UNIVERSIDADE DE SÃO PAULO – USP
ESCOLA DE ENGENHARIA DE SÃO CARLOS / FACULDADE DE MEDICINA DE
RIBEIRÃO PRETO / INSTITUTO DE QUÍMICA DE SÃO CARLOS
PROGRAMA DE PÓS-GRADUAÇÃO INTERUNIDADES BIOENGENHARIA

Alexandre Defelicibus

**Koala: sistema para integração de
métodos de predição e análise de
estruturas de proteína**

Alexandre Defelicibus

**Koala: sistema para integração de
métodos de predição e análise de
estruturas de proteína**

Dissertação de mestrado apresentada ao Programa de Pós-Graduação Interunidades Bioengenharia - Escola de Engenharia de São Carlos / Faculdade de Medicina de Ribeirão Preto / Instituto de Química de São Carlos da Universidade de São Paulo como parte dos requisitos para a obtenção do título de Mestre em Ciências.

Área de concentração: Bioengenharia

Orientador: Prof. Dr. Alexandre Cláudio Botazzo Delbem

São Carlos

2016

Trata-se da versão corrigida da dissertação.

AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTE TRABALHO,
POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS
DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

D313k Defelicibus, Alexandre
Koala: sistema para integração de métodos de
predição e análise de estruturas de proteínas /
Alexandre Defelicibus; orientador Alexandre Cláudio
Botazzo Delbem. São Carlos, 2015.

Dissertação (Mestrado) - Programa de Pós-Graduação
Interunidades Bioengenharia e Área de Concentração em
Bioengenharia -- Escola de Engenharia de São Carlos;
Faculdade de Medicina de Ribeirão Preto; Instituto de
Química de São Carlos, da Universidade de São Paulo,
2015.

1. fluxo de trabalho. 2. PSP. 3. Galaxy. 4.
reprodutibilidade. 5. Koala. I. Título.

Programa de Pós-Graduação
Interunidades Bioengenharia
EESC / FMRP / IQSC

ALEXANDRE DEFELICIBUS

Título: "Koala: sistema para integração de métodos de predição e análise de estruturas de proteína"

DISSERTAÇÃO APRESENTADA AO PROGRAMA DE PÓS - GRADUAÇÃO INTERUNIDADES EM BIOENGENHARIA - EESC/FMRP/IQSC DA UNIVERSIDADE DE SÃO PAULO PARA OBTENÇÃO DO TÍTULO DE MESTRE EM CIÊNCIAS NA ÁREA DE BIOENGENHARIA.

Data da defesa: 25/01/2016

Prof. Dr. Alexandre Cláudio Botazzo Delbem (Orientador)

Instituto de Ciências Matemáticas e de Computação/ICMC - USP

Assinatura:

Resultado:

APROVADO

Prof. Dr. Carlos Dias Maciel

Escola de Engenharia de São Carlos/EESC-USP

Assinatura:

Resultado:

APROVADO

Prof. Dr. Paulo Sérgio Lopes de Oliveira

Centro Nacional de Pesquisa em Energia e Materiais – Laboratório Nacional de Biociências

Assinatura:

Resultado:

APROVADO

Homologado em: ____/____/____

Agradecimentos

Primeiramente gostaria de agradecer a Deus por me abençoar com uma família maravilhosa e amigos especiais à minha volta, e por me dar a oportunidade de desenvolver esse trabalho.

Agradeço a toda a minha família pelo apoio oferecido desde o início do mestrado, me incentivando e me encorajando, especialmente meus pais Celso e Sônia.

Agradeço ao meu professor e orientador Dr. Alexandre Cláudio Botazzo Delbem, por me orientar neste trabalho, auxiliando-me em todas as fases desse projeto, além de todo aprendizado sobre algoritmos evolutivos, pesquisa e escrita.

Agradeço ao professor Dr. Rodrigo A. Faccioli, por ter contribuído muito neste trabalho, e por todos os ensinamentos na área de bioinformática e proteínas.

Agradeço ao pessoal do LCR-ICMC, com quem pude dividir algumas dúvidas sobre proteína; ao pessoal da secretaria do programa de pós-graduação pelo apoio e orientação; ao Tiago do CEFAP-USP por contribuições pontuais a respeito de estruturas de proteína.

Agradeço também a minha namorada Érica, pelo apoio e paciência nos dias difíceis dessa pesquisa, foi muito importante para mim.

À CAPES, pelo fomento deste trabalho.

Enfim, agradeço a todos que contribuíram pela elaboração deste trabalho e não foram citados aqui, não diminuindo de forma alguma o valor de sua contribuição.

Resumo

Defelicibus, Alexandre. **Koala: sistema para integração de métodos de predição e análise de estruturas de proteína**, 2016. 156f. Dissertação de mestrado – Programa de Pós-Graduação Interunidades Bioengenharia - Escola de Engenharia de São Carlos/ Faculdade de Medicina de Ribeirão Preto/ Instituto de Química de São Carlos, Universidade de São Paulo, São Carlos, 2016.

A Biologia Computacional tem desenvolvido algoritmos aplicados a problemas relevantes da Biologia. Um desses problemas é a *Protein Structure Prediction* (PSP). Vários métodos têm sido desenvolvidos na literatura para lidar com esse problema. Porém a reprodução de resultados e a comparação dos mesmos não têm sido uma tarefa fácil. Nesse sentido, o *Critical Assessment of protein Structure Prediction* (CASP), busca entre seus objetivos, realizar tais comparações. Além disso, os sistemas desenvolvidos para esse problema em geral não possuem interface amigável, não favorecendo o uso por não especialistas da computação. Buscando reduzir essas dificuldades, este trabalho propões o Koala, um sistema baseado em uma plataforma web, que integra vários métodos de predição e análises de estruturas de proteínas, possibilitando a execução de experimentos complexos com o uso de fluxos de trabalhos. Os métodos de predição disponíveis podem ser integrados para a realização de análises dos resultados, usando as métricas RMSD, GDT-TS ou TM-Score. Além disso, o método *Sort by front dominance* (baseado no critério de optimalidade de Pareto), proposto nesse trabalho, consegue avaliar predições sem uma estrutura de referência. Os resultados obtidos, usando proteínas alvo de artigos recentes e do CASP11, indicam que o Koala tem capacidade de realizar um conjunto relativamente grande de experimentos estruturados, beneficiando a determinação de melhores estruturas de proteínas, bem como o desenvolvimento de novas abordagens para predição e análise por meio de fluxos de trabalho.

Palavras-chave: fluxo de trabalho, PSP, Galaxy, reprodutibilidade, Koala.

Abstract

Defelicibus, Alexandre. **Koala: system for integration of methods for protein structures prediction and analysis**, 2016. 156f. Master Thesis – Programa de Pós-Graduação Interunidades Bioengenharia - Escola de Engenharia de São Carlos/ Faculdade de Medicina de Ribeirão Preto/ Instituto de Química de São Carlos, University of São Paulo, São Carlos, 2016.

Computational Biology has developed algorithms applied to relevant problems from Biology. One of these problems is Protein Structure Prediction (PSP). Several methods have been developed on the literature to deal with this problem. However, the reproduction of results and the comparison of the methods have not been an easy task. Accordingly, the Critical Assessment of protein Structure Prediction (CASP), has among his objectives, perform these comparisons. Besides, the developed systems for this problem have low usability, not benefiting the investigation of various methods by non experts. In order to minimize those difficulties, this project proposes Koala, a web-based system that integrates several algorithms applied to PSP and analysis, allowing the execution of complex experiments by using workflows. The prediction methods can be integrated to perform some analysis of the results, by using the RMSD, GDT-TS and TM-Score metrics. Moreover, the Sort by front dominance method (based on the criterion of Pareto optimalidad), proposed on this work, can evaluate predictions with no reference structure. The results obtained, using target proteins from recent articles and CASP11, indicate that Koala has the capability to execute a relatively large set of organized experiments, benefiting determining of better protein structures, as well as the development of new approaches for prediction and analysis through workflows.

Keywords: workflow, PSP, Galaxy, reproducibility, Koala.

Lista de figuras

Figura 1	Estrutura básica de um aminoácido.	30
Figura 2	Estrutura das cadeias laterais R dos aminoácidos	31
Figura 3	Ligação peptídica	32
Figura 4	Estrutura Primária	32
Figura 5	Estrutura Hélice- α	33
Figura 6	Estrutura Folha- β	34
Figura 7	Estruturas β -turn e β -hairpin	35
Figura 8	Representação da estrutura terciária de uma proteína (PDB 1VII) . . .	35
Figura 9	Representação computacional do problema de predição.	37
Figura 10	Diagrama do ProtPred-Gromacs	40
Figura 11	Biblioteca do ProtPred-EDA	42
Figura 12	Métodos propostos no ProtPred-EDA	44
Figura 13	Fluxograma do AEMMT	45
Figura 14	Subpopulações usadas no AEMMT	46
Figura 15	Geospiza GeneSifter fluxo experimental	50
Figura 16	Interface de usuário do ChIPseeqer	51
Figura 17	Arquitetura do ABrowse	52
Figura 18	Arquitetura padrão do SeqAdapt	53
Figura 19	Fluxograma do LabKey Server	54
Figura 20	Tela principal do <i>Galaxy</i> sendo executado pela <i>internet</i>	57
Figura 21	Páginas do Galaxy	60
Figura 22	Galaxy Workspace	61
Figura 23	Galaxy Workflow	62
Figura 24	Número de membros das listas de emails do Galaxy	63
Figura 25	Número de usuários registrados no Galaxy	63
Figura 26	Esquema geral do do Koala	66

Figura 27	Modelo de interação usuário-computador.	67
Figura 28	Processo de desenvolvimento de projeto de interfaces.	68
Figura 29	Exemplo de Fronteira de Pareto	70
Figura 30	Diagrama da <i>cloud computing</i>	73
Figura 31	Servidor Koala.	74
Figura 32	Configuração do Pulsar na <i>cloud</i>	75
Figura 33	Processo de compartilhamento de jobs	76
Figura 34	Módulo <i>Download PDB</i>	78
Figura 35	Módulo <i>Rename Atoms</i>	78
Figura 36	Módulo <i>Check PDB Structures</i>	79
Figura 37	Módulo <i>Split PDB</i>	80
Figura 38	Módulo <i>Merge PDB</i>	81
Figura 39	Ferramenta 2PG <i>Build Conformation.</i>	82
Figura 40	Ferramenta 2PG <i>Mono.</i>	83
Figura 41	Ferramenta 2PG <i>Random Algorithm to PSP.</i>	84
Figura 42	Ferramenta 2PG <i>Monte Carlo Metropolis.</i>	85
Figura 43	Ferramenta 2PG NSGA-II.	86
Figura 44	Módulo ProtPred-EDA RW	87
Figura 45	Módulo ProtPred-EDA MC	89
Figura 46	Módulo ProtPred-EDA EDA	90
Figura 47	Módulo ProtPred-EDA rBOA	91
Figura 48	Módulo ProtPred-EDA GA	92
Figura 49	Módulo ProtPred-EDA DE	93
Figura 50	Módulo MEAMT <i>Build Population.</i>	94
Figura 51	Módulo MEAMT.	95
Figura 52	Ferramenta 2PG <i>Sort by Front Dominance</i>	96
Figura 53	Ferramenta <i>Dominance Ranking</i>	98
Figura 54	Ferramenta 2PG <i>Sort Method by Front Dominance</i>	99
Figura 55	Ferramenta <i>Calculate RMSD</i>	100
Figura 56	Ferramenta <i>Calculate TM-Score</i>	101
Figura 57	Ferramenta <i>Calculate GDT-TS</i>	101
Figura 58	Estruturas terciárias das proteínas nativas utilizadas nos experimentos	104
Figura 59	Estruturas terciárias da proteína 2EVQ preditas	109
Figura 60	Estruturas terciárias da proteína 1A11 preditas	110
Figura 61	Estruturas terciárias da proteína 1VII preditas	111
Figura 62	Estruturas terciárias da proteína 1ERC preditas	112
Figura 63	Gráfico da Fronteira de Pareto da proteína 2EVQ	118
Figura 64	Gráfico da Fronteira de Pareto da proteína 1A11	119
Figura 65	Gráfico da Fronteira de Pareto da proteína 1VII	120

