padrões gerais observados em nível molecular, onde a homologia entre sequências de aminoácidos envolve uma semelhança estrutural e funcional, além disso proteínas homólogas apresentam regiões internas conservadas, sendo que as principais diferenças estruturais entre proteínas homólogas ocorrem nas regiões das alças, que são regiões mais externas (Filho; Alencastro, 2003).

Em caso de falhas nas técnicas experimentais, a modelagem de proteínas por homologia é a única maneira de se obter informações estruturais. O método de modelagem de proteínas por homologia consiste em várias etapas consecutivas geralmente repetidas até que um modelo satisfatório seja obtido e estas etapas são: identificar e selecionar proteínas-molde, alinhar as sequências de resíduos, construir as coordenadas do modelo, otimizar e a validar o mesmo (Marti-Renom, *et al.*, 2000). Depois que uma estrutura homóloga conhecida for identificada poderá ser modelada utilizando-se uma variedade de procedimentos de modelagem por homologia, a Figura 12 mostra o esquema geral desta modelagem.

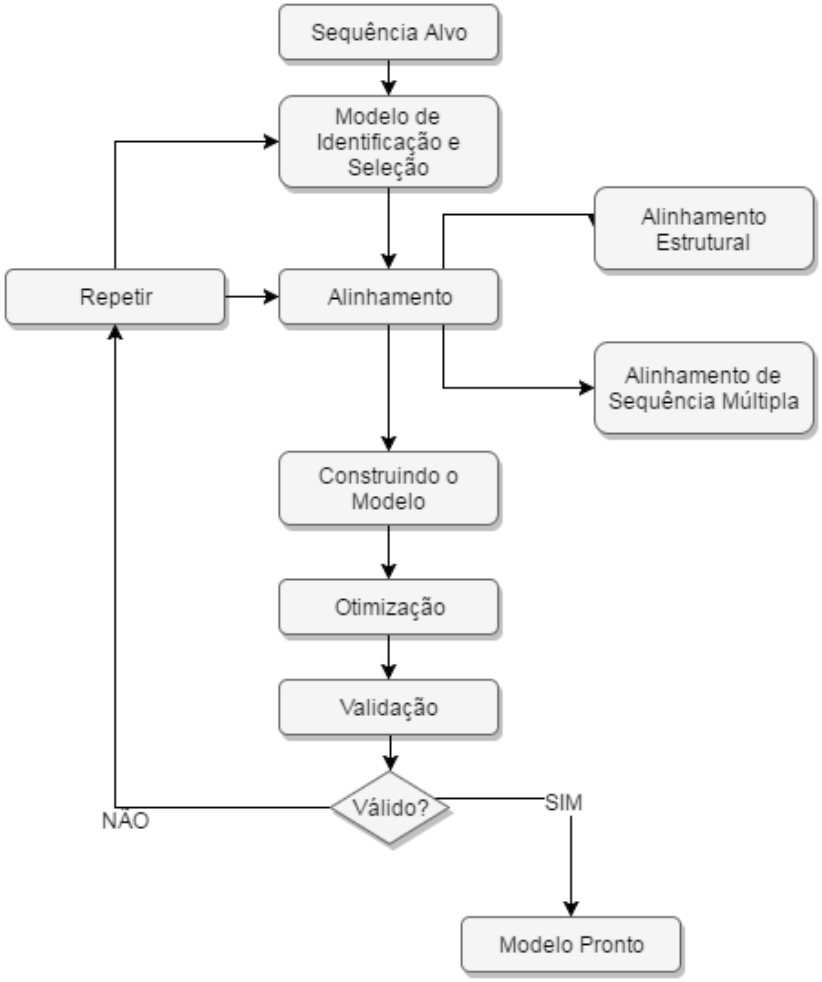

Figura 12 – A modelagem por homologia apresenta quatro etapas: a procura e seleção dos *templates* (sequências de proteínas homólogas), o alinhamento com o *template* alvo, a construção e otimização do modelo e por fim, a avaliação e validação da estrutura gerada. Caso necessário a estrutura final pode passar por um processo de refinamento (Bishop, *et al.* 2008).

2.7 Etapas da modelagem por Homologia

Inicia-se pela procura de *templates* em um banco de dados de estruturas de proteínas, PDB, usando-se como parâmetro de entrada uma sequência primária de estrutura não determinada experimentalmente (alvo) para que esta seja alinhada com possíveis sequências homólogas de estruturas conhecidas depositadas no PDB (*templates*). Uma vez obtida uma lista de *templates* potenciais usando-se um ou mais métodos de busca, é necessário selecionar os *templates* que são apropriados para o problema de modelagem em particular.

Geralmente, seleciona-se os modelos com a identidade mais elevada, isto é, porcentagem mais alta de resíduos idênticos e um menor número de gaps (espaçamentos que podem ser inseridos entre os resíduos para que caracteres semelhantes, por algum critério sejam

alinhados em colunas sucessivas) no alinhamento. Na construção de um complexo de proteína-ligante, o *template* escolhido que contém um ligante semelhante é provavelmente mais importante que a resolução do modelo.

2.7.1 Alinhamento Alvo / Template

Uma vez selecionado o *template*, um método deve ser utilizado para executar o alinhamento entre a proteína-alvo com a proteína-*template*. O alinhamento é um dos principais passos na modelagem, pois é dele que são extraídas as restrições espaciais para a construção do modelo. É possível nos métodos de modelagem molecular comparativa utilizar-se de variadas faixas de identidade, sempre relacionando-se o modelo gerado a partir de uma identidade sequencial com sua utilização. Para sequências de proteínas relacionadas com identidade superior a 40% de identidade residual, o alinhamento será mais preciso. Regiões de baixa similaridade local de sequências, são comuns quando a identidade total da sequência está abaixo de 40% (Saqi *et al*., 1998). Alinhamentos abaixo de 30% começam a apresentar muitas falhas com grandes extensões de gaps e erros nos alinhamentos.

2.7.2 Construção do Modelo

Tendo sido realizado o alinhamento entre a sequência do alvo e o *template*, obtém-se o o modelo construído utilizando-se da modelagem molecular comparativa (Šali & Blundell, 1993). Usa-se a distância geométrica e técnicas de otimização para satisfazer as restrições espaciais obtidas do alinhamento implementadas pelo programa Modeller. Esta deriva muitas distâncias e restrições de ângulos diedros no alinhamento da sequência alvo com o modelo da estrutura tridimensional.

As restrições espaciais, na sequência, alvo são obtidas da análise estatística das relações entre várias características da estrutura da proteína. Vários modelos ligeiramente diferentes podem ser calculados variando a estrutura inicial. Outros fatores como seleção de *template* e um alinhamento preciso, têm um grande impacto na construção do modelo e em sua precisão, especialmente para modelos baseados em uma identidade sequencial abaixo de 40%.

2.7.3 Avaliação dos Modelos

A qualidade do modelo predito determina a informação que pode ser extraída dele, sendo essencial estimar a precisão do modelo tridimensional da proteínal para interpreta-lo. O modelo pode ser avaliado como um todo bem como em regiões individuais, com base na similaridade entre as sequências do *template* e do alvo, observando resíduos importantes em regiões da proteína como o sítio ativo e sua conservação (Sánchez & Šali, 1998). As características de um modelo que são checadas por programas como o Procheck (Laskowski *et al*., 1993), incluem comprimento de ligação, ângulo de ligação, ligação peptídica e planaridade de anéis da cadeia lateral, quiralidade, ângulos de torção da cadeia principal e cadeia lateral e choques entre pares de átomos não ligados.

Há, também, métodos para testar modelos 3D que, implicitamente, carregam muitas características espaciais compiladas de estruturas de proteínas a alta resolução. Estes métodos são baseados nos perfis e potenciais estatísticos de força (SippL, 1993; Luthy *et al*., 1992). Os programas avaliam o ambiente químico de cada resíduo em um modelo com respeito ao ambiente químico esperado como encontrado em estruturas de raios-x à alta resolução.

Apesar do nivel de alta qualidade em suas predições, a técnica de modelagem comparativa por homologia apresenta limitações. A primeira diz respeito à impossibilidade de realizar a predição de novas formas de enovelamento. Isto é explicado pelo fato de tal metodologia estar presa às formas de enovelamento conhecidas e armazenadas no banco de dados de estruturas (PDB). A segunda limitação está no fato de não ser possível estudar o processo de envovelamento da proteína, ou seja, o caminho que a proteína percorre do seu estado desenovelado até o seu estado enovelado e funcional (estado nativo).

As aplicações de modelos moleculares, determinados por modelagem molecular comparativa estão diretamente relacionadas à precisão dos modelos com relação à identidade entre o alvo e o *template*. A qualidade da predição das cadeias laterais pode ser medida pelo RMSD (*Root Mean Ssquare Deviation)* que mede a diferença estrutural entre duas estruturas sobrepostas.

A modelagem comparativa é uma poderosa ferramenta da biologia computacional na elucidação de estruturas proteicas. Trata-se de uma técnica extremamente acurada, de rápida execução e ocupa uma posição de destaque na genômica estrutural. É a ferramenta mais bem sucedida de predição de estruturas tridimensionais de proteínas, apresenta-se como o método

de predição com maior acurácia nos resultados finais (Martim-Remon, 2000).

2.8 Modelagem Baseada em Conhecimento: *Threading*

A modelagem por homologia restringe-se a um espaço relativamente pequeno de estruturas terciárias conhecidas e pela necessidade de similaridades de entrea a sequência alvo e a do tempate (ou molde) obtida do PDB. Pode-se se encontrar proteínas com baixa similaridade na sequência, mas que possuem estrutura terciária e funções similares. Proteínas não evolutivamente relacionadas (não homólogas), também podem ter estruturas similares, assumindo o mesmo tipo de dobramento. Devido a esse fato, motivou o desenvolvimento do método baseado em conhecimento *(knowledge-based*) *threading*.

Essa abordagem faz o reconhecimento de padrões de enovelamento via alinhavamento ou *threading*, considera que o número de dobramentos possíveis é finito e ao invés de procurar por todo o espaço conformacional. Se limita a buscar pelo melhor dobramento possível, dentre as proteínas que possuem estrutura conhecida (Leach, 2001). Para tanto, os métodos de *threading* alinham uma sequência proteica de busca diretamente nas estruturas tridimensionais conhecidas, buscando assim encontrar o dobramento da proteína de interesse.

Esta técnica consiste em uma dada sequência de resíduos de aminoácidos, é construída uma biblioteca de padrões de enovelamento. Se fragmentos da sequência da proteína-alvo alinhar-se bem à esta forma de enovelamento, é possível deduzir um alinhamento, mesmo que não haja informação suficiente para construir um modelo 3D completo. Após, com as informações obtidas de proteínas com estruturas conhecidas, são construídos modelos estruturais. Com base no valor retornado de uma função objetivo, cada modelo estrutural obtém uma pontuação (*escore*) e com base nesta pontuação todas as conformações construídas são classificadas (ranqueadas) e o modelo 3D final é escolhido. Este método é utilizado para identificar homologias que não podem ser descobertas por um alinhamento par a par de sequências de proteínas.

A Figura 13 representa genericamente um método de predição de estruturas baseado no reconhecimento de padrões de enovelamento.

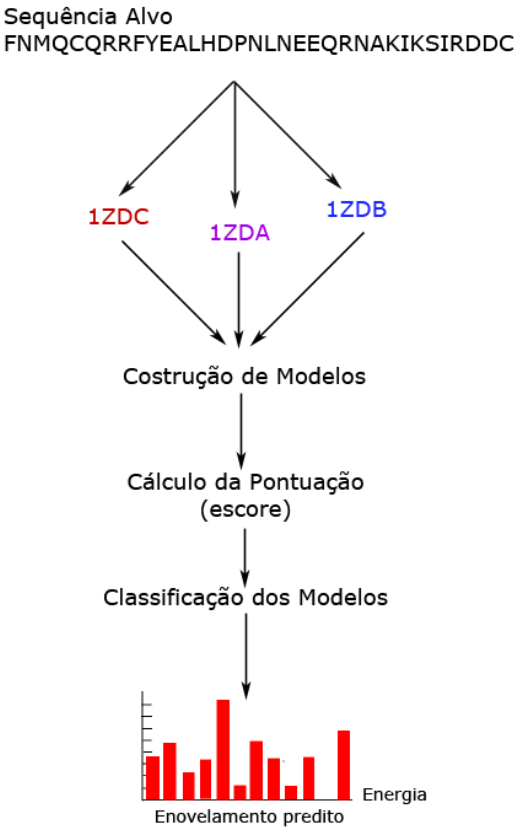

Figura 13 – Representação esquemática do processo de modelagem *threading*. Constução de uma biblioteca de enovelamentos, construção dos modelos, a energia de cada modelo é calculada e as conformações são pontuadas (*escore*), por fim os modelos são classificados (ranqueados). Adaptada (Dorn, 2008).

Sendo assim, sua principal vantagem é a possibilidade de modelar proteínas de médio e grande porte sem que exista a estrutura de uma proteína homóloga determinada. Contudo, por se basear em padrões de alinhavamento, o *threading* nem sempre apresenta resultados satisfatórios.

As técnicas de *threading* são utilizadas quando a proteína não tem sequência com alta similaridade, mas pode ter uma estrutura tridimensional semelhante (Lemer, Rooman, e Wodak, 1995).

2.9 Predição *Ab initio*

O método de predizer a estrutura tridimensional de proteínas a partir de sua estrutura primária, é conhecida como modelagem de proteínas *ab initio* (Simons, 1999). Este método não

utiliza informações de bases de dados estruturais de proteínas, ou seja, essa abordagem não depende do conhecimento prévio de estruturas de proteínas, sejam essas homólogas ou cadastradas no PDB. O objetivo do método é encontrar o valor mínimo global da energia livre de uma proteína que corresponderia, ou à estrutura nativa, ou a uma conformação funcional da mesma (Osguthorpe, 2000). Esse método utiliza o fato que estruturas podem ser reconstruídas utilizando bibliotecas relativamente pequenas de estruturas-modelos de segmentos curtos.

Métodos computacionais para predição *ab initio* de estruturas podem ser classificados em duas categorias principais: métodos de predição baseados em simulações fisicamente realistas normalmente são bem menos eficientes, sob o ponto de vista computacional, que abordagens baseadas em conhecimento. Esta metodologia permite que as diversas propriedades moleculares que dependam da energia total do sistema ou de diferenças de energia (por exemplo, parâmetro de rede, compressibilidade, distâncias interatômicas, energia de formação e outros) possam ser determinadas com precisão. Os principais desafios para método do tipo *ab initio* são a minimização da função de avaliação para proteínas, assim como o crescimento exponencial do espaço de busca conforme o aumento da quantidade de resíduos da proteína.

Os métodos *ab initio* conseguem prever novas formas de enovelamento, pois, não se limitam apenas à proteínas que possuem sua estrutura já conhecida como na modelagem comparativa por homologia e no alinhavamento (Dorn, 2008). Predizer estruturas 3D através deste método possuem dois problemas, o cálculo da energia de uma dada conformação e quanto a estratégia de busca utilizada para encontrar todas as possíveis conformações (Tramontano, 2006). Outro problema se refere à dimensão do espaço de busca conformacional (Levinthal, 1968). Este problema é conhecido e frequentemente referenciado por vários autores como o *Paradoxo de Levinthal.*

Metodologias baseadas apenas no conhecimento de sequências de estruturas similares de proteínas já conhecidas podem melhorar a eficiência de métodos *ab initio*. Esta metodologia considera o processo de alinhavamento de uma proteína em sua estrutura nativa um processo puramente físico que depende apenas da sequência de aminoácidos e do ambiente em que se encontra (Floudas, 2006).

2.10 Ângulos de Torção

A conformação de uma proteína pode ser descrita, quantitativamente, em termos dos

ângulos internos de rotação em torno das ligações entre os átomos da cadeia principal. A conformação dos átomos da cadeia principal de uma proteína é determinada por um par de φ e ψ de ângulos para cada aminoácido.

Os ângulos torcionais correspondem ao ângulo *phi* e *psi* de cada resíduo, que são plotados no gráfico de Ramachandran, o qual fornece informações acerca da qualidade do arranjo estrutural (Mount, 2001). O princípio de que dois átomos não podem ocupar o mesmo espaço limita os valores dos ângulos conformacionais.

A utilização de ângulos de torção para representar a conformação de uma cadeia polipeptídica tem ainda uma outra grande vantagem, o número de variáveis para manipulação é menor do que o número de variáveis presentes em representações baseadas em coordenadas cartesianas, isto torna mais eficiente de um ponto de vista computacional, o tratamento destas variáveis.

O mapa (Figura 14), exemplifica a variação possível dos ângulos φ (*phi*) e ψ (*psi*), de -180º a 180º. O ângulo de torção ômega (ω), em torno da ligação peptídica, normalmente assume o valor de 180º (conformação *trans*) e, ocasionalmente ω = 0º (conformação *cis*) (Lesk, 2008). As regiões do mapa que identificam as conformações permitidas dependem do raio de Van der Waals escolhido para calculá-las (Voet e Voet, 2006). Em termos de enovelamento, as regiões do mapa representam padrões de torção da cadeia polipeptídica para elementos da estrutura secundária como folhas β e hélices α (Voet e Voet, 2013).

Cada aminoácido integra com três ligações para a espinha dorsal da cadeia. A ligação peptídica é planar e rígida é o resultado da estabilização e não permite a rotação. Isso restringe o número de conformações que uma proteína pode adotar.

As rotações podem ocorrer sobre o C α - C, cujo ângulo de rotação é o psi (ψ), e sobre o N-C α, cujo ângulo de rotação é o phi (φ). Por convenção, um grupo R é frequentemente usado para indicar um aminoácido de cadeia lateral (círculos laranjas). A conformação dos átomos da cadeia principal de uma proteína é determinada por um par de φ e ψ de ângulos para cada aminoácido. No gráfico de Ramachandran, cada ponto representa um par de ângulos observada numa proteína. (Alberts *et al.*, 2002)

As regiões do mapa de Ramachandran também estão associadas a conformações de resíduos. No gráfico de Ramachandran, cada ponto representa um par de ângulos observada numa proteína conforme mostra a Figura 14 (Alberts *et al.*, 2008), onde a região mais favorável

é representada em vermelho, região permitida em amarelo, região ainda aceitável em amarelo claro e região não permitida em branco. O canto superior em vermelho trata-se de região favorável para folhas β e no centro direito e esquerdo em vermelho para hélices α, respectivamente.

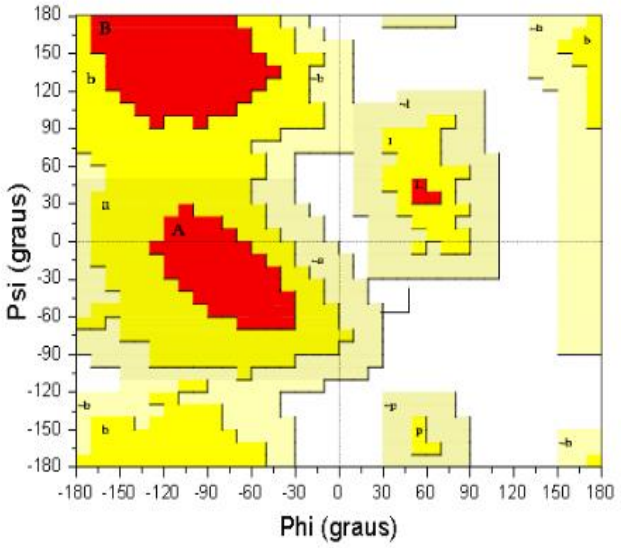

Figura 14 – Gráfico de Ramachandran gerado pelo Procheck. As regiões representadas em vermelho, são as mais favoráveis, as representadas em amarelo, são as favoráveis, as representadas em bege, são as menos favoráveis e as regiões brancas, são as desfavoráveis (Laskowski, 1993).

2.11 O Método CReF – Predição *Ab initio* da Estrutura 3D de Proteínas

Predizer a estrutura proteica a partir do conhecimento de sua sequência primária, sem o auxílio de estruturas de referência pré-determinadas, tem sido um dos maiores desafios. Apesar do avanço computacional nas técnicas de predição de estruturas de proteínas ainda não existe uma ferramenta capaz de predizer a estrutura tridimensional para uma grande variedade de proteínas.

O problema de Predição de Estruturas Tridimensionais de Proteínas (PEP) através de métodos computacionais pode ser visto como uma das questões centrais da Biologia Molecular, ainda sem uma solução completa (Pedersen e Moult, 1996; Zhang, 2008). O conhecimento de estruturas tridimensionais é de vital importância para o entendimento das funções das diferentes proteínas, as quais são fundamentais para a maioria dos processos biológicos e tem sido alvo

de crescente interesse por parte de pesquisadores. Por essa razão, PSP permanece um desafio, uma vez que diversas estruturas não podem ser preditas e/ou determinadas eficientemente com os métodos existentes. Dois pontos cruciais que devem ser considerados para o problema de PSP são: a escolha do método de otimização e a escolha da função de energia.

A disponibilidade de estruturas no banco de dados público PDB, repositório de estruturas de proteínas já resolvidas, tem favorecido a utilização de técnicas baseadas em conhecimento, as quais têm obtido sucesso para várias proteínas. O grande, porém, desse tipo de abordagem é que nem todas as proteínas possuem similaridade no PDB.

Existe métodos de modelagem por similaridade que realizam predições bem-sucedidas, mas esses métodos só são possíveis para proteínas que possuem estruturas similares já conhecidas. O conhecimento que se obtém de estruturas já conhecidas utilizam técnicas de métodos *ab initio* (do latim "desde o começo"). A caracterização do que constitui um método *ab initio* nem sempre pode ser feita de maneira precisa: alguns dos métodos de modelagem por similaridade que realizam predições bem-sucedidas são assim classificados. Nesse sentido, Dorn & Norberto de Souza (2008, 2010) propuseram um novo algoritmo para predizer a estrutura 3D aproximada de uma proteína ou polipeptídio.

O método CReF (Dorn e Norberto de Souza, 2008) usa técnicas de mineração de dados para agrupar dados de estruturas determinadas experimentalmente, utilizando intervalos de variação angular para representar uma conformação através da manipulação das informações estruturais. Esse método leva em consideração que uma proteína é composta por uma sequência de aminoácidos. O princípio que dois átomos não podem ocupar o mesmo lugar no espaço limitam os valores dos ângulos conformacionais. A modelagem no CReF inicia-se pela procura de moldes na base de dados do PDB (Berman *et al*., 2000) *(http://www.rcsb.org/pdb*), usando-se como parâmetro de entrada, a sequência primária de estrutura não determinada experimentalmente (alvo), realizando a predição de sua estrutura 3D aproximada.

2.11.1 Etapas do CReF

1 – Fragmentação da sequência alvo: na primeira etapa o CReF inicialmente pega a sequência alvo (aquela que se busca para fazer a predição, através da sequência de aminoácidos) e divide consecutivamente e continuamente em fragmentos com 5 resíduos de aminoácidos, formando um conjunto com todos os possíveis fragmentos.

2 – Busca por proteínas molde no PDB: para cada fragmento obtido na etapa anterior procura-se por fragmentos molde na base de dados do PDB. Essa busca é realizada com a versão web do programa BLASTp (*Basic Local Alignment Search Tool*) (Altschul *et al*., 1997) que permite a identificação de fragmentos homólogos (moldes) ao fragmento submetido (alvo).

3 – Cálculo dos ângulos de torção dos dupletos: para cada arquivo obtido do PDB, calcula-se para cada fragmento os ângulos de torção do aminoácido central do fragmento molde

4 – Agrupamento de dupletos: o conjunto de tuplas de um fragmento é submetido a um processo de agrupamento que busca identificar as regiões onde as tuplas molde concentram-se no mapa de Ramachandran. Esse agrupamento baseia-se no método probabilístico EM (*Expectation Maximization*) (Witten e Frank, 2005) que analisa as diferentes distribuições de probabilidades para cada grupo, buscando identificar o conjunto de grupos mais favoráveis em uma coleção de dados.

5 – Representação dos ângulos de torção na forma de intervalos: a partir dessa etapa cada fragmento passa ser representado na forma de intervalos de variação, com isso reduz-se drasticamente o espaço de busca conformacional. Estes intervalos são constituídos pelo valor do desvio padrão calculados a partir dos dupletos de um grupo.

6 – Classificação dos grupos em regiões ocupadas no mapa de Ramachandran: nesta etapa os valores de todos os clusters dos grupos de *ki* são rotulados para cada representação na forma de intervalo de variação angular (Dorn e Norberto de Souza, 2008). Todo processo é baseado segundo a região do mapa de Ramachandran que ele ocupa, conforme biblioteca baseada na pesquisa de Thornton (1992) e seus colaboradores (Laskowski *et al*., 1993; Morris *et al*., 1992) que divide o mapa de Ramachandran em 11 regiões preferenciais.

7 – Predição da estrutura secundária: realiza-se a predição da estrutura secundária da sequência alvo K, por meio da determinação da região do mapa de Ramachandran que os ângulos de torção (*phi* e *psi*) de cada aminoácido possivelmente estarão ocupando. O CReF utilizou um consenso obtido entre três métodos de predição (Dorn, 2008).

8 – Construção da conformação inicial na forma de intervalos: inicia-se a construção da conformação inicial, com base nas informações de cada grupo *ki* de cada

fragmento si. Elege-se um grupo para representar o i-ésimo aminoácido da sequência K (Dorn, 2008) guiadas pela predição da estrutura secundária. Essa escolha, não aleatória, é guiada pela predição da estrutura secundária e obrigatoriamente respeita a definição de duas regras.

9 – Otimização das regiões de volta: a determinação da forma de enovelamento de proteínas depende das regiões de volta (Fiser *et al*., 2000), sendo que as estruturas irregulares (voltas e alças) são os tipos de estruturas secundárias mais difíceis de serem previstas, pois elas podem aparecer em qualquer área do mapa de Ramachandran. O método CReF, conhecendo isto, optou por otimizar pela redução do intervalo da conformação inicial, somente as regiões de volta identificadas na predição da estrutura secundária (Dorn, 2008).

A ideia do método CReF é que as estruturas aproximadas preditas sejam boas o suficiente para serem submetidas a protocolos de refinamento pelas técnicas de simulação pela dinâmica molecular (Dorn e Norberto de Souza, 2008).

2.12 Dinâmica Molecular

Dinâmica Molecular (DM) é uma das técnicas computacionais mais versáteis para o estudo de macromoléculas biológicas (Alonso; Bliznyuk; Gready, 2006). Simulação de Dinâmica Molecular (DM) é uma das técnicas computacionais mais versáteis para o estudo de macromoléculas biológicas. As simulações por DM são simulações que modelam a proteína como sendo um sistema newtoniano de átomos ligados por molas, que estima o movimento dos átomos ligados entre si considerando as interações electrostáticas e os parâmetros relativos às energias de alongamento, de torsão e de dobragem da ligação química (FUJTISU, 2003). Por isso, para além das interacções electrostáticas entre os átomos, há ainda a considerar o cálculo dos parâmetros relativos às energias de alongamento, de torsão e de dobragem da ligação química.

As simulações DM determinam aproximadamente o movimento real do sistema por meio de algum campo de força cuja energia cinética do sistema é derivada da velocidade atômica, essa por sua vez é influenciada pela temperatura da simulação. Desta forma, as simulações DM tem por objetivo estudar o movimento do sistema, sendo por isso empregados para a validação de modelos 3D (FUJTISU, 2003).O método da DM, assim como a de Monte

Carlo, é observar a evolução do sistema dado através da determinação do movimento das partículas individuais.

Essas simulações determinam aproximadamente o movimento real do sistema por meio de algum campo de força cuja energia cinética do sistema é derivada da velocidade atômica, essa, por sua vez, é influenciada pela temperatura da simulação. Desta forma, a dinâmica molecular tem por objetivo estudar o movimento do sistema, sendo por isso empregados para a validação de modelos 3D (Fujtisu, 2003).

É o mais detalhado método de simulação que computa a movimentação das moléculas individuais. As equações de movimento de Newton descrevem as posições e momentos das partículas. As equações de movimento para estas partículas que interagem via potenciais intra e intermoleculares podem ser resolvidas usando vários métodos de integração. O processo de simulação molecular possui também a finalidade de otimizar as posições das cadeias laterais, refinar a geometria obtida e validar o modelo construído (Gibas, 2001).

2.12.1 O Pacote de Programa AMBER 14

O AMBER 14 (*Assisted Model Building with Energy Refinement)* refere-se a um programa utilizado para conduzir simulações d conduzir cálculos de Mecânica e Dinâmica Molecular de biomoléculas em determinados campos de força, os quais se diferem pela natureza das equações, assim como detalhes das suas parametrizações. O campo de força representa um ambiente adequado para a determinação estrutural dos potenciais de proteínas juntamente com o conjunto de parâmetros adotados.

A etapa inicial para a utilização do AMBER 14 é a identificação do arquivo de entrada, que deve estar no formato PDB. Nesta fase, o programa LEAP (*Long-range Energy Alternatives Planning)* analisa a proteína inserida no sistema e tem a função de gerar dois arquivos importantes para o funcionamento do AMBER 14 que são eles: o arquivo topológico e o arquivo de coordenada. As informações para a construção destes dois arquivos vêm de um banco de dados contido no próprio Amber, onde estão descritos a topologia dos aminoácidos essenciais (Case *et al.*, 2006).