Figura 66	Gráfico da Fronteira de Pareto da proteína 1ERC	121
Figura 67	Gráfico de Ramachandran para a solução do <i>Sort</i> para a proteína 2EVQ	123
Figura 68	Gráfico de Ramachandran para a solução com menor RMSD para a proteína 2EVQ	124
Figura 69	Comparação entre as estruturas de melhor <i>ranking</i> e com menor RMSD para a proteína 2EVQ	124
Figura 70	Gráfico de Ramachandran para a solução do <i>Sort</i> para a proteína 1A11	125
Figura 71	Gráfico de Ramachandran para a solução com menor RMSD para a proteína 1A11	126
Figura 72	Comparação entre as estruturas de melhor <i>ranking</i> e com menor RMSD para a proteína 1A11	126
Figura 73	Gráfico de Ramachandran para a solução do <i>Sort</i> para a proteína 1VII	127
Figura 74	Gráfico de Ramachandran para a solução com menor RMSD para a proteína 1VII	128
Figura 75	Comparação entre as estruturas de melhor <i>ranking</i> e com menor RMSD para a proteína 1VII	128
Figura 76	Gráfico de Ramachandran para a solução do <i>Sort</i> para a proteína 1ERC	129
Figura 77	Gráfico de Ramachandran para a solução com menor RMSD para a proteína 1ERC	130
Figura 78	Comparação entre as estruturas de melhor <i>ranking</i> e com menor RMSD para a proteína 1ERC	130
Figura 79	Histograma das 10 melhores soluções para os alvos do CASP	133
Figura 80	Histograma com todas as soluções de todos os alvos	134
Figura 81	Esquema geral de execução do <i>workflow Comparison of predictions with reference</i>	136
Figura 82	Esquema geral de execução do <i>workflow Ranking of solutions by Sort by Front Dominance</i>	136
Figura 83	<i>Workbench</i> do <i>workflow</i> para predição e análise com referência	137
Figura 84	<i>Workbench</i> do <i>workflow</i> para predição e análise sem referência	138
Figura 85	Execução do <i>workflow</i> com análise com referência	139
Figura 86	Execução do <i>workflow</i> com análise sem referência	140
Figura 87	Page "Predições e análises no Koala" pública	142

Lista de tabelas

Tabela 1	Relação dos vinte aminoácidos padrões nas proteínas	30
Tabela 2	Relação dos métodos de predição.	47
Tabela 3	Relação de <i>frameworks</i> encontrados na literatura	55
Tabela 4	Proteínas utilizadas nos experimentos	104
Tabela 5	Parâmetros para os algoritmos do 2PG	106
Tabela 6	Parâmetros para os algoritmos do ProtPred-EDA	106
Tabela 7	Parâmetros para o algoritmo MEAMT	106
Tabela 8	Resumo dos parâmetros para os experimentos	107
Tabela 9	Quantidade de soluções geradas para a proteína 2EVQ	108
Tabela 10	Quantidade de soluções geradas para a proteína 1A11	108
Tabela 11	Quantidade de soluções geradas para a proteína 1VII	108
Tabela 12	Quantidade de soluções geradas para a proteína 1ERC	113
Tabela 13	<i>Ranking</i> dos 10 melhores RMSD calculados para a proteína 2EVQ . . .	114
Tabela 14	<i>Ranking</i> dos 10 melhores RMSD calculados para a proteína 1A11 . . .	115
Tabela 15	<i>Ranking</i> dos melhores RMSD calculados para a proteína 1VII	115
Tabela 16	<i>Ranking</i> dos melhores RMSD calculados para a proteína 1ERC	115
Tabela 17	Valores de RMSD disponíveis na literatura	116
Tabela 18	<i>Ranking</i> das 10 melhores soluções para a proteína 2EVQ pelo método Sort	117
Tabela 19	<i>Ranking</i> das 10 melhores soluções para a proteína 1A11 pelo método Sort	118
Tabela 20	<i>Ranking</i> das 10 melhores soluções para a proteína 1VII pelo método Sort	119
Tabela 21	<i>Ranking</i> das 10 melhores soluções para a proteína 1ERC pelo método Sort	120
Tabela 22	Relação das melhores soluções apresentadas pelo Sort e pelo RMSD . .	122

Tabela 23	Taxa de sucesso do método Sort para os as proteínas usadas nos experimentos	131
Tabela 24	Resultados apresentados pelo CASP11 para os 10 <i>targets</i> utilizados . .	132
Tabela 25	Taxa de sucesso do método Sort para os dados do CASP	135

Lista de siglas

2PG ProtPred-GROMACS

AE Algoritmo Evolutivo

AEMO Algoritmo Evolutivo Multiobjetivo

CASP *Critical Assessment of protein Structure Prediction*

CE Computação Evolutiva

ChIP-Seq *Chromatin immunoprecipitation followed by NGS*

GO *Gene Ontology*

NGS *Next Generation Sequencing*

PDB *Protein Data Bank*

PSP *Protein Structure Prediction*

SOA *Service-Oriented Architecture*

SQL *Structured Query Language*

Sumário

1	Introdução	21
1.1	Motivação e Relevância do Trabalho	23
1.2	Objetivos e Contribuições da Pesquisa	26
1.3	Organização da Dissertação	27
2	Proteínas	29
2.1	Aminoácidos	29
2.2	Estrutura primária	32
2.3	Estrutura secundária	33
2.4	Estrutura terciária	34
2.5	Representação computacional da proteína	36
2.6	Problema de predição da estrutura terciária	36
3	Métodos de predição	39
3.1	ProtPred-Gromacs	39
3.2	ProtPred-EDA	41
3.3	AEMMT	44
3.4	Comparação entre os métodos	46
4	Sistemas para biologia computacional	49
4.1	GeneSifter	49
4.2	Genomatix	50
4.3	ChIPseeqer	50
4.4	ABrowse	52
4.5	SeqAdapt	52
4.6	LabKey Server	53
4.7	Galaxy	54
4.8	Comparação dos sistemas	54

4.8.1	Framework Galaxy	55
4.8.2	Considerações de uso	63
5	Metodologia proposta	65
5.1	Projeto de Interfaces	66
5.2	Métodos para avaliação de resultados	68
5.2.1	<i>Ranking</i> de dominância	68
5.2.2	Fronteira de pareto e dominância	69
5.3	Métricas para comparação de estruturas	70
5.3.1	RMSD	70
5.3.2	TM-Score	71
5.3.3	GDT-TS	71
5.4	Aspectos técnicos de instalação e desenvolvimento	72
5.4.1	Instalação do Koala no Cloud USP	72
5.4.2	Pulsar	74
5.5	Módulos desenvolvidos	77
5.5.1	Pré-processamento dos dados	77
5.5.2	Algoritmos para predição de estruturas	81
5.5.3	Módulos de Análises	95
6	Resultados	103
6.1	Experimentos	105
6.1.1	Comparação com estrutura de referência	113
6.1.2	Comparação sem estrutura de referência	116
6.2	Workflow	135
6.3	Reprodutibilidade	140
6.4	Considerações Finais	143
7	Conclusões	145
7.1	Trabalhos Futuros	147
	Referências	149

Introdução

Os recentes avanços nas tecnologias de pesquisas envolvendo biologia e computação têm gerado uma grande quantidade de dados que precisam ser, de alguma forma, analisados. Essas pesquisas podem fazer com que um único laboratório gere uma quantidade de dados muito maior do que se produzia anos atrás, criando um cenário favorável para novas descobertas. Essas informações formam vários bancos de dados que são compartilhados por toda a comunidade científica e são usados como base para novos experimentos e pesquisas em diversas áreas, como genômica e proteômica (FIELDS, 2001) (FIELDS; JOHNSTON, 2002).

Biologia Computacional é o termo utilizado para as investigações que aplicam computação nas áreas biológicas com intuito de solucionar parte de seus problemas. O sequenciamento de DNA, RNA e de proteínas são alguns dos problemas que têm se destacado pelo uso da computação (MAOJO et al., 2010). A predição de estruturas de proteínas é outro problema relevante, para o qual ainda não existe uma solução apropriada mesmo com todo trabalho computacional desenvolvido para se lidar com diversos aspectos complicados presentes nesse problema (DILL; MACCALLUM, 2012).

O PSP é um problema NP-Completo, exigindo tempo exponencial para encontrar a solução no espaço de busca (SETUBAL, 1997), assim não é viável encontrar computacionalmente a melhor solução por meio de um algoritmo de busca exaustiva, uma vez que o tempo de processamento seria impraticável, a não ser para instâncias do problema pequenas. Por outro lado, esse problema pode ser interpretado como de otimização, em que a busca pela melhor solução é realizada em um espaço de possíveis soluções, segundo um critério de avaliação para cada uma encontrada. Com isso, a geração de novas soluções é orientada pelo critério podendo reduzir o número de soluções que precisam ser geradas para se encontrar o ótimo.

Nesse contexto, diversos sistemas computacionais têm sido desenvolvidos para auxiliar na modelagem computacional de problemas como o PSP, dentre eles destacam-se o GROMACS (HESS et al., 2008) e o Faunus (LUND; TRULSSON; PERSSON, 2008). Esses sistemas

têm sido desenvolvidos em projetos *open-source*¹ e que fornecem ferramentas computacionais que auxiliam no processo de modelagem molecular a partir de dados obtidos por métodos experimentais como NMR (do inglês *Nuclear Magnetic Resonance*) (NIXON et al., 1986) e Cristalografia de Raio X (BLUNDELL, 1976).

Os sistemas computacionais para Biologia Computacional são, em sua maioria, complexos e de difícil manipulação, principalmente por pesquisadores que não possuem conhecimento técnico em computação. Os sistemas aplicados em Biologia Computacional geralmente são *scripts* que possuem dependência de bibliotecas externas e são executados por linha de comando, como o BLAST, usado para alinhamento e pesquisas de sequências de DNA (ALTSCHUL et al., 1990). Para aumentar a usabilidade desses sistemas, têm sido cada vez mais comum o desenvolvimento de ferramentas computacionais baseadas em *web*, os quais possibilitam maior facilidade de acesso para diversos pesquisadores. O BLAST NCBI (JOHNSON et al., 2008) é um exemplo de que disponibiliza uma ferramenta por meio de uma interface amigável para o usuário. Além de disponibilizar ferramentas de execução para esses programas, existem desafios relacionados também com a integração e transferência de dados biológicos. Segundo (DAVIDSON; OVERTON; BUNEMAN, 1995), os repositórios de dados em geral utilizam nomenclaturas, estruturas de dados e tipos de arquivos específicos. Para resolver isso, foi iniciado o desenvolvimento de bancos de dados com padrões definidos (RITTER et al., 1994) (MAYER et al., 2014).

Da mesma forma que existe a necessidade de integrar diferentes bancos de dados biológicos, integrar as ferramentas usadas para manipular tais bancos também é importante, uma vez que isso possibilita ao investigador explorar um conjunto maior de métodos favorecendo melhores resultados para uma pesquisa. Por exemplo em (HU et al., 2005), é proposto o VisANT, um sistema computacional baseado em *web* para visualização e análise de vários tipos de redes de interação e associação biológicas, que disponibiliza ferramentas de análises e integração com diferentes bancos de dados.

Em resumo, um conjunto amplo de métodos computacionais e bancos de dados integrados possibilitam que o pesquisador execute um plano de experimentos mais elaborado. Além disso, os experimentos podem ser realizados através de uma interface amigável ao usuário, encapsulando toda a lógica de instalação e configuração, o que favorece que um número de pesquisadores possam executar planos experimentais relativamente complexos.

Uma forma de execução automática de vários passos desses experimentos é através de *workflows* científicos. Muitos estudos têm sido feitos para demonstrar a importância e necessidade dos *workflows* científicos para as pesquisas, apontando três principais motivações para a sua construção (CYBOK, 2006) (TAYLOR et al., 2014):

- Algumas aplicações computacionais complexas geralmente exigem a criação de um *workflow* colaborativo;

---

¹Sistemas computacionais com o código aberto e gratuito, podendo ser modificado mas não comercializado.

- Muitos cientistas carecem de conhecimento mínimo para utilizar os recursos computacionais atuais e infraestrutura;
- As especificações de um *workflow* podem ser reutilizadas, modificadas e compartilhadas, uma vez que elas estejam definidas.

Nesse sentido, esse trabalho desenvolveu o Koala, um ambiente integrado com diversas ferramentas, as quais podem ser conectadas para a execução de fluxos de trabalho automático. Todos os experimentos realizados nesse ambiente podem ser compartilhados no servidor, tornando-os acessíveis e reproduzíveis. A reprodutibilidade de resultados na ciência é um requisito para tornar a pesquisa mais transparente e confiável (CASADEVALL; FANG, 2010), e todo o processo realizado no experimento é reproduzível.

1.1 Motivação e Relevância do Trabalho

Uma vez que o PSP ainda é um problema em aberto, vários métodos têm sido desenvolvidos na literatura para lidar com esse problema. Porém, a reprodução de resultados e comparação dos mesmos não têm sido uma tarefa fácil. Nesse sentido, um evento bianual, o CASP (*Critical Assessment of protein Structure Prediction*), busca entre seus objetivos realizar tais comparações (MOSIMANN; MELESHKO; JAMES, 1995), criando uma competição entre vários métodos que são aplicados ao problema de predição da estrutura de proteínas, evidenciando assim o estado da arte em modelagem de estrutura de proteínas.

Por outro lado, o projeto WeFold aplica a metodologia denominada "*coopetition*": a cooperação de vários grupos, cada com os seus métodos aplicados ao PSP (KHOURY et al., 2014). Os participantes do WeFold competem e cooperam entre si, combinando os diferentes componentes e abordagens dos programas utilizados por cada um, para a geração de um novo sistema, integrando diversos programas para a geração de um sistema mais completo. O resultado dessa "*coopetition*" foi testado no CASP10 (2012) conseguindo bons resultados (KHOURY et al., 2014).

Na edição do CASP11 (2014), um dos algoritmos participantes foi o ProtPred-GROMACS (2PG), um *framework* de computação evolutiva multiobjetivo aplicado ao PSP (FACCIOLI; BORTOT; DELBEM, 2014). O 2PG é classificado como um Algoritmo Evolutivo Multiobjetivo (AEMO), uma técnica computacional de otimização aplicada no problema de PSP. Além dos Algoritmos Evolutivos (AEs), o algoritmo Monte Carlo Metropolis também é utilizado em problemas biológicos (METROPOLIS; ULAM, 1949). Neste contexto, trabalhos do Laboratório de Computação Reconfigurável (LCR) do ICMC-USP mostram que os AEMOs podem contribuir significativamente para o PSP. Em (LIMA et al., 2007) é apresentado um algoritmo evolutivo aplicado ao PSP chamado ProtPred, o qual utiliza uma abordagem *ab initio* para a predição da estrutura terciária.

Em um dos algoritmos propostos por (BRASIL; DELBEM; BONETTI, 2011) utiliza-se uma abordagem de dominância nas soluções encontradas, uma extensão do algoritmo NSGA-II, em que soluções não-dominadas são consideradas melhores. O outro algoritmo baseia-se em tabelas como sendo subpopulações, estendendo o algoritmo MEAT (do inglês *Multiobjective Evolutionary Algorithm on Tables*), e cada uma dessas tabelas representa um objetivo. Em (BRASIL; DELBEM; SILVA, 2013), um novo algoritmo evolutivo multiobjetivo com muitas tabelas (MEAMT, do inglês *Multiobjective evolutionary algorithms with many tables*) foi proposto baseado na limitação de outros algoritmos em trabalharem com mais de 4 objetivos ao mesmo tempo, demonstrando resultados satisfatórios. Integrando o ProtPred com outro *framework*, foi proposto o 2PG, o qual utiliza do GROMACS para o cálculo de dinâmica molecular (FACCIOLI et al., 2011) (FACCIOLI et al., 2012).

Porém, cada um desses *softwares* tem as suas especificações quanto a instalação, configuração, dependências de outros programas e execução, desfavorecendo o uso por outros pesquisadores sem conhecimento técnico em computação. Ademais, isso dificulta que eles sejam usados pela comunidade científica, não podendo contribuir com diversas outras pesquisas. Portanto, as contribuições desses trabalhos não podem ser difundidas devido a questões técnicas de uso e instalação.

Nesse cenário, sistemas baseados em *web* oferecem ferramentas para execução de programas que realizam alguma análise biológica, seja de DNA ou proteínas. O projeto GROMACSIMUN tem como objetivo fornecer uma interface gráfica para o *framework* GROMACS, facilitando o acesso aos *softwares* disponíveis (CAMPBELL; CHEN., 2014). O serviço Rosetta, através da interface *Robetta* (['http://rosetta.bakerlab.org/'](http://rosetta.bakerlab.org/)), disponibiliza os métodos tanto *ab initio* quanto de comparações para as análises de proteína (KIM; CHIVIAN; BAKER, 2004) (SIMONS et al., 1997). O servidor *I-TASSER* (['http://zhanglab.ccmb.med.umich.edu/I-TASSER/'](http://zhanglab.ccmb.med.umich.edu/I-TASSER/)), disponibiliza ferramentas para análises de estrutura e funções proteicas, possibilitando a geração de predições da estrutura 3D da proteína em alta qualidade e predições das funções biológicas das moléculas informadas a partir de sequências de aminoácidos (ZHANG, 2008). O servidor *QUARK* (['http://zhanglab.ccmb.med.umich.edu/QUARK/'](http://zhanglab.ccmb.med.umich.edu/QUARK/)), oferece ferramenta para análises de *folding ab initio* de proteína e predição de sua estrutura, gerando modelos 3D a partir da sequência de aminoácidos (XU; ZHANG, 2012).

Entretanto, essas ferramentas disponibilizadas *on-line* são relativamente complexas pois requerem certo treinamento para serem usadas adequadamente. Esse aspecto, em geral, restringe o usuário a adotar uma ferramenta em detrimento de outras, uma vez que em geral se requer um longo período para se trabalhar adequadamente com a ferramenta. Outra consequência é que essa dificuldade restringe comparações de resultados obtidos por outras ferramentas. Em geral tais comparações são feitas pela cooperação entre laboratórios que dominam ferramentas diferentes. Nesse sentido, esse trabalho propõe o desenvolvimento do Koala, para integrar várias ferramentas, que usualmente estão em

laboratórios diferentes, em um único ambiente que possui acesso simples ao uso integrado de ferramentas.

Existem alguns *frameworks* para armazenamento e compartilhamento de resultados que poderiam ser utilizados para desenvolver serviços de PSP em um ambiente integrado. Alguns desses *frameworks* são de uso comercial, como o Geospiza (PERKINELMER, 2014) e o Genomatix (GENOMATIX, 2014). Essas soluções oferecem serviços baseados na *web*, possibilitando baixar o *software*, fazer a instalação em um servidor local e disponibilizar o serviço, porém são soluções pagas.

Como alternativas não comerciais, existem alguns projetos *open-source*. Dentre esses, pode-se citar o SEQADAPT (BURDICK et al., 2010), o LabKey Server (NELSON et al., 2011), o ChIPseeqer (GIANNOPOULOU; ELEMENTO, 2011), o ABrowse (KONG et al., 2012) e o Galaxy (BLANKENBERG et al., 2010) (GOECKS; NEKRUTENKO; TAYLOR, 2010).

O Galaxy destaca-se nesse contexto por ser um *framework* que oferece várias formas de instalação e utilização (servidor *web*, servidor local e *cloud*). Uma instalação local é importante caso a pesquisa exija a manipulação de arquivos muito grandes (genoma completo, por exemplo) cuja transferência por meio da *internet* seja demorada ou inviável. Além disso, o Galaxy possui algumas características interessantes, como recursos que beneficiam a transparência, acessibilidade e reprodutibilidade dos experimentos. Por exemplo, um experimento pode ser compartilhado publicamente, detalhando a pesquisa, mostrando como o resultado foi obtido e possibilitando que outros pesquisadores a refaçam. Usuários sem conhecimentos específicos em computação conseguem utilizar o sistema. Além disso, o Galaxy possibilita que as informações sejam mantidas em um só lugar, com isso as análises podem ser facilmente refeitas. O Galaxy também possui controle de acesso e facilidades para customização, uma vez que as ferramentas podem ser escritas em qualquer linguagem de programação.

Para a comunidade científica, a princípio, é importante que um sistema tenha as seguintes características (BURDICK et al., 2010):

- *Open-source*. O sistema precisa ter o código aberto para que cada usuário possa alterá-lo conforme as suas necessidades;
- Padronizado. O sistema deve seguir um padrão de codificação que facilite o desenvolvimento de novas funcionalidades e integração com bases de dados;
- Adaptável. O sistema tem que ser flexível e de fácil adaptação, garantindo que modificações relativamente simples possam ser feitas sem exigir conhecimentos detalhados sobre a arquitetura interna do sistema;
- Modificável. O usuário deve poder alterar o sistema por meio de fácil reutilização dos procedimentos já implementados, para não se perder tempo com codificações complexas e detalhes técnicos.

1.2 Objetivos e Contribuições da Pesquisa

O objetivo global deste trabalho de mestrado é o desenvolvimento de um sistema de integração de métodos de predição e análises de estruturas de proteínas, de forma que disponibilize várias ferramentas aplicadas ao problema de PSP para predição e análise, gerando um ambiente integrado com acesso simples à essas ferramentas. O objetivo secundário é o desenvolvimento de métodos para aplicar o conceito de Dominância de Pareto, a fim de criar um *ranking* com soluções geradas por vários algoritmos preditores, selecionando as soluções com base em uma combinação de dois objetivos, como uma alternativa para as métricas de comparação de estruturas de proteínas, como o RMSD, que possuem dependência de uma estrutura de referência.

Para que esse sistema esteja em acordo com as necessidades da comunidade científica, além das características citadas acima, ele precisa garantir a reprodutibilidade das execuções que são realizadas no sistema, de forma que outros pesquisadores possam refazê-las, usando os mesmos dados e parâmetros utilizados. Em (SANDVE et al., 2013), são listadas 10 regras básicas para a reprodutibilidade de pesquisa computacional:

- Para cada resultado, manter salvo os passos de como foi produzido;
- Evitar manipulação manual de arquivos;
- Salvar as versões exatas de todos os programas externos usados;
- Controle de versão de todos os *scripts* personalizados;
- Salvar todos os resultados intermediários, se possível em arquivos com padrões;
- Para análises com números randômicos, ter salvo a 'semente' utilizada;
- Sempre salvar e disponibilizar os dados utilizados em gráficos;
- Gerar resultados com hierarquia, permitindo que seja informado mais detalhes para ser verificado;
- Conectar os resultados com suas descrições textuais;
- Prover o acesso público a *scripts*, dados e resultados.

Este trabalho de mestrado contribuiu para a área da Biologia Computacional por meio do desenvolvimento do Koala e todos os seus módulos. Todos os algoritmos foram implementados de forma eficiente para que seja criado um ambiente integrado, permitindo o uso de fluxos de trabalho (*workflows*) e garantindo a reprodutibilidade dos experimentos realizados no servidor.

1.3 Organização da Dissertação

O Capítulo 2 apresenta os conceitos sobre aminoácidos, com a sua estrutura química e uma lista com os 20 aminoácidos que constituem as proteínas. As proteínas, por sua vez, são apresentadas em seguida, abordando as suas três estruturas e sua representação computacional. Além disso, é apresentado o problema de predição da estrutura de proteína.

O Capítulo 3 aborda alguns métodos disponíveis na literatura que são aplicados no problema de predição da estrutura de proteína. As suas características, diferenças e relevância são também apresentadas nesse capítulo, por meio de testes comparativos entre eles.

O Capítulo 4 apresenta *softwares* (*frameworks*) para a criação de uma infraestrutura computacional que permite a criação e alteração de ferramentas e *pipelines* para execução de algoritmos aplicados na Biologia Computacional. As características de cada um são, também, sintetizadas e comparadas. Um dos *frameworks* é detalhado, pois foi usado neste trabalho para a instalação de uma plataforma com diversos serviços aplicados ao PSP.

O Capítulo 5 descreve o ambiente Koala, o sistema proposto por esse trabalho de mestrado, destacando os aspectos técnicos da instalação e configuração, além de apresentar os módulos desenvolvidos para esse sistema.

O Capítulo 6 detalha os resultados obtidos com a utilização do sistema desenvolvido neste trabalho. E, por fim, o Capítulo 7 apresenta as considerações finais deste trabalho de mestrado.

CAPÍTULO 2

Proteínas

As proteínas são biomoléculas que estão presentes em todas as partes de uma célula, sendo as mais abundantes nos seres vivos. Elas podem desempenhar diversas funções biológicas e possuem muitas propriedades e atividades distintas, assumindo cada uma um papel biológico específico. As proteínas são hétero-polímeros cujas unidades são constituídas a partir de um alfabeto de 20 aminoácidos (MURRAY et al., 2014). A Seção 2.1 abordará aspectos relevantes sobre os aminoácidos.

As proteínas podem assumir quatro formas diferentes com relação a sua estrutura espacial. Essas diferentes classificações são organizadas hierarquicamente: estrutura primária, secundária e terciária. Existe também a quaternária, que é a união de proteínas que se encontram na estrutura terciária. Cada uma delas será detalhada nas próximas seções que seguem, a saber a Seção 2.2, Seção 2.3 e a Seção 2.4, exceto a estrutura quaternária, por não ser abordada neste trabalho.

É importante ressaltar que, para uma proteína exercer efetivamente a sua função específica no organismo, ela precisa estar na estrutura espacial terciária.

2.1 Aminoácidos

Os aminoácidos, por sua vez, são moléculas orgânicas que possuem uma estrutura química básica formada por um átomo de carbono, denominado carbono α , e outros quatro componentes ligados ao carbono central, que são: um átomo de hidrogênio (H), um grupo carboxílico (COOH), um grupo amina (NH₂) e uma cadeia lateral **R**, que é diferente para cada aminoácido (CAMPBELL; SMITH; PETERS TIMOTHY, 2006). A Figura 1 representa essa estrutura básica.

Todas as proteínas, independente do ser vivo, são formadas essencialmente por um alfabeto de 20 aminoácidos, que são denominados resíduos, pois ocorre a perda de átomos das estruturas do aminoácido no processo de formação da proteína (ligação peptídica).

A Tabela 1 apresenta uma descrição de todos os 20 aminoácidos e seus códigos. Os aminoácidos podem ser representados por um código de uma ou três letras. A estrutura

Figura 1: Estrutura química básica de um aminoácido (JUNIOR; FRANCISCO, 2006)

primária é representada pelo arquivo FASTA, que é um formato de arquivo que contém na primeira linha a definição de uma proteína ou de um gene (ID), e uma sequência de caracteres onde cada um corresponde a um aminoácido com código de uma letra (MURRAY et al., 2014).