Assim, ao submeter à molécula ao AMBER 14, o programa LEAP prossegue com a "preparação" da molécula corrigindo átomos não descritos anteriormente e neutralizando a carga total da molécula através do potencial Coulombiano que neutraliza a molécula com a

adição monoatômica de íons, podendo essa rede ser estendida quando se utiliza solvente para os programas de cálculos como o Sander (Case *et al*., 2006).

O Sander (*Simulated Annealing with NMR-Derived Energy Restraints)* representa a principal módulo do AMBER 14, pois neste módulo se conduz os principais parâmetros operacionais para a realização de cálculos de minimização, dinâmica molecular e deslocamentos químicos para RMN (Ressonância Magnética Nuclear). Através do algoritmo *Steepest Descent,* para cálculo da minimização de energia seguido de cálculos de Gradiente, ambos necessários para refinamentos por mecânica molecular e determinação do mínimo local (Case *et al*., 2006). Assim como, o algoritmo Verlet é empregado para cálculos de trajetória atômica na dinâmica molecular (Pearlman et al., 1995).

O algoritmo Verlet é uma integração numérica da equação Newtoniana do movimento. Esta simulação conserva a energia total e o volume do sistema (Fujtisu, 2003). Após o refinamento do modelo a etapa seguinte é a validação do modelo. O primeiro passo da validação consiste na checagem da qualidade do enovelamento, sendo este parâmetro muito ligado a qualidade do molde utilizado.

Há uma série de programas que analisam a estereoquímica do modelo. Dentre estes programas pode-se incluir o Procheck (*http://www.ebi.ac.uk/thornton-srv/software/PROCHECK/*). Este programa, assim como outros similares, usualmente verificam os comprimentos de ligação, ângulos de ligação, ligação peptídica, planaridade de anéis, quiralidade, ângulos torsionais da cadeia principal e lateral, contato estérico entre átomos não ligados. Após a otimização da proteína molde é checada a qualidade estrutural e a qualidade estereoquímica de uma estrutura de proteína gerando análises gráficas sobre a geometria espacial da proteína, resíduo por resíduo. Através de mapas de Ramachandran, os aminoácidos da conformação são analisados em relação às regiões energeticamente favoráveis.O programa Molprobity (Cheng *et al*., 2005) também é utilizado para validação dos modelos gerados. Dentre várias análises fornecidas pelo programa, também representa esta validação através do gráfico de Ramachandran.

O pacote do AMBER 14, também possui um programa chamado Ptraj que é utilizado para processar e analisar conjuntos de coordenadas 3D lidas de uma série de arquivos de coordenadas de entrada. Para cada conjunto de coordenadas lido, uma sequência de ações pode ser executada em uma ordem especifica, de acordo com configurações pré-estabelecidas. Após o processamento de toda a entrada, arquivos de trajetória podem ser escritos, como por

exemplo, no formato PDB (Case *et al.*, 2006). Para exemplificar, a Figura 15 mostra trechos dos arquivos utilizados pelo Ptraj e do arquivo resultante.

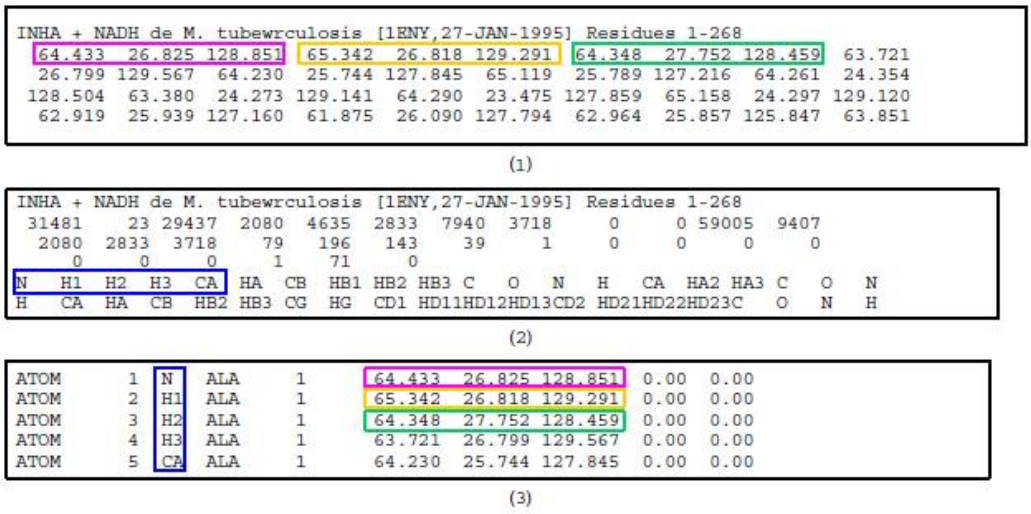

Figura 15 – A primeira figura (1) mostra trecho de um arquivo de saída da simulação por dinâmica molecular; a segunda (2) mostra trecho do arquivo de topologia utilizado pelo Ptraj e a última (3), arquivo PDB gerado após a execução do Ptraj (Machado, 2011).

A ilustração em *shell script* (Figura 16) com os comandos de execução do Ptraj. Caso exista a necessidade de se incluir mais arquivos de entrada, os comandos serão inseridos manualmente.

```
#!/bin/csh -f
trajin ~/INHA_NADH/DINAM/CRD/mdcp_0050ps.crd.gz
trajin ~/INHA_NADH/DINAM/CRD/mdcp_0100ps.crd.gz
trajin ~/INHA_NADH/DINAM/CRD/mdcp_0150ps.crd.gz
trajin ~/INHA_NADH/DINAM/CRD/mdcp_0200ps.crd.gz
trajout mdcp.pdb pdb nobox
center :1-268 mass origin
image origin center
strip :269-9407
```

Figura 16 – Exemplo de comandos de execução com *shell script* do Ptraj (Machado, 2011).

Para a descrição destes tipos de experimentos citados, *workflows* científicos representam uma alternativa atraente. Aliados a serviços *web*, pode-se criar um ambiente com independência e interoperabilidade entre as diversas aplicações científicas e os diversos bancos de dados.

3 *WORKFLOW* CIENTÍFICO

A busca pelo conhecimento faz com que os pesquisadores procurem formas para aprimorar a qualidade dos experimentos científicos e reduzir o tempo necessário para a sua execução. A adoção de técnicas que permitam atingir elevados ganhos de produtividade e qualidade na condução de experimentos científicos pode ser vista como um diferencial competitivo. Cada vez mais a ciência tem feito uso de procedimentos computacionais com o intuito de lidar com o aumento constante dos volumes de dados e manipulações necessárias aos experimentos científicos *in-virtuo* e *in-silico*, usualmente processados por simulações computacionais e analisados via técnicas de visualização (Travassos e Barros 2003) (Deelman, 2009).

Um cientista define o seu *workflow* como um modelo usando uma notação de compreensão bastante intuitiva. A partir desse modelo, o sistema de gerenciamento é capaz de executar o experimento de forma automática, com pouca ou nenhuma intervenção do cientista, utilizando, para isso, a infraestrutura computacional disponível.

Um *workflow* significa automatizar procedimentos, onde documentos, informação ou tarefas são passadas entre os participantes de acordo com um conjunto pré-definido de regras. Apesar de um *workflow* poder ser manualmente organizado, na prática a maioria, são organizados dentro de um contexto de um sistema de informação para prover um apoio automatizado aos procedimentos (Sommerville, 2007).

As instituições de pesquisa ainda dependem exclusivamente da capacidade individual dos cientistas no encadeamento dos programas necessários para a execução de experimentos, com isso torna-se sujeito a falhas e improdutivo, especialmente em se tratando de experimentos complexos, que podem envolver muitos programas, grande quantidade de dados.

Além de *scripts* prontos, os pesquisadores utilizam-se de linguagens de *shell scripts* para implementar os seus *workflows*, devido a facilidade que estas oferecem para fazer chamadas de programas. Cada vez mais se conta com a colaboração de profissionais de informática na definição e execução de *workflows* para auxiliar os pesquisadores em aplicações científicas que demandam grande poder computacional.

Como o *workflow* científico representa a definição das sequências de processos que manipulam dados de modo a construir uma simulação, ele é o principal recurso do experimento

científico. Entretanto, neste contexto, um experimento se caracteriza pela composição e execução de diversas variações de um *workflow,* que incluem a mudança de dados de entrada, parâmetros, programas ou ainda a combinação delas (Ogasawara *et al*. 2008, Oliveira *et al*. 2008). Para representar e apoiar o desenvolvimento do experimento científico, é necessário registrar-se as variações dos *workflows* associados a um experimento, pois o resultado final do experimento será obtido com resultados de variações dos w*orkflows*.

Pode-se dizer que um *workflow* científico nada mais é que a descrição de uma série de passos que, dado uma entrada, resultam na resolução de um problema no âmbito científico. Muitas vezes esses passos não são simples, por possuírem muitas dependências uns dos outros sendo necessárias análises entre eles. As principais etapas envolvidas no desenvolvimento de um *workflow* científico, segundo Wainer *et al.* (1997) são:

1. Modelagem do *workflow*: o *workflow* é definido como um conjunto de atividades agrupadas, utilizando programas já existentes para executar essa etapa;

2. Execução do *workflow*: a execução do *workflow* é feita geralmente por meio da execução de cada uma de suas atividades, na qual a saída de uma atividade corresponde à entrada da próxima. São comuns de aparecerem em *workflows* científicos atividades como testes no fluxo de execução, desvios, restrições, tratamentos de erros, entre outras;

3. Visualização dos resultados: os resultados, tanto parciais quanto finais, podem ser visualizados e analisados, e ainda utilizados para a geração de ajustes e de novas re-execuções do *workflow*;

4. Finalização do *workflow*: pode-se considerar como último passo da modelagem de um processo utilizando *workflows* científicos o momento em que o experimento atingir o seu objetivo, produzindo resultados relevantes e, assim, algum tipo de conhecimento científico poderá ser inferido desses resultados.

Cada vez mais *workflows* científicos permitem a automatização de procedimentos, onde documentos, informações ou tarefas são passadas entre os participantes de acordo com um conjunto predefinido de regras.

Segundo Ludascher (2009), um *workflow* científico tem seu ciclo de vida composto por quatro fases conforme mostra a Figura 17:

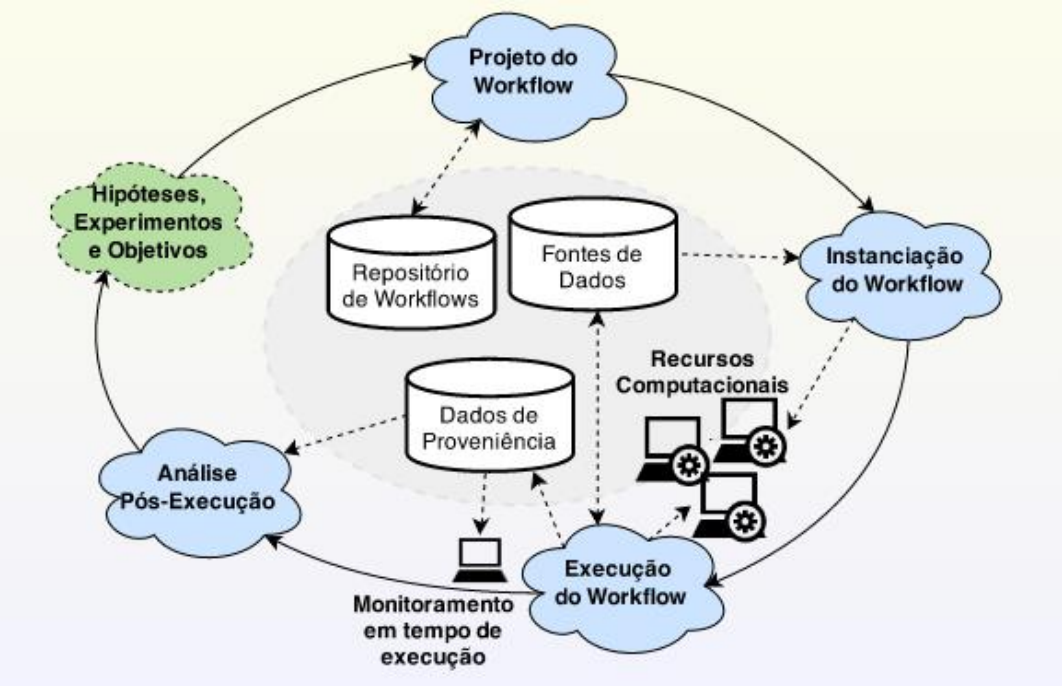

Figura 17 – Representação do ciclo de vida de um *workflow* Ludascher (2009).

- **Projeto:** construção de um modelo a partir de uma hipótese científica que será testada, ou de um experimento com um objetivo específico.
- **Instanciação:** preparação necessária para uma execução particular do *workflow*. A cada nova execução do *workflow*, precisa-se indicar quais são os dados de entrada e os parâmetros definidos para a execução. Pode ocorrer a necessidade de haver alguma seleção e alocação dos recursos computacionais. Dependendo do caso, é necessário transferir os dados de entrada para os computadores onde a execução será realizada.
- **Execução:** corresponde à execução das atividades que compõem o *workflow* nos recursos computacionais disponíveis. Ocorre o processamento dos dados de entrada e produzem novos dados que, por sua vez, podem ser processados por outras atividades. Durante a execução, recomenda-se que os dados gerados e as informações de proveniência sejam registrados, para consultas do cientista do estado atual e até mesmo identificar possíveis problemas da execução. As informações de proveniência incluem além dos dados de entrada, os parâmetros definidos pelo cientista na instanciação do *workflow*, o registro de todas as atividades já executadas na instância, seus respectivos tempos de início e de término, os recursos empregados em sua execução, as referências para os dados usados como entrada e saída de cada atividade. Essas informações

possibilitam que um dado experimento seja reproduzido tal como ele foi realizado da primeira vez. Elas possibilitam também que, em caso de falha na execução do *workflow*, a execução possa ser retomada do ponto onde parou.

- **Análise Pós-execução**: é a fase da inspeção e interpretação dos resultados obtidos a partir da execução do *workflow*. Pode-se obter com a análise dos dados de saída a rejeição da hipótese científica e na análise de informações de proveniência é possível identificar-se as atividades que geram gargalos na execução do *workflow*.

Em bioinformática, *workflows* científicos são usados para também descrever experimentos *in silico* tendo seu uso crescido muito nos últimos anos. Estes *workflows* são construídos através de uma sequência de passos que caracterizam um fluxo de execução onde, cada um destes passos usam programas de terceiros. Para cada execução do *workflow* é considerado um experimento e, de acordo com o resultado alcançado, os parâmetros podem ser revistos e o *workflow* é novamente executado ou, em alguns casos, é preciso realizar algumas execuções parciais novamente (Silva *et al*, 2006). A execução de um determinado *workflow* pode não ser concluída ou não apresentar resultados conclusivos, mas recomenda-se manter registrado os experimentos com execução bem-sucedida, bem como aquelas defeituosas (Meyer *et al*., 2004).

Workflows científicos diferem dos de negócio em diversos aspectos (Cavalcanti, 2005), em especial na bioinformática. Eles caracterizam-se pelo alto grau da intervenção humana em sua execução realizados por meio de especificação e composição de atividades que podem ser definidas de diversas formas como por exemplo usando linguagens de *scripts* ou, mais recentemente serviços *web* (Cavalcanti, 2005). Uma grande diferença entre os *workflows* de negócios e os científicos é o fato de que um *workflow* científico nem sempre está completamente definido antes de seu início (Wainer, 1997) e são executados com um grau de flexibilidade muito superior ao feito no contexto de negócios.

3.1 Importância do *Workflow* para a Bioinformática

As ferramentas de informática têm um papel fundamental na análise dos dados biológicos, a quantidade e complexidade dos dados vêm tornando a análise dos dados cada vez mais custosa, assim, a necessidade de análises *in silico* cresce constantemente, exigindo cada vez mais dos programas acurácia e eficiência na sua utilização. O SWfMS (*Scientific Workflow*

Management System) é um sistema criado, especificamente para gerenciar *workflows* no campo científico. Para que experimentos científicos em larga escala possam ser gerenciados, é necessário que um conjunto de funcionalidades esteja presente. Dentre essas funcionalidades está o apoio à composição dos experimentos, que inclui a concepção de *workflows* científicos.

Muitas tarefas descritas pelos pesquisadores da utilização de vários programas para conclusão de um resultado, e este dado resultante pode ser a coleção de entrada de outro programa. No entanto, a composição destes programas não é uma tarefa simples de ser realizada pelos pesquisadores, dificultando para análises mais complexas, revelando a importância do uso de *workflows* em Bioinformática.

Geralmente os pesquisadores utilizam linguagens de *scripts* para implementar os seus *workflows*, tendo em vista a facilidade que estas linguagens oferecem para fazer chamadas de programas. Entretanto, os *scripts* são muito específicos e difíceis de serem reutilizados e de se fazer manutenção. Por exemplo, o acréscimo e a remoção de programas, o acesso aos dados, o desenvolvimento de analisadores sintáticos, e a configuração de parâmetros geralmente não são tarefas triviais. Sendo assim, torna-se importante a colaboração de profissionais de informática para auxiliar os pesquisadores na definição e execução de seus *workflows*.

Alguns desses gerenciadores oferecem apoio à gerência de dados de proveniência, a parâmetros e aos artefatos que participaram na geração do dado (Shoshani e Rotem, 2009). Dentre este tipo de tecnologias pesquisadas é possível destacar o Kepler e o Taverna, sistemas que possuem mecanismos distintos para gerenciar dados envolvidos nos *workflows.*

Existem alguns motivos por não se usar um sistema gerenciador de *workflow* já existente para se gerar um *workflow* de refinamento, entre eles esta a questão da compatibilidade com outros sistemas. A criação do modelo de *workflow* proposto atende esta questão, que além de cumprir a tarefa de integrar todas as etapas de refinamento de predição da estrutura 3D utilizando o AMBER 14, também preocupou-se em facilitar a integração com os resultados de outros sistemas já existentes. Desta forma, mantém-se o mesmo padrão de desenvolvimento de todas as ferramentas utilizadas no ambiente de pesquisa Laboratório de Bioinformática, Modelagem e Simulação de Biossistemas (LABIO).

Outro motivo por não fazer-se uso de sistemas gerenciadores, deve-se à quantidade de recursos que os mesmos apresentam e que não seriam utilizados, com isso geraria em algum momento uma perda de desempenho no tempo de processamento durante a realização do processo. O modelo de *workflow* desenvolvido contempla o necessário para realizar o processo

de refinamento de predição de estrutura realizada pelo AMBER 14.

3.2 Experimentos científicos

A concepção é a principal etapa de um experimento que tem por objetivo a modelagem e especificação de *workflows* em seus diversos níveis de abstração e no registro de variações do *workflow* no contexto do experimento. Frequentemente, *workflows* são classificados em dois níveis, concreto e abstrato (Deelman *et al.*, 2009). Um *workflow* abstrato é modelado sem estar preso a programas e nem à definição de recursos computacionais, utiliza-se de diagramas de atividades UML *(Unified Modeling Language*) (Pressman 2004) ou outro para modela-los. Um *workflow* concreto é uma instância específica de um *workflow* abstrato para resolver um determinado problema.

Para facilitar a concepção do *workflow*, os sistemas gerenciadores de *workflows* precisariam apoiar a especificação e modelagem de *workflows* nesses diferentes níveis de abstração. Esses sistemas permitem a construção e execução de sequências de tarefas, e em geral disponibilizam uma interface gráfica que abstrai toda ou grande parte da programação que está por trás do fluxo, ajudando o cientista a construir um *workflow* sem precisar entender de programação. Pode-se citar como exemplo, o sistema de *workflow* como o Taverna (Hull *et al*., 2006) e o Kepler (Altintas *et al.*, 2006) que apoiam a modelagem de *workflows* com baixo nível de abstração, muito próximo ao nível concreto de especificação.

3.3 Reutilização de experimentos científicos

A reutilização de experimentos científicos desafia a se tirar o máximo de vantagens dos *workflows* ou experimentos previamente elaborados para compor-se novos *workflows* ou experimentos. Durante o experimento científico podem ocorrer duas situações, o experimento ter que ser executado repetidamente, alterando os dados de entrada e analisando o comportamento do modelo de acordo com essas mudanças, re-executando apenas variando-se os parâmetros (Goderis *et al*., 2008). Na segunda situação, o pesquisador pode não concordar com os resultados alcançados e explorar o uso de programas alternativos, que neste caso carecem de recursos para apoiar a experimentação.

4 SISTEMA GERENCIADOR DE WORKFLOW CIENTÍFICO (SGWfC)

Com o propósito de automatizar e gerenciar a construção e execução dos *workflows* científicos foram desenvolvidas ferramentas computacionais denominadas sistemas de gerência de *workflows* científicos (SGWfC) (Hey *et al*., 2009).

Com o crescente aumento do volume de dados na pesquisa aliada a diversas formas de explorá-los e mantê-los, têm demandado ferramentas que suportem todo o ciclo de pesquisa, fazendo-se uso cada vez mais deste tipo de ferramenta.

A automação de *workflows* pode fornecer as informações necessárias para a reprodutibilidade científica, a derivação, a geração de dados e o compartilhamento de resultados em um ambiente de pesquisa colaborativo (Oinn *et al.,* 2007). Sendo umas das vantagens dos SGWfC possuir toda infraestrutura necessária para executar, definir e monitorar as execuções dos *workflows* científicos tanto local quanto remotamente (Silva, 2011).

Atualmente, existem dezenas de SGWfC disponíveis sendo descrito nos próximos subcapítulos 2 sistemas deste tipo, que possuem grande aceitação na comunidade científica: o Kepler e oTaverna. São sistemas centralizados, que suportam *workflows* científicos do tipo grafo direcionado acíclico (do inglês, DAG - *directed acyclic graph*) e elaborados sob o paradigma do software livre, o que favorece uma contínua melhoria da ferramenta.

4.1 Kepler

O Kepler (Ludascher *et al.*, 2006) é um SGWfC construído como uma extensão do *framework* Ptolemy (Ptolemy, 2015), que promove a construção de *data-flows* baseados na conexão de componentes de software chamados atores. No Ptolemy a semântica da comunicação entre os atores é definida por um componente chamado de diretor, que descreve sob qual modelo de computação o *workflow* será executado. Inicialmente o Kepler (Altintas *et al*., 2004) armazenava a definição dos *workflows* em uma linguagem de modelagem baseada em XML (*Extensible Markup Language*) chamada MoML (*Modeling Markup Language*). A nova versão do sistema incluiu várias funcionalidades de proveniência, principalmente o registro de execuções e definições em banco de dados relacional.

O Kepler planeja o apoio à proveniência de dados através de classes que observam a

execução do *workflow*, armazenando os dados gerados durante a mesma e pode ser configurado para os parâmetros de granularidade e destino do dado. A gravação dos dados de proveniência em banco está habilitada por padrão na distribuição da ferramenta, sendo que o usuário pode optar pelo desligamento dessa funcionalidade a qualquer momento.

A estratégia do Kepler de armazenar os dados intermediários possibilita a re-execução de um *workflow* cujos parâmetros eventualmente tenham sido alterados, ou que tenham obtido falhas em alguns de seus passos, aproveitando os dados de etapas anteriores sem a necessidade de executá-las novamente. A Figura 18 mostra a interface gráfica do Kepler através de um MWC (Modelo de Workflow Científico).

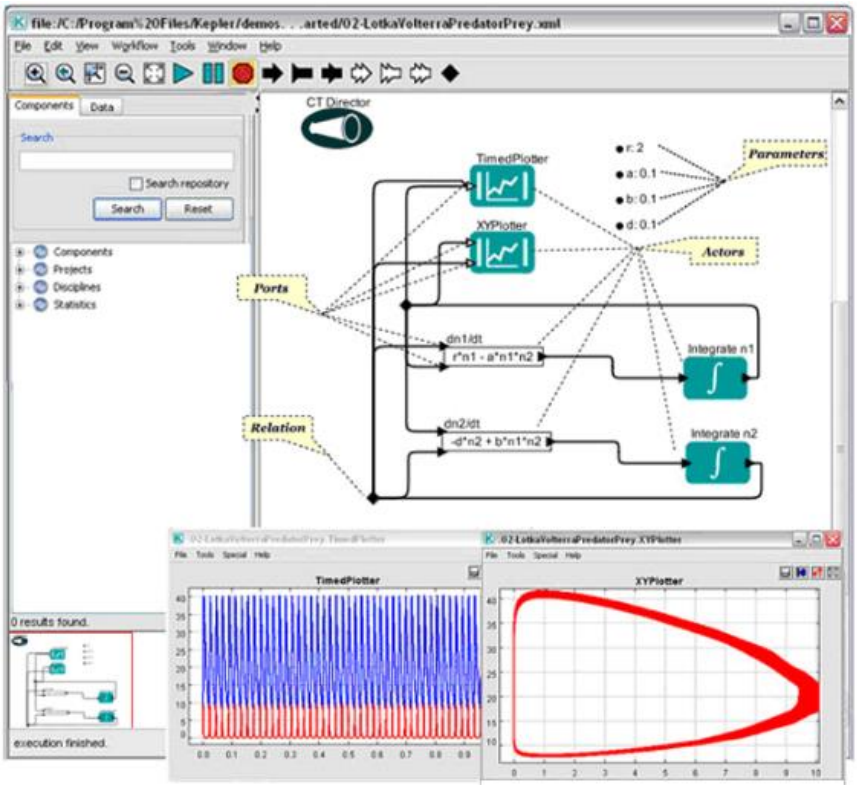

Figura 18 – Exemplo de workflow modelado com o Kepler. Figura extraída do site do Kepler: *https://kepler-project.org/users/sample-workflows/.*

O sistema Kepler simplifica o esforço necessário para modelar e analisar dados científicos, ao usar uma representação gráfica dos seus processos. Essa representação, do MWC, descreve o fluxo de dados entre os componentes da modelagem. Destacam-se a seguir as caraterísticas do SWC Kepler:

- Possui um conceito de modelos computacionais, que descrevem como serão o comportamento da execução de um MWC;
- tem uma interface gráfica amigável, com um conceito de componentes organizados em blocos de construção, onde o usuário move os blocos, ou componentes, para a área de composição de MWCs;
- possui um procedimento de instalação prático;
- é de código aberto escrito na linguagem Java;
- possui uma documentação instrutiva para criação de novos componentes.

Usuários do Kepler com pouco conhecimento de computação científica conseguem criar MWCs com os componentes padrões do sistema, ou ainda modificar os modelos existentes para adaptar às suas necessidades. Outro exemplo de tarefas externas são os componentes do Kepler que permitem o acesso às tecnologias de computação distribuída, que são usadas pelos usuários para compartilhar seus dados e MWCs com outros usuários por meio de repositórios, além da computação de alto desempenho. A utilização desses componentes, diminuindo a complexidade do uso dessas tecnologias, automatizando a comunicação com os servidores Grid[2].

A interface de Grid fornecida pelo Kepler é para o sistema Globus, que é uma comunidade que desenvolve tecnologias para Grid e permite que pessoas compartilhem computação de alto desempenho, banco de dados e outras ferramentas corporativas.

Com o Kepler é possível criar *workflows* hierárquicos, possibilitando que atividades complexas sejam definidas a partir de componentes mais simples. Uma particularidade do Kepler é que a ordem de execução dos atores em um *workflow* é determinada por uma entidade independente, chamada diretor (Ludascher *et al*.,2006). Outra característica interessante é que o sistema pode notificar o usuário sobre alterações no *workflow* de forma assíncrona.

Os MWCs no sistema Kepler não necessitam de compilação para serem executados, com isso permitem que seus usuários examinem e visualizem os dados durantes sua execução. Essa funcionalidade auxilia na modelagem e permite que se altere facilmente os parâmetros do

2 Grid: no contexto computacional trata do compartilhamento de recursos relacionados e soluções de problemas, em organizações virtuais distribuídas geograficamente (Lauschner, 2005).

modelo, produzindo uma maior variedade de resultados experimentais.

4.2 Taverna

Assim como o Kepler, o WfMS (*Workflow Management Systems*) Taverna (Hull *et al*., 2006), conforme mostra a Figura 19, tela inicial, é também um sistema de código aberto que permite a automação de métodos experimentais utilizando serviços de vários domínios científicos utilizando interfaces baseadas em *web-services* (Duncan *et al*., 2006). O Taverna fornece um mecanismo para validação do MWC que é definido na linguagem Scufl (*Simple Conceptual Unified Flow Language)* (Taverna Project, 2007).