Tabela 1: Relação dos vinte aminoácidos padrões nas proteínas.

Aminoácido	Cód. 3 letras	Cód. 1 letra	Origem do Cód. 1 Letra
Alanina	Ala	A	Alanine
Cisteína	Cys	C	Cysteine
Acido Aspártico	Asp	D	asparDic acid
Fenilalanina	Phe	F	Fenylalanine
Acido Glutâmico	Glu	E	gluEtamic acid
Glicina	Gly	G	Glycine
Histidina	His	H	Histidine
Isoleucina	Ile	I	Isoleucine
Lisina	Lys	K	letra antes do <i>L</i>
Leucina	Leu	L	Leucine
Metionina	Met	M	Methionine
Asparagina	Asn	N	asparagiNe
Prolina	Pro	P	Proline
Glutamina	Gln	Q	Q-tamine
Arginina	Arg	R	aRginine
Serina	Ser	S	Serine
Treonina	Thr	T	Theorine
Valina	Val	V	Valine
Triptofano	Trp	W	tWo rings
Tirosina	Tyr	Y	tYrosine

Cada aminoácido presente nas proteínas difere-se pela cadeia lateral *R*, no que se refere a estrutura, tamanho, cargas elétricas e solubilidade em água. Além disso, essas cadeias são responsáveis por forças estabilizadoras que surgem a partir de interações fracas com outras moléculas, permitindo que as proteínas cheguem em sua estrutura nativa, que é o estado em que a proteína pode exercer alguma função no organismo (MURRAY et al., 2014). Esse processo é conhecido como enovelamento ou *folding*, e não é totalmente compreendido. A Figura 2 demonstra as estruturas das cadeias laterais **R** de todos os 20 aminoácidos presentes nas proteínas.

Figura 2: Estrutura das cadeias laterais **R** dos 20 aminoácidos presentes nas proteínas. Fonte: <http://www.mdsaude.com/2012/05/o-que-e-proteina.html>. Acesso: 10/05/2014.

De acordo com a estrutura química do radical **R**, ou simplesmente cadeia lateral, ligado ao aminoácido, o mesmo pode ser classificado em três diferentes classes: estritamente apolares ou hidrofóbico (compostos que não se dissolvem na água); estritamente polares ou hidrofílico (compostos que se dissolvem em contato com a água); e por fim, são os aminoácidos com comportamento apolares e polares, também chamados de anfipáticos (CAMPBELL; SMITH; PETERS TIMOTHY, 2006).

As proteínas são formadas pela união de vários aminoácidos. No processo de formação da proteína, os aminoácidos unem-se formando um polímero longo, contínuo e não ramificado, chamado de hétero-polímero, ou cadeia polipeptídica. O resultado da cadeia polipeptídica é a geração de um polímero chamado polipeptídico, formado por todos os aminoácidos que se uniram através das ligações peptídicas. As ligações peptídicas são geradas a partir de ligações covalentes¹ do grupo carboxila de um aminoácido (COOH) com o grupo amina do outro aminoácido (NH₂). Neste processo, existe a liberação de uma molécula de água (H₂O) (KARP, 2002). A Figura 3 demonstra o processo de formação de uma Ligação Peptídica.

A diferença entre proteínas e polipeptídios é estritamente semântica, pois, por definição, todas as proteínas são polipeptídios. Porém, é de uso comum da palavra polipeptídio

¹Ligação química onde ocorre o compartilhamento de elétrons entre os átomos.

Figura 3: Ligação peptídica (JUNIOR; FRANCISCO, 2006).

apenas para algumas sequências de aminoácidos.

2.2 Estrutura primária

A sequência linear de aminoácidos de uma proteína é chamada de estrutura primária. É essa sequência de aminoácidos que diferencia uma proteína de outra, sendo o nível de organização molecular mais simples e mais importante, pois é a partir deste que se origina o arranjo espacial da molécula (MURRAY et al., 2014). A Figura 4 representa essa estrutura.

A quantidade de aminoácidos presente nessa estrutura determina a quantidade de resíduos da proteína e como eles estão ligados (Ligação peptídica). Utilizando desse conhecimento sobre a quantidade de resíduos, tem-se dois termos para se designar acerca das cadeias formadas pelos aminoácidos. Uma cadeia pequena, com cerca de 30 resíduos, é chamada de peptídio. Já uma cadeia mais longa, com uma quantidade de resíduos superior a 4 mil, é chamada de polipeptídio.

Como exemplo, pode-se considerar a sequência da aminoácidos da proteína 1VII que foi extraída do *Protein Data Bank* (PDB)². A sequência MLSDEDFKAVFGMTRSA FANLPLWKQQNLKKEKGLF é a estrutura primária dessa proteína, formada pelos aminoácidos: Metionina, Leucina, Serina, ácido Aspártico, ácido Glutâmico, Fenilalanina, Lisina, Alanina, Valina, Glicina, Treonina, Arginina, Asparagina, Prolina, Triptofano e Glutamina, podendo cada um ser repetido ou não dentro da sequência.

Figura 4: Estrutura Primária é a sucessão de resíduos dos aminoácidos, nesse caso abreviados por uma letra (Adaptado de (RUSSEL, 2009)).

²O PDB é uma das principais bases de dados de proteínas com estrutura terciária determinada por meio de métodos experimentais

2.3 Estrutura secundária

A estrutura secundária está relacionada à conformação local de uma porção de um polipeptídeo, o que corresponde ao arranjo tridimensional de aminoácidos localizados mais próximos dentro da estrutura primária (MURRAY et al., 2014). Existem dois tipos principais de estruturas secundárias:

1. Hélice- α ;
2. Folhas- β ;

A Hélice- α é uma estrutura que possui a forma helicoidal, ou na forma de bastão. Ela é estabilizada por interações de hidrogênio entre os grupos NH e CO da cadeia principal. Cada volta da hélice é formada por 3.6 resíduos de aminoácidos, gerando um comprimento aproximado de 12 resíduos (JUNIOR; FRANCISCO, 2006). A Figura 5 representa essa estrutura.

Figura 5: Estrutura Hélice- α de mão direita (Adaptado de (RUSSEL, 2009)).

As Folhas- β são estruturas formadas por cadeias- β , as quais possuem em média entre 5 e 10 resíduos. Cada segmento da cadeia principal interage com outro, resultando em uma estrutura achatada. Ela apresenta uma cadeia principal distendida, não possibilitando a ocorrência de interações de hidrogênio com outras cadeias- β , o que resulta na liberação de O₂ do grupo da carboxila e Nitrogênio da cadeia principal (BRANDEN; TOOZE, 1991). A Figura 6 representa essa estrutura.

Essas estruturas são estabilizadas por interações de hidrogênio que ocorre no *backbone*³ das proteínas (MURRAY et al., 2014), e podem ser subdivididas em outras duas formas, diferenciando-se pelo ângulo formado nas interações de hidrogênio dentro da cadeia- β (MURRAY et al., 2014):

---

³O *backbone* é conhecido também por cadeia principal da proteína, formada pelos carbonos- α de todos os resíduos que formam a proteína.

Figura 6: Estrutura Folha- β (Adaptador de (EMBL, 2015)). A seta indica a direção da folha- β , que é da extremidade chamada de N-terminal (o último átomo é um Nitrogênio) para a C-terminal (o último átomo é um Carbono).

1. Paralelas;
2. Antiparalelas.

Em alguns casos de estrutura proteica, pode acontecer de existir somente duas cadeias- β antiparalelas. Essas estruturas são chamadas de estruturas secundárias modificadas (do inglês *secondary structure motif*). Um exemplo dessa estrutura é a grampo de cabelo (β -*hairpin*, em inglês). O β -*hairpin* é formado por duas cadeias- β antiparalelas interligadas por um *loop*, ou uma volta (*turn*). A estrutura β -*turn* tem um importante papel nas estruturas tridimensionais, conectando cadeias- β , folhas- β e hélices- α (LODISH et al., 2004). A Figura 7a representa um exemplo de uma β -*turn* contendo o aminoácido Glicina, e a Figura 7b mostra a estrutura do β -*hairpin*.

2.4 Estrutura terciária

A estrutura terciária da proteína é o resultado do dobramento total de todos os resíduos de aminoácidos que compõe a proteína.

Essa estrutura representa o arranjo tridimensional que é assumido pela sua cadeia polipeptídica devido a composição das cadeias laterais dos aminoácidos. Essa conformação hierárquica está ligada à maneira com que as estruturas secundárias estão arranjadas tridimensionalmente e como os aminoácidos interagem entre si, formando interações de hidrogênio, pontes eletrostáticas e interações hidrofóbico-hidrofílico (MURRAY et al., 2014).

A estrutura terciária também é conhecida como estrutura nativa da proteína, ou enovelamento nativo, e é a forma que confere as atividades biológicas à proteína, ou seja, as suas diversas funções no organismo.

Figura 7: Estruturas β -turn(a) e β -hairpin(b). Em (a) é detalhado a estrutura química da *beta-turn*, destacando a ponte de hidrogênio que a estabiliza. Em (b) exibe uma estrutura tridimensional no padrão *ribbon*.

As proteínas, em sua estrutura tridimensional, estão na forma que possuem um mínimo de energia livre, de acordo com a solução que se encontram. Neste estado, os resíduos apolares se afastam do meio aquoso, o que forma o centro hidrofóbico da proteína. Similarmente, nesse estado, as proteínas se tornam favoráveis as interações entre os aminoácidos polares da superfície hidrofílica com o solvente. A Figura 8 apresenta essa estrutura.

Figura 8: Representação da Estrutura terciária de uma proteína (PDB 1VII) (*Protein Data Bank* - (<http://www.rcsb.org/pdb/home/home.do>)).

Devido a todos esses fatores e pelo fato da estrutura terciária da proteína não ser regular, faz com que a tarefa de descrevê-la se torne algo complexo e difícil. Por isso, muitos pesquisadores têm investigado o dobramento proteico, mas ainda com muitas questões não resolvidas.

Uma alternativa para solução dessa questão é categorizada por meio de arranjos topológicos dos elementos da estrutura secundária. Uma propriedade da estrutura terciária é a

existência de uma superfície topográfica complexa permitindo que a proteína interaja com pequenas moléculas que podem ligar-se em fendas, ou com macromoléculas, com as quais a proteína pode ter regiões de topologia complementar de carga (positiva ou negativa) (CAMPBELL; SMITH; PETERS TIMOTHY, 2006) (MURRAY et al., 2014).

O estudo para determinar as estruturas terciária é muito importante, pois a estrutura e a função estão estritamente relacionadas. Alguns detalhes desse problema serão apresentados na Seção 2.6.

2.5 Representação computacional da proteína

As proteínas podem ser representadas computacionalmente de acordo com as suas conformações tridimensionais. De acordo com cada conformação da proteína, é possível estudar as inferências que ocorrem nas suas propriedades. Essas conformações podem ser representadas de duas formas (BAKER; CHAN, 1996):

- **Coordenadas Internas:** a proteína é representada de acordo com dados de Ligação entre dois átomos, o valor do Ângulo de Ligação com um terceiro átomo, e o valor do Ângulo formado com um quarto átomo;
- **Coordenadas Cartesianas:** a proteína é representada de acordo com a posição tridimensional de cada átomo pertencente à sua composição.

Essas conformações podem ser simuladas alterando-se os valores do comprimento de Ligação, e nos Ângulos de Ligação e torcionais.

2.6 Problema de predição da estrutura terciária

O problema de predição da proteína (PSP) vem sendo abordado desde a década de 70, mas até agora não foi encontrada uma solução (DILL; MACCALLUM, 2012), sendo um dos maiores desafios da bioinformática estrutural ainda em aberto, mesmo com vários avanços nos últimos 10 anos (DRAKE, 2015).

Na literatura existem trabalhos que demonstram que, durante o processo de *folding*, a proteína pode apresentar diversas conformações e que seria inviável uma abordagem aleatória para percorrer todo o espaço de busca que essas conformações podem assumir, caracterizando um problema de otimização.

O processo de *folding*, ou enovelamento, é o processo de formação da estrutura terciária da proteína. Para esse processo, algumas propriedades físicas precisam ser levadas em consideração (DILL; MACCALLUM, 2012):

- Rigidez da cadeia principal do aminoácido;

>2LZT:A|PDBID|CHAIN|SEQUENCE
KVFGRCLEAAAMKRHGLDNYRGYSLGNWVCAAKFESNFNTQATNRNTD
GSTDYGILQINSRWWCNDGRTPGSRNLCNIPCSALLSSDITASVNC AKKIV
SDGNGMNAWVAWRNRCKGTDVQAWIRGCRL

(a) Arquivo FASTA

ATOM	1	N	LYS	A	1	2.967	4.770	13.995	1.00	9.18	N
ATOM	2	CA	LYS	A	1	2.755	5.653	12.837	1.00	9.51	C
ATOM	3	C	LYS	A	1	3.345	7.017	13.183	1.00	9.91	C
ATOM	4	O	LYS	A	1	3.180	7.464	14.332	1.00	10.78	O
ATOM	5	CB	LYS	A	1	1.256	5.826	12.530	1.00	9.46	C
ATOM	6	CG	LYS	A	1	0.967	6.938	11.512	1.00	11.37	C
ATOM	7	CD	LYS	A	1	-0.492	7.111	11.149	1.00	11.62	C
ATOM	8	CE	LYS	A	1	-0.664	8.337	10.256	1.00	11.53	C
ATOM	9	NZ	LYS	A	1	-2.121	8.432	9.994	1.00	14.80	N
ATOM	10	N	VAL	A	2	4.001	7.627	12.208	1.00	8.34	N
ATOM	11	CA	VAL	A	2	4.566	8.957	12.347	1.00	8.02	C
ATOM	12	C	VAL	A	2	3.699	9.921	11.521	1.00	6.94	C
ATOM	13	O	VAL	A	2	3.686	9.773	10.283	1.00	8.43	O
ATOM	14	CB	VAL	A	2	6.042	8.990	11.950	1.00	8.15	C
ATOM	15	CG1	VAL	A	2	6.657	10.380	12.190	1.00	9.59	C
ATOM	16	CG2	VAL	A	2	6.889	7.939	12.606	1.00	9.62	C
ATOM	17	N	PHE	A	3	3.030	10.846	12.182	1.00	6.53	N
ATOM	18	CA	PHE	A	3	2.129	11.804	11.541	1.00	5.73	C
ATOM	19	C	PHE	A	3	2.883	12.964	10.882	1.00	4.74	C
ATOM	20	O	PHE	A	3	3.906	13.457	11.333	1.00	4.86	O

(b) Arquivo PDB

(a) Arquivo FASTA

(b) Arquivo PDB

Figura 9: Representação computacional do problema de predição da estrutura terciária da proteína. Em (a) é exibido a sequência primária da proteína 2LZT, onde cada caractere representa um aminoácido. Em (b) mostra cada átomo dos aminoácidos com seus valores angulares, descritos no arquivo PDB.

- Interações entre os aminoácidos que compõem a proteína;
- Interações eletrostáticas;
- Forças de dispersão (Van der Waals);
- Restrição de volume;
- Pontes de hidrogênio e dissulfeto;
- Interações dos aminoácidos com o meio aquoso.

Na Biologia Computacional, o processo de *folding* pode ser demonstrado utilizando dois tipos de arquivos, FASTA e PDB, os quais representam as estruturas primária e terciária, respectivamente, da proteína. A Figura 9a mostra o arquivo FASTA da proteína Lisozima (PDBID: 2LZT) e a Figura 9b exibe um trecho do arquivo PDB da mesma proteína, onde é possível ver os átomos de cada resíduo e outros valores.

A predição do processo de *folding* por meio de métodos computacionais na Biologia Computacional pode ser dividido em três técnicas principais (BAXEVANIS; OUELLETTE, 2001):

- **Homologia:** esta técnica prediz a estrutura terciária de uma proteína desconhecida com base em outra proteína com estrutura conhecida, chamada de homóloga, que possuem um ancestral comum;
- **Threading:** Ao contrário da homologia, esta técnica é aplicada quando uma proteína não possui nenhuma outra proteína homóloga, mas com uma estrutura tridimensional similar;

- ***Ab initio***: Ao contrário dos dois métodos anteriores, esse método não precisa ter conhecimento de nenhuma outra estrutura, predizendo a estrutura da proteína levando em consideração os aspectos físicos e não experimentais, a partir da estrutura primária. Pode ser aplicado mesmo sem a existência de uma proteína homóloga ou similar.

Métodos de predição

Neste capítulo serão abordados alguns métodos aplicados no problema de predição da estrutura de proteína, os quais têm uma ferramenta de execução com uma interface amigável para o usuário no servidor criado, além de estarem integrados com outras ferramentas de análises.

A Seção 3.1 descreve o *framework* 2PG e alguns dos seus programas utilizados. Nas Seções 3.2 e 3.3 são apresentados dois *softwares* diferentes aplicados ao PSP desenvolvidos no LCR-ICMC, ProtPred-EDA e AEMMT. Por fim, a Seção 3.4 sumariza as informações de alguns métodos disponíveis no servidor criado neste trabalho.

3.1 ProtPred-Gromacs

O ProtPred-Gromacs (2PG) é um *framework* de computação evolutiva, com algoritmos mono e multiobjetivos, aplicado ao problema de predição de estrutura da proteína (PSP) (FACCIOLI et al., 2011) (FACCIOLI et al., 2012) (FACCIOLI; BORTOT; DELBEM, 2014). O 2PG utiliza o *framework* GROMACS (SPOEL et al., 2009) para realizar os cálculos dos *fitness*¹ dos indivíduos. O Gromacs (GROningem MACHine for Chemical Simulations) é utilizado em modelagem molecular para simulações e minimização de energia em sistemas moleculares.

A representação da proteína no sistema é realizada utilizando as coordenadas Cartesianas, e não as coordenadas internas. Com isso, o sistema trabalha com valores para os indivíduos em nível atômico. Ademais, o 2PG utiliza os campos de força CHAMM27 (FELLER; MACKERELL, 2000), (KLAUDA et al., 2005) e AMBER99 (WANG; CIEPLAK; KOLLMAN, 2000).

Para execução de cada algoritmo disponível, o 2PG utiliza um arquivo de parâmetros contendo informações de diretórios, arquivos, opções avançadas para customizar a execução dos algoritmos, entre outras. Esses parâmetros foram utilizados como campos de

---

¹Na computação evolutiva, *fitness* é o valor de uma solução que determina o quão boa ela é para o problema.

entrada nas ferramentas desenvolvidas no servidor criado. Com essa abstração, o usuário não precisa se preocupar em preencher corretamente o arquivo de parâmetro, saber qual é o nome específico do parâmetro a ser preenchido ou até mesmo correr o risco de incluir informações incorretas, o que poderia gerar resultados errôneos ou erros no *software*. A Figura 10 mostra o diagrama genérico de execução do 2PG.

Figura 10: Diagrama do ProtPred-Gromacs. Como entrada (*input*) o 2PG recebe uma sequência de aminoácidos e um arquivo PDB como população inicial, além de um arquivo com os parâmetros do algoritmo. A partir desses arquivos, o 2PG carrega os parâmetros e o arquivo de população e calcula a *fitness* dos indivíduos utilizando o GROMACS. No final, vários modelos são gerados como resposta do algoritmo. (Adaptado de (FACCIOLI, 2012)).

O 2PG possui diversos algoritmos baseados em otimização implementados e aplicados ao PSP. Para esses métodos foram criadas algumas ferramentas para execução no servidor, e integrados com outros algoritmos e ferramentas de análises, em uma infraestrutura computacional baseada em *cloud*. Os algoritmos utilizados foram:

- **Build Conformation:** Um método para a criação de conformações aleatórias de proteínas, que são utilizadas como parâmetro de entrada como população inicial para os outros algoritmos preditores do 2PG. A partir de uma sequência de aminoácidos, esse algoritmo cria N modelos aleatórios, no padrão de arquivo PDB;

- **Algoritmo Evolutivo Mono Objetivo:** Um algoritmo genético que realiza a busca da melhor solução utilizando como base uma população inicial aleatória. É totalmente *ab initio*, pois não se baseia em outras conformações conhecidas de proteínas homólogas. Utilizando de mutações e recombinações entre indivíduos da população, o método converge para encontrar a melhor solução baseado em um *fitness* de acordo com o objetivo selecionado;
- **Random Algoritmo:** Um método que gera soluções aleatórias a partir de uma população inicial. Não utiliza nenhum critério de busca ou refinamento das soluções (mutação). Como resultado, o algoritmo gera diversos modelos conformacionais aleatórios;
- **Monte Carlo-Metrolopis:** É um algoritmo estatístico utilizado para calcular as probabilidades das soluções geradas aleatoriamente e muito utilizado em problemas complexos na biologia, física e matemática, pois não é possível obter uma resposta determinística, já que seria inviável o processamento desses problemas (HROMKOVIC, 2010). Este algoritmo foi inicialmente proposto por (METROPOLIS et al., 1953) e generalizado por (HASTINGS, 1970);
- **Non-dominated Sorting Genetic Algorithm-II (NSGA-II):** Um algoritmo multiobjetivo para classificação de não-dominados que possui complexidade computacional igual a $O(MN^2)$, em que M corresponde ao número de objetivos e N é o tamanho da população. O NSGA-II converge para encontrar as melhores soluções próximas à fronteira de Pareto ótima. Foi proposto por (DEB et al., 2002) e é muito utilizado no PSP, pois este é um problema multiobjetivo;
- **Sort By Front Dominance:** É um método que cria um *ranking* utilizando Fronteira de Pareto com diversas soluções geradas por vários algoritmos aplicados ao PSP. A partir de uma combinação de dois objetivos, as soluções são agrupadas em fronteiras e ordenadas de acordo com a dominância de cada uma sobre as outras. A solução que está classificada na fronteira 0 e posição 1 é a melhor solução encontrada, pois não é dominada por nenhuma outra.

3.2 ProtPred-EDA

ProtPred-EDA é um algoritmo de Estimação de Distribuição (EDA) específico para o PSP. Foi baseado no ProtPred proposto por (LIMA et al., 2007), com alterações para trabalhar com modelos probabilísticos e outras abordagens evolutivas propostas por Bonetti (BONETTI, 2015). A Figura 11 demonstra a biblioteca disponível no ProtPred-EDA.

Figura 11: Biblioteca do ProtPred-EDA. Em verde está representado o código-fonte, as bibliotecas de terceiros está em azul e os parâmetros de entrada personalizáveis estão representados pela cor laranja. O bloco em destaque (exec) é o binário responsável por interligar todas os outros blocos (BONETTI, 2015).

O ProtPred-EDA utiliza uma modelagem puramente *ab initio* e *full-atom*, e oferece diversos algoritmos aplicados ao PSP, os quais estão disponíveis para execução no servidor criado (BONETTI, 2015):

- **Busca Aleatória:** A Busca Aleatória (RW, do inglês *Random Walk*) é um algoritmo que gera soluções aleatórias, dentro de um intervalo definido pelas variáveis do problema. Tem como característica a geração de soluções bem diversificadas (PEARSON, 1905);
- **Monte Carlo:** O método Monte Carlo (MC) consiste em receber amostras (ou soluções) aleatórias e depois então analisar o resultado. É usado em problemas de otimização pois utiliza um critério de aceitação ou rejeição de uma amostra (METROPOLIS; ULAM, 1949);
- **Algoritmo Genético:** O Algoritmo Genético (GA, do inglês *Genetic Algorithm*) é baseado na teoria da evolução de Darwin (HOLLAND, 1975). Neste GA adaptado para o PSP, possui uma operação de recombinação e uma de mutação;

- **Evolução Diferencial:** A Evolução Diferencial (DE, do inglês *Differential Evolution*) pertence a classe dos AEs e tem funcionamento semelhante a de um GA, diferenciando somente nos operadores de recombinação e mutação (STORN; PRICE, 1997);
- **Algoritmo de Estimação de Distribuição:** Os Algoritmos de Estimação de Distribuição (EDA, do inglês *Estimation of Distribution Algorithms*) é a classificação dada aos AEs que utilizam um modelo probabilístico para estimar variáveis que são usadas na geração de novas soluções dentro de um conjunto de soluções promissoras. O ProtPred-EDA possui três modelos probabilísticos específicos para o PSP:
  - **UNIO:** O método *Univariate model-based Optimization* (UNIO) utiliza um modelo probabilístico simples, pois não considera relacionamento de variáveis (BONETTI, 2015);
  - **KDEO:** O algoritmo *Kernel Density Estimation model-based Optimization* (KDEO) utiliza um modelo probabilístico bivariado KDE com duas dimensões, chamado de KDE2D (BONETTI, 2015). Neste caso, o modelo trata o relacionamento entre variáveis;
  - **FGMO:** O método *Finite Gaussian Mixtures model-based Optimization* (FGMO) utiliza um modelo probabilístico bivariado para a criação das estimativas utilizadas para amostrar novas soluções. Para isso, é utilizada Misturas Finitas Gaussianas (FGM, do inglês *Finite Gaussian Mixtures*), o qual utiliza de várias distribuições Gaussianas para estimar a densidade dos dados multimodais (BONETTI, 2015);
  - **hEDA:** O EDA hierárquico (hEDA) utiliza uma método de agrupamento hierárquico para dividir o problema (cadeia completa de aminoácidos da proteína) em subproblemas menores, de forma que seja possível tratar cada subproblema de forma independente. Ele utiliza um dos modelos probabilísticos citados anteriormente para estimar e amostrar novas soluções;
  - **rBOA:** O algoritmo *real-valued Bayesian Optimization Algorithm* (rBOA) também separa um problema maior em subproblemas, avaliando os resultados de cada subproblema. Utiliza um modelo probabilístico com estrutura de grafos, conhecida como rede Bayesiana (BONETTI, 2015).