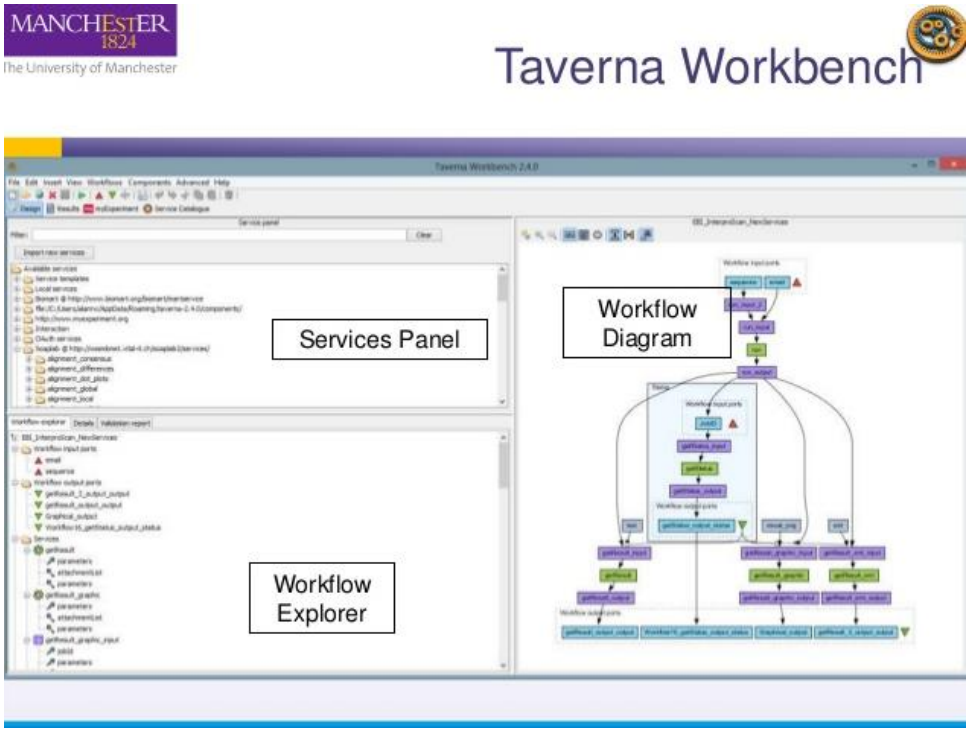

Figura 19 – Tela principal do Taverna. Figura extraída do manual do Taverna: *www.taverna.org.uk/.*

A linguagem *Scufl* permite que o usuário descreva uma tarefa conceitual através de uma entidade simples, além de ter um mecanismo de tratamento de falha básico. O foco desse sistema é disponibilizar para cientistas de bioinformática, com pouco conhecimento em computação, ferramentas para criação de análises complexas de forma que possam ser gerenciadas a partir de um computador pessoal.

O Taverna registra dados de proveniência em banco de dados relacional, porém o banco não foi desenvolvido para que o usuário final consulte a proveniência. A persistência em banco da proveniência é opcional, o usuário pode optar por manter os dados apenas em cache. Neste caso os dados serão apagados ao sair.

Um *workflow*, gerado pelo Taverna, é especificado como um grafo direcionado. Os vértices do grafo (processadores) são os serviços que implementam as atividades. Cada arco conecta um par de entrada-saída e denota dependência de dados entre as atividades. Há também ligações de controle que indica que uma atividade só pode terminar depois da execução de outra.

5 TRABALHOS RELACIONADOS

A criação de *workflows* científicos não é uma tarefa trivial. Existe uma grande dificuldade em definir formalmente os experimentos científicos. Criar *workflows* científicos com serviços *web* disponíveis ainda é uma tarefa complexa, pois os cientistas nem sempre são da área de Ciência da Computação, e por isso, não têm facilidade para utilizar recursos computacionais para definir seus experimentos científicos.

A aplicação de *workflows* científicos em Bioinformática tem recebido atenção crescente na comunidade de pesquisa nos últimos anos. Biólogos estão frequentemente pesquisando como um organismo responde a mudanças no seu ambiente, expressas por meio do comportamento dos seus genes. Os centros de pesquisa, em geral, possuem definidos um processo ou *workflow* para o estudo de sequências genômicas. Atualmente já existe uma grande quantidade de serviços disponíveis o que aumenta a dificuldade para que os cientistas encontrem aqueles que realmente os interessam. Neste cenário, diversos esforços vêm sendo feitos no sentido de automatizar este processo de descoberta de serviços.

As ferramentas de informática surgem com o objetivo de fornecer suporte ao desenvolvimento de *workflows* científicos. Exemplos do uso de *workflow* científico em Bioinformática podem ser vistos nos estudos que seguem.

O trabalho desenvolvido por Machado *et al.*, (2007) busca melhorar o suporte computacional para a realização dos experimentos, criando um *workflow* científico para automatizar o processo de execução de experimentos de docagem molecular, considerando a flexibilidade do receptor, conforme mostra a Figura 20.

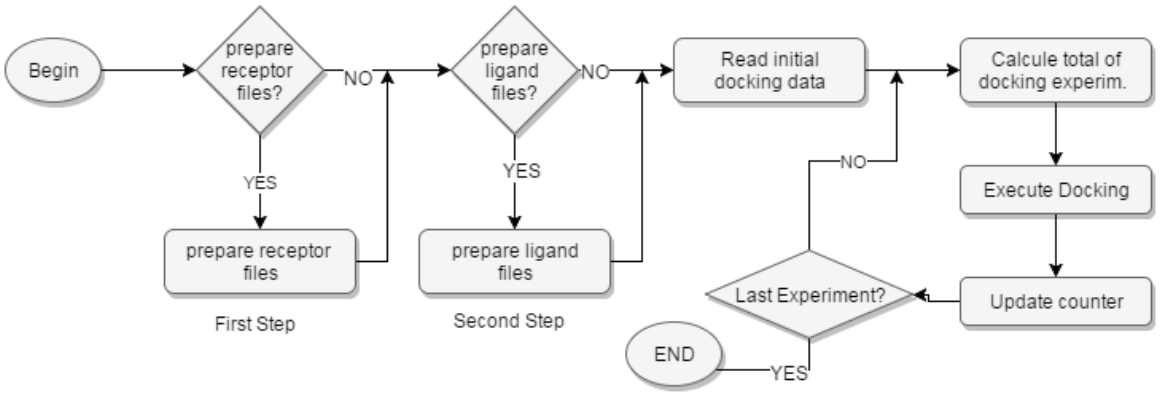

Figura 20 – Processo de CADD com flexibilidade explícita do receptor (Machado *et al.*, 2007).

Com isso, os experimentos puderam ser executados de forma mais automática. Antes, execuções desse tipo no LABIO, eram manuais ou com o auxílio de *scripts* básicos, que necessitavam serem modificados a cada execução de diferentes experimentos receptor-ligante.

O estudo apresentado por Weske *et al., (1995)* trata-se de um *workflow* para modelar o processo de agrupamento de fragmentos de DNA (*Deoxyribonucleic Acid*) utilizado no melhoramento do sequenciamento de genomas, com o objetivo de validar uma ferramenta desenvolvida para a definição e execução de *workflows* científicos

Encontrar e interpretar uma sequência (fragmentos de DNA) de bases de um organismo é uma tarefa importante e fundamental em biologia molecular. Atualmente, pequenas sequências de DNA podem ser geradas semi-automaticamente, utilizando recursos específicos. Porém, o fato de existir muitos erros dentro das sequências, dificultam o estudo sobre estas sequências, gerando com isso resultados de um sequenciamento de bases não totalmente correto.

Uma tarefa dos cientistas é corrigir esses erros e assim produzir um conjunto de dados com um maior nível de qualidade, para isso utilizam-se de alguns procedimentos para executá-las através de *workflows*. O resultado dessa atividade é avaliado por um humano, se a validação ocorrer positiva a análise da sequência pode ser feita. Caso encontre erros, novos fragmentos devem ser gerados e unidos para formar diferentes sequências que se aproximem da sequência a que se quer chegar.

Encontrou-se, na literatura, outras publicações que envolvem *workflow* científico, porém, não dizem qual a abordagem, entre elas cita-se a de Mattoso *et al.*, (2009) cujo artigo aborda os desafios da computação relacionados à modelagem computacional e ao uso de simulação, em especial a necessidade de novas tecnologias para apoiar essa categoria de experimentos. O artigo faz referência a falta de métodos para concepção, em especial, no nível abstrato.

O trabalho desenvolvido por Ludäscher *et al*. (2006), apresenta uma arquitetura com o objetivo de fornecer suporte ao desenvolvimento de *workflows* científicos, onde é usada a tecnologia de *microarrays* de DNA para determinar o nível de expressão de umconjunto de genes. Esses genes são marcados com cores fluorescentes sendo que, quanto maior esse número de pontos fluorescentes, mais alto o nível de expressão do gene, que após é selecionado um subconjunto de genes que seja parecido com os genes que obtiveram um maior nível de expressão. Foi possível modelar o processo de identificação de promotores de genes, gerando

o que chamou de PIW (*Promoter Identification Workflow)*.

Na visão da reutilização das informações de um experimento científico, existem algumas iniciativas como myExperiment (De Roure *et.al*., 2009) e Vistrails (Callahan *et al*., 2006) que permitem o armazenamento e posterior uso de *workflows* científicos por outros pesquisadores ou grupos de pesquisa. Porém, os *workflows* científicos armazenados encontram-se representados em nível concreto, já definidos para um SGWfC e ligados a uma infra-estutura computacional específica, neste caso pode-se necessitar de adaptações e acarretar, assim, problemas no seu uso.

6 RESULTADOS: DESENVOLVIMENTO DO MODELO DE *WORKFLOW*

Neste capítulo é descrito toda a modelagem e implementação do modelo de *workflow* científico desenvolvido, descrevendo detalhadamente cada uma de suas etapas. As pesquisas realizadas para esse trabalho, demonstraram que se utilizando do simulador de dinâmica molecular AMBER 14 e submetendo as etapas de protocolo de refinamento desenvolvido, a partir dos resultados obtidos na predição do método CReF, é possível obter um refinamento das estruturas geradas pelo método CReF, com resultados mais consistentes da predição inicialmente obtida por este método.

O uso de *workflows* científicos, além de fornecer o apoio necessário ao ciclo de execução e análise, torna possível a criação de um ambiente com independência entre as diversas aplicações científicas, projetados para realizar experimentos *in silico* com o intuito de processar e analisar uma grande quantidade de dados usando simulação computacional. Os experimentos são organizados como uma sequência de etapas que caracterizam um fluxo de execução, onde em cada etapa utilizam-se diferentes *softwares* ou processos.

A determinação experimental via difração de raio x é cara e demorada, geralmente ainda com limitações devido a estes fatos, tentar fazer *in silico* é uma grande oportunidade para solucionar este problema. Utilizando-se de um pequeno raio de corte e torcendo para que a trajetória da dinâmica molecular, obtida via cristalografia por difração de raios x, conduza a uma conformação predita *in silico* mais próxima à experimental do polipeptídio.

6.1 Implementação do Modelo de *Workflow*

O modelo descrito é importante para contribuir com o aperfeiçoamento dos resultados do método CReF para processo de refinamento que atualmente são executadas através do AMBER 14, que por ser executado manualmente, não permite realizar diversas vezes os experimentos e análises em um curto espaço de tempo. Sendo assim haverá com o uso do *workflow* um ganho no tempo necessário para a sua execução e obter-se resultados mais consistentes de predição.

O protocolo de refinamento realizado manualmente com a utilização *de shell scripts* ocasiona ao usuário uma grande limitação na quantidade de experimentos que poderá realizar.

Isso ocorre porque a configuração manual do protocolo é muito trabalhosa e envolve peculiaridades no detalhamento da configuração, principalmente em resíduos de aminoácidos do polipeptídio predito pelo CReF que irá receber restrições de movimentação em ângulos diedros. Dessa forma, têm-se problemas para definir a ordem correta em que as etapas deverão ser executadas. É possível executar o processo utilizando parâmetros diferentes, monitorar a execução do mesmo, gerar análises de resultados, o que torna este cenário sujeito a falhas e improdutivo, especialmente em se tratando de experimentos complexos, que envolvem muitos programas e grande quantidade de dados.

Com o objetivo de melhorar este processo, foi criado um modelo de *workflow* científico que automatiza boa parte das configurações do AMBER 14.

Esse modelo resume-se basicamente em cinco etapas:

1. Construção do comando inicial;

2. Criação dos arquivos;

3. Criação do arquivo de restrições;

4. Início da simulação da dinâmica molecular;

5. Análise automática dos resultados obtidos através da trajetória da dinâmica molecular.

A configuração de todos os requisitos necessários para realização do processo de dinâmica molecular fica menos custosa computacionalmente com a utilização de solvente de água implícito no GB (*Generalized Born*). Com isso o tempo necessário para o processamento da dinâmica molecular pelo computador através do AMBER 14 torna-se bem reduzido.

O método utiliza-se de um pequeno raio de corte para que os cálculos computacionais da trajetória da dinâmica molecular tenham um ganho de tempo e conduzam a uma conformação predita *in silico*, mais próxima à experimental do polipeptídio, obtida via NMR ou cristalografia por difração de raios x. Para que se tenha na forma *in silico* um esboço de método candidato a auxiliar paralelamente à determinação experimental de estrutura terciária de polipeptídio, com aperfeiçoamento dos resultados do método CReF.

O objetivo principal do protocolo de refinamento é aproximar as regiões de alças e voltas do polipeptídio predito pelo CReF comparado a conformação de alças e voltas da versão do polipeptídio obtida experimentalmente e depositada no PDB. Este refinamento agora será

realizado através do modelo de *workflow,* tendo suas configurações para esta execução supervisionadas pelo usuário. A Figura 21, mostra de uma forma geral as etapas da automatização do processo para o refinamento utilizado pelo *workflow.*

Figura 21 – Etapas do processo de refinamento. O ciclo inicia com a entrada do polipeptídio, após a leitura dos arquivos de entrada para a preparação da simulação, a criação do arquivo com as restrições, a execução do refinamento e por último os arquivos com a análise dos resultados.

Para integrar cada etapa do *workflow* usou-se a função "*os.system()*", comando da linguagem de programação *Python*. Esta função é utilizada para executar comandos em *shell script*, permitindo, com isso, a criação de todos os comandos de cada etapa ou programas que serão executados, durante a execução do *workflow*. Nos comandos são informados todos os parâmetros necessários para iniciar a execução de cada etapa.

O resultado obtido com o uso de *workflow,* irá auxiliar na análise de dados da trajetória da dinâmica molecular realizada pelo AMBER 14, que teve início com o resultado da predição de estrutura 3D obtido pelo método CReF, se gerou alguma conformação *in silico* mais próxima do polipeptídio determinada experimentalmente, concluindo com a representação do mapa de Ramachandran, juntamente com o cálculo do RMSD para a análise.

6.2 Elicitação de Requisitos

Nessa fase definiu-se os requisitos necessários para modelagem e o conjunto de funcionalidades para o desenvolvimento do modelo de *workflow*. O produto dessa fase é o Documento de Requisitos (Apêndice A).

Durante a etapa de concepção de um *workflow*, é muito importante ter o envolvimento dos usuários, pois eles têm o papel de fornecer os requisitos necessários para o desenvolvimento do *workflow* e, também, de validar o produto gerado na elicitação de requisitos. Afinal, ele possui o conhecimento do domínio e pode determinar como o experimento é representado em termos de atividades, resultados e ferramentas, para tanto, utilizou-se de entrevistas com os usuários.

Para apoiar a tarefa de especificação dos requisitos funcionais, realizou-se discussões entre integrantes do grupo do LABIO. Todos os requisitos especificados com o grupo, passaram por uma análise para avaliar sua viabilidade e consequentemente refinados. Após este processo concluído foi gerado o documento dos requisitos funcionais com a lista das suas funcionalidades em alto nível, para as etapas de refinamento de proteínas e da interface *web* de resultados. Para criação do Documento de Requisitos com os requisitos funcionais foi utilizado o processo de escrita em linguagem natural, para assim garantir que não haja ambiguidade nas informações descritas, garantindo clareza e precisão nos requisitos elicitados.

O documento final, apresenta além dos requisitos funcionais, os requisitos não funcionais, as definições para questões de segurança, desempenho, usabilidade, confiabilidade, padrões e necessidades operacionais.

6.3 Arquitetura do Modelo de *Workflow* Desenvolvido

Neste capítulo descreve-se cada etapa para execução do *workflow*. Para aplicar as funcionalidades do modelo de *workflow* desenvolvido, escolheu-se para estudo de caso a proteína 1GAB, executando-se, assim, todas as etapas que compõem o refinamento desta proteína. Utilizou-se o *software* de simulações moleculares AMBER 14, sendo que o método utilizado para implementação deste *workflow* permite usar as versões 9 até 14.

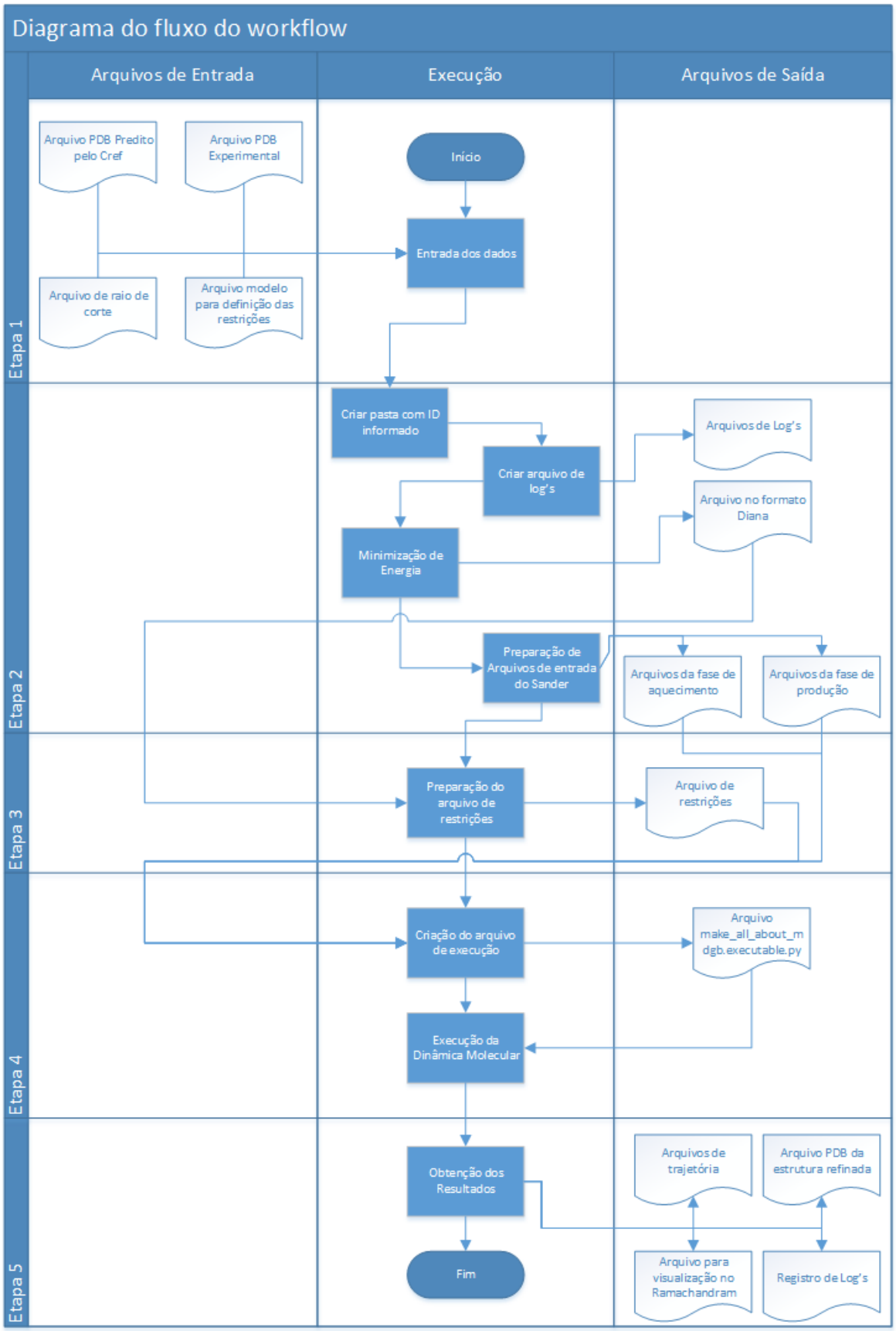

Figura 22 - Diagrama de todas as etapas, suas composições de arquivos que servem de entrada e arquivos gerados como saída.

Devido ao fato de serem mantidos os parâmetros e os padrões de arquivos na versão 14 do AMBER, e necessitando apenas realizar-se algumas verificações para adaptar-se os arquivos gerados pela ferramenta, é o que possibilitou usar o *workflow* nessas versões.

O processo inicia com a estrutura 3D aproximada gerada pelo método CReF. Em uma máquina com sistema operacional Linux, envia-se um comando de *shell script* pelo terminal, onde são informados os parâmetros para iniciar a dinâmica molecular, e o arquivo gerado como resultado do CReF.

Para melhor entendimento da composição de cada etapa do *workflow*, a Figura 22 apresenta um diagrama de todas as etapas, suas composições de arquivos que servem de entrada e arquivos gerados como saída. Dessa maneira fica mais fácil acompanhar a explicação de cada etapa do *workflow* nos capítulos que seguem.

6.4 Detalhes da Implementação: Estudo de Caso

Este capítulo apresenta os detalhes da implementação do modelo de *workflow*. Para demonstrar o processo de refinamento com a aplicação do modelo de *workflow* foi utilizada como estudo de caso a proteína alvo 1GAB, determinada por NMR, composta por 53 resíduos de aminoácidos. O resultado obtido do experimento está descrito no Capítulo 7.1.2.

Para testar a eficiência do modelo de *workflow* desenvolvido foram submetidas outras duas proteínas a 1YWJ composta por 28 resíduos de aminoácidos e 1GPT composta por 47 resíduos de aminoácidos, ambas determinadas por NMR. As três proteínas submetidas ao protocolo de refinamento possuem características diferenciadas entre si, e o resultado obtido nos experimentos estão descritos no Capítulo 7.

6.4.1 Etapa 1: Entrada de Dados

O processo da etapa de entrada de dados é composto pelo arquivo PDB predito pelo CReF, o arquivo PDB experimental do repositório de estruturas de proteínas já resolvidas e o arquivo do raio corte, que servirão para compor a entrada de dados desta etapa, conforme mostra a Figura 23.

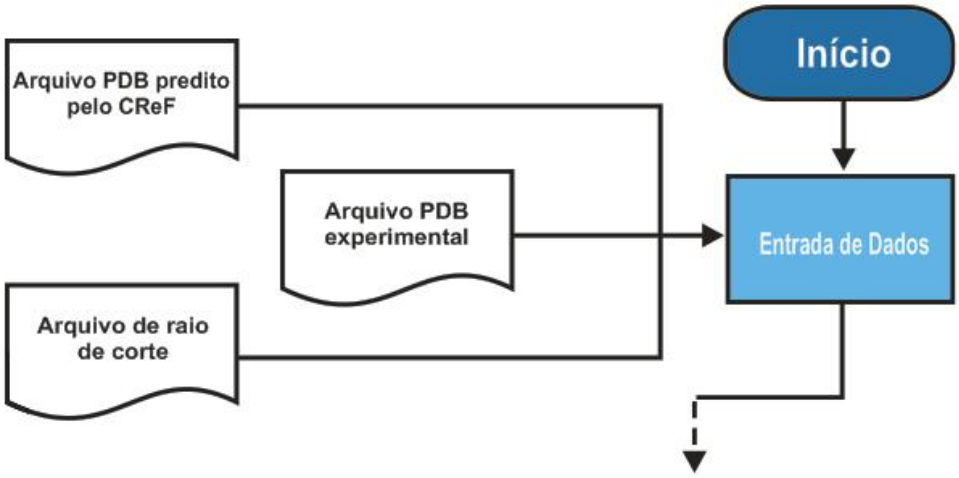

Figura 23 – Estrutura do processo da etapa 1 do refinamento.

Para iniciar o processo de refinamento utilizando-se o modelo de *workflow* com a proteína 1GAB, que servirá como estudo de caso, o usuário precisa executar um comando em *shell script* composto por vários parâmetros. Alguns desses parâmetros informados pelo usuário são obrigatórios. Os demais parâmetros que são opcionais definem o valor padrão para execução do refinamento.

Para compor o *script* de execução que dará início ao *workflow*, no Quadro 1, estão descritos todos os possíveis parâmetros de entrada que poderão ser informados pelo usuário. A coluna da esquerda exibe o comando para execução e a coluna da direita descreve sua funcionalidade, assim como qual o valor padrão estipulado pelo *workflow*, se o mesmo não for utilizado no *script* inicial de execução.

Quadro 1 – Descrição dos parâmetros e sua respectiva funcionalidade.

Comandos	**Descrição**
-id	Informa uma identificação exclusiva para polipeptídio, que será uma sub-pasta dentro da pasta de saída do *workflow*.
-predictedPdb	Informa uma estrutura predita no formato PDB. Este argumento torna-se opcional se uma sequência de estrutura primária polipeptídio se for informado através de um arquivo

	passado pelo argumento " *-proteinSequenceFile*".
-a	Pode ser usado para informar um arquivo contendo códigos (código em formato similar ao do programa DSSP *(Dictionary of Protein Secondary Structure, http://swift.cmbi.ru.nl/gv/dssp/*) de estrutura secundária, para fazer restrições dos ângulos diedros.
-cutRgbmaxFromFile	Requer um arquivo de entrada com informações do raio de corte e *rgbmax* para subestágios da fase de aquecimento e subestágios da fase de produção da dinâmica molecular.
-amberVersion	Versão 14 do AMBER.
-pmemd	Informa que a dinâmica molecular será computada pelo *pmemd* ao invés do Sander
-molprobityThreadsNumber	Número de threads que serão criadas cada uma realizando uma análise simultânea através de chamadas individuais ao programa MOLPROBITY (*http://molprobity.biochem.duke.edu/).*
-np	Número de processadores (número de núcleos a serem usados pelo AMBER 14, para a dinâmica molecular). Caso não informado será utilizado apenas um processador.
-experimentalPdb	Pode ser usado para informar a estrutura experimental em formato de PDB. Se informado, o programa irá gerar arquivos que facilitem análise RMSD. Gerando com isso os arquivos automaticamente, que executam o cpptraj para calcular a RMSD.
-finalTemp	Parâmetro para informar a temperatura final da fase de aquecimento. Caso não informado esse argumento, a temperatura final será a do padrão 325 Kelvin (K).
-tempAddition	Parâmetro para informar a adição de temperatura dos subestágios da fase de aquecimento. Caso não for informado será usado o padrão de 50 Kelvin (K).

-ntt	Parâmetro do AMBER 14 para escala de temperatura
-heatDuration_ps	É possível informar a duração total (em picossegundos) da fase de aquecimento através deste argumento. Caso não seja informado, será adotado o padrão de 100 picossegundos.
-dt	Passo de tempo (picossegundos). Maiores temperaturas significam aumento velocidades e mais distância percorrida entre cada avaliação.
-nscm	Define o nome nscm do AMBER 14 do Sander.
-ntpr	Informa o valor padrão de "500" para " NTPR" que será utilizado na fase de aquecimento e de produção para os resultados que serão impressos a cada passo.
-igb	Sinaliza o uso de solvente implícito *Generalized Born*.
-mdDuration_ps	Informa a duração da dinâmica molecular (em picossegundos) para a fase de produção. Quando não informado, o valor padrão de " 50.000 " picossegundos será usado.
-proteinSequenceFile	Pode ser usado para informar um arquivo de estrutura primária de polipeptídio com a sequência de polipeptídio principal. Este arquivo deve ter uma linha com a sequência primária polipeptídio.
-cut	Usado para especificar o ponto de corte de átomos não ligantes, em *Angstroms*.
-rgbmax	Indica a distância máxima entre pares de átomos

Para dar início à execução do *workflow*, criou-se o comando formado pelos parâmetros conforme mostra a Figura 24.

Para melhor explicar os comandos (destacados em negrito), cada parâmetro utilizado foi separado linha a linha. Nesse *script* inicial de execução, os comandos são preenchidos com os parâmetros de acordo com o teste de refinamento que se escolha realizar, possibilitando com

isso uma ampla variedade de experimentos, tendo em vista a flexibilidade de escolhas dos parâmetros que irão compor o refinamento.

Após concluir o comando com os parâmetros escolhidos, executa-se o mesmo tendo o início com o arquivo programado em *Python*, que receberá todos os parâmetros informados.

```
./mdgb_automation.py
 -id 1GAB_P_F4
-predictedPdb
./INPUT_FILES/1GAB_P_F4_Y28r.cref.predicted.and.backbone.trespassing.aromatic.ring.repaired__OK.pdb
 -experimentalPdb ./INPUT_FILES/1GAB_EXPERIMENTAL0_WITHOUT_OXT.pdb
-a ./INPUT_FILES/secondary_structs_rst.in
-np 4
-cutRgbmaxFromFile ./INPUT_FILES/cutAndRgbmax.in
-mdDuration_ps 50000
```

Figura 24 – Comandos para iniciar o estudo de caso do *workflow*, específico para a proteína 1GAB.

Tem início então a execução do *workflow*, com a preparação dos arquivos que serão utilizados pelo AMBER 14, de forma que é calculado um *snapshot* da dinâmica molecular a cada um picossegundo transcorrido da trajetória. Como mencionado anteriormente, nem todos os parâmetros descritos são obrigatórios. Quando não informados, utilizam-se de valores padrões para a execução.

A seguir, uma breve descrição dos parâmetros utilizados para execução do refinamento escolhido como caso de estudo, os parâmetros estão todos destacados em negrito no quadro a cima.

- **-id:** recebe o valor "1GAB_P_F4" como identificador desta dinâmica.

- **-predictedPdb:** recebe o conteúdo: "./INPUT_FILES/1GAB_P_F4_Y28r.cref.predicted.pdb". Este será um dos arquivos de entrada que exibe o caminho onde encontra-se a estrutura predita pelo método CReF que será refinada pelo *workflow*. O arquivo sempre deverá ser informado no formato PDB, outro formato não é permitido. Aqui alerta-se, que tal arquivo de predição a ser informado (formato PDB), deve ter sua estrutura revisada pelo usuário. A

Figura 25 mostra um dos trechos do arquivo PDB, que neste caso será o da proteína 1GAB utilizado pelo *workflow*.

```
ATOM   2 CA THR  1    2.001    1.311    0.000  1.00  0.00
```

Figura 25 – Trecho do arquivo PDB da proteína 1GAB.

Basicamente serão utilizadas todas as linhas onde estão descritas as coordenadas 3D da proteína. Para tornar melhor compreendido este arquivo, descreve-se o significado de cada coluna do arquivo mostrado na Figura 25.

– **ATOM**: indica que esta linha é composta pelas características do átomo em questão, a coluna que apresenta o número;

– **2**: é o número do átomo;

– **CA**: índica o átomo;

– **THR**: sigla do aminoácido;

– **1**: indica que é o primeiro aminoácido;

– **Destacado em amarelo**: indica as coordenadas 3D para futura montagem da proteína tridimensional.

- **-experimentalPdb**: parâmetro que está recebendo o conteúdo "./INPUT_FILES/1GAB _EXPERIMENTAL.pdb". Este é outro dos arquivos de entrada, exibe o caminho onde encontra-se a estrutura experimental no formato PDB sem o oxigênio terminal, dispondo da mesma composição do arquivo predito pelo CReF apresentado anteriormente. Posteriormente este arquivo servirá para facilitar a análise do RMSD através do Ptraj.

- **-a**: parâmetro que está recebendo o modelo de arquivo informado pelo usuário, que indica onde ocorrerá as restrições nos ângulos de torção. Com o intuito de facilitar o usuário para informar estas restrições, foi definido um arquivo modelo, onde ele basicamente se tiver interesse de restringir uma medida de ângulo especifico, deverá apenas retirar o carácter de comentário (#) da linha onde deseja aplicar a restrição. O objetivo principal deste arquivo é simplificar a inserção das informações, porque numa etapa posterior durante a execução do

workflow será feita a leitura deste arquivo e gerado o arquivo de restrições no formato que o AMBER 14 necessita para sua execução. A Figura 26 apresenta modelo do arquivo que será utilizado para informar as restrições.

```
1.  #DSSPSecStructure: consecutiveMinAmount  consecutiveMax  OMEGAfreedom PHIfreedom PSIfreedom forceConstant_rk2    forceConstant_rk3
2.   H:              1                      ...              0            0           0           2.0                 2.0  #H: alfa-helix
3.   G:              1                      ...              0            0           0           2.0                 2.0  #G: 310-helix
4.   I:              1                      ...              0            0           0           2.0                 2.0  #I: pi-helix
5.  # E:             1                      ...              0            0           0           2.0                 2.0  #E: beta-strand
6.  # B:             1                      ...              0            0           0           2.0                 2.0  #B: beta-bridge
7.  # T:             1                      ...              0            0           0           2.0                 2.0  #T: beta-turn
8.  #  S:            1                      ...              0            0           0           2.0                 2.0  #S: bend
9.  #***Insert secondary structures non selected as comments (#Yes or #No)? 'answer in the next line!!!
10. #Yes
```

Figura 26 – Modelo do arquivo utilizado para informar as restrições.

Como descrito anteriormente é possível perceber que temos algumas linhas do arquivo de restrições onde no início consta o carácter de comentário "#" (linhas 5, 6, 7, e 8). As linhas que apresentam este carácter no início, indica que o comando após não será executado, dessa forma isso é dito que a linha está "comentada".

Quando a linha recebe o carácter de comentário indica que esta estrutura secundária não receberá nenhum tipo de restrição de movimentos, já as estruturas secundárias que não apresentarem este carácter (linhas 2, 3, e 4) no início, receberão restrições de movimento. Para entender melhor este arquivo (Figura 26), cada coluna e sua respectiva função será detalhado no Quadro 2.

Quadro 2 – Descrição dos parâmetros, seu valor e sua respectiva funcionalidade.

Coluna	**Valor**	**Descrição**
***Tag* de comentário**	Não preenchido	Inserção ou não do carácter de comentário para definição da restrição.
DSSPSecStructure:	H:	Estrutura secundária.

consecutiveMinAmount	1	Quantidade mínima a considerar de resíduos consecutivos para a estrutura secundária detectada pelo DSSP.
consecutiveMax	...	Quantidade máxima a considerar de resíduos consecutivos para a estrutura secundária detectada pelo DSSP. Caso não queira restringir a quantidade máxima usa-se "...".
OMEGAfreedom	0	Restrição especifica para o ângulo Omêga do resíduo, com o valor informado o usuário pode definir uma liberdade de movimento para o ângulo do resíduo. Se o usuário informar 50 neste campo, então o ângulo computado automaticamente terá sua restrição "relaxada" em 25 graus para menos e 25 graus para mais.
PHIfreedom	0	Restrição especifica para o angulo *phi* do resíduo, com o valor informado o usuário pode definir uma liberdade de movimento para o ângulo do resíduo. Caso seja informado o valor 80 para este campo, o ângulo computado automaticamente, terá sua restrição "relaxada" em 40º graus para menos e 40º, para mais.
PSIfreedom	0	Restrição especifica para o angulo *psi* do resíduo. Com o valor informado o usuário pode definir uma liberdade de movimento para o ângulo do resíduo. Se o usuário informar, por exemplo, o valor 120 neste campo, então o ângulo computado automaticamente, terá sua restrição "relaxada" em 60º graus para menos e 60º para mais.
forceConstant_rk2	2.0	Constante de força rk2 do arquivo de restrição. O valor informado deve ser na unidade *kcal/mol.*
forceConstant_rk3	2.0	Constante de forca rk3 do arquivo de restrição. O valor informado deve ser na unidade *kcal/mol*.

Como última opção para o usuário deste arquivo modelo para informação das restrições,

ele possui a alternativa de geração do arquivo de restrições no formato esperado pelo AMBER 14 só com as estruturas restritas ou com todas, para isso, no final do arquivo escolhe-se a opção "#Yes" ou "#No" para a pergunta ***"#******Insert secondary structures non selected as comments (#Yes or #No)? 'answer in the next line!!!"*. Esse é o único caso em que o carácter "#" não serve como comentário. Se a opção escolhida foi com todas as estruturas, as que se encontram comentadas neste arquivo também estarão comentadas no arquivo gerado pelo *workflow*, possibilitando assim ao usuário, analisar o arquivo e realizar algumas alterações caso queira, antes de executar o refinamento.

- **-np:** o parâmetro está recebendo o valor 4 indicando o número de processadores que será utilizado para o processo de refinamento da proteína 1GAB.

- **-cutRgbmaxFromFile**: o parâmetro está recebendo o conteúdo "./INPUT_FILES/ cutAndRgbmax.in", indicando o caminho onde encontra-se o arquivo "cuAndRgbmax.ini". Este arquivo tem como objetivo, facilitar a interação do usuário com o *workflow*. Para isso utiliza-se o exemplo apresentado na Figura 27, onde o usuário tem a opção de alterar conforme o valor escolhido para o cut (raio de corte) e o rgbmax.

```
# MD sub stage                                          cut      rgbmax
heat                                                   8.06       10.0
heat  cut rgbmax for misinformed stage                 8.06       10.0
production                                             8.06       10.0
10.0production__cut_rgbmax_for_misinformed_stage       8.06       10.0
```

Figura 27 – Modelo de arquivo para facilitar a interação do usuário com o *workflow* para alterar o valor das opções cut (raio de corte) e rgbmax.

- Tem-se apenas dois parâmetros para a fase de aquecimento (amarelo) e dois para a de produção (verde), sendo que em cada um deles é informado o cut e o rgbmax. Para a fase de aquecimento tem-se o parâmetro *"heat"*, que indica o cut e o rgbmax para o primeiro arquivo, já o *"heat__cut_rgbmax_for_misinformed_stage"* indica o cut e o rgbmax para arquivos restantes. Para a fase de produção aplica-se o parâmetro *"production"* e *"production__cut_rgbmax_for_misinformed_stage"*, com a mesma finalidade da fase

anterior. Neste experimento está sendo usado nos parâmetros cut e rgbmax os valores 8,06 e 10,0 para todos os parâmetros.

- -mdDuration_ps - o parâmetro está recebendo o valor 50.000 que indica o tempo de duração da dinâmica (em picosegundos) para a fase de produção.

Após a explicação detalhada de todos os parâmetros utilizados como entrada para o processo de refinamento da proteína 1GAB, tem-se o conhecimento necessário para se dar início à execução do *workflow*, passando para a Etapa 2 do processo.

6.4.2 Etapa 2: Leitura dos arquivos de entrada e preparação do ambiente

Tem início a etapa de leitura dos arquivos de entrada e preparação do ambiente, conforme ilustra a Figura 28.

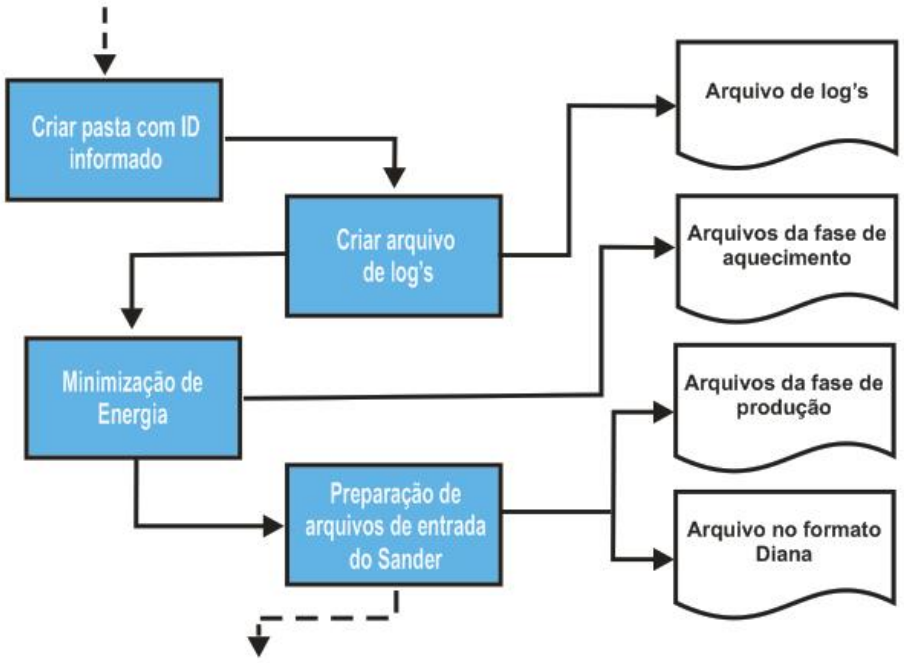

Figura 28 – Estrutura do processo da etapa 2 do refinamento.

O primeiro passo é a criação da pasta com o nome da dinâmica molecular, gerado pelo *workflow* com o identificador informado pelo usuário onde irão conter além dos resultados após o termino da dinâmica, todos os arquivos de registros de *log's* de cada etapa realizada, a pasta

com os arquivos de minimização de energia, a pasta com os arquivos de entrada do Sander e uma pasta com os arquivos de restrições. Após, é realizada a leitura da conformação polipeptídica terciária 3D predita pelo CReF para submissão à dinâmica de refinamento através do Sander do AMBER 14, arquivo informado no parâmetro de entrada "–predictedPdb". Para iniciar a etapa de refinamento é necessário realizar a minimização de energia sobre cada átomo, que servirá como estrutura de partida para o início das simulações de dinâmica molecular. Na sequência, conforme mostra a Figura 29, é explicado cada fase desta etapa.

6.4.2.1 Criação dos arquivos de minimização de energia

A minimização de energia é um dos métodos mais simples para otimizar uma estrutura, sendo utilizada como ferramenta na preparação da estrutura inicial para a dinâmica molecular. Tem como objetivo encontrar o mínimo local de uma dada estrutura.

Antes de começar a dinâmica molecular precisa-se realizar a minimização da nossa estrutura para obter-se a estrutura de partida, preparando as posições de hidrogênio de modo que a dinâmica molecular, quando iniciada, seja estável. Segue na Figura 29 os parâmetros de entrada para a minimização de energia:

```
1. # Energy minimization for molecule 1GAB_P_F4.
2. &cntrl
3. imin=1,    maxcyc=100,  ncyc=10,
4. ntpr=10,
5. cut=16., rgbmax=16.,igb=1, ntb=0,
 /
```

Figura 29 – Parâmetros de entrada para a minimização de energia.

Os parâmetros utilizados nos arquivos de entrada para o Sander estão destacados em negrito. Inicialmente é colocado na parte superior de cada arquivo da fase de minimização um comentário, sinalizado pelo carácter "#", indicando o id da proteína que foi definida pelo usuário (linha 1). Essa minimização de energia não tem interação com o usuário, portanto não existe a possibilidade de alterar-se qualquer parâmetro no momento do *script* de início do

refinamento. Caso queira fazer qualquer alteração o usuário terá que abrir o arquivo gerado e modificar os parâmetros que escolher.

O parâmetro **"***cut"* especificado em *Angström*, indica o raio de corte para as interações não ligantes. O parâmetro *"rgbmax"* indica a distância máxima entre pares de átomos. O parâmetro *"igb"* define o modelo de solvatação implícita do modelo *Generalized Born* e, por último, *"ntb***"**, que define um sistema molecular sem condições periódicas de contorno (todos na linha 5). Para esta etapa se não forem modificados os parâmetros dentro do código-fonte do *workflow* ou após sua criação dentro da pasta onde é salvo, será gerado o arquivo de minimização neste padrão (Figura 29). A minimização de energia só pode ser realizada pelo Sander e não existe nenhuma possibilidade de realizá-la com o PMEMD (*Particle Mesh Ewald Molecular Dynamics*, parte da plataforma AMBER 14), sem que o código-fonte do *workflow* seja alterado.

Após a conclusão da criação do arquivo de minimização de energia, inicia-se a criação dos arquivos de configurações do AMBER 14, e também a preparação da execução do Sander. A configuração é realizada em duas etapas, sendo que na primeira são criadas as entradas necessárias para a fase de aquecimento da dinâmica molecular e a segunda, as entradas para a fase de produção. Estas duas etapas fazem parte da configuração do Sander para sua execução.

6.4.2.2 Criação dos arquivos da fase de aquecimento

É nesta fase que acontece o preparo do arquivo que servirá de entrada para o Sander, no momento da execução da dinâmica molecular. Nesta fase a proteína alvo será aquecida gradativamente até a temperatura final estipulada para realização da dinâmica molecular. A Figura 30, mostra o segundo arquivo da fase de aquecimento gerado para o refinamento da proteína 1GAB.

São apresentados todos os parâmetros de entrada necessários para compor o arquivo de configuração de extensão ".in" do Sander para a realização da fase de aquecimento.

```
1. #Heating of 1GAB_P_F4 from 50.000000 Kelvin to 100.000000 Kelvin from 15 ps
   to 30 ps.
2. &cntrl
3. imin=0, irest=0, ntx=1,
4. nmropt=1,
5. nstlim=7500, dt=0.002000, nscm=1000,
```

```
6.  ntc=2, ntf=2,
7.  ntt=1,
8.  tempi=50.000000, temp0=100.000000,
9.  ntpr=500, ntwx=500,
10. ntb=0, igb=1,
11. cut=8.06, rgbmax=10.0
12.   /
```

Figura 30 – Arquivo da fase de aquecimento gerado para o refinamento da 1GAB.

Caso esses parâmetros não sejam informados no comando que inicia o *workflow*, serão atribuídos os valores padrões. Todos os parâmetros apresentados na Figura 30 podem ter seus valores alterados. A primeira linha está com o carácter de comentário (#) e exibe as informações resumidas sobre o arquivo para facilitar o entendimento do usuário. A palavra "*heating*" é o indicador de que o arquivo é da fase de aquecimento, seguido tem-se o id da proteína informado pelo usuário (nesse exemplo, é "1GAB_P_F4"). Após é informado o intervalo de aquecimento deste arquivo, que nesse experimento é de 50 K até 100 K, e por último o intervalo de duração desse subestágio de aquecimento em picossegundos, que para esse exemplo é de 15ps a 30ps. A primeira linha resume todas as ações desse arquivo.

Após as informações definidas (linha 1) seguem todos os parâmetros que compõem o arquivo da fase de aquecimento (linhas 2 a 12). Alguns desses parâmetros foram descritos conforme Quadro 1, no Capitulo.5.4.1.

Para realizar a montagem do arquivo, o *workflow* executa três cálculos. O primeiro cálculo serve para identificar quantos arquivos da fase de aquecimento serão gerados, para isso usa-se o valor padrão de 50 K ou outro valor que o usuário informa através do parâmetro "*–tempAddition*". Para este estudo de caso especificamente, a cada intervalo de aquecimento a temperatura é elevada em 50 K, no exemplo do arquivo da Figura 30, mostra que a temperatura está subindo de 50 K para 100 K. Com base na definição deste intervalo usa-se o valor da temperatura final, como este valor não foi informado no comando que inicia a execução do *workflow*, fica então definido o valor padrão de 325 K de temperatura final. Pode-se ainda definir outro valor de temperatura final através do parâmetro *"–finalTemp*".

Com estas duas informações é executada uma operação de divisão entre a temperatura final e o valor do intervalo de aquecimento, obtendo-se como resultado o valor 6.5, indicando

que se têm seis arquivos com temperatura sendo adicionado 50 K a cada arquivo de aquecimento, e o arquivo final sendo aquecido 25 K para alcançar os 325 K de temperatura final.

O segundo cálculo a ser executado é para identificar qual intervalo de picossegundos terá cada arquivo gerado. São utilizados dois parâmetros do arquivo da fase de aquecimento (linha 5), o *"nstlim",* indicando a quantidade de passos de simulação (valor de "7500"), e o *"dt"*, que indica o tempo de integração ou passo de tempo (valor de 0.002ps). O resultado da multiplicação entre esses dois valores é de 15ps que indica quantos picossegundos terá o subestágio para o arquivo gerado.

O terceiro cálculo é executado com base no valor informado para o aumento de temperatura de cada subestágio ("*–tempAddition*"), que nesse caso é de 50 K. O valor atribuído ao parâmetro "*tempi"* (temperatura inicial) do subestágio é de 50 K. Como foi definido o valor de 50 K para o aumento de temperatura de cada subestágio, o valor final deste subestágio será de 100 K atribuído ao parâmetro *"temp0*" (linha 8).

Por último (linha 11), têm-se os parâmetros de *"cut",* este indicando que apenas irão ser consideradas as interações não ligantes que estejam dentro de um raio de 8.06 *Angströms* e o parâmetro *"rgbmax"* que adota a distância máxima entre pares de átomos. Esses valores são extraídos do arquivo *"cutAndRgbmax.in*", informado no comando que dá início ao *workflow* já descrito no Capítulo 5.4.1, conforme mostra a Figura 27.

Existem outros parâmetros que também devem ser analisados como o "*ntt"* (linha 7) que indica a escala de temperatura e tempo de relaxamento. Os parâmetros *"ntb"* que indica o método de malha de partículas PME (*Particle Mesh Ewald*)[3] de eletrostáticas infinitas, de valor zero porque não está sendo aplicado, e o *"igb"*, de valor igual a "1" indicando que será aplicado o mesmo raio padrão no *Leap* (linha 10), e por último o parâmetro *"irest"* (linha 3) de valor 0 (zero) para os subestágios de aquecimento e 1 para os de produção.

6.4.2.3 Criação dos arquivos da fase de produção

Como na fase anterior, nesta fase acontece o preparo dos arquivos que também servirão

3 Método que divide os somatórios infinitos usados em Coulomb em duas partes (Case *et al.*, 2005).

de entrada para o Sander, dando assim a continuidade na preparação dos arquivos para realização da dinâmica molecular. Aqui a proteína alvo não terá variação de temperatura, sendo mantida a temperatura final da fase de aquecimento para realizar a dinâmica molecular e a trajetória da dinâmica molecular. Para o tempo total de duração será assumido o valor de 50.000ps, caso o parâmetro *"-mdDuration_ps"* não tenha sido informado. A Figura 31, mostra o primeiro arquivo da fase de produção gerado para o refinamento da 1GAB.

```
1.  # Production phase of 1GAB_P_F4 at 325.000000 Kelvin from 100 ps to 5000
2.  &cntrl
3.  imin=0, irest=1, ntx=5,
4.  nmropt=1,
5.  nstlim=2450000, dt=0.002000, nscm=1000,
6.  ntc=2, ntf=2,
7.  ntt=1,
8.  tempi=325.000000, temp0=325.000000,
9.  ntpr=500, ntwx=500,
10. ntb=0, igb=1,
11. cut=8.06, rgbmax=10.0
12. /
```

Figura 31 – Arquivo da fase de produção gerado para o refinamento da 1GAB.

Para realizar a fase de produção é preciso informar os parâmetros de entrada necessários para compor o arquivo de extensão *"in"* do Sander, conforme apresenta a Figura 31. O usuário pode alterar esses parâmetros de acordo com a necessidade. Este arquivo será usado para gerar os arquivos de extensão ".crd" da trajetória da dinâmica molecular. Caso esses parâmetros não sejam informados no comando que dá início ao *workflow*, serão atribuídos os valores padrões.

O arquivo de configuração apresenta a primeira linha comentada (#) onde escreve um breve resumo sobre o arquivo para ajudar o usuário. A palavra *"Production"* sinaliza que o arquivo faz parte da fase de produção, seguido pelo "id" da proteína informado pelo usuário (1GAB_P_F4), após é informa-se a temperatura que será mantida para a fase de produção (325 K) e por último o intervalo de picossegundos desse subestágio (de 100ps a 5000ps), da trajetória da dinâmica molecular. Portanto nesta linha está resumida todas as ações deste arquivo.

Na linha 8 está informada a temperatura inicial (*"tempi"*) e temperatura final (*"temp0"*) são iguais a 325 K, ou seja, não há variação de temperatura. Uma diferença do arquivo da fase

de aquecimento é o parâmetro *"irest"* com o valor 1, que identifica que este arquivo é da fase de produção.

Para realizar a montagem do arquivo, o *workflow* executa alguns cálculos. O primeiro cálculo é para saber qual intervalo de picossegundos terá cada arquivo gerado. Para obter esta informação são utilizados dois parâmetros do arquivo da fase de produção, o *"nstlim"* com valor de 2450000, indicando a quantidade de passos de simulação e o *"dt"* que indica o tempo de integração com o valor atribuído de 0,002 (linha 5).

Com estes dois valores é executado uma operação de multiplicação para obter-se o resultado de quantos picossegundos terá um subestágio da fase de produção. Para os valores atribuídos ao *"nstlim"* e "*dt"* anteriormente, tem-se como resultado desta operação, 4900ps para o primeiro arquivo que sempre iniciará em 100ps (linha 1). Os restantes dos arquivos terão o intervalo de 5000ps para os próximos arquivos de subestágios gerados.

O segundo cálculo realizado é para identificar quantos arquivos da fase de produção serão gerados. Para efetuar este cálculo utiliza-se o tempo de duração da dinâmica molecular *("–mdDuration_ps"),* como não foi informado fica definido o valor padrão de 50000ps. Este valor será dividido pelo tempo total de duração de cada fase que será de 5000ps tendo como resultado 10 arquivos para a fase de produção.

Diferente da fase de aquecimento onde temos um cálculo para os paramentos *"tempi"* e *"temp0"*, na fase de produção esses parâmetros irão manter uma temperatura constante de 325 K, temperatura está definida pelo usuário para realização da dinâmica molecular.

Por último, temos os parâmetros de *"cut"*, indicando que apenas serão consideradas as interações não ligantes dentro de um raio de 8.06 *Angströms* e o parâmetro "*rgbmax"* que adota a distância máxima entre pares de átomos (linha 11). Esses valores são preenchidos com base no arquivo *"cutAndRgbmax.in*" informado no comando que dá início ao *workflow*.

6.4.3 Etapa 3: Criação do arquivo de restrições

Tem início a etapa da criação do arquivo que corresponde às restrições conforme ilustra a Figura 32 o processo desta fase. Este é o arquivo mais importante gerado pelo *workflow,* onde constam todas as restrições dos movimentos de resíduos dos ângulos diedros *Phy*, *Psy* e *Omega*, deixando as alças/voltas, hélices-alfas e folhas-betas, restritas para a realização da dinâmica

molecular.

Figura 32 – Processo da etapa da criação dos arquivos que correspondem às restrições.

O processo de geração do arquivo de restrições começa a partir do arquivo *"secondary_structs_rst.in"*, conforme mostra a Figura 33, informado no comando em *shell scripts* que dá início ao *workflow*. Todas as restrições de ângulos que foram aplicadas neste arquivo foram inseridas na geração do novo arquivo de restrições agora com um formato esperado pelo AMBER 14. Para criar-se manualmente um arquivo com as restrições no formato entendível pelo AMBER 14 é muito trabalhoso e complexo, criou-se um arquivo modelo de inserção de restrições de ângulos, conforme mostra a Figura 33. Com isso, torna mais fácil para o usuário, identificar onde serão aplicadas as restrições nos ângulos.

Na execução da Etapa 3, este arquivo modelo é lido e aplicada as restrições para realização da dinâmica molecular e após gerado um novo arquivo entendível pelo AMBER 14.

```
1.  #DSSPSecStructure: consecutiveMinAmount  consecutiveMax  OMEGAfreedom PHIfreedom PSIfreedom forceConstant_rk2    forceConstant_rk3
2.  H:                 1                                     0            0          0          2.0                 2.0  #H: alfa-helix
3.  G:                 1                                     0            0          0          2.0                 2.0  #G: 310-helix
4.  I:                 1                                     0            0          0          2.0                 2.0  #I: pi-helix
5.  # E:               1                ...                  0            0          0          2.0                 2.0  #E: beta-strand
6.  # B:               1                ...                  0            0          0          2.0                 2.0  #B: beta-bridge
7.  # T:               1                ...                  0            0          0          2.0                 2.0  #T: beta-turn
8.  # S:               1                ...                  0            0          0          2.0                 2.0  #S: bend
9.  #***Insert secondary structures non selected as comments (#Yes or #No)? 'answer in the next line!!!
10. #Yes
```

Figura 33 – Arquivo modelo utilizado para informar as restrições (secondary_structs_rst.in"), transcrito do Capítulo 5.4.1.

Como mostra a Figura 33, do arquivo *"secondary_structs_rst.in",* as duas últimas linhas destacadas em vermelho (linhas 9 e 8) disponibiliza a opção de resposta para geração do arquivo completo ou não de restrições ("#Yes" ou "#No").

Caso a resposta seja "#Yes", a geração do arquivo será da forma completa. Este arquivo irá conter todas as restrições de ângulos diedros que serão aplicadas ao refinamento (quadro verde) e as restrições que não serão (quadro azul) da Figura 33. As restrições que não serão aplicadas terão o carácter de comentário ("#") adicionado no início de cada linha das restrições indicando que a mesma não será aplicada ao refinamento. Com isso, o usuário tem a opção de editar e aplicar alguma restrição que no início do processo de refinamento não seria aplicada. Neste estudo de caso será gerado o arquivo completo de restrições, por consequência da opção de resposta do usuário ter sido *"#Yes"* (linha 10).

As Figuras 34 e 35 mostram as duas opções geradas do arquivo completo, com e sem a aplicação de restrições nos ângulos diedros.

```
1. # H (alfa-helix)
2. # 3 ASP OMEGA:  (2 ILE CA)-(2 ILE C)-(3 ASP N)-(3 ASP CA) 165.6 165.6
3. # Sequence number = 3, residue name = D, dihedral = OMEGA
4.  &rst     iat =     17,     32,     34,     36
5.       r1 = 164.600000, r2 = 165.600000, r3 = 165.600000, r4 = 166.600000
6.       rk2 = 2.000000, rk3 = 2.000000
&end
```

Figura 34 – Trecho extraído do arquivo de restrições.

A Figura 34, representa uma restrição de ângulo diedro que será aplicada ao refinamento.

```
1. ## T (beta-turn)
2. ## 9 ASN OMEGA:  (8 LYS CA)-(8 LYS C)-(9 ASN N)-(9 ASN CA) 172.6 172.6
3. # Sequence number = 9, residue name = N, dihedral = OMEGA
4. # &rst     iat =    127,    145,    147,    149
5. #     r1 = 171.600000, r2 = 172.600000, r3 = 172.600000, r4 = 173.600000
6. #     rk2 = 2.000000, rk3 = 2.000000
&end
```

Figura 35 – Trecho extraído do arquivo de restrições.

Já a Figura 35, mostra uma restrição de ângulo diedro que não será aplicada ao refinamento.