A Figura 12 exhibe os métodos de otimização propostos para o PSP e como eles podem ser combinados com os modelos probabilísticos disponíveis no *framework*.

Figura 12: Métodos de otimização propostos no ProtPred-EDA para o PSP. (BONETTI, 2015).

3.3 AEMMT

O Algoritmo Evolutivo Multiobjetivo com muitas tabelas (AEMMT) (em inglês *Multi-objective Evolutionary Algorithm with many tables* (MEAMT)) é um sistema aplicado ao PSP desenvolvido no ICMC-USP usando como base o ProtPred proposto por (LIMA et al., 2007), e foi adaptado para a utilização no problema de predição de proteínas com uma abordagem diferente, usando a técnica de algoritmo evolutivos multiobjetivo baseado em tabelas (AEMT), no qual os objetivos são representados por subpopulações como tabelas, além de tabelas de ponderação entre os objetivos (BRASIL, 2012). A Figura 13 mostra o fluxograma de execução do AEMMT.

O AEMMT utiliza quatro objetivos (funções de energia) para prever as estruturas de proteínas: Van der Waals, solvatação, eletrostática e interações de hidrogênio (*vdw*, *charge*, *solv* e *hbond*, respectivamente). Além dos 4 objetivos, possui uma tabela com os indivíduos não-dominados (ND) e uma com a ponderação dos quatro objetivos. Nessa configuração, verificou-se que a diversidade das soluções é maior do que com dois ou

Figura 13: Fluxograma do AEMMT. Como entrada, o algoritmo recebe a sequência de aminoácidos, o tamanho da população e a quantidade de gerações, condição de parada do algoritmo. Uma população inicial com possíveis conformações a partir da sequência de aminoácidos é criada. Subpopulações para cada objetivo são criadas usando como base a população inicial. Em cada geração, indivíduos são selecionados por torneio para compor a nova população. No final, os modelos são gerados com base nas subpopulações finais. (Adaptado de (BRASIL; DELBEM; SILVA, 2013)).

três objetivos, entretanto as estruturas secundárias folhas- β não foram satisfatoriamente preditas, além de um desempenho ruim em relação à Fronteira de Pareto, uma vez que em um espaço de quatro dimensões o número de soluções necessárias para amostrar a fronteira é muito alto (BRASIL, 2012).

A partir disso, foi verificado que o uso de funções ponderações como critérios de seleção seria melhor para o algoritmo. Com isso, foram criadas combinações duplas e triplas entre os objetivos, totalizando 16 tabelas com subpopulações no algoritmo, onde cada subpopulação é verificada qual a sua contribuição para o algoritmo, e as que mais contribuíram são selecionadas para a próxima geração. A Figura 14 mostra as subpopulações utilizadas no AEMMT.

O AEMMT possui duas ferramentas disponíveis no servidor criado, uma para a criação da população inicial utilizada no algoritmo e uma para o próprio algoritmo, que realiza a

Figura 14: Subpopulações usadas no AEMMT, com um total de 16 tabelas, resultado da combinação de quatro objetivos (seis por combinação dupla e quatro por combinação tripla) (BRASIL, 2012).

predição da estrutura da proteína. Esse algoritmo obteve ótimos resultados em relação a outros preditores, como mostrado em (BRASIL; DELBEM; SILVA, 2013).

3.4 Comparação entre os métodos

Existem diversos métodos disponíveis na literatura aplicado ao PSP, cada um com abordagens diferentes. Neste trabalho foram escolhidos alguns desenvolvidos no Laboratório de Computação Reconfigurável (LCR) do ICMC-USP. Esses métodos possuem características diferentes, e por isso podem ser aplicados a problemas distintos, como proteínas pequenas ou só com uma estrutura secundária. Por isso, eles foram escolhidos para terem uma ferramenta nessa versão do sistema.

O objetivo de ter uma ferramenta para esses serviços em nosso servidor é oferecer, ao usuário, um ambiente mais completo, disponibilizando vários métodos que podem ser executados todos no mesmo lugar. Ademais, o usuário pode comparar os resultados de cada método mais facilmente, uma vez que as ferramentas de análise podem ler qualquer PDB gerado pelos algoritmos. Assim, o objetivo de integrar vários preditores fica evidente, resultando em uma plataforma mais diversificada em relação a disponibilidade de algoritmos aplicados ao PSP diferentes. A Tabela 2 sumariza os métodos e algumas de suas características.

Tabela 2: Relação dos métodos de predição. *A maior proteína com que o 2PG já trabalhou foi uma com 595 resíduos, um dos alvos (T0793) do CASP11 (2014), mas não existe restrições quanto ao número de aminoácidos (FACCIOLI; BORTOT; DELBEM, 2014).

Método	Princípio de predição	Modelo de proteína	Quantidade de resíduos	Acesso
2PG	<i>ab initio</i>	<i>all-atomic</i>	-*	Download
ProtPred-EDA	<i>ab initio</i>	<i>all-atomic</i>	-	Download
AEMMT	<i>ab initio</i>	<i>all-atomic</i>	-	Download

Esses métodos foram escolhidos para terem uma ferramenta no sistema por sua relevância e por terem sido desenvolvidos no nosso laboratório, além de que cada um tem uma característica importante, como mostrado na tabela 2. O *Protein Structure Prediction Center*² desenvolve anualmente um espécie de competição em relação aos métodos aplicados ao PSP, com o objetivo de estabelecer o atual estado da arte para esse problema, identificar o que tem melhorado ao longo dos anos e indicar onde os próximos esforços deverão ser focados. Essa competição, chamada de *Critical Assessment of protein Structure Prediction* (CASP), está na versão 11, realizada em 2014 com a participação do 2PG, trabalhando com os alvos sugeridos pela competição, obtendo bons resultados (FACCIOLI; BORTOT; DELBEM, 2014).

²Organizado pelo *US National Institute of General Medical Sciences* (NIH/NIGMS)

Sistemas para biologia computacional

Como foi abordado no capítulo anterior, existem vários métodos aplicados ao PSP, cada um com as suas particularidades de uso, instalação, configuração e funcionamento. Assim, o objetivo desse trabalho é a criação de um sistema que faça a integração de vários métodos e tenha recursos para analisar as soluções geradas por cada um deles, avaliando-as e indicando, através de métodos estatísticos, qual é a melhor.

Para que esses dados gerados nas pesquisas em Biologia Computacional sejam processados e analisados, é necessário o uso de uma infraestrutura computacional capaz de manipular esses dados de forma eficiente. Para isso, muitos *softwares* estão disponíveis para suprir esta demanda tanto para usuários finais, sem o conhecimento técnico computacional, quanto para programadores especialistas em desenvolvimento de sistemas para a pesquisa biológica.

Nesse cenário, cada vez mais bancos de dados com informações biomédicas estão interligados, com a possibilidade de compartilhamento de dados e colaboração entre usuários (ZHANG; CHEUNG; TOWNSEND, 2009). Algumas dessas abordagens são apresentadas nas Seções 4.1, 4.2, 4.3, 4.4, 4.5, 4.6 e 4.7, e na Seção 4.8 é realizada a comparação entre essas abordagens. A Seção 4.8.1 descreve com mais detalhes o *framework* utilizado nesse trabalho.

4.1 GeneSifter

O Geospiza's GeneSifter é um gerenciador de dados e análises para NGS (do inglês, *Next Generation Sequencing*). Esse sistema fornece uma variedade de ferramentas desenvolvidas de acordo com as necessidades encontradas na NGS e exclusivamente para esse contexto. Além disso, suporta várias plataformas (Illumina, Roche) e aplicações (RNA-Seq, ChIP-Seq) para análise e sequenciamento de código genético.

Modelo de *software* sendo executado nas nuvens (*Cloud computing*), criação de *pipelines* para análises com o uso de uma interface amigável, possibilidade de execução de testes em banco de dados de forma simples e acessível por qualquer dispositivo e sistema ope-

racionais são algumas das características mais relevantes do sistema (BASKIN et al., 2009). A Figura 15 mostra o fluxo de execução do sistema.

Figura 15: Geospiza GeneSifter fluxo experimental. GeneSifter abrange todas as fases de experimentos *NextGen* e por isso, os dados podem ser coletados, armazenados, analisados e permanentemente associados com amostras dentro de um único computador. É fácil visualizar este processo contínuo dividindo-o em cinco fases distintas (Adaptado de (BASKIN et al., 2009)).

4.2 Genomatix

Genomatix é uma solução integrada (*hardware* e *software*) também com foco em gerenciamento e análise para a NGS. Na parte de *software*, o Genomatix oferece diversas ferramentas de análise de dados genômicos por meio de uma interface de execução e algumas por linha de comando. No *hardware*, algumas configurações com processadores de alto desempenho e memória estão disponíveis (atualmente, a configuração oferecida é uma estação com 1 processador Intel Quadcore, 32GB de RAM e 4TB de HD).

Fornecer a infraestrutura de processamento necessária para as análises e um conjunto de sistemas integrados são características importantes desta solução. Além disso, a possibilidade de execução das ferramentas, tanto por meio de uma interface quanto por modo texto, dá uma liberdade maior ao usuário. Por fim, o sistema permite que o usuário crie seus próprios *pipelines* de análises dos dados (TEAM, 2011).

4.3 ChIPseeqer

ChIPseeqer é uma solução que disponibiliza uma grande quantidade de ferramentas para processamento e análise ChIP-Seq (do inglês, *Chromatin immunoprecipitation followed by NGS*) de dados genômicos, sendo possível realizar todas as fases dessa análise com o uso dessas ferramentas, que podem ser acessadas de duas formas, por meio de uma interface gráfica e por linha de comando, oferecendo mais flexibilidade de uso.

O recurso disponível ao usuário de combinar vários módulos (ferramentas) disponíveis para a criação de *workflows* específicos para um objetivo ou uma hipótese é a característica mais importante deste *framework*. Com isso, o usuário pode utilizar cada módulo de

uma forma diferente, alcançando resultados e performances diferentes (GIANNOPOULOU; ELEMENTO, 2011). A Figura 16 mostra a interface gráfica do sistema.

Figura 16: Interface de usuário do ChIPseeqer. Em (A) o usuário pode manipular todos os parâmetros das ferramentas disponíveis no *framework*; em (B) mostra os resultados das ferramentas, que consiste em gráficos ou uma tabela que resume as análises realizadas e as estatísticas do que foi executado (Adaptado de (GIANNOPOULOU; ELEMENTO, 2011)).

4.4 ABrowse

ABrowse é um sistema que é executado em um navegador e oferece uma experiência de navegação, pesquisa e análise ao usuário, além de disponibilizar acesso a diversos bancos de dados genômicos e permitir que sejam realizados trabalhos colaborativos entre usuários. O ABrowse também pode ser usado por sistemas que fornecem dados através de páginas customizadas.

Para facilitar a análise pelo usuário, os resultados das pesquisas realizadas neste sistema podem ser enviados para outros *softwares*, como o Galaxy. O ABrowse, que é um *genome browser*¹, é indispensável não apenas para visualização mas também como uma plataforma de acesso a diversas fontes de dados (KONG et al., 2012). A Figura 17 mostra a arquitetura desse sistema.

Figura 17: Arquitetura do ABrowse. O ABrowse possui três camadas: a camada de interação com o usuário, a de solicitação/processamento de dados e a de banco de dados. Além disso, um serviço *web* está disponível para acesso de dados abertos, e tudo o que estiver armazenado no Abrowse pode ser acessado a partir de aplicações de bioinformática externas, para realização de análises posteriores (Adaptado de (KONG et al., 2012)).

4.5 SeqAdapt

SeqAdapt é um sistema que utiliza uma infraestrutura baseada em serviço (SOA, do inglês *Service-Oriented Architecture*)², ou seja, existe um componente central que controla

¹ *Genome Browser* é uma interface gráfica que exibe informações sobre bancos de dados biológicos e genômicos.

² Service-Oriented Architecture (SOA), é baseada na computação distribuída, de forma que as funcionalidades de um sistema devem ser disponibilizadas por meio de serviços conectados através de uma interface ou barramento.

todos os serviços, fazendo com que o sistema se torne flexível e amplamente customizável.

O *software* SeqAdapt permite que novos serviços sejam adicionados pelos usuários à infraestrutura já existente, possibilitando que cada usuário customize-o de acordo com as necessidades de suas pesquisas. O SeqAdapt tem código aberto (*open source*), segue padrões de desenvolvimento de *software* e estrutura de dados, não requer conhecimentos específicos acerca da infraestrutura utilizada para ser customizado e permite que o usuário verifique a execução do código. Essas são as características principais do sistema (BURDICK et al., 2010). A Figura 18 mostra a arquitetura padrão do SeqAdapt.