De acordo com o que foi apresentado na Figura 33, observa-se:

- **Aplicada a restrição -** onde aplicou-se as restrições de ângulos retirando-se o carácter de comentário (#) das linhas que contém "helix" (linhas 2, 3 e 4) destacadas em verde, foi gerada a restrição dos ângulos sem comentários como mostra a Figura 34 (linhas 4, 5 e 6) destacadas em verde, para serem aplicadas ao refinamento.

- **Não aplicada a restrição** - as linhas que contém "beta" e "bend" (linhas 5, 6, 7 e 8) destacadas em azul, onde manteve-se os comentários, também foram inseridas as restrições no arquivo completo de restrição, mas foi incluído o carácter de comentário (#) no início de cada restrição. Como mostra a Figura 35 foi gerada a restrição dos ângulos com comentários (linhas 4, 5 e 6), destacados em azul indicando que não serão aplicados.

Conhecendo a origem de onde foi extraída as informações que gerou o arquivo de restrições, será explicado cada linha de uma restrição utilizando-se a Figura 34.

A linha 1 com a informação "# H (alfa-helix)", no arquivo de restrição com formato reconhecido pelo AMBER 14, é possível o usuário identificar rapidamente onde está sendo aplicada a restrição. Na linha 2 é adicionada a informação "# 3 ASP OMEGA:" que serve para indicar em qual dos ângulos diedros será aplicado a restrição, e o restante dos dados constantes nesta linha são originados do arquivo PDB, informados no comando que dá início ao *workflow* após a minimização de energia. A linha 3 informa o número de sequência do resíduo, o nome e o ângulo para aplicação da restrição.

Com os resultados extraídos da minimização de energia o *workflow* gera um arquivo no formato Diana. A Figura 36, mostra uma linha extraída desse arquivo.

```
3        ASP        OMEGA        165.641200        165.641200
```

Figura 36 – Trecho extraído do arquivo gerado da minimização de energia.

A primeira informação indica o número do resíduo (“3”); a sigla ASP representa o nome do aminoácido, composto por três letras; a palavra OMEGA indica o ângulo diedro e os dois últimos dados com os números “165.641200” indica valores dos ângulos diedros. Os valores dos ângulos do arquivo Diana são parte das informações que compõe o arquivo de restrição destacadas em amarelo na Figura 36 e Figura 34 (linha 2).

E por último, apresenta a composição das restrições (linhas 4, 5 e 6). Para gerar os ângulos de restrições (r1, r2, r3 e r4) é utilizado um módulo do AMBER 14 denominado “Make_ANG_RST”. Este módulo recebe o arquivo Diana para gerar as restrições de cada ângulo. Como resultado obtêm-se todos os ângulos de r1, r2, r3 e r4. Para os ângulos r1 e r4 está sendo aplicada uma flexibilidade para as torções, isso será sempre executado em todos os ângulos como mostra a Figura 37.

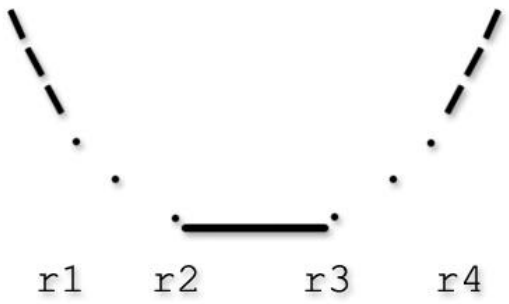

Figura 37 – Nesta figura as barras “\” e “/” representam os diedros inferiores, o “.” refere-se à flexibilidade do ângulo diedro e o “_” corresponde ao ângulo em que se aplica a restrição.

Com este arquivo completo de restrições de ângulos diedros é possível estabelecer as restrições da seguinte forma. O programa DSSP (*Database of Secondary Struture in Proteins*) identifica as conformações como hélices-alfas ou fitas-betas detectadas pelo CReF. São então adicionadas as restrições nessas hélices-alfas e fitas-betas para que durante a dinâmica molecular não ocorram alterações em suas conformações, como por exemplo, deixar de ter aspecto e conformação de hélice, pois caso isso ocorra, mudaria completamente a topologia da proteína, mas cada resíduo pode mudar suas coordenadas *x, y e z* no espaço.

6.4.4 Etapa 4: Execução do refinamento

Após concluir o processo de criação de todos os arquivos realizados nas etapas 2 e 3, é gerado pelo *workflow* o arquivo “*make_all_about_mdgb.executable.py”* que dará início à

dinâmica molecular. A Figura 38 ilustra o processo dessa etapa.

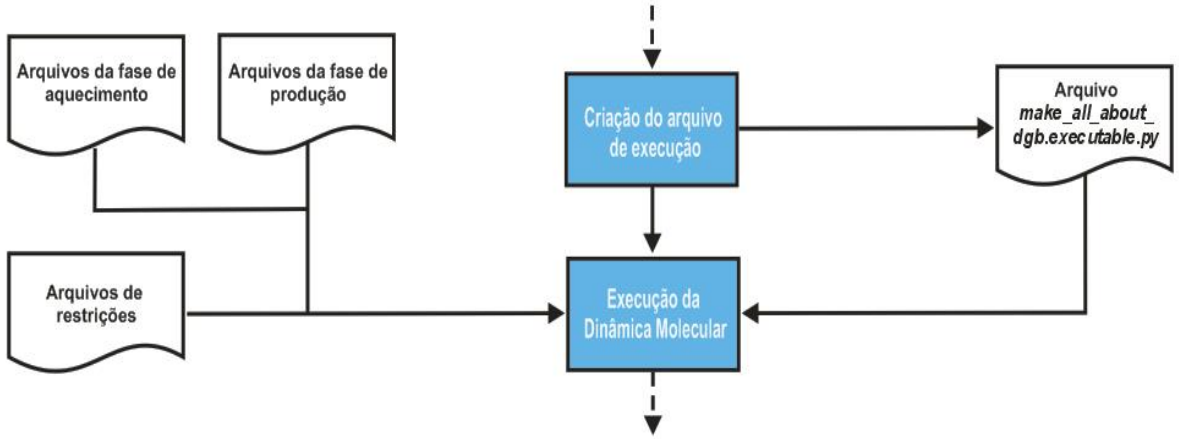

Figura 38 – Processo de execução do refinamento da etapa 4.

Esse arquivo automatizará todas as execuções de *shell scripts* do processo de refinamento utilizando os arquivos gerados nas etapas anteriores. A execução de todo o processo de refinamento agora é executada com apenas a chamada em uma única linha deste arquivo gerado, tornando o processo mais rápido e diminuindo a probabilidade de erros gerados pelo usuário.

O arquivo "*make_all_about_mdgb.executable.py*" é executado pelo usuário permitindo que o mesmo tenha a possibilidade de editar qualquer arquivo gerado pelo *workflow* antes do processo de execução do refinamento, como por exemplo, o arquivo de restrições.

O arquivo de execução é composto por todos comandos em *shell scripts* necessários para a execução da dinâmica molecular. A Figura 39, apresenta-se uma linha desse arquivo para exemplificar uma das chamadas que foi automatizada pelo *workflow*. Neste caso, está sendo executado um dos arquivos gerados na fase de produção.

```
mpirun -np %d sander.MPI -O -i "%s/mdgb00017_production_phase_45000ps_to_50000ps.in
```

Figura 39 – Linha de comando em *shell script* extraída do arquivo de execução *"make_all_about_mdgb.executable.py".*

Para execução dos arquivos é utilizado o *shell script* que inicia com o comando "*mpirun*". Este comando determina qual o tipo de máquina e a quantidade de núcleos de processadores que serão utilizados para execução do arquivo. Após é informado no parâmetro *"-np",* que inicia o *workflow*, o número de núcleos a serem utilizados pela dinâmica molecular.

Por último, segue a execução do MPI do Sander com a chamada de um dos arquivos da fase de produção através do parâmetro *"-i"*.

A conformação da trajetória da dinâmica molecular com a RMSD mais aproximada à experimental é extraída automaticamente através do *workflow* de refinamento com a execução do arquivo "*make_all_about_mdgb.executable.py"*.

6.4.5 Etapa 5: Arquivos de análise de resultados

Tendo concluído o processo da dinâmica molecular (etapa 4) executada pelo Sander do AMBER 14, é possível obter-se vários resultados para posterior análise. A Figura 40 mostra o processo da etapa 5.

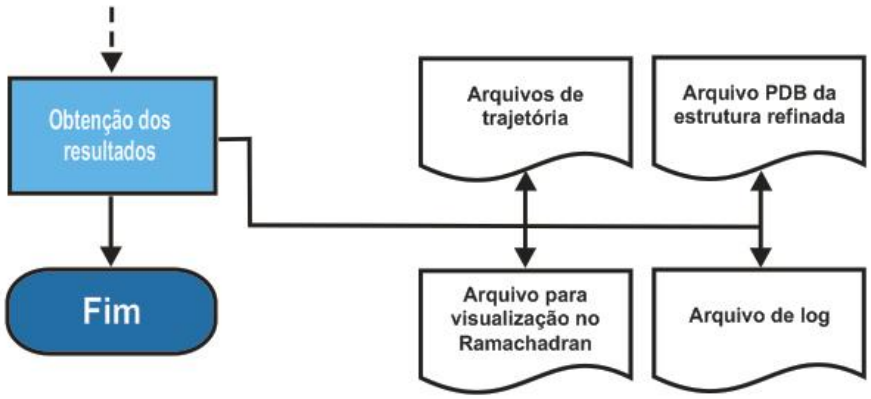

Figura 40 – Processo de execução do refinamento da etapa 5.

Para facilitar a visualização desses resultados, desenvolveu-se uma interface *web*, separando cada conjunto de arquivos, descrita no Capítulo 6.

Neste capítulo apresentam-se todos os arquivos de resultados, obtidos após a proteína alvo ser submetida ao protocolo de refinamento do *workflow,* e as formas de análise disponíveis para usuário. Os resultados deste estudo de caso serão apresentados no Capítulo 7.1.2, sendo que neste será apresentado apenas os diferentes tipos de resultados para cada experimento submetido ao protocolo de refinamento.

6.4.5.1 Gráfico de Ramachandran

Um dos arquivos de resultados gerado, por exemplo, o arquivo no formato PDB refinado

da proteína alvo, contém o resultado do cálculo da melhor RMSD (desvio médio quadrático) em relação a conformação experimental e a trajetória interna da dinâmica molecular (cada frame da trajetória). Isto é, o frame da dinâmica molecular que obtiver menor RMSD em relação a todas as conformações experimentais oriundas do arquivo PDB é extraído da trajetória concatenando com a estrutura experimental e a predita do CReF, que deu início a dinâmica molecular em um único arquivo PDB, e este é enviado para o programa PROCHECK para geração dos gráficos de Ramachandran.

Com isso tem-se como resultado os dados indicativos de quantidades de resíduos populados nas regiões das hélices-alfas e folhas-betas, caso tenha, indicando se existe resíduos habitando regiões desfavoráveis. Todos esses indicativos são possíveis para análises, através da geração do mapa de Ramachandran, indicando se possui uma tendência ao polipeptídio ser desfavoravelmente enovelado.

Pode-se também analisar, se com o refinamento o polipeptídio teve uma melhor abrangência nas regiões do mapa em relação ao não refinado. Os resultados obtidos são sempre comparados em relação à conformação experimental mais aproximada ao da trajetória.

Esta comparação busca o melhor frame da trajetória da dinâmica molecular com todos os arquivos experimentais que serão utilizados para os cálculos da RMSD, com cada conformação experimental constante no arquivo do PDB. O cálculo da RMSD, é executado automaticamente pelo programa Ptraj do AMBER 14.

6.4.5.2 Visualização da trajetória da Dinamica Molecular

Outros arquivos gerados para análise, tais como as estruturas polipeptídicas de alça/volta, hélices-alfas, folhas-betas, são os arquivos como *"prmtop"* (parâmetros da topologia da molécula) e o "crd" (indicam os frames da trajetória). Com estes dois arquivos é possível visualizar artificialmente via *software* VMD (*Visual Molecular Dynamics*), toda a trajetória da dinâmica molecular gerada. O VMD é um programa de visualização molecular para demonstração, animação e análise de sistemas de macromoléculas usando gráficos 3D *(http://www.ks.uiuc.edu/Research/vmd/).* A animação permite que o usuário acompanhe em um vídeo, todos os movimentos realizados pela proteína durante o refinamento, podendo-se realizar as análises de cada frame.

Anteriormente os arquivos no formato "crd" eram gerados separadamente de cada passo

da trajetória da dinâmica molecular, sendo então informados um a um ao VMD, para montar toda a trajetória artificial da proteína alvo. O *workflow* desenvolvido facilita ao usuário agrupar em um único arquivo todos os arquivos com extensão "crd" contendo toda a trajetória da proteína alvo, facilitando assim a manipulação para a visualização da trajetória.

6.4.5.3 Visualização tridimensional das estruturas refinadas

Por último, resulta o arquivo no formato PDB com a visualização tridimensional da proteína após o refinamento, onde é possível realizar a análise de qualidade comparando-se os resultados entre a estrutura refinada e a experimental. A visualização será apresentada no Capítulo 7, dos resultados dos experimentos. Para realizar esta análise, utilizou-se o programa Pymol, versão 1.6 *(www.pymol.com),* que permite a visualização e manipulação de estruturas tridimensionais de proteínas, de moléculas menores, de superfícies densas, apresentando a modelagem em 3D de alta resolução e detalhes.

Para facilitar o usuário, após concluído o refinamento, é gerado pelo *workflow,* com base no mesmo método do mapa de Ramachandran, o cálculo utilizando o melhor RMSD em relação a estrutura experimental e a trajetória interna da dinâmica molecular. Como resultado é criado o arquivo no formato PDB com a estrutura tridimensional, com o frame de melhor refinamento comparado com a estrutura experimental.

6.4.5.4 Arquivos de *Log's*

Todo o processo realizado pelo *workflow* fica armazenado nos arquivos de *log's* que servirão para futuras consultas auxiliando na detecção e correção de problemas. Nestes arquivos são armazenados de forma agrupada todos os arquivos gerados nas etapas 2 e 3 separadamente do processo de refinamento do *workflow*, assim como todos os comandos executados automaticamente através do arquivo que inicia a dinâmica molecular.

A primeira etapa de armazenamento dos *log's* fica no arquivo onde se tem todas as informações do processo de refinamento como data e valores de entrada para as variáveis indicadas no comando de entrada do workflow.

Na última etapa é criaodo o restante dos arquivos que são:

- Um arquivo contendo todos os arquivos gerados de forma agrupada na fase de

aquecimento.

- Um arquivo contendo todos os arquivos gerados de forma agrupada na fase de produção
- Um arquivo contendo todas as restrições aplicadas para o processo de refinamento.

Dessa forma é possível reproduzir a mesma dinâmica molecular em outro momento. Porque todos os arquivos e dados necessários são salvos, com isso o usuário pode comparar outras dinâmicas já realizadas entre si como forma de analise e criação de novas dinâmicas moleculares.

7 RESULTADOS: AMBIENTE WEB

Para facilitar a interação do usuário com os resultados do *workflow*, desenvolveu-se uma interface *web* que disponibiliza os resultados de forma organizada, dessa maneira ajuda na visualização da pasta gerada pelo refinamento. Assim, o usuário não precisa navegar pelas diversas pastas geradas em seu sistema operacional a procura dos arquivos de resultados. Os capítulos a seguir descrevem cada interface e suas funcionalidades.

7.1 Interface de *Login*

Somente usuários autorizados terão acesso, para isso, o administrador faz o cadastro identificando-o com um nome e uma senha. Já cadastrado, o usuário acessa a tela de *login* conforme exibe a Figura 41, e faz sua autenticação possibilitando visualizar os resultados dos refinamentos.

Figura 41 – Tela de *Login*, onde o usuário informa nome e senha para ter acesso a tela inicial.

Completada a autenticação, o usuário é direcionado para a tela inicial do sistema onde no menu à esquerda, tem todas as opções disponíveis para o usuário logado.

7.2 Tela inicial do sistema

A tela inicial apresenta-se conforme mostra a Figura 42, com as opções disponíveis de acordo com o perfil cadastrado pelo administrador à cada usuário. A seguir será descrita a tela inicial com todas as opções de navegação para o perfil deste usuário.

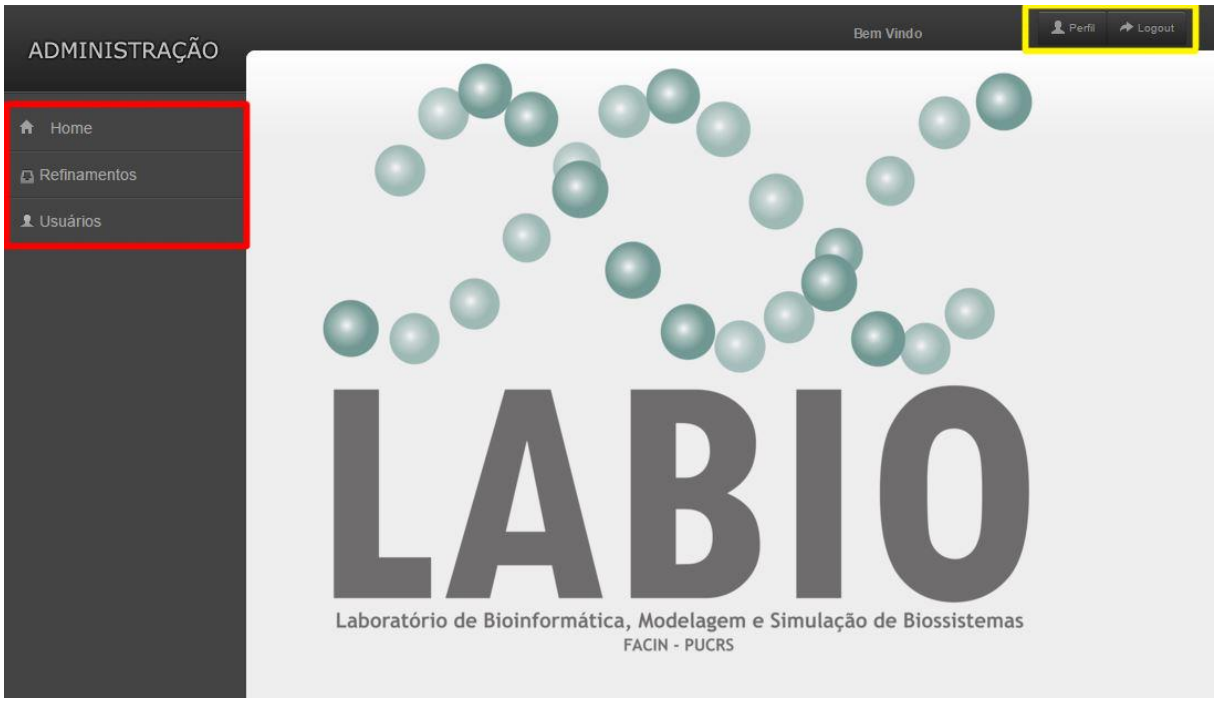

Figura 42 – Tela inicial com os menus disponibilizados para exibição, de acordo com o perfil de cada usuário.

- **Menu Lateral** – Conforme mostra o menu da esquerda, destacado em vermelho, todos os usuários terão acesso à opção de retornar à tela inicial (*Home*), de navegar pela lista de refinamentos (Refinamentos) sendo que este último item do menu (Usuários) só será visível para usuários autorizados pelo administrador.

- **Menu Superior Direito –** O menu superior à direita, destacado em amarelo, disponibiliza ao usuário editar e/ou alterar seus dados no botão "Perfil" e para encerrar a seção de forma segura o botão "*Logout*".

7.3 Modulo de Resultados dos Refinamentos

Este módulo disponibiliza os resultados de todos os refinamentos realizados. Para acessar este módulo o usuário deve clicar no botão “Refinamentos”, após clicar no subitem “Listagem”, com isso será exibido todos os resultados dos refinamentos realizados em ordem alfabética.

A ordenação é realizada com base no identificador informado pelo usuário, através do parâmetro “-id” que identifica o refinamento. A Figura 43 mostra a tela do módulo de resultados dos refinamentos.

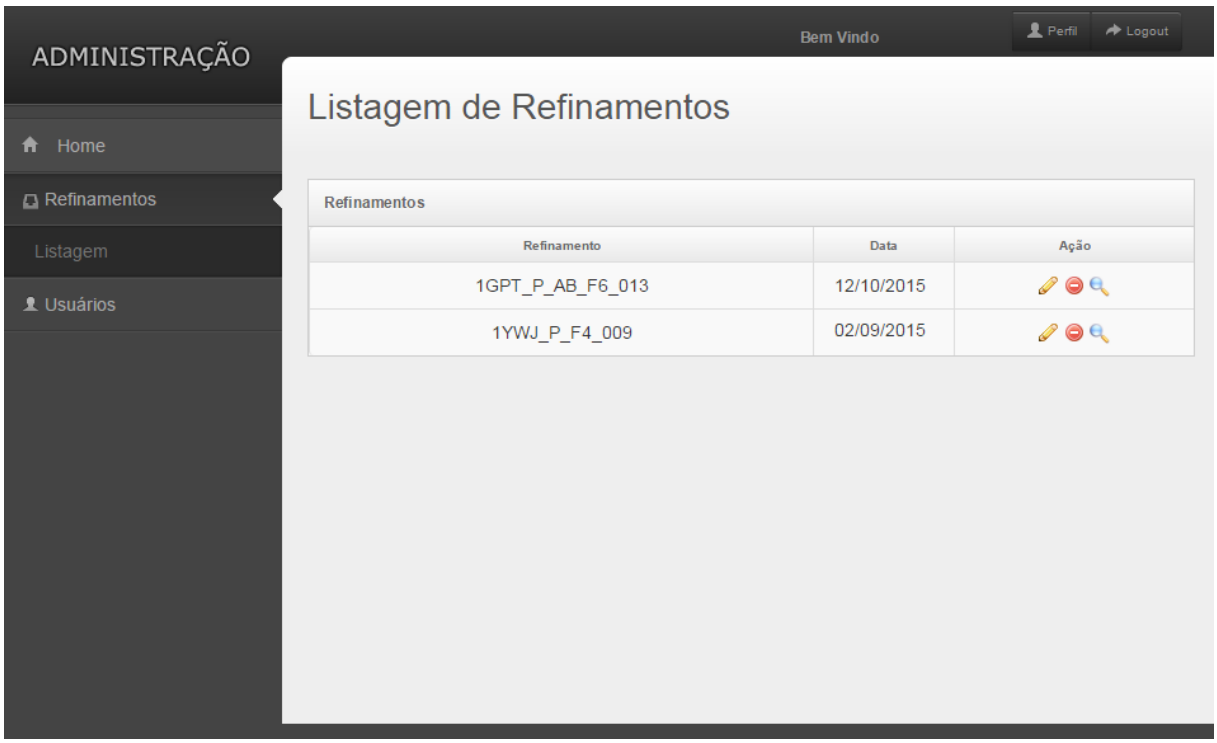

Figura 43 – Tela com a lista de todos os refinamentos realizados, ordenados pelo “id” que identifica o refinamento.

Como mostra a Figura 43, a tela exibe uma tabela com a lista ordenada dos refinamentos dividida em três colunas. A primeira coluna identifica o “id”, ou seja, o nome definido pelo usuário para o refinamento; na segunda, mostra a data em que foi realizado o refinamento a última coluna (Ação) disponibiliza algumas ações que podem ser realizadas para cada refinamento, conforme descritas a seguir:

- **Lápis** – A ação de edição, representada pelo ícone de um lápis, permite ao usuário editar o nome (“id”) do refinamento caso queira alterá-lo. O uso desta ação será mais detalhado no Capítulo 6.4, Módulo: Edição do Refinamento.
- **Círculo Vermelho** – Este ícone permite que se exclua o refinamento.
- **Lupa** – Permite ao usuário visualizar os detalhes do refinamento com todos os arquivos que correspondem ao refinamento selecionado. O uso desta ação será mais detalhado no Capítulo 6.5, Módulo: Detalhes do Refinamento.

7.4 Ação: Edição do Refinamento (Lápis)

A ação de edição, permite ao usuário alterar o nome do refinamento inserido no comando de início do *workflow*. Ao clicar neste ícone será direcionado para o módulo de edição em uma nova tela, Figura 44.

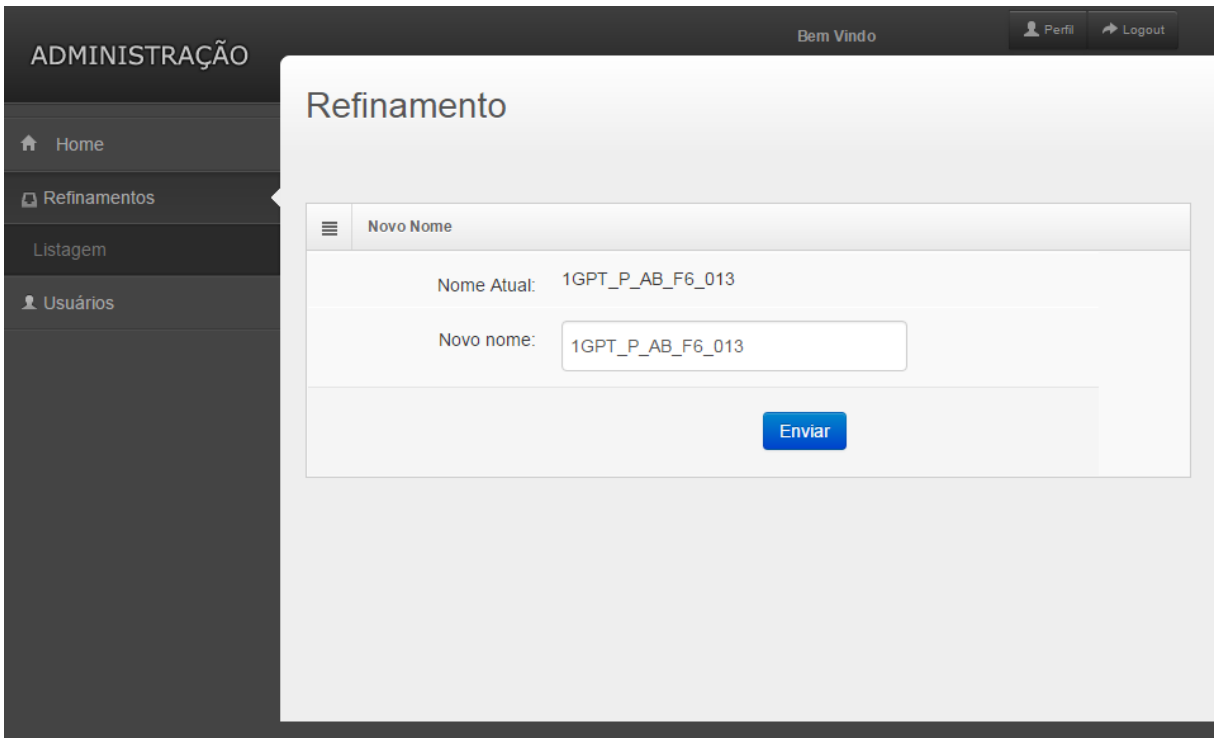

Figura 44 – Tela de edição para alterar o nome do refinamento.

Nesta tela apresenta o nome atual do refinamento e o campo onde o usuário irá inserir o novo nome para substituição. O campo “Nome Atual” está informado o nome inserido anteriormente. Para facilitar a troca do nome é carregado o nome atual dentro do campo “Novo

Nome", para caso o usuário deseje realizar uma pequena alteração no nome anterior evitando ter que reescrever todo o nome novamente.

Antes de editar o novo nome, o sistema verifica se não existe outro refinamento com o mesmo nome. Caso esteja disponível o novo nome, será efetuada a alteração que, após realizada, retorna para tela de "Resultado dos Refinamentos", informando que a troca foi realizada com sucesso.

7.5 Ação: Detalhes do Refinamento (Lupa)

Esta ação permite ao usuário acessar todos os arquivos gerados pelo o *workflow* de um determinado refinamento. Para acessar este módulo, o usuário deve clicar no ícone da lupa na linha do refinamento escolhido. Com isso, será direcionado para uma nova tela onde são listados todos os arquivos separados por tipos que o identificam dentro do processo de refinamento. Cada tipo de arquivo está agrupado em uma mesma tabela, sendo que, a primeira coluna, informa o nome do arquivo, e a segunda coluna, o ícone representado por um círculo com uma seta no centro, para o usuário salvar o arquivo em sua máquina. Essa tela, por listar muitos arquivos, torna-se muita extensa exigindo o uso da barra de rolagem para visualizar sua totalidade e, portanto, não teria como apresentá-la aqui. Então, para facilitar a visualização, separou-se em partes a sequência da tela, conforme descreve-se a seguir:

- **Parte 1: Refinamento e Restrições** – A tela de apresentação tem início conforme mostra a Figura 45.

Figura 45 – Tela inicial do detalhamento do refinamento selecionado.

Na parte superior da tela mostra o nome do refinamento informado pelo usuário, no caso, YWJ_P_F4_009, destacado em amarelo, para situar o usuário os detalhes de qual refinamento está visualizando.

O quadro destacado em vermelho, Figura 45, é apresentado o melhor refinamento obtido com base no arquivo experimental e, no quadro em verde, o arquivo de restrições utilizados para realização do refinamento. A ordem da apresentação desses resultados será sempre nesta ordem, conforme mostra a Figura 45, que representa a tela de "Detalhes do Refinamento".

- **Parte 2: Arquivos da fase de aquecimento –** Na sequência da rolagem da barra, aparece a listagem com todos os arquivos gerados para a fase de aquecimento, de acordo com os cálculos efetuados na Etapa 2 do processo de refinamento. Os arquivos foram nomeados pelo *workflow* quando foram gerados com o nome "*heating phase*", que indica que o arquivo é da fase de aquecimento, na sequência da sua nomenclatura apresenta o intervalo de picossegundos de cada arquivo e por último, a extensão referente à fase de aquecimento ".in", conforme mostra a Figura 46.

Figura 46 – A sequência da Figura 45 após rolagem da barra, mostra os arquivos da fase de aquecimento.

- **Parte 3: arquivos da fase de produção –** Na sequência da rolagem da barra, aparece a listagem com todos os arquivos gerados para a fase de produção, de acordo com os cálculos efetuados na Etapa 2 do processo de refinamento. Os arquivos foram nomeados pelo *workflow* quando foram gerados com o nome "*production phase*", que indica que o arquivo

é da fase de produção, na sequência da sua nomenclatura apresenta o intervalo de picossegundos de cada arquivo e por último, a extensão referente à fase de produção“.in”, conforme mostra a Figura 47.

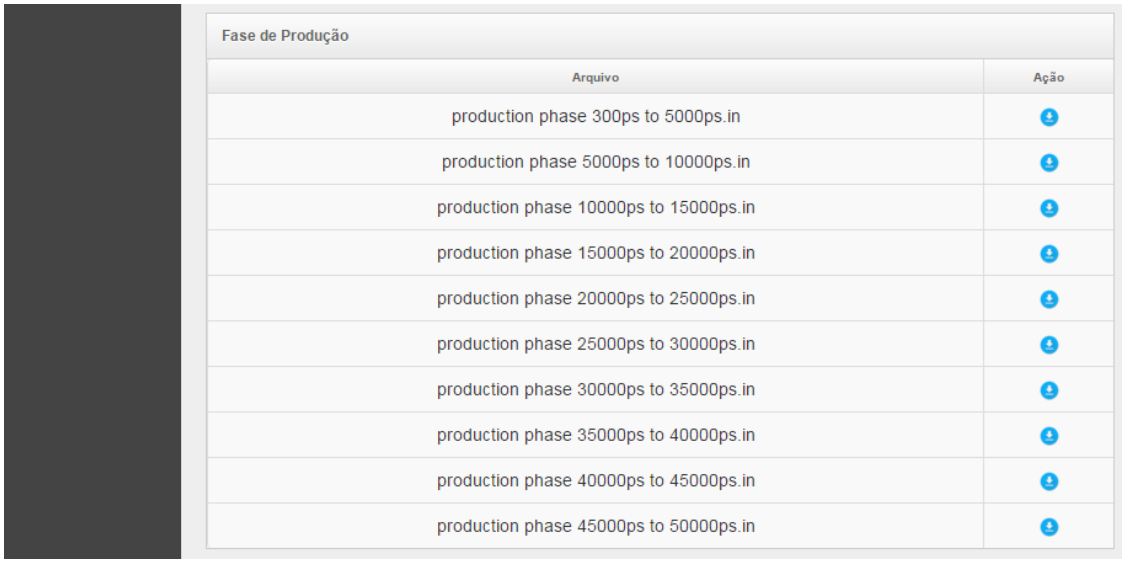

Figura 47 – A sequência da Figura 45 após rolagem da barra, mostra os arquivos da fase de produção.

- **Parte 4 – Arquivos para visualização artificial da trajetória:** Na sequência da rolagem da barra, aparece a listagem com todos os arquivos para a visualização artificial da trajetória utilizando o programa VMD. Os arquivos com a extensão “crd” contém todas as coordenadas da dinâmica molecular e o arquivo de extensão “*prmtop*” contém os parâmetros da topologia da proteína, conforme mostra a Figura 48.

Figura 48 – A sequência da Figura 45 após rolagem da barra, mostra os arquivos para a visualização artificial da trajetória utilizando o programa VMD.

- **Parte 5 – Arquivos gerados de análise do Prochek:** Na sequência da rolagem da barra, aparece a listagem com todos os arquivos gerados de análise do Prochek, conforme mostra a Figura 49. Todos os arquivos com a extensão “ps” contêm diferentes tipos de análises do refinamento utilizando o mapa de Ramachandran. O mapa de Raamachandran com a distribuição dos resíduos nas regiões de hélice-alfa e folha-beta é um desses arquivos, utilizado nos resultados dos experimentos para comprovar a eficiência do refinamento.

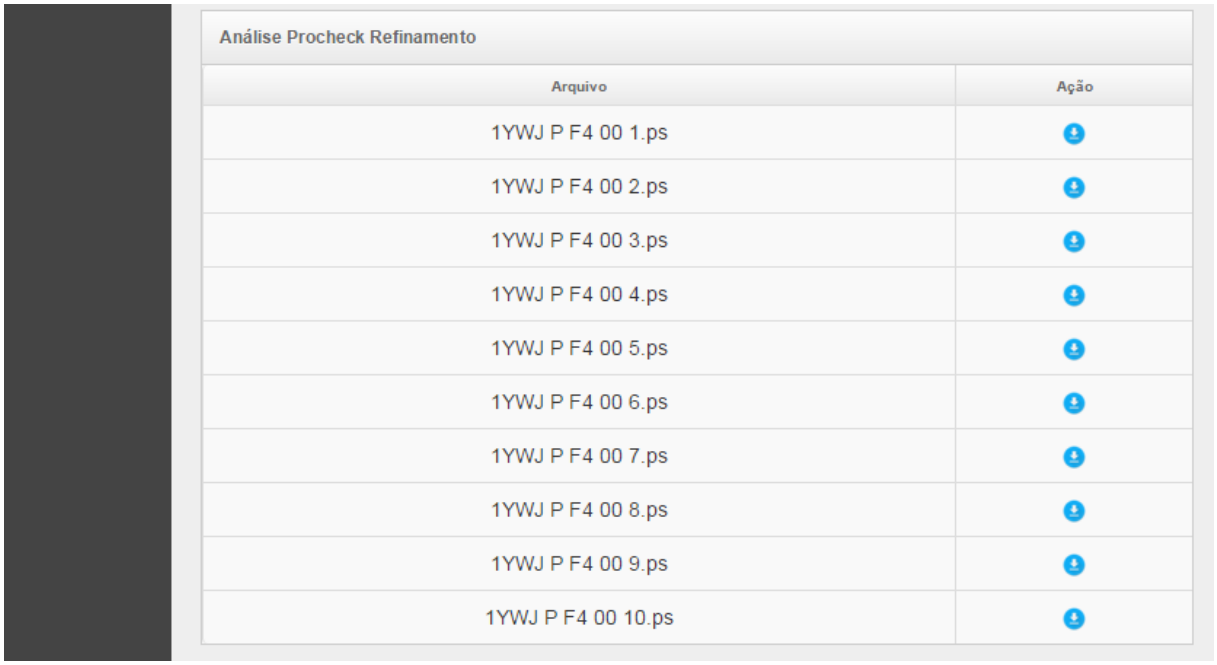

Figura 49 – A sequência da Figura 45 após a última rolagem da barra, mostra os arquivos gerados de análise do Prochek

- **Parte 6 – Arquivo de Log:** Na sequência da última rolagem da barra, aparece em um único arquivo de log todos os arquivos de *log’s* gerados durante a execução do *workflow*.

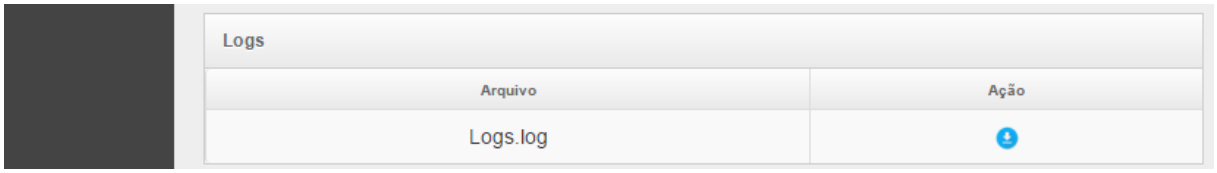

Figura 50 – A sequência da Figura 45 após a última rolagem da barra, mostra o arquivo gerado de Log

7.6 Módulo de Controle de Usuários

Este módulo permite, aos administradores, o cadastro de um novo usuário e o controle de acesso ao ambiente de resultados dos refinamentos gerados pelo o *workflow*. Para acessar este módulo, o usuário que possui permissão cadastrada, deve clicar no último botão no menu da esquerda “Usuários”, após clicar no subitem “Controle de Usuários”, com isso será exibida a tela de cadastro e gerenciamento de usuários já cadastrados, como mostra a Figura 51.

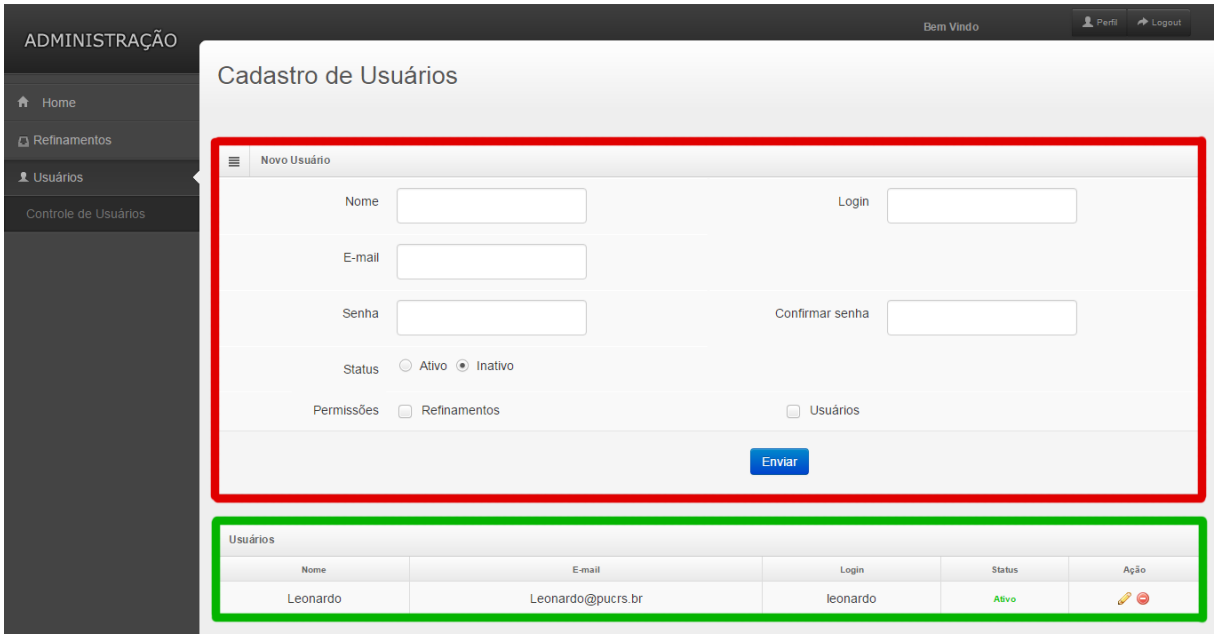

Figura 51 –Tela de Cadastro de Usuários.

Como mostra a Figura 51, destacado em vermelho, refere-se ao cadastro de novos usuários. Para efetuar o cadastro, deve-se preencher todos os campos solicitados (nome, e-mail, *login*, senha e confirmação de senha), e no campo “Status”, marcar uma das duas opções, “Ativo”, para usuários que podem efetuar *login* e consultar os refinamentos realizados, ou “Inativo”, aqueles que, mesmo estando cadastrados, não terão acesso. Dessa forma, mantém-se o cadastro de todos os envolvidos, mesmo usuários que por algum motivo não estejam utilizando o ambiente de resultados no momento, mas que, futuramente, venham utilizá-lo novamente. Se fosse excluído, teria que cadastrá-lo novamente, dessa maneira basta só ativá-lo da opção “Status”.

E por último, tem-se a opção das permissões do usuário, onde será marcado todos os itens do menu que ele poderá acessar.

Na parte inferior da tela, destacado em verde, aparece a tabela com a listagem dos usuários cadastrados e alguns dos seus dados como nome, e-mail e *login*. A última coluna da tabela tem a opção para o administrador de alterar os dados cadastrados ou excluir cada usuário apresentado na listagem.

8 RESULTADOS: EXPERIMENTOS UTILIZANDO O MODELO DE *WORKFLOW*

Este capítulo tem por objetivo descrever os resultados dos experimentos realizados com o modelo de *workflow* desenvolvido, para desta forma validar a eficiência do processo demonstrando que os resultados foram muito satisfatórios. Para realização dos experimentos utilizou-se um conjunto de três proteínas com características diferentes e com refinamentos já conhecidos. Assim, comprovou-se que, após submeter estas mesmas proteínas ao refinamento automatizado do modelo de *workflow*, foram obtidos os mesmos resultados do experimento executado de forma manual.

8.1.1 Definição do conjunto de proteínas utilizadas

No intuito de aprimorar e conseguir alcançar um bom refinamento, foi definido um conjunto de três proteínas para teste. A escolha desse conjunto tomou por base artigos que descreviam a realização de testes de refinamento bem-sucedidos. Além de considerar proteínas já utilizadas em testes, foram selecionadas aquelas com diferentes classes estruturais (α, β e αβ) e com menos de 53 resíduos. As proteínas selecionadas foram:

- 1GAB: Cadeia A da proteína PAB (*Escherichia coli*), determinada por NMR, considerada proteína pequena (53 aminoácidos) classe SCOP α, (Johansson *et al.*,1997);

- 1YWJ: Estrutura do domínio FBPWW1 (*Homo sapiens*), determinada por NMR, considerada proteína pequena (28 aminoácidos), classe SCOP β, (Pires *et al.*, 2005).

- 1GPT: Estrutura em solução das tioninas gama 1-H e gama 1-P de cevada (*Hordeum vulgare*) e endosperma de trigo determinada por 1H-NMR: um motivo estrutural comum a proteínas tóxicas de artrópodes. Determinada por NMR, considerada proteína pequena (47 aminoácidos), classe SCOP pequenas proteínas (αβ), (Bruix *et al.*, 1993).

8.1.2 Experimento 1: Refinamento da Proteína 1GAB

Para este experimento foi realizado o refinamento da proteína cujo o código PDB é 1GAB, composta 53 resíduos de aminoácidos, tendo uma cadeia tripla de hélices-α e interligadas por suas alças e voltas correspondentes. Sua estrutura 3D foi obtida experimentalmente através de ressonância magnética nuclear (NMR).

Para formar o comando de execução da dinâmica molecular, foram utilizados os arquivos da predição da estrutura 3D aproximada gerada pelo método CReF, da estrutura experimental, das restrições, o arquivo com a informação do raio de corte, o parâmetro de duração da dinâmica molecular e o "id" do refinamento.

Após iniciar a dinâmica molecular é executada a minimização de energia com 500 passos de tempo, com o propósito de relaxar distorções nas ligações químicas, nos ângulos entre ligações e nos contatos de van der Waals.

A seguir, são executadas as fases de aquecimento e produção. Na fase de aquecimento a temperatura foi aumentada gradativamente em 50 K a cada arquivo gerado, até atingir a temperatura final definida de 325 K, e o tempo total de duração de 30.000ps.

Na fase de produção é mantida a temperatura final da fase de aquecimento para a execução da dinâmica molecular sendo utilizado o tempo total de duração de 50.000ps, que nas duas fases foi aplicado um passo de tempo de integração de 0.002ps.

Para o arquivo referente ao raio de corte foi informado o valor de 8,6 Angströms, que especifica o ponto de corte de átomos não ligantes. O solvente foi tratado implicitamente utilizando *Ge]neralized Born*, que reduz o custo computacional, e por último foi utilizado o arquivo modelo para indicação das restrições dos ângulos diedros, onde são aplicadas as restrições nas hélices-α .

Com a configuração e criação dos arquivos conforme descritos, a proteína alvo 1GAB é então submetida às etapas do refinamento obtendo-se os resultados apresentados a seguir.

O primeiro resultado a ser avaliado é para identificar o quanto próximo é a semelhança da proteína alvo 1GAB refinada com a conformação experimental. Os arquivos no formato PDB da conformação refinada e da experimental foram submetidos ao programa Pymol, para gerar as duas conformações tridimensionais. Desta maneira é possível comparar as duas conformações que foram alinhadas, paralelamente, de modo que, a cor rosa representa a

conformação refinada e, a azul, a experimental, conforme mostra a Figura 52.

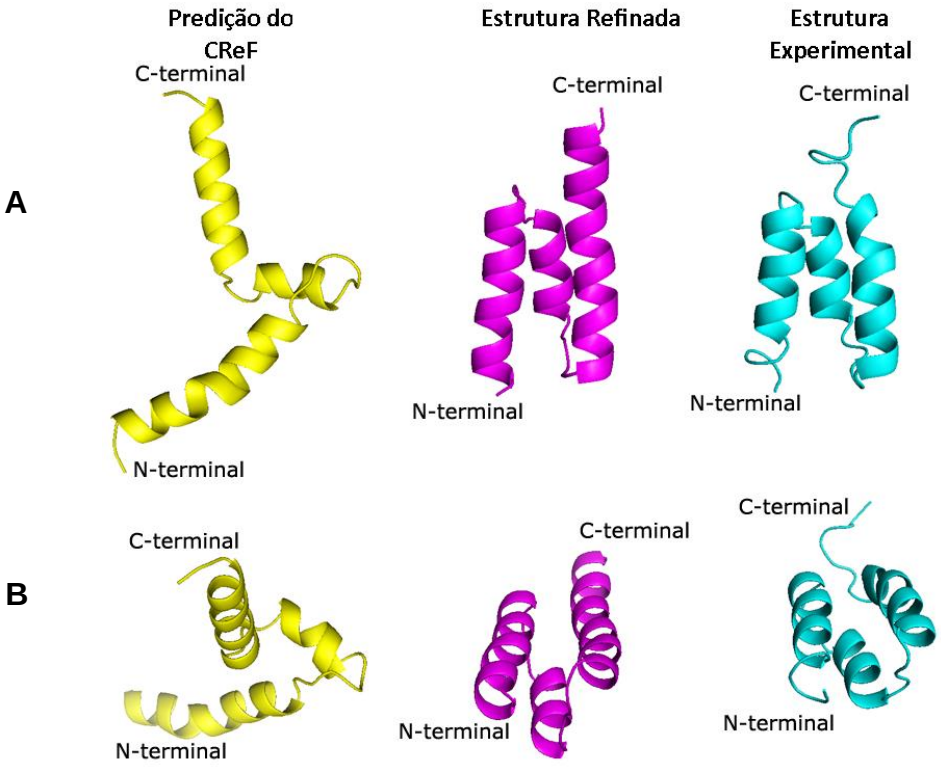

Figura 52 – A sequência de imagens (letra A) representa três perspectivas da estrutura da proteína 1GAB, onde em amarelo indica a predição inicial do CReF, rosa o seu refinamento e em azul sua estrutura experimental. A sequência de imagens (letra B) representa três perspectivas diferentes da letra A. Em todas as estruturas utiliza-se do "C-terminal" e "N-terminal" para mostrar o movimento realizado em cada perspectiva.

Observa-se que esta simulação teve seu início com uma estrutura predita pelo método CReF e não com uma conformação já refinada. Conforme mostra a Figura 52, visualizada em várias perspectivas (A e B) observa-se claramente que a proteína alvo 1GAB com a melhor conformação refinada a partir do passo de tempo 4,85ps, tem sua simililaridade muito próxima da conformação experimental. Com base nas restrições aplicadas, todas as hélices-α foram mantidas e suas alças/voltas estão muito semelhantes com a da conformação experimental.

Também realizou-se outra análise para avaliar a qualidade do modelo gerado, para isso utilizou-se do mapa de Ramachamdran o qual fornece informações acerca da qualidade do arranjo estrutural, através da análise esterioquímica da estrutura experimental (Figura 53), a estrutura predita pelo CReF (Figura 54) e a estrutura refinada (Figura 55).

Os mapas representados nas figuras que seguem, ratificam o resultado satisfatório do

refinamento da proteína alvo 1GAB através do modelo de *workflow* usado.

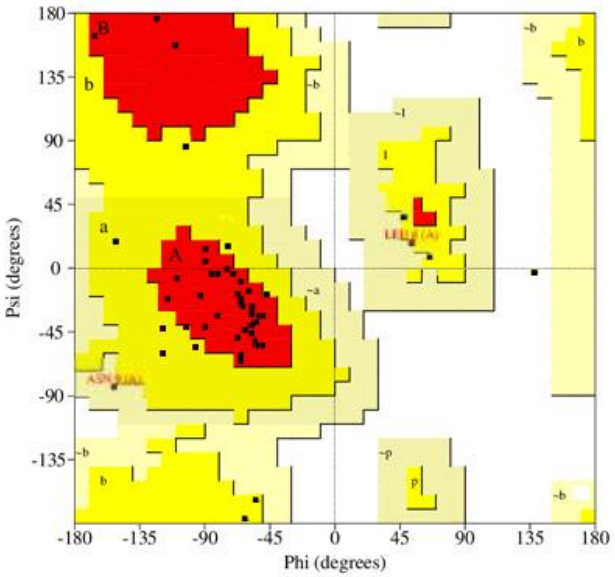

Figura 53 – Mapa de Ramachandran da estrutura experimental, com 36 resíduos em regiões mais favoráveis (A, B e L) contabilizando 72%, com 12 resíduos em regiões permitidas (a,b,l,p) contabilizando 24%, com 2 resíduos em região ainda aceitável (~a,~b,~l,~p) contabilizando 4% e nenhum resíduo em regiões não permitidas contabilizando 0%. Foram excluídas da contabilização dos resíduos de glicina e a prolina, com eles totalizariam os 53 resíduos.

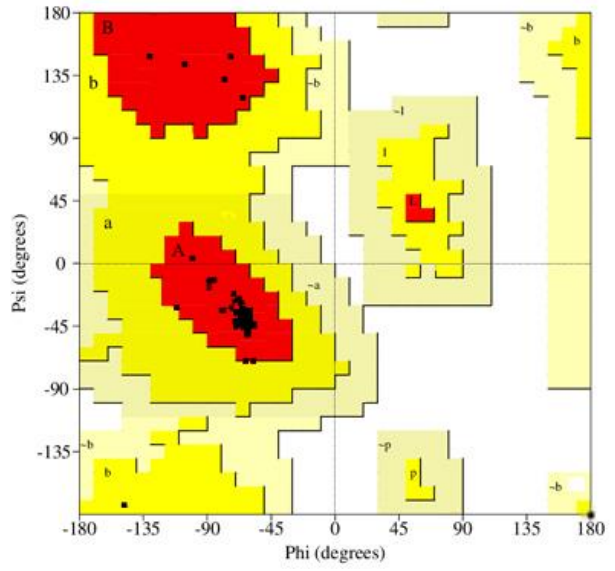

Figura 54 – Mapa de Ramachandran da estrutura predita pelo CReF, com 45 resíduos em regiões mais favoráveis (A, B e L) contabilizando 90%, com 5 resíduos em regiões permitidas (a,b,l,p) contabilizando 10%, com nenhum resíduo em região ainda aceitável (~a,~b,~l,~p) contabilizando 0% e nenhum resíduo em regiões não permitidas contabilizando 0%. Foram excluídas da contabilização dos resíduos de glicina e a prolina, com eles totalizariam os 53 resíduos.

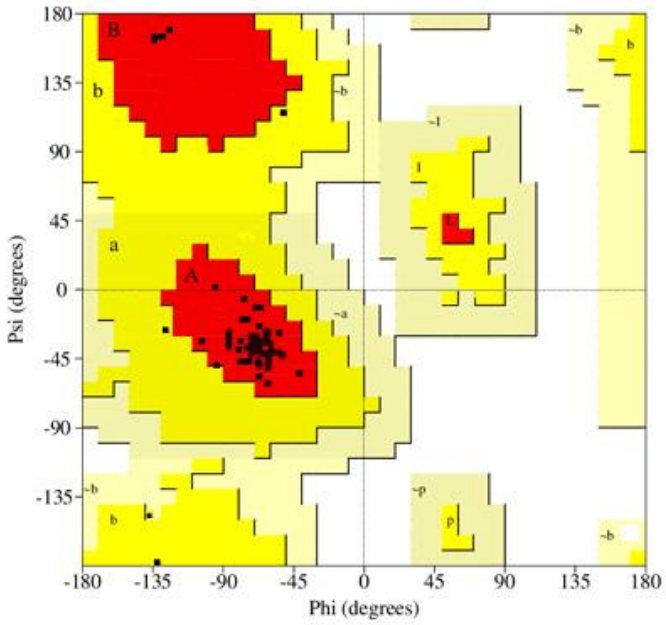

Figura 55 – Mapa de Ramachandran da estrutura refinada, com 47 resíduos em regiões mais favoráveis (A, B e L) contabilizando 94%, com 3 resíduos em regiões permitidas (a,b,l,p) contabilizando 6%, com nenhum resíduo em região ainda aceitável (~a,~b,~l,~p) contabilizando 0% e nenhum resíduo em regiões não permitidas contabilizando 0%. Foram excluídas da contabilização dos resíduos de glicina e a prolina, com eles totalizariam os 53 resíduos.

Desta forma, é possível confirmar a partir dos mapas de Ramachandran anteriores, que, após o refinamento da estrutura alvo 1GAB, obteve-se uma grande quantidade de pontos nas regiões mais favoráveis (A, B e L) comparado com a estrutura predita pelo CReF e, de uma forma geral, a estrutura refinada, ocupou as mesmas regiões do mapa da estrutura experimental, com maior quantidade de resíduos em regiões mais favoráveis.

Com base na imagem gerada do refinamento comparada com a experimental e nos mapas de Ramachandran, conclui-se que, a proteína após ser submetida ao *workflow* de refinamento realizou o refinamento com base na predição do CReF.

8.1.3 Experimento 2: Refinamento da Proteína 1YWJ

Para este experimento, foi realizado o refinamento da proteína cujo o código PDB é 1YWJ, composta 28 resíduos de aminoácidos, tendo uma cadeia tripla de três folhas-β antiparalelas e interligadas por suas alças e voltas correspondentes. Sua estrutura 3D foi obtida

experimentalmente através de ressonância magnética nuclear (NMR).

Para formar o comando de execução da dinâmica molecular, foram utilizados os arquivos da predição da estrutura 3D aproximada gerada pelo método CReF, da estrutura experimental, das restrições, o arquivo com a informação do raio de corte, o parâmetro de duração da dinâmica molecular e o "id" do refinamento.

Após iniciar a dinâmica molecular é executada a minimização de energia com 500 passos de tempo, com o propósito de relaxar distorções nas ligações químicas, nos ângulos entre ligações e nos contatos de Van der Waals.

A seguir, são executadas as fases de aquecimento e produção. Na fase de aquecimento a temperatura foi aumentada gradativamente em 50 K a cada arquivo gerado, até atingir a temperatura final definida de 300 K, e o tempo total de duração de 50.000ps.

Na fase de produção é mantida a temperatura final da fase de aquecimento para a execução da dinâmica molecular, sendo utilizado o tempo total de duração de 50.000ps, que nas duas fases, foi aplicado um passo de tempo de integração de 0.002ps.

Para o arquivo referente ao raio de corte foi informado o valor de 8,3 Angströms, que especifica o ponto de corte de átomos não ligantes. O solvente foi tratado implicitamente utilizando *Generalized Born*, que reduz o custo computacional, e, por último, foi utilizado o arquivo modelo para indicação das restrições dos ângulos diedros, onde são aplicadas as restrições apenas nas folhas β.

Com a configuração e criação dos arquivos conforme descritos, a proteína alvo 1YWJ é então submetida às etapas do refinamento obtendo-se os resultados apresentados a seguir.

O primeiro resultado a ser avaliado é para identificar o quanto próximo é a semelhança da proteína alvo 1YWJ refinada com a conformação experimental. Os arquivos no formato PDB da conformação refinada e da experimental foram submetidos ao programa Pymol, para gerar as duas conformações tridimensionais. Desta maneira, é possível comparar as duas conformações que foram alinhadas paralelamente de modo que a cor rosa representa a conformação refinada e a azul a experimental, conforme mostra a Figura 56.