Figura 18: Arquitetura padrão do SeqAdapt. O sistema SeqAdapt consiste de um certo número de serviços e programas de análise que podem ser substituídos para permitir um elevado nível de personalização. Rastreamento padrão de amostra, gerenciamento de dados e análise de componentes estão disponíveis. Estes componentes podem ser substituídos por outros sistemas de acordo com as necessidades individuais. Todas as interfaces entre os componentes são padronizadas usando REST e JSON (tecnologias de troca de informação entre a estação do usuário e o servidor), possibilitando a criação de aplicações personalizadas para a utilização dos componentes (Adaptado de (BURDICK et al., 2010)).

4.6 LabKey Server

LabKey Server é uma aplicação *web* dividida em módulos, os quais são usados para integração de dados, análise e colaboração. Cada módulo é independente e pode ser utilizado, desabilitado, atualizado ou removido a qualquer momento, encapsulando a lógica da aplicação, os dados e a interface para o usuário que, por sua vez, pode inserir novas funcionalidades de análise, tipos de dados ou outros recursos utilizando o código aberto fornecido.

Fácil extensibilidade e customização de interfaces, análises e visualização; fácil interoperabilidade com fontes internas e externas de dados; práticas ágeis de desenvolvimento de *software*; uso de um método baseado em uma plataforma definida, fazendo com que os custos e erros com o sistema sejam diminuídos, e, mantendo uma estabilidade do sistema

e suporte são algumas das características deste sistema (NELSON et al., 2011). A Figura 19 mostra um diagrama da aplicação.

Figura 19: Fluxograma do LabKey Server. O LabKey integra várias origens de dados para serem utilizados no servidor. Diversas ferramentas de análises e APIs permitem que cientistas, pesquisadores e programadores criem uma rede de colaboração dentro de um único ambiente (Adaptado de (SERVER, 2014)).

4.7 Galaxy

Galaxy é um conjunto de componentes de *softwares* reutilizáveis que podem ser integrados em outras aplicações, encapsulando suas funcionalidades de forma que o usuário tenha acesso somente a interfaces de ferramentas computacionais, que podem ser executadas em vários ambientes de execução, e trabalha com ferramentas e suas relações.

Uma aplicação criada com esse *framework* disponibiliza o acesso à uma ferramenta através de uma interface *web* amigável, a qual possui parâmetros de execução para uma pesquisa computacional. O servidor Galaxy é uma execução dessa aplicação com determinadas ferramentas disponíveis (GOECKS; NEKRUTENKO; TAYLOR, 2010).

4.8 Comparação dos sistemas

A Tabela 3 sumariza as características de todos os *frameworks* anteriormente citados, mostrando que as soluções LabKey e Galaxy são as melhores, por serem gratuitas, altamente customizáveis e terem várias formas de acesso e suporte a execução nas nu-

vens. Como visto, existem diversas soluções, gratuitas ou comerciais, para a questão de infraestrutura e análises.

Tabela 3: Relação de *frameworks* encontrados na literatura

<i>Framework</i>	Sistema Operacional	Nível de customização	Formas de acesso	Licença	Facilidade de paralelização
GeneSifter	Todos	Instalação Ferramentas <i>Workflow</i>	Web Cloud	Comercial	Não
Genomatix	N/A	<i>Workflow</i>	Local Hardware	Comercial	Não
ChIPseeqer	Linux	Instalação <i>Workflow</i>	Local	Gratuito	Não
ABrowse	Linux	Instalação <i>Workflow</i>	Web	Gratuito	Não
SeqAdapt	Todos	Ferramentas <i>Workflow</i>	Web Cloud Local	Gratuito	Sim
LabKey Server	Todos	Ferramentas <i>Workflow</i>	Web Cloud Local	Gratuito	Sim
Galaxy	Linux	Instalação Ferramentas <i>Workflow</i>	Web Cloud Local	Gratuito	Sim

A única restrição do Galaxy em relação ao LabKey refere-se ao sistema operacional, que oferece suporte dos desenvolvedores somente para Linux, mas é possível instalar em Windows, porém o trabalho é bem maior. Como o Galaxy possui características importantes, como a facilidade de integração de *software*, documentação abrangente, uma comunidade de desenvolvimento e uso muito grande, além de suas formas de customização e acesso, ele será detalhado na Seção 4.8.1.

4.8.1 Framework Galaxy

O *Framework* Galaxy é desenvolvido na linguagem Python, com suporte para as versões 2.4 a 2.6. O *framework* pode ser baixado utilizando Mercurial ou Git³, ou se não houver esses programas instalado, está disponível pelo endereço “<http://getgalaxy.org/>”. Nesse pacote, está incluso um servidor *web* e um banco de dados SQL (do inglês, *Structured Query Language*). Diversas atualizações são liberadas frequentemente pelo grupo de desenvolvedores do Galaxy através do controle de versões e o próprio *framework* gerencia automaticamente as dependências de atualização.

³Ferramentas de controle de versões para desenvolvedores de software.

4.8.1.1 Descrição

O *Galaxy* é uma plataforma *web*, *open source* para pesquisas sobre biologia computacional e biomédica, com foco em bioinformática, genômica computacional e análises de dados biológicos.

O Projeto *Galaxy* é desenvolvido pelos laboratórios NSF (*National Science Fundation*), NHGRI (*National Human Genome Research Institute*), *The Huck Institutes of the Life Sciences*, *The Institute for CyberScience at Penn State* e *Emory University*. Ele pode ser acessado pelo endereço: <http://usegalaxy.org/>. Este é um dos projetos *open-source* para pesquisas em bioinformática com um grande volume de dados e resultados que mais cresce no mundo, possui mais de 50 mil usuários cadastrados mundialmente (GALAXY, 2015).

A plataforma *Galaxy* pode ser acessada basicamente de quatro formas:

1. Servidor público do projeto;
2. Instância Local;
3. *Cloud*;
4. Servidores públicos de organizações.

A instância pública do *Galaxy* pode ser acessada via *internet* por qualquer pessoa ou instituto que queira utilizar alguma ferramenta disponível em seus servidores. Essas ferramentas ou foram criadas pelo grupo de desenvolvedores do projeto ou por outros usuários que enviaram para a instância pública uma cópia, tornando-a acessível a todos. Para utilizar os recursos do *site* não é necessário nenhum cadastro prévio, desde que não se tenha a intenção de armazenar alguma informação no sistema a fim de utilizá-la posteriormente. Nesse caso, é necessária a criação de uma conta no servidor. Todo o processamento e armazenamento são feitos pelos servidores nos quais o sistema público está sendo executado, ou seja, toda a estrutura computacional permanece encapsulada frente ao usuário, o qual não precisa saber quando ou onde suas pesquisas serão executadas, importando somente os resultados obtidos. O servidor público do Galaxy oferece uma cota de armazenamento de 5GB para usuários não cadastrados e uma de 250GB para usuários com uma conta registrada. A Figura 20 mostra a tela principal de uma instância pública.

A instância local pode ser baixada e instalada em qualquer computador, servidor ou *cluster*. Isso é possível por ser uma plataforma *open source*. Nessa instalação, é possível fazer diversas configurações com relação aos diretórios onde serão armazenados os arquivos resultantes das ferramentas, o processamento ou *job*⁴, qual banco de dados utilizar (por padrão, o *Galaxy* utiliza o *SQLite*), entre outras. Uma outra possibilidade é a alteração de uma ferramenta já existente para atender algum requisito específico para uma pesquisa,

---

⁴Um *job* é um processo para o *Galaxy*, gerado por uma ferramenta.

Figura 20: Tela principal do *Galaxy* sendo executado pela *internet*.

ou a criação de novas. Todas as ferramentas da instância pública estão presentes na instância local.

Existe também a opção de utilizar o *Galaxy* nas nuvens e funciona, basicamente, como uma instância local. Entretanto, é preciso criar uma conta na *Amazon Elastic Compute Cloud* (EC2). É utilizado o modelo de pagamento *pay-as-you-go*⁵ e definido a quantidade de espaço e instâncias que se deseja utilizar.

E por fim, outras organizações possuem uma instância do *Galaxy* sendo executado em um servidor público, o qual é disponibilizado para que qualquer pessoa possa utilizar as ferramentas específicas do laboratório. Isso é necessário, pois por mais completa que a plataforma seja, ela não seria capaz de suprir as necessidades de todos em suas pesquisas. Como exemplo pode ser citado o servidor: *Genomic Hyperbrowser*, que disponibiliza ferramentas estatísticas para calcular variações biológicas em bancos de dados genômicos. Pode ser acessado pelo endereço: <http://hyperbrowser.uio.no/hb/>.

A facilidade com que as ferramentas disponíveis podem ser utilizadas, a *user-friendly interface* da plataforma como uma característica interessante, a possibilidade de personalização da instalação e desenvolvimento de novas funcionalidades fazem com que o sistema ganhe um espaço satisfatório em relação aos sistemas mais utilizadas em pesquisas na bioinformática.

4.8.1.2 Características

O *Galaxy* oferece diversas ferramentas para análises com uma grande quantidade de dados, é uma plataforma gratuita, transparente, acessível e compartilhável. Este sistema permite que usuários sem conhecimentos em programação executem, de forma simples, as ferramentas disponíveis em seu servidor público, podendo definir quais parâmetros forem

⁵Forma de pagamento que varia de acordo com a quantidade de espaço adquirido.

necessários para a sua pesquisa. Outra característica interessante é que os resultados de uma pesquisa podem ser compartilhados e acessados por outros usuários, assim essas informações podem ser utilizadas por outras pessoas, em outras pesquisas, sendo possível o entendimento dessas análises. Além disso, toda pesquisa pode ser executada quantas vezes necessária, com ou sem alteração. Essas características são detalhadas abaixo.

Acessibilidade

O *framework* Galaxy, por ser uma plataforma baseada na *web*, oferece uma acessibilidade muito grande para os seus usuários, pois a única exigência, ou requisito, para acessar as ferramentas e recursos disponíveis no servidor público do Galaxy é um navegador (*web browser*) com conexão com a *internet*, sendo isso, um fator que diferencia o Galaxy de diversas outras ferramentas baseadas em linha de comando (*command line*).

Este framework oferece diversos recursos para seus usuários sem exigir que eles tenham conhecimento específico em computação ou que tenham que aprender como o sistema funciona internamente, ou mesmo o funcionamento de uma única ferramenta, fazendo com que os usuários tenham liberdade e facilidade no uso do sistema.

Disponibilizando uma interface *web* unificada e fácil de utilizar para os usuários, o Galaxy permite a importação de dados genômicos (*dataset*) para o ambiente de trabalho do usuário (*workspace*) a partir de diversas fontes de dados, como seus próprios arquivos locais ou a partir de banco de dados de genes, como o NCBI (*National Center for Biotechnology Information*, <http://www.ncbi.nlm.nih.gov/>) e a aplicação de diversas ferramentas de análises computacional para esses dados. Além disso, o *framework* é aberto, o que permite que novas ferramentas sejam criadas e incorporadas ao sistema, tornando-o ainda mais acessível e customizável.

Uma ferramenta é uma parte do sistema que tem à ela associado um interpretador escrito em qualquer linguagem de programação (Python, Perl, C, Java, etc) que define a sua funcionalidade e tem o controle de parâmetros de entrada e saída (*input* e *output*, respectivamente). Dessa forma, o Galaxy incorpora a ferramenta criada ao *framework*, gera uma interface com as opções definidas e a executa de forma abstrata, pois o modelo de integração do Galaxy permite isso.

O *framework* pode se mostrar menos flexível para pesquisadores que possuem conhecimento computacional do que se fosse trabalhar em uma linguagem de programação diretamente, mas o comportamento de desenvolvimento da ferramenta faz com que a computação se torne acessível, adicionando transparência e reprodutibilidade para aqueles pesquisadores e usuários que são avessos a utilizar ferramentas baseadas em linha de comando (GOECKS; NEKRUTENKO; TAYLOR, 2010).

Transparência

Pesquisas transparentes e abertas são essenciais para o processo da ciência (BLANKENBERG et al., 2010). Em um projeto de pesquisa utilizando o Galaxy, em que são importados dados e executadas diversas ferramentas, os usuários geram diversos dados de resultados, caracterizando-se como os históricos de cada passo da análise. Esses dados podem ser armazenados no Galaxy, estando disponíveis para serem executados novamente ou serem compartilhados. Essa possibilidade de armazenamento e compartilhamento dos resultados e históricos gera uma transparência para experimentos computacionais.

A transparência nos resultados é possível através do Galaxy de três maneiras:

1. Modelo de compartilhamento de itens (banco de dados, históricos e fluxos de trabalho) em um repositório público de itens publicados;
2. *Framework web* para visualização de itens publicados e compartilhados;
3. Páginas *web* personalizadas contendo detalhes da pesquisa.

O modelo de compartilhamento dos itens gerados na pesquisa permite ao usuário compartilhar os resultados obtidos, os dados utilizados, e outras informações pertinentes ao projeto com qualquer outro usuário do Galaxy, através de *links* da *internet*. Cada item publicado possui um *link* que fica disponível em uma lista de itens publicados no repositório público do Galaxy, onde é possível aos usuários pesquisarem um item por diversos filtros, como autor, *tag* ou anotações (*annotations*), de acordo com o seu interesse. Esses itens publicados são exibidos em um *framework web* contendo os dados pertinentes a cada item, seja um *dataset* ou *history*. Em cada página do *link* publicado, é permitido ao usuário fazer o *download* dos resultados obtidos ou mesmo importá-los para o seu *workspace* e assim utilizá-los em uma outra ferramenta.