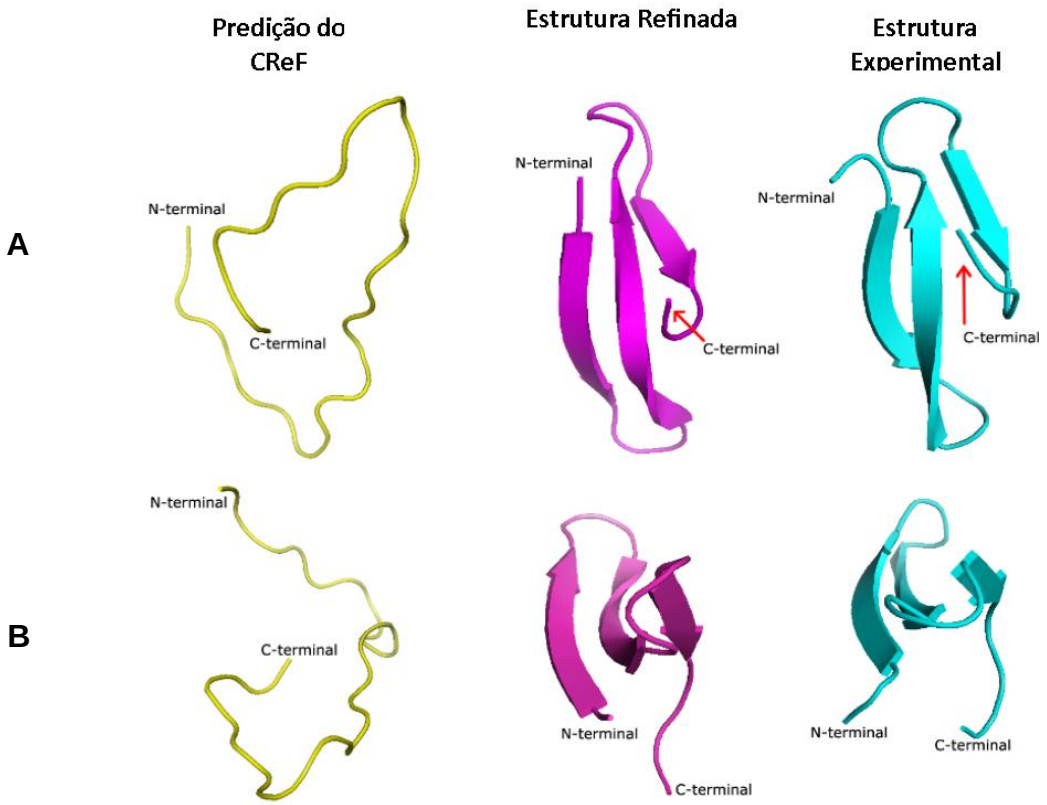

Figura 56 – A sequência de imagens (letra A) representa três perspectivas da estrutura da proteína 1YWJ, onde em amarelo indica a predição inicial do CReF, rosa o seu refinamento e em azul sua estrutura experimental. A sequência de imagens (letra B) representa três perspectivas diferentes da letra A. Em todas as estruturas utiliza-se do "C-terminal" e "N-terminal" para mostrar o movimento realizado em cada perspectiva.

Observa-se que, esta simulação, teve seu início com uma estrutura predita pelo método CReF e não com uma conformação já refinada. Conforme mostra a Figura 56, visualizada em várias perspectivas (A e B) observa-se claramente que a proteína alvo 1YWJ com a melhor conformação refinada a partir do passo de tempo 11,31ps, tem sua simililaridade muito próxima da conformação experimental. Com base nas restrições aplicadas, todas as folhas-β foram mantidas e suas alças/voltas estão muito semelhantes com a da conformação experimental.

Também realizou-se outra análise para avaliar a qualidade do modelo gerado, para isso, utilizou-se do mapa de Ramachamdran, o qual fornece informações acerca da qualidade do arranjo estrutural, através da análise esterioquímica da estrutura experimental (Figura 57), a estrutura predita pelo CReF (Figura 58) e a estrutura refinada (Figura 59).

Os mapas representados nas figuras a seguir, ratificam o resultado satisfatório do refinamento da proteína alvo 1YWJ através do modelo de *workflow* usado.

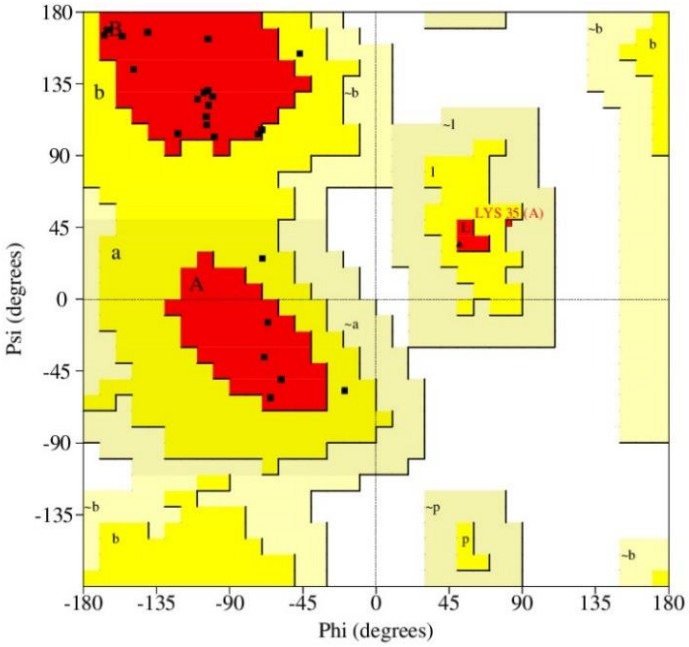

Figura 57 – Mapa de Ramachandran da estrutura experimental, com 21 resíduos em regiões mais favoráveis (A,B e L) contabilizando 87.5%, com 2 resíduos em regiões permitidas (a,b,l,p) contabilizando 8.3%, com 1 resíduos em região ainda aceitável (~a,~b,~l,~p) contabilizando 4.2% e nenhum resíduo em regiões não permitidas contabilizando 0%. Foram excluídas da contabilização dos resíduos de glicina e a prolina, com eles totalizariam os 28 resíduos.

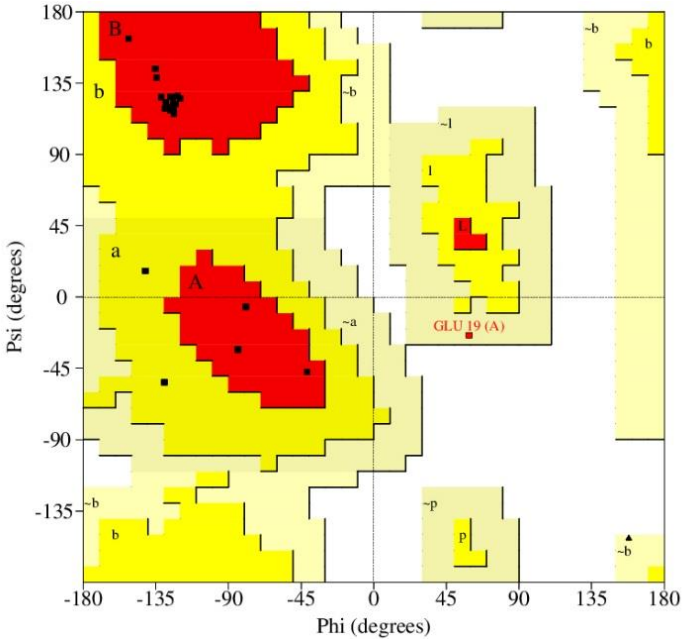

Figura 58 – Mapa de Ramachandran da estrutura predita pelo CReF, com 21 resíduos em regiões mais favoráveis (A, B e L) contabilizando 87.5%, com 2 resíduos em regiões permitidas (a,b,l,p) contabilizando 8.3%, com 1 resíduos em região ainda aceitável (~a,~b,~l,~p) contabilizando 4.2% e nenhum resíduo em regiões não permitidas contabilizando 0%. Foram excluídas da contabilização dos resíduos de glicina e a prolina, com eles totalizariam os 28 resíduos.

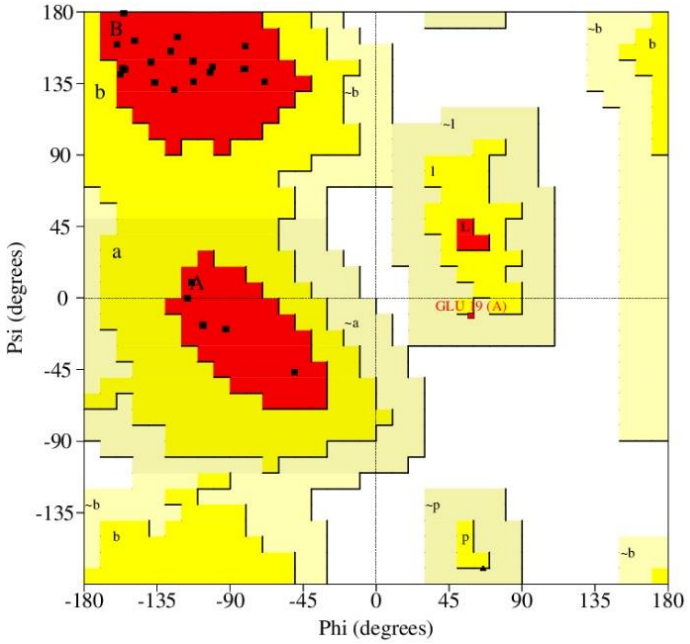

Figura 59 – Mapa de Ramachandran da estrutura refinada, com 23 resíduos em regiões mais favoráveis (A, B e L) contabilizando 95.8%, com nenhum resíduo em regiões permitidas (a,b,l,p) contabilizando 0%, com 1 resíduos em região ainda aceitável (~a,~b,~l,~p) contabilizando 4.2% e nenhum resíduo em regiões não permitidas contabilizando 0%. Foram excluídas da contabilização dos resíduos de glicina e a prolina, com eles totalizariam os 28 resíduos.

Desta forma, é possível confirmar a partir dos mapas de Ramachandran anteriores, que, após o refinamento da estrutura alvo 1YWJ, obteve-se uma grande quantidade de pontos nas regiões mais favoráveis (A, B e L) comparado com a estrutura predita pelo CReF e, de uma forma geral, a estrutura refinada ocupou as mesmas regiões do mapa da estrutura experimental, com maior quantidade de resíduos em regiões mais favoráveis.

Com base na imagem gerada do refinamento comparada com a experimental e nos mapas de Ramachandran, conclui-se que, a proteína após ser submetida ao *workflow* de refinamento realizou o refinamento com base na predição do CReF.

8.1.4 Experimento 3: Refinamento da Proteína 1GPT

Para este experimento foi realizado o refinamento da proteína cujo o código PDB é 1GPT, composta 36 resíduos de aminoácidos, tendo uma hélice α, uma cadeia tripla de folhas-β e interligadas por suas alças e voltas correspondente. Sua estrutura 3D foi obtida experimentalmente através de ressonância magnética nuclear (NMR).

Para formar o comando de execução da dinâmica molecular, foram utilizados os arquivos da predição da estrutura 3D aproximada gerada pelo método CReF, da estrutura experimental, das restrições, o arquivo com a informação do raio de corte, o parâmetro de duração da dinâmica molecular e o “id” do refinamento.

Após iniciar a dinâmica molecular é executada a minimização de energia com 500 passos de tempo, com o propósito de relaxar distorções nas ligações químicas, nos ângulos entre ligações e nos contatos de van der Waals.

A seguir, são executadas as fases de aquecimento e produção. Na fase de aquecimento, a temperatura foi aumentada gradativamente em 50 K a cada arquivo gerado, até atingir a temperatura final definida de 325 K, e o tempo total de duração de 50.000ps.

Na fase de produção é mantida a temperatura final da fase de aquecimento para a execução da dinâmica molecular, sendo utilizado o tempo total de duração de 50.000ps, que, nas duas fases, foi aplicado um passo de tempo de integração de 0.002ps.

Para o arquivo referente ao raio de corte, foi informado o valor de 8,3 Angströms, que especifica o ponto de corte de átomos não ligantes. O solvente foi tratado implicitamente utilizando *Generalized Born*, que reduz o custo computacional, e por último foi utilizado o arquivo modelo para indicação das restrições dos ângulos diedros, onde são aplicadas as restrições nas hélices-α e folhas β.

Com a configuração e criação dos arquivos conforme descritos, a proteína alvo 1GPT é então submetida às etapas do refinamento, obtendo-se os resultados apresentados a seguir.

O primeiro resultado a ser avaliado é para identificar o quanto próximo é a semelhança da proteína alvo 1GPT refinada com a conformação experimental. Os arquivos no formato PDB da conformação refinada e da experimental foram submetidos ao programa Pymol, para gerar as duas conformações tridimensionais. Desta maneira, é possivel comparar as duas conformações que foram alinhadas paralelamente de modo que, a cor rosa representa a conformação refinada e, a azul, a experimental, conforme mostra a Figura 60.

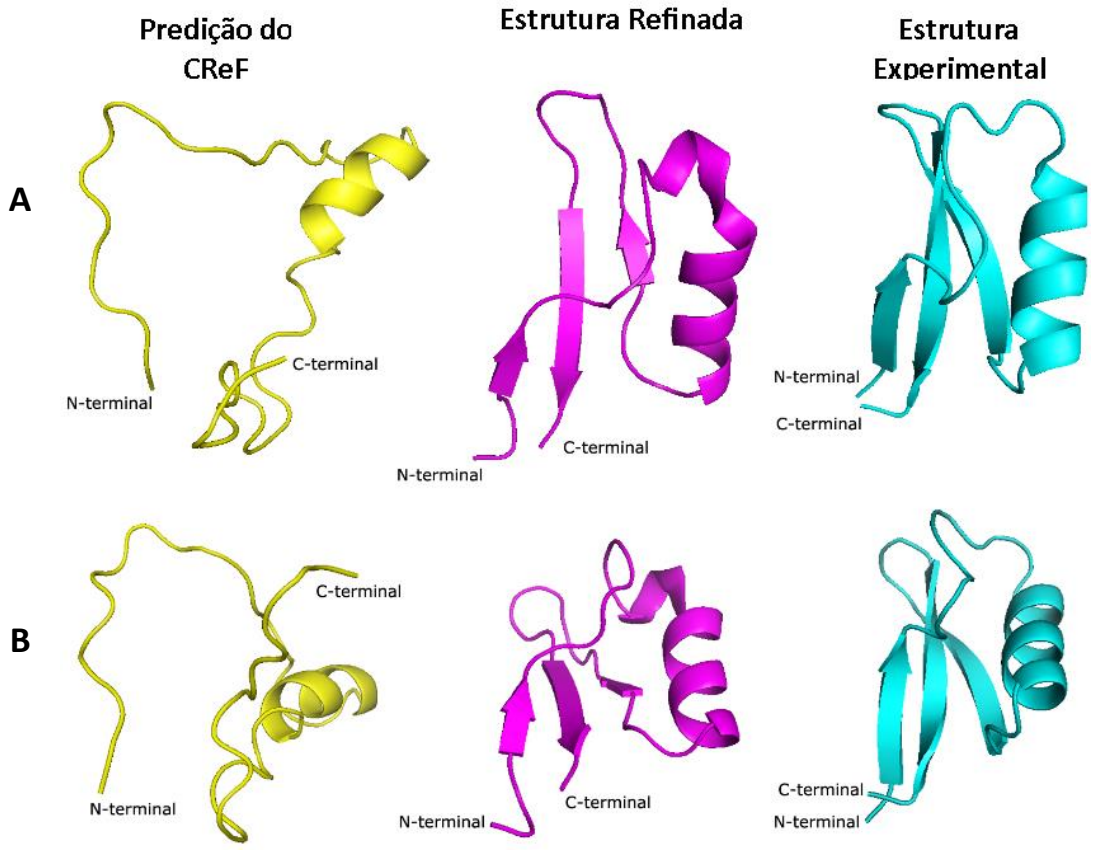

Figura 60 – A sequência de imagens (letra A) representa três perspectivas da estrutura da proteína 1GAB, onde em amarelo indica a predição inicial do CReF, rosa o seu refinamento e em azul sua estrutura experimental. A sequência de imagens (letra B) representa três perspectivas diferentes da letra A. Em todas as estruturas utiliza-se do "C-terminal" e "N-terminal" para mostrar o movimento realizado em cada perspectiva.

Observa-se, que esta simulação, teve seu início com uma estrutura predita pelo método CReF e não com uma conformação já refinada. Conforme mostra a Figura 60, visualizada em várias perspectivas (A e B), observa-se claramente que a proteína alvo 1GPT com a melhor conformação refinada a partir do passo de tempo 45,56ps, tem sua simililaridade muito próxima da conformação experimental. Com base nas restrições aplicadas, todas as hélices-α e folhas-β foram mantidas e suas alças/voltas estão muito semelhantes com a da conformação experimental.

Também realizou-se outra análise para avaliar a qualidade do modelo gerado, para isso utilizou-se do mapa de Ramachamdran o qual fornece informações acerca da qualidade do arranjo estrutural, através da análise esterioquímica da estrutura experimental (Figura 61), a estrutura predita pelo CReF (Figura 62) e a estrutura refinada (Figura 63).

Os mapas representados nas figuras a seguir, ratificam o resultado satisfatório do

refinamento da proteína alvo 1GPT através do modelo de *workflow* usado.

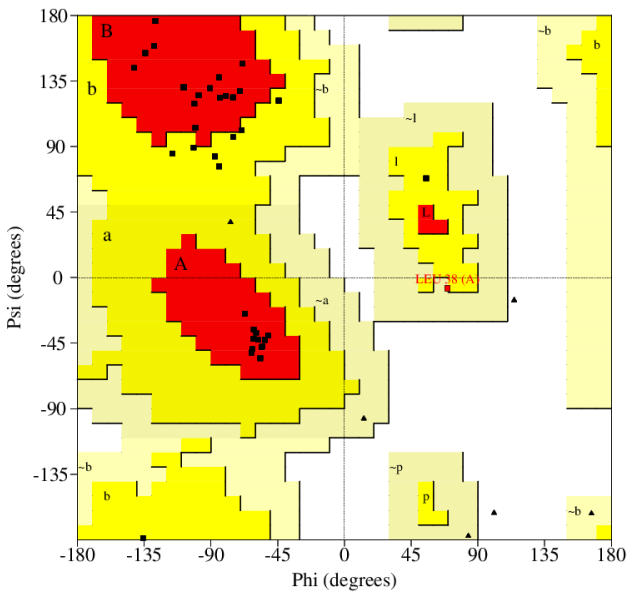

Figura 61 – Mapa de Ramachandran da estrutura experimental, com 27 resíduos em regiões mais favoráveis (A, B e L) contabilizando 75%, com 8 resíduos em regiões permitidas (a,b,l,p) contabilizando 22.2%, com 1 resíduos em região ainda aceitável (~a,~b,~l,~p) contabilizando 2,8% e nenhum resíduo em regiões não permitidas contabilizando 0%. Foram excluídas da contabilização dos resíduos de glicina e a prolina, com eles totalizariam os 47 resíduos.

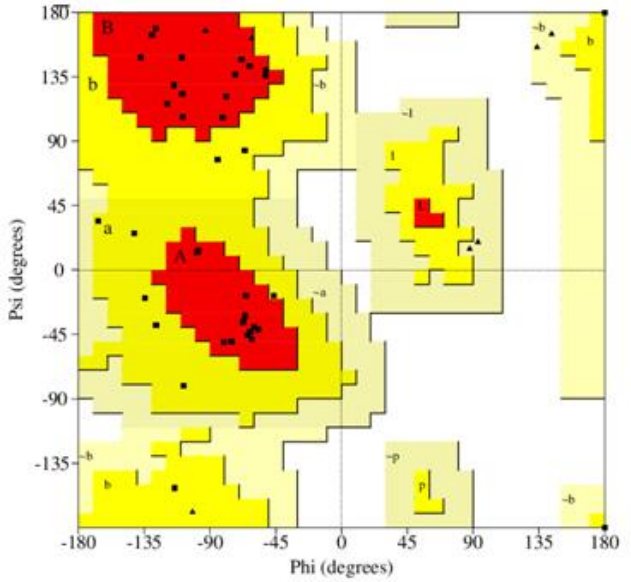

Figura 62 – Mapa de Ramachandran da estrutura predita pelo CReF, com 26 resíduos em regiões mais favoráveis (A, B e L) contabilizando 72.2%, com 10 resíduos em regiões permitidas (a,b,l,p) contabilizando 27.8%, com nenhum resíduo em região ainda aceitável (~a,~b,~l,~p) contabilizando 0% e nenhum resíduo em regiões não permitidas contabilizando 0%. Foram excluídas da contabilização dos resíduos de glicina e a prolina, com eles totalizariam os 47 resíduos.

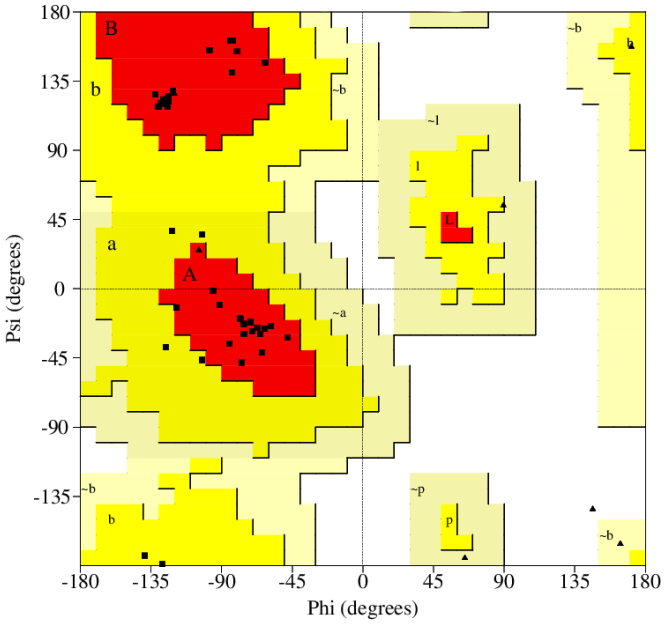

Figura 63 – Mapa de Ramachandran da estrutura refinada, com 30 resíduos em regiões mais favoráveis (A, B e L) contabilizando 83.3%, com 6 resíduos em regiões permitidas (a,b,l,p) contabilizando 16.7%, com nenhum resíduo em região ainda aceitável (~a,~b,~l,~p) contabilizando 0% e nenhum resíduo em regiões não permitidas contabilizando 0%. Foram excluídas da contabilização dos resíduos de glicina e a prolina, com eles totalizariam os 47 resíduos.

Desta forma é possível confirmar a partir dos mapas de Ramachandran anteriores, que após o refinamento da estrutura alvo 1GPT obteve-se uma grande quantidade de pontos nas regiões mais favoráveis (A, B e L) comparado com a estrutura predita pelo CReF e, de uma forma geral a estrutura refinada ocupou as mesmas regiões do mapa da estrutura experimental, com maior quantidade de resíduos em regiões mais favoráveis.

Com base na imagem gerada do refinamento comparada com a experimental e nos mapas de Ramachandran, conclui-se que, a proteína após ser submetida ao *workflow* de refinamento realizou o refinamento com base na predição do CReF.

9 CONSIDERAÇÕES FINAIS E DISCUSSÕES

O compartilhamento de dados, análises e recursos computacionais tem se tornado cada dia mais comum entre os grupos de pesquisas do meio científico. Os recursos computacionais, aplicados às atividades da pesquisa científica, possibilitam a colaboração entre os usuários, permitindo que eles se preocupem apenas com as atividades de pesquisa, sem se importar como as informações são armazenadas ou outro tipo de problema para executar suas tarefas.

O uso de experimentação apoiada por computação é uma realidade e, a simulação por *workflows* científicos, é importante na obtenção de resultados que possam contribuir para a criação da base de conhecimento com resultados mais precisos. Sendo assim, configura-se que é fundamental o uso de abordagem que permita ao pesquisador documentar e gerar estes estudos experimentais de forma organizada e correta. Um *workflow* representa os processos experimentais científicos.

Com o intuito de contribuir nesta área, surgiu a possibilidade de melhorar os resultados da predição 3D aproximada de proteínas geradas pelo método CReF. Para que isso fosse possível, foi utilizado o método de refinamento de proteínas, que tinha esse processo realizado manualmente, apresentando com isso dificuldades na usabilidade e execução do refinamento.

Os resultados aqui apresentados são decorrentes da experimentação computacional, envolvendo a proteína 1GAB (53 resíduos), que foi escolhida como um estudo de caso, e as proteínas 1YWJ (28 resíduos) e 1GPT (47 resíduos) para completar os testes de funcionalidade do modelo de *workflow* desenvolvido.

Estes demonstraram que o *workflow* executa corretamente o refinamento dos diferentes experimentos, de maneira simples e eficiente. Se os experimentos fossem realizados manualmente em *shell scripts*, ocasionaria ao usuário uma grande limitação na quantidade de experimentos que poderiam ser realizados, e teriam sidos executados de maneira bem mais lenta e complicada. Dessa forma, haveria a possibilidade de seus resultados não ter um final satisfatório, devido a erros que o usuário poderia cometer durante o processo de refinamento se feito de forma manual.

Com a utilização do modelo desenvolvido, esse processo pode ser executado de forma mais simples, onde a tarefa de cada uma das etapas ocorre em tempo de execução e não há a necessidade da manipulação dos *shell scripts*, com isso, o usuário abandona a execução manual dos experimentos do processo de refinamento e aplica a abordagem de *workflows* científicos.

A interface *web* desenvolvida para a análise dos resultados simplificou muito o trabalho dos usuários, que antes precisavam ir no diretório do seu sistema operacional e abrir cada diretório para visualizar os arquivos de resultados.

O Quadro 3 mostra como era o processo realizado de forma manual e após, ser automatizado.

Quadro 3 –Quadro comparativo do processo manual com o *workflow* desenvolvido

	Manual	***Workflow***
Entrada de dados	Executada através da linha de comando, onde informam-se todos os parametros separadamente, sem nenhum arquivo ou mecanismo de auxilio, que facilite a inserção dos parâmetros de entrada da dinâmica molecular.	Executada através da linha de comando, com o auxilio de arquivos modelos, de fácil entendimento para o usuário, facilitando a inserção dos parâmetros de entrada da dinâmica molecular.
Criação dos arquivos da fase de aquecimento	Executada de forma manual com ou sem auxilio de um arquivo modelo.	Executada de forma totalmente automatizada. A criação dos arquivos é realizada com base nos parâmetros de entrada do *workflow*.
Criação dos arquivos da fase de produção	Executada de forma manual com ou sem auxilio de um arquivo modelo.	Executada de forma totalmente automatizada. A criação dos arquivos é realizada com base nos parâmetros de entrada do *workflow*.
Criação do arquivo de restrições	Executada de forma manual com ou sem auxilio de um arquivo modelo.	Executada de forma totalmente automatizada. A criação do arquivo tem como base, as informações inseridas no aquivo modelo, informado no comando de entrada do *workflow*.
Execução da dinamica molecular	O usuário executa de forma manual todos os comandos necessários, do início ao fim, para realização do processo de dinâmica molecular.	Executada de forma semiautomática. Após o usuário conferir os arquivos gerados, basta que se execute, um arquivo que contém todos os comandos necessários, do inicio ao fim,

		para realização da dinâmica molecular.
Arquivos de Resultados	Os resultados são gerados em vários arquivos, sendo estes, organizados pelo usuário.	Os resultados são organizados em diferentes pastas, de acordo com suas funções dentro do processo de refinamento. Além da utilização da página *web* que facilita o acesso pelo usuário.
Criação dos arquivos de *Log's*	Se o usuário não manter uma organização, do que esta sendo executado em cada momento, não haverão registros de *log's*.	São armazenadas todas as informações necessárias para reprodução do mesmo processo de refinamento em outro momento.

Durante a pesquisa, encontrou-se outros trabalhos que se utilizam de *workflows* científicos com o objetivo de melhorar os resultados na área da bioinformática, visto que os experimentos em geral, são complexos de serem realizados por requererem uma grande quantidade de execuções e encadeamentos de programas. Percebeu-se, também que alguns deles não possuem *log's* para guardar as etapas da execução, porém na revisão tradicional da literatura técnica, não foi possível identificar uma abordagem de concepção de *workflow* científico para o refinamento de proteinas.

O trabalho desenvolvido por Machado *et al.*, (2007) assim como o modelo de *workflow* científico desenvolvido, é executado através de etapas e buscam apoiar computacionalmente para a realização dos experimentos, para automatizar o processo de execução dos mesmos. Antes, execuções desse tipo no LABIO, eram manuais ou com o auxílio de *scripts* básicos, que necessitavam serem modificados a cada execução de diferentes experimentos de docagem molecular. No método desenvolvido por Machado (2011) para a predição da conformação e da energia envolvida na interação entre ligantes e suas proteínas-alvo ou receptores, um dos maiores problemas era o tempo necessário para executá-lo tendo desenvolvido um método com o objetivo de contribuir para a seleção de conformações do receptor de forma a acelerar a execução dos experimentos, assim como o modelo de *workflow* desse trabalho, também objetivou a redução do tempo de execução dos experimentos da predição de proteínas.

No contexto de experimentos científicos computacionais não foi encontrado uma solução de apoio à execução automática e interativa de um *workflow* científico para o refinamento de proteinas. Justifica-se por não se ter usado sistemas gerenciadores de *workflow*

para geraração automática de refinamento como o Kepler e Taverna, estudados aqui, por estes não oferecerem compatibilidade com outros sistemas e ferramentas do LABIO.

Outro motivo por não fazer-se uso de sistemas gerenciadores, deve-se à quantidade de recursos que os mesmos apresentam e que não seriam utilizados, o que geraria, em algum momento uma perda de desempenho no tempo de processamento durante a realização do processo de refinamento. O modelo de *workflow* desenvolvido contempla o necessário para realizar o processo de refinamento da predição de estrutura realizada pelo método CReF.

Os experimentos foram realizados em dois computadores com configurações distintas, conforme segue:

1. – Intel Core i5-430M processor 2.26GHz with Turbo Boost Technology up to 2.53 GHz).
   – Memória: 4 GB of SDRAM DDR3 D at 1600 MHz.
   – HD: 320GB Hard Disk, SATA (7200 RPM).
   – Versão do software: AMBER 14.
   – Sistema operacional: Ubuntu / Linux.
2. – Processador: Intel® Processor Core™ i7-3770 (3.4GHz to 3.9GHz with Turbo Boost 2.0, 8 Threads, 8Mb Cache).
   – Memória: 16 GB of SDRAM DDR3 D at 1600 MHz.
   – HD: 2TB Hard Disk, SATA 3Gb/s (7200 RPM).
   – Versão do software: AMBER 14.
   – Sistema operacional: Ubuntu / Linux

Durante o processo de refinamento realizado com as três proteínas, utilizando o modelo de *workflow* desenvolvido com as configurações de *hardware* citadas, ficou evidente um ganho de tempo quatro vezes mais rápido nas execuções, quando comparado ao processo executado de forma manual. O tempo do processo manual de preparação dos arquivos, o qual não se utiliza de nenhum modelo, depende muito do conhecimento e da habilidade do usuário, dura em torno de 1 hora ou mais. Utilizando-se o *workflow*, e conforme as configurações do *hardware* da máquina, leva-se em média dez minutos, para a mesma preparação de arquivos. A execução da dinâmica molecular não é contabilizada, porque tempo utilizado será o mesmo para ambos os métodos.

Esta constatação do tempo nas execuções sempre será diferenciada, pois existem

variáveis a serem consideradas que influenciam na duração do processo completo do refinamento, como a configuração do *hardware*, o tamanho da proteína, que se for grande e executada em uma configuração baixa, o tempo de execução do refinamento será maior. Este ganho de tempo foi pressuposto que o usuário, ao executar de forma manual, não inseriu nenhum comando errado ou fora da sequência; caso isso ocorra o ganho de tempo seria maior. Pode-se dizer que conseguiu-se diminuir o tempo das execuções, e que a variação desse tempo depende de fatores externos ao *workflow*.

Os bons resultados obtidos durante o desenvolvimento desse trabalho foram fatores motivadores para considerar que a abordagem foi validada e o seu objetivo alcançado.

9.1 Contribuições

As principais contribuições desta pesquisa foram:

- Documentação da concepção do *workflow* científico desenvolvido, composta por um conjunto de etapas e parâmetros para que, a partir de uma proteína predita pelo método CReF, chegar-se ao seu refinamento.
- Execução automatizada do processo de refinamento de proteínas através de parâmetros, reduzindo consideravelmente o uso de *shells scripts*.
- Minimização de erros no processo de refinamento.
- Possibilidade de execução de mais experimentos em menos tempo.
- Resultados apresentados em uma interface web, o que facilita a visualização para as análises.

9.2 Trabalhos Futuros

A considerar pelos resultados motivadores dessa dissertação, trabalhos futuros poderão abordar os seguintes problemas:

- Execução de experimentos utilizando o *workflow* para outras classes de proteínas com estruturas mais complexas.
- Integração do *workflow* desenvolvido com o método CReF.

REFERÊNCIAS

Alberts B.; Bray D.; Jonhson A.; Lewis J.; Raff M.; Roberts K. “Fundamentos da Biologia Celular: Uma Introdução à Biologia Molecular da Célula”. Artmed, 2008, 2ª edição.

Alonso, H.; Bliznyuk, A. A.; Gready, J. E. “Combining docking and molecular dynamic simulations in drug design”. Med. Res. Rev., nº 26, 2006, pp. 531-568.

Altintas, I.; Berkley, C.; Jaeger, E.; Jones, M.; Ludascher, B.; Mock, S. "Kepler: An Extensible System for Design and Execution of Scientific Workflows". In: 16th International Conference on Scientific and Statistical Database Management, 2004, pp. 423-424.

Altschul, S. F.; Madden, T. L.; Schäffer, A. A.; Zhang, J.; Zhang, Z.; Miller, W.; Lipman, D. J. “Gapped BLAST And PSI-BLAST: A New Generation of Protein Database Search Programs”. Nucleic Acids Research, 1997.

Branden, C.; Tooze, J. “Introduction to Protein Structure”. Garlang Publishing Inc., 1998, 2ª edição.

Berg, J. M.; Tymoczko J. L.; Stryer, L. Trad. Moreira, A. J. M. da Silva e et al. “Bioquímica”. Guanabara Koogan, 2008, 6ª edição.

Berman, H. M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bath, T. N.; Weissig, H.; Shindyalov, I. N.; Bourne, P. E. “The Protein Data Bank”. Nucleic Acids Research, 2000.

Bruix, M.; Jimenez, M. A.; Santoro, J.; Gonzales, C.; Colilla, F. J.; Mendez, E. “Solution Structure of Gama 1- H And Gamma 1-P Thionins From Barley And Wheat Endosperm Determined by 1H-NMR: A Structural Motif Common to Toxic Arthropod Proteins”. Biochemistry, 1993, pp. 715-724.

Callahan, S. P.; Freire, J.; Santos, E.; Scheidegger, C. E.; Silva, C. T. Vo, H. T. “VisTrails: Visualization Meets Data Managemen”. In: ACM SIGMOD International Conference on Management of Data, 2006, pp. 745-747.

Case, D.A; Darden, T.A.; Cheatham III, T.E.; Simmerling, C.; Wang, B. e et al.: “AMBER 9: User Manual”. University of California, 2006.

Cattley S.; Arthur J.W.: “BioManager: The Use of Abioinformatics Web Application as a Teaching Tool in Undergraduate Bioinformatics”. Brief Bioinform, 2007.

Cavalcanti, M.C.; Targino, R.; Baião, F.; Rössle, S. C.; Bisch, P. M.; Pires, P. F.; Campos, M. L. M.; Mattoso, M. “Managing Structural Genomic Workflows Using Web Services”. Data & Knowledge Engineering, 2005, pp. 45-74.

Cheng, J.; Randall, A.; Sweredoski, P.; Baldi, M. “SCRATCH: A Protein Structure And Structural Feature Prediction Server”. Nucleic Acids Research, 2005, pp. 45-74.

Chothia, C; Lesk, A. M. “The Relation Between the Divergence of Sequence and Structures in Proteins”. The EMBO Journal, 1986, pp. 823-826.

Dall’Agno, K.C. da M. “Um Estudo Sobre a Predição da Estrutura 3D Aproximada de Proteínas Utilizando o Método CReF com Refinamento”. Dissertação de Mestrado. PUCRS - Pontifícia Universidade Católica do Rio Grande do Sul, 2012.

da Silveira, N. F. “Bioinformática Estrutural Aplicada ao Estudo de Proteínas Alvo do Genoma do Mycobacterium Tuberculosis”. Tese de Doutorado em Biofísica Molecular. UNESP - Universidade Estadual Paulista, 2005.

Deelman, E.; Gannon, D.; Shields, M.; Taylor, I. “Workflows and e-Science: An Overview of Workflow System Features and Capabilities”. Future Generation Computer Systems, 2009, pp. 528-540.

De Roure, D.; Goble, C. And Stevens, R. “The Design and Realisation of the myExperiment Virtual Research Environment for Social Sharing of Workflows”. Future Generation Computer Systems, 2009, pp. 561-567.

Dorn, M. “Uma Proposta para a Predição Computacional da Estrutura 3D Aproximada de Polipeptídeos com Redução do Espaço Conformacional Utilizando Análise de Intervalos”. Dissertação Mestrado em Ciência da Computação. PUCRS - Biblioteca Central, 2008.

Filho, S; Andrade, O.; Alencastro, R. B. “Protein Homology Modeling”. Química Nova, 2003, 16ª edição.

Floudas, C. A.; Fung, H. K.; Mcallister, S. R.; Mnnigmann, M.; Rajgaria, R.: “Advances in Protein Structure Prediction and de Novo Protein Design: A review”. Chem. Eng. Sci., 2006, pp. 966-988.

Foresman, J. B.; Frisch, E. “Exploring Chemistry with Electronic Structure Methods”. Gaussian, 1996, 2ª edição.

Fujitsu, BioMedCache 6.1. “User Guide”. Fujitsu Limited, 2003.

Gibas, G.; Jambeck, P. “Desenvolvendo Bioinformática”. Campus, 2001, 1ª edição.

Gil, Y.; Deelman, E.; Ellisman, M.; Fahringer, T.; Fox, G.; Gannon, D.; Goble, C.; Livny, M.; Moreau, L. e et al. “Examining the Challenges of Scientific Workflows”, 2007, pp. 24-32.

Ginalski, K.: “Comparative modeling for protein structure prediction”. Current Opinion in Structural Biology, 2006, pp. 172-177.

Goderis, A.; De Roure, D.; Goble, C.; Bhagat, J.; Cruickshank, D.; Fisher, P.; Michaelides, D.; Tanoh, F. “Discovering Scientific Workflows: The My Experiment Benchmarks”. IEEE Transactions on Automation Science and Engineering, 2008.

Gonçalves, F.A. “Nutrição Humana”. Fundação Calouste Gulbenkian, 1994, 2ª edição.

Hey, T.; Tansley, S.; TollE, K. “The Fourth Paradigm: Data-Intensive Scientific Discovery”. Microsoft Research, 2009, pp. 113-119.

Hull, D.; Wolstencroft, K.; Stevens, R.; Goble, C.; Pocock, M. R.; Li, P.; Oinn, T. “Taverna: A Tool for Building and Running Workflows of Services”. Ucleic Acids Research, 2006, pp. 1-18.

Johansson, M. U.; de Chateau, M.; Wikstrom, M.; Forsen, S.; Drakenberg, T.; Bjorck, L. “Solution Structure of the Albumin-Binding GA Module: A Versatile Bacterial Protein Domain”. Journal of Molecular Biology, 1997, pp. 859-865.

Jones, D. T.; Taylor, W. R.; Thornton, J. M. “A New Approach to Protein Fold Recognition. Nature”, 1992, pp. 86-89.

Junqueira L. C.; Carneiro J. “Biologia Celular e Molecular”. Guanabara Koogan, 2005, 8ª edição.

Kerian, K. Medicine Newbie, School of Medical Sciences, Kelantan, Malaysia. Capturado em: http://medicinenewbie.blogspot.com.br/2011/05/professional-i-exam-2-may-2011-part-4.html. Outubro 2015.

Khouri, C. M. B. “Modelos Escondidos de Markov para Classificação de Proteínas”. Dissertação de Mestrado em Ciências da Computação. UFPE - Universidade Federal de Pernambuco, 2002.

Laskowski, R. A.; MacArthur, M. W.; Moss, D. S.; Thornton, J. M. “Procheck: A Program to Check the Stereochemical Quality of Protein Structures”. Journal of Applied Crystallography, 1993, pp. 283-291.

Leach A. R. “Molecular Modelling: Principles and Applications”. Pearson Education Limited, 2001, 2ª edição.

Lee, J.; Kim, S. Y.; Joo, K.; Kim, I. “Prediction of Protein Tertiary Structure Using Profesy, a Novel Method Based on Fragment Assembly and Conformational Space Annealing”. Proteins: Structure and Bioinformatics, 2004, pp. 704-714.

Lehninger, A. L.; Nelson, D. L.; Cox, M. M. “Princípios da Bioquímica”. Sarvier, 2005, 4ª edição.

Lemer, C.; Rooman, M.J. e Wodak, S.J. “Protein Structure Prediction by Threading Methods: Evaluation of Current Techniques”. Proteins: Structure and Genetics, 1995.

Lesk, A. M. ‘Introduction to Protein Architecture”. Oxford University Press, 2001.

Lesk, A. M. “Introdução à Bioinformática”. Tradução de Ardala Elisa Breda Andrade e et al. (LABIO/FACIN/PUCRS), Artmed, 2008, 2ª edição.

Levinthal C. “Mossbauer Spectroscopy in Biological Systems”. In: Meeting Held at Allerton House. University of Illinois, 1969, pp. 22-24.

Ludascher, B.; Altintas, I.; Berkley, C.; Higgins, D.; Jaeger-Frank, E.; Jones, M.; Lee, E.; Tao, J.; Zhao, Y. “Scientific Workflow Management and the Kepler System”. Concurrency and Computation: Practice and Experience, 2006.

Ludascher, B.; Weske, M.; McPhillips, T.; Bowers, S. “Business Process Management”. Springer, 2009.

Luscombe, N. M.; Greenbaum, D.; Gerstein, M. “What is Bioinformatics? A Proposed Definition and Overview of the Field”. Methods of Information in Medicine, 2001, pp. 346-358.

Machado, K. S.; Schroeder, E. K.; Ruiz, D D.; Souza, O.N. “Automating Molecular Docking With Explicit Receptor Flexibility Using Scientific Workflows”. In: Brazilian Symposium on Bioinformatics. Lecture Notes in Computer Science, 2007.

Machado, K. S. “Seleção Eficiente de Conformações de Receptor Flexível em Simulações de Docagem Molecular”. Tese de Doutorado em Computação. PUCRS - Faculdade de Informática, 2011.

Marinho, A.; Murta, L.; Werner, C.; Braganholo, V.; Cruz, S. M. S. D.; Mattoso, M.: "A Strategy for Provenance Gathering in Distributed Scientific Workflows". In: IEEE International Workshop on Scientific Workflows, 2009, pp 344-347.

Marti-Renom M.A.; Stuart A.C.; Fiser A.; Sanchez R.; Melo F.; Sali, A.: “Comparative Protein Structure Modeling of Genes and Genomes”. Rev. Anual Review of Biophysics and Biomolecular Structure, 2000, pp. 291-325.

Mattoso, M.; Werner, C.; Travassos, G.; Braganholo, V.; Murta, L. “Gerenciando Experimentos Científicos em Larga Escala”. In: SEMISH - CSBC, 2008, pp.121-135.

Morris, A. L.; MacArthur, M. W.; Hutchinson, E. G.; Thornton, J. M. “Stereochemical Quality of Protein Structure Coordinates”. Structure, Function and Bioinformatics, 1992, pp. 345-364.

Mount, W. D. “Bioinformatics. Sequence and Genome Analysis”. Cold Spring Harbor Laboratory Press, 2001.

Oinn, T.; Li, P.; Kell, D. B.; Goble, C.; Goderis, A.; Greenwood, M.; Hull, D.; Stevens, R.; Turi, D.; Zhao, J. “Taverna/myGrid: Aligning a Workflow System with the Life Sciences Community”, Springer, 2007, pp. 300-319.

Oliva, G. “Bioinformática: Perspectivas na Medicina”. Gazeta Médica da Bahia, 2008, pp. 52-58.

Oliveira, F.; Murta, L.; Werner, C.; Mattoso, M. “Using Provenance to Improve Workflow Design”. In: 2nd International Provenance and Annotation Workshop - IPAW, 2008, pp. 136-143.

Osguthorpe, D. “Ab Initio Protein Folding”. Current Opinion In Structural Biology, 2000, pp. 146-152.

Pauling, L.; Corey, R.; Branson, H. “The Structure of Proteins: Two Hydrogenbonded Helical Configurations of the Polypeptide Chain”. PubMed, 1951, pp. 205-234.

Pedersen J. T.; Moult, J.: “Genetic Algorithms for Protein Structure Prediction”. Current Opinion In Structural Biology, 1996, pp. 227-259.

Pearlman D. A.; Case D. A.; Caldwell J. W.; Ross W. S.; Cheatman, T. E.; DeBolt, S.; Ferguson, D.; Seibel, G.; Kollman, P. “Amber, a Package of Computer Programs for Applying Molecular Mechanics, Normal Mode Analysis, Molecular Dynamics and Free Energy Calculations to Simulate the Structural and Energetic Properties of Molecules”. Computer Physics Communications, 1995, pp. 1-41.

Pressman, R. S. “Software Engineering: A Practitioner's Approach”. Makron Books, 2004, 6ª edição.

Prosdocimi, F.; Cerqueira, G. C.; Binneck, E.; Silva, A. F.; Reis, A. N.; Junqueira, A. C. M.; Santos, A. C. F.; Nbani, A.; Wust, C. I.; Filho, F. C.; Kessedjian, J. L.; Petretski, J. H.; Camargo, L. P.; Ferreira, R. G. M.; Lima, R. P.; Pereira, R. M.; Jardim, S.; Sampaio, V. S.; Flatschart, A. V. F. “Bioinformática: Manual do Usuário”. Biotecnologia Ciência e Desenvolvimento, 2002, pp. 12-25.

Ptolemy Project. Capturado em: http://ptolemy.eecs.berkeley.edu/, abril 2015.

Santos, D. F.: “Alinhamento Múltiplo de Proteínas Via Algoritmo Genético Baseados em Tipos Abstratos de Dados”. Dissertação de Mestrado. UFAL - Universidade Federal de Alagoas, 2008, pp. 1-119.

Saqi, M. A.; Russell, R. B.; Sternberg, M. J. “Misleading Local Sequence Alignments: Implications for Comparative Protein Modelling”. National Center for Biotechnology Information, 1998, pp. 627-630.

Silva, F. N., Cavalcanti, M. C., Dávila, A. M. R. “In Services: Data Management for In Silico Workflows”. In: 17th International Conference on Database and Expert Systems Applications, 2006, pp. 206-210.

Silva, V. B.; Silva, C. H. T. P. “Modelagem Molecular de Proteínas-Alvo por Homologia Estrutural. Revista Eletrônica de Farmácia, 2011, pp. 15-26.

Sippl, M.J. “Recognition of errors in three-dimentional structrures of proteins. Proteins”, 1993, pp. 355-362.

Simons, K.; Ruczinki, I.; Kooperberg, C.; Fox, B.; Bystroff, C.; Baker, D. “Improved Recognition of Native-Like Structures Using a Combination of Sequence-Dependent and Sequence-Independent Features of Proteins”, 1999, pp. 82-95.

Sommerville, I. “Engenharia de Software”. Pearson, 2007, 8ª edição.

Stryer, L.; Berg, J. M., Tymoczko, J.L. “Biochemistry”, International Student Edition, 1988, 3º edição.

Tramontano, A. “Protein Structure Prediction”. Johnwiley and Sons Inc., 2006, 1ª edição.

Travassos, G. H.; Barros, M. O. “Contributions of In Virtuo and In Silico Experiments for the Future of Empirical Studies in Software Engineering”. In: 2nd Workshop on Empirical Software Engineering the Future of Empirical Studies in Software Engineering, Roma, 2003, pp. 117-130.

Voet, D.; Voet, J. G.: “Bioquímica”. Artmed, 2006, 3ª edição.

Wainer, J. M.; Weske, G.; Vossen and Medeiros, C. B. “Scientific Workflow Systems”. In: NSF Workshop on Workflow and Process Automation Information Systems, Atenas, Grecia, 1996.

Weske, M.; Vossen, G.; Medeiros C. “Scientific Workflow Management: WASA Architecture and Applications”. In: 6th DEXA Conference, London, Inglaterra, 1995.

Westhead, D. R.; Parish, J. H.; Twyman, R. M. “Bioinformatics. InstantNotes”. Bios Scientific Publishers Limited, Oxford, 2002, 253p.

Witten, I. H.; Frank, E.: “Data Mining: Practical Machine Learning Tools and Techniques”. Morgan Kaufmann Series in Data Management Systems, 2005, 2ª edição.

APÊNDICE A – DOCUMENTO DE REQUISITOS

UM MODELO DE *WORKFLOW* CIENTÍFICO PARA O REFINAMENTO DA ESTRUTURA 3D APROXIMADA DE PROTEÍNAS

Levantamento de Requisitos

Versão: 1.0

Data: 17/04/2015

Localização: Labio- PUCRS

Histórico de revisões do modelo

Versão (XX.YY)	Data (DD/MMM/YYYY)	Autor	Descrição	Localização
01.01	17/ABR/2015	Leonardo	Versão inicial	
01.02	09/MAI/2015	Leonardo	Formatação do doc. e revisão para fechar uma versão.	
01.03	25/MAI/2015	Leonardo	Formato final	
01.04	27/MAI/2007	Leonardo	Versão revisada	

Índice

1. **INTRODUÇÃO** **4**
2. **VISÃO GERAL DO PRODUTO** **7**
3. **RESTRIÇÕES** **8**
4. **REQUISITOS FUNCIONAIS** **9**
5. **REQUISITOS NÃO FUNCIONAIS** **11**

1. Introdução

1.1. Propósito

Este documento especifica os requisitos de software que serão utilizados para a construção do modelo de workflow científico para o refinamento da estrutura 3D aproximada de proteínas, fornecendo aos desenvolvedores as informações necessárias do projeto e sua implementação, assim como para a realização dos testes e homologação do workflow.

O documento especifica todos os requisitos funcionais e não funcionais do workflow e foi preparado levando-se em conta o protocolo de refinamento de proteínas.

1.2. Público Alvo

Este documento se destina aos arquitetos de software, engenheiros de software e testadores.

1.3. Escopo

Este documento realiza a elicitação de requisitos do modelo de workflow científico para o refinamento da estrutura 3D aproximada de proteínas.

1.4 Benefícios esperados do produto

Neste tópico serão apresentados todos os benefícios esperados com a produção do projeto.

Número		Valor para o cliente
1	Refinamento de uma proteína predita pelo método CReF.	Essencial
2	Automatização do processo de refinamento de proteínas.	Importante
3	Ambiente de visualização dos resultados.	Importante
4	Economia de tempo no processo de refinamento de proteínas.	Importante

1.5 Problema identificado

O protocolo de refinamento realizado manualmente com a utilização *de shell scripts* ocasiona ao usuário uma limitação para realizar uma grande quantidade de experimentos. Isso ocorre porque a configuração manual do protocolo é muito trabalhosa e envolve peculiaridades no detalhamento da configuração, principalmente em resíduos de aminoácidos do polipeptídeo predito pelo CReF que irá receber restrições de movimentação em ângulos diedros. Dessa forma, têm-se problemas para definir a ordem correta em que as etapas devem ser executadas. É possível executar o processo utilizando parâmetros diferentes, monitorar a execução do mesmo, gerar análises de resultados, o que torna este cenário sujeito a falhas e improdutivo, especialmente em se tratando de experimentos complexos, que envolvem muitos programas e grande quantidade de dados.

1.6 Definições e Abreviações

A correta interpretação deste documento exige o conhecimento de algumas convenções e termos específicos, que são descritos a seguir.

1.6.1 Definições

Nome		Valor para o cliente
Feature	Característica principal do sistema	**FEAT**
Scenario	Característica secundária de uma Feature	**SCE**
Requisitos Funcionais	Correspondem à listagem de todas as coisas que o sistema deve fazer	**RF**
Requisitos Não-Funcionais	São restrições que se coloca sobre como o sistema deve realizar seus requisitos funcionais	**NF**

1.6.2 Abreviações

Abreviação	
AMBER	Assisted Model Building with Energy Refinement
CReF	Central Residue Fragment-based Method

1.7 Referências

Esta seção é destinada à descrição das referências utilizadas pelo documento, como por exemplo, URLs e livros. Ver exemplo a seguir:

[1] Marinho, A, ProvManagé: Uma abordagem para gerenciamento de proveniência e monitoramento de workflows científicos; Proposta de Dissertação de Mestrado, COPPE/UFRJ, 2009.

1.8 Visão geral do documento

- **Na seção 2** apresenta uma visão geral do sistema, caracterizando qual é o seu escopo e descrevendo seus usuários.
- **Na seção 3** especifica as restrições dos requisitos levantados.
- **Na seção 4** são enumerados todos os requisitos funcionais, e
- **Na seção 5** os não-funcionais do sistema.

2. Visão Geral do Produto

Nas últimas décadas, com o avanço da computação, os experimentos científicos passaram a utilizar ferramentas computacionais para facilitar a sua execução (Marinho, 2009). Com este cenário a experimentação começa a sofrer uma transformação, os experimentos passaram a possuir uma grande quantidade de programas e a sua manipulação tornou-se mais complexa. O objetivo de facilitar a abstração de um experimento, permitindo que se obtenha uma estruturada de atividades que visando um determinado resultado, chamados então de *workflows* científicos.

Nesse contexto, o principal objetivo desse trabalho consiste no desenvolvimento de um *workflow* científico que auxilie os pesquisadores (ou usuários) em tarefas de experimentos *in silico*, com intuíto de automatizar ao máximo a configuração de todos os requisitos necessários do AMBER 14 a fim de realizar o processo de dinâmica molecular, reduzindo-se o tempo necessário de execução.

2.1 Descrição dos usuários

Existem basicamente 3 tipos de usuários que podem ter acesso às funcionalidades do sistema:

Número	Característica	Definição
1	Cientistas, Bioinformatas, Pesquisadores	Usuário Final
2	Pesquisadores	Autores do Documento de requisitos e também pesquisadores da comunidade científica
3	Profissionais da Computação	Programadores, projetistas, designers que utilizem o Documento de Requisitos para desenvolver softwares para Bioinformática

3. Restrições

O propósito desta seção é apresentar a descrição de todas as restrições de operação e propriedades gerais.

3.1 Restrições de Ambiente

- Restrição 1: Este trabalho necessita de um sistema operacional LINUX para ser executado.
- Restrição 2: Todos os pacotes de AMBER 14 devem estar instalados corretamente no sistema operacional LINUX para que sejam executadas todas as análises de resultados.
- Restrição 3: Sistema deve ser desenvolvido na linguagem *Python*.

3.2 Restrições de Usabilidade

- Restrição 1: O ambiente *web* desenvolvido para visualização dos resultados deve levar em consideração critérios de usabilidade.

3.3 Restrições de Legais

- Restrição 1: Os produtos desenvolvidos a partir desse documento deverão estar de acordo com as leis e regulamentos vigentes na época de seu desenvolvimento.

3.4 Restrições de Segurança

- Restrição 1: O projeto desenvolvido deve estar de acordo com questões de segurança e tolerância a erros, levando-se em consideração conforme descritos os requisitos.

4. Requisitos Funcionais

São descritos os requisitos funcionais do projeto a ser implementado. Para melhor clareza, as funcionalidades são agrupadas e descritas nas subseções a seguir.

4.1. RF001 - Refinar Proteína

Neste requisito é realizado o refinamento da proteína predita pelo método CReF. Este refinamento resume-se basicamente em 5 etapas: construção do comando inicial, criação dos arquivos, criação do arquivo de restrições, início da simulação da dinâmica molecular e por fim a análise automática dos resultados obtidos através da trajetória da dinâmica molecular.

4.2. RF002 – Login de Acesso

Neste requisito é realizada a autenticação do usuário para acesso a interface de visualização dos resultados do refinamento. Para realizar a autenticação serão informados o nome de usuário e a senha previamente cadastradas pelo administrador.

4.3. RF003 – Cadastrar Usuários

Neste requisito é apresentada uma tela para cadastro de novo usuário para acesso a interface de visualização de resultados. Para realizar o cadastro de um novo usuário deve ser informado o nome do usuário, o e-mail, o nome de usuário que será utilizado para efetuar o login, a senha de acesso e em quais itens o usuário terá permissão. Junto com o cadastro será apresentada a listagem de todos os usuários já cadastrados para que seja possível alterar seus dados.

4.4. RF004 – Manter Perfil

Neste requisito é apresentada uma tela para o usuário alterar todos os seus dados cadastrados pelo administrador. Estes dados estão descritos no RF003.

4.5. RF005 – Listar Resultados

Neste requisito permite ao usuário acessar todos os arquivos gerados pelo workflow de um determinado refinamento. Os arquivos serão listados separados por tipos que

o identificam dentro do processo de refinamento. Cada tipo de arquivo está agrupado em uma mesma tabela, sendo que na primeira coluna informa o nome do arquivo e na segunda coluna o ícone representado por um círculo com uma seta no centro, para o usuário salvar o arquivo em sua máquina.

5. Requisitos Não Funcionais

Descreve os requisitos não-funcionais do sistema. Os requisitos são descritos nas próximas subseções. Tais como:

5.1 RNF001 - Segurança

Somente o administrador do sistema ou usuários previamente autorizados poderão realizar configurações ou alterações no método de refinamento, assim como liberar o acesso ao ambiente de consulta de resultados.

5.2 RNF002 - Desempenho

Requisito essencial ao projeto, embora seja considerado por corresponder a um fator de qualidade de software. Usuários da área de bioinformática tem necessidade de resultados rápidos em suas pesquisas.

5.3 RNF003 - Usabilidade

Interface *web* responsável por apresentar os resultados para o usuário deve contemplar os critérios mínimos de usabilidade facilitado assim a interação do usuário com o ambiente desenvolvido.

5.4 RNF004 - Confiabilidade

O projeto deve estar disponível sempre que necessário para o usuário. Por não se tratar de um sistema crítico, não há necessidade de que esteja no ar por tempo integral.

5.5 RNF005 - Padrões

O sistema deve ser desenvolvido em HTML/CSS para o ambiente de visualização de resultados devendo ser compatível com so *browsers* de mercado, e na linguagem *Python* para o desenvolvimento funcional da ferramenta.

5.6 RNF006 - Operacionais

Como se trata de uma aplicação que será executada através de *shell scripts* ela deve ser configurada em um sistema operacional LINUX com o pacote do AMBER 14 instalado corretamente. O hardware para execução irá interferir apenas no tempo de resposta em que o sistema irá processar o refinamento, não sendo impedimento de execução em maquinas com menor capacidade de processamento.