As páginas personalizáveis são a principal e mais completa forma de comunicação de resultados do Galaxy. Essas páginas garantem acessibilidade, transparência e a reprodutibilidade de pesquisas computacionais. Com essas páginas é possível descrever detalhes completos da pesquisa, quais as ferramentas utilizadas, os passos que foram executados e seus respectivos *output*, e também podem conter itens do Galaxy, como *datasets*, *histories* e *workflow* utilizados (Figura 21).

Essas páginas exibem cada detalhe da pesquisa, como textos, imagens e os itens do Galaxy. Cada item pode ser visualizado individualmente e expandido se houver a necessidade de mais detalhes dos passos executados em cada histórico ou *workflow*. Todas essas ações são permitidas sem a necessidade do usuário deixar a página, ou abrir uma outra para cada item. Os itens informados nas páginas podem ser utilizados por outros usuários, importando-os para o seu *workspace*. É possível importar um *history* e começar a utilizá-lo em outra ferramenta, ou importar um *workflow* e *datasets*, modificá-los e executá-los, tudo ao mesmo tempo, na mesma interface.

Figura 21: Páginas do Galaxy. É demonstrado os detalhes da pesquisa e os itens do Galaxy gerados e disponibilizados. *Links* para salvar, importar ou exibir com mais detalhes estão disponíveis em cada item (Adaptado de (GOECKS; NEKRUTENKO; TAYLOR, 2010)).

Compartilhável

Quando uma análise é executada no Galaxy, os detalhes pertinentes a cada passo são salvos no que é chamado de *history* (histórico). Cada histórico pode ser executado novamente com ou sem alterações nos parâmetros da ferramenta. Como dito na Seção 4.8.1.2, esses históricos podem ser compartilhados e reproduzidos por outros usuários, garantindo a reprodutibilidade do *framework* e transparência (BLANKENBERG et al., 2010).

Todos os itens do Galaxy têm espaço para a inserção de anotações, ou seja, descrições sobre os dados utilizados e os resultados obtidos. A cada execução de uma ferramenta são geradas itens para cada passo executado, que pode conter todos os detalhes informados na sua geração, como *datasets* importados, parâmetros informados, e *outputs* gerados. Essas anotações são salvas nos históricos, ficam visíveis e disponíveis para edição.

Uma única anotação, como um *history* ou *workflow*, demonstra os passos executados na pesquisa, mas ainda assim não é suficiente para que outro usuário entenda todo o processo. Para isso, são utilizadas *tags* para descrever o porquê dos passos executados. Essas *tags* auxiliam também nas consultas por itens compartilhados no repositório público do Galaxy e garantem a reprodutibilidade da plataforma, pois elas são características importantes para que outros usuários encontrem um item desejado e use-o. O reuso de informações é outra atividade necessária para reprodutibilidade. É possível visualizá-las no painel de histórico do Galaxy no *workspace* do usuário. Além disso, é possível criar

novas *tags* e anotações para um histórico no próprio *workspace*, sem a necessidade de sair desse ambiente. A Figura 22 representa um *workspace* do Galaxy.

Figura 22: Galaxy *Workspace*. Ambiente de trabalho do Galaxy. O painel A é o de ferramentas. O B é onde mostra a ferramenta em execução e seus parâmetros. E o C corresponde ao *History* (Histórico) de todas as análises executadas (Adaptado de (GOECKS; NEKRUTENKO; TAYLOR, 2010)).

As definições de execução de uma ferramenta, considerando os valores para os seus parâmetros e o *dataset* utilizado, podem ser definidos em uma espécie de *template* no Galaxy, o que é chamado de *Workflow* (fluxo de trabalho). O usuário pode configurar um *workflow* para uma ferramenta ou para um conjunto, conectando-as, para ser executada quantas vezes for necessário. Para cada execução, a mesma ferramenta com os mesmos parâmetros para um mesmo conjunto de dados vai ser executada. Existe uma interface visual *user-friendly* para a criação desse *workflow*.

O *workflow* pode ser configurado para ser executado interligando qualquer ferramenta mediante uma verificação de compatibilidade, fazendo com que o resultado de uma ferramenta (*output*) seja utilizado com um *dataset* para outra (*input*). Um *workflow* também pode ser gerado a partir de um *history* de uma ferramenta previamente executada. O pro-

cesso de execução do *workflow* segue as mesmas regras de execução de uma ferramenta simples. Em sua execução, ele gera um grupo de *datasets* no *history* do *workspace*, o qual o usuário pode inserir *tags*, anotações, executar novamente, ou compartilhar. A Figura 23 representa um *workflow* do Galaxy.

Figura 23: Editor do Galaxy *Workflow*. O editor do Galaxy *workflow* possui quatro áreas: a A corresponde ao painel de ferramentas; a B é painel de edição; a C é a barra de navegação; e a D é o painel de detalhes do *workflow*. As ferramentas do painel de ferramentas podem ser arrastadas para dentro da área de edição, e assim configurar cada passo a ser executado (Adaptado de (GOECKS; NEKRUTENKO; TAYLOR, 2010)).

4.8.1.3 Estatísticas

O *Framework* Galaxy é muito utilizado na comunidade de pesquisas biomédicas, demonstrando ser uma ótima plataforma para novos projetos. As estatísticas disponíveis de usuários, acessos, linhas de códigos e artigos publicados sobre e com o uso do Galaxy vêm confirmando-o como um sistema robusto e completo.

Com início do desenvolvimento do projeto em 2006, o Galaxy vem crescendo ao longo dos anos. Uma comunidade de usuários e desenvolvedores é mantida pelos administradores do sistema com o uso de listas de emails (*mailing lists*, em inglês), a qual é utilizada para a troca de mensagens a respeito de dificuldades, dúvidas, novidades e ajuda entre os membros e usuários ativos do Galaxy. A Figura 24 mostra a quantidade de usuários registrados nas listas a partir do ano de 2011, quando esse recurso começou a ficar disponível.

O Galaxy mantém uma instância pública que permite que qualquer usuário a acesse e execute as ferramentas disponíveis. Com uma cota disponível para cada usuário de 250GB, é possível utilizar a infraestrutura de supercomputadores oferecida pelo Galaxy e executar até 6 *jobs* simultaneamente para diversas ferramentas. A Figura 25 mostra o crescimento de usuários registrados na instância pública.

Figura 24: Número de membros das listas de emails do Galaxy (Adaptado de (GALAXY, 2015)).

Figura 25: Número de usuários registrados na instância pública do Galaxy (Adaptado de (GALAXY, 2015)).

4.8.2 Considerações de uso

O nível de abstração para os usuários de sistemas para biologia computacional pode ser considerado alto, pois em sua maioria, não é exigido nenhum conhecimento computacional acerca do funcionamento do *framework* para utilizá-lo. Estudos na área de engenharia de *software* mostram que os desenvolvedores tendem a ser mais produtivos quando utilizam de um alto nível de abstração no desenvolvimento, como linguagens de programação, bibliotecas e ambiente de desenvolvimento de *software* completos e ricos (BROOKS, 1987).

Sendo assim, o Galaxy se destaca por gerar uma produtividade alta para seus usuários.

Porém, esse nível de abstração pode fazer com que detalhes importantes sejam encobertos, e, por isso, passem despercebidos pelo usuário, podendo causar problemas na execução do sistema. Por exemplo, ao utilizar uma ferramenta disponível no Galaxy sem que seja utilizado os dados e parâmetros corretos, o usuário pode obter um resultado não esperado e, com isso, julgar que a ferramenta não serve para o seu experimento. Isso ocorre pelo fato de que o Galaxy não possui todos os pacotes de ferramentas, dados, ou aceitação de todos os tipos de dados que os usuários possam precisar. Por outro lado, todas as outras características do Galaxy, como produtividade e transparência, superam essas limitações de análise.

Outro problema para quem utiliza um conjunto muito grande de dados é o poder computacional necessário para as análises. Para tal, as instâncias locais seriam mais apropriadas, pois o Galaxy seria instalado em uma infraestrutura própria, com o intuito de resolver algum problema específico de um laboratório (SCHATZ, 2010).

Metodologia proposta

Este capítulo descreve o desenvolvimento do sistema Koala, um sistema computacional baseado em web para predição e análises de estruturas de proteínas, para integração de algoritmos aplicados ao Problema de Predição de Estrutura de Proteínas (PSP) e ferramentas de análises de resultados.

O Koala foi desenvolvido usando como base o *framework* Galaxy, descrito no Capítulo 4, como sendo um sistema para pesquisas biomédicas amplamente utilizado por pesquisadores de todo o mundo e com uma quantidade muito extensa de recursos computacionais, além de ser *open-source*, ter uma interface *web* amigável e funcionar em *cluster* e *cloud*.

O sistema desenvolvido oferece o acesso a diversos algoritmos aplicados ao PSP, através de uma interface amigável, e toda a lógica dos *softwares* utilizados fica encapsulada para o usuário do sistema. Os módulos (ferramentas) desenvolvidos podem ser interligados e executados de forma sequencial (*workflow*), fazendo com que as pesquisas possam ser executadas de maneira intuitiva. Além disso, tudo o que é feito no sistema pode ser compartilhado no servidor para outros usuários, favorecendo a reprodutibilidade das pesquisas. Esses processos serão detalhados nas próximas seções.

A Seção 5.1 aborda as orientações para o desenvolvimento de interfaces de sistemas *web*, para garantir a usabilidade por parte do usuário. A Seção 5.2 detalha os métodos que foram utilizados em alguns módulos desenvolvidos no sistema, com o intuito de comparar os resultados de alguns algoritmos. A Seção 5.3 descreve os módulos que utilizam métricas de comparação de estruturas, com o objetivo de quantificar, através de um *score*, o quanto uma estrutura predita é similar a uma estrutura de referência. A Seção 5.4 aborda os detalhes da instalação, configuração e estrutura técnica computacional do sistema desenvolvido. E, por fim, a Seção 5.5 detalha todos os módulos desenvolvidos para o Koala.

A Figura 26 apresenta um esquema geral do Koala, os tipos de arquivos de entrada e saída suportados, as bibliotecas externas utilizadas e os *frameworks* que estão disponíveis no servidor, além de demonstrar a infraestrutura computacional disponível. O item 'base' representa a parte lógica utilizada em todas as ferramentas desenvolvidas.

Figura 26: Esquema geral do Koala, detalhando os tipos de arquivos suportados, as bibliotecas utilizadas, a infraestrutura computacional e os *frameworks* que fazem parte deste sistema.

O nome Koala foi escolhido no intuito de evitar o uso de acrônimos a partir de palavras ligadas ao contexto do sistema. Além disso, o animal Koala é um bicho dócil e sua interação com humanos é realizada de forma tranquila. Espera-se que o Koala Server também ofereça uma interação com o usuário mais amigável.

5.1 Projeto de Interfaces

No processo de desenvolvimento de *software*, o design de interfaces amigáveis se tornou importante ao longo dos anos, pois muitos sistemas computacionais são utilizados por usuários sem conhecimento técnico, como é o caso na área da Biologia Computacional.

Nas primeiras décadas do desenvolvimento computacional, a preocupação com a interface não existia. Segundo (NORMAN, 1988), uma mudança de atitude era necessária, pois, antes, as pessoas deveriam se adaptar às tecnologias que elas gostariam de utilizar, estudando-as, mas agora as tecnologias deveriam se adaptar às pessoas (usuários).

Para isso, muito tem se estudado sobre a interação entre homem e computador, a

qual é muito importante para que o usuário consiga utilizar o *software* da melhor forma possível, executando ações e interpretando a resposta do programa (PREECE et al., 1994). A Figura 27 demonstra esse processo de interação entre usuário e sistema.

Figura 27: Modelo de interação usuário-computador (Adaptado de (BARBOSA; SILVA, 2010)).

De acordo com (MORAN, 1981), a interface do usuário deve ser entendida como sendo a parte de um sistema computacional com a qual uma pessoa entra em contato de forma física, perceptiva e conceitualmente. Ou seja, existem alguns componentes, físico e conceitual, com os quais o usuário interage, aumentando a importância de uma boa interface.

A qualidade de uma interface bem projetada, que garante ao usuários recursos de utilização fácil e transparente, pode ser chamada de usabilidade, que pode ser definida como a medida de quanto um sistema computacional facilita o aprendizado, ajuda o usuário a se lembrar dos recursos que aprendeu, reduz a probabilidade de erros e os deixa satisfeitos com o sistema (DONAHUE; WEINSCHENK; NOWICKI, 1999).

O que os usuários querem, ao utilizar um *software*, são boas ferramentas (sistemas computacionais, operacional, de linguagens e de entradas de dados são alguns exemplos) que sejam fáceis de aprender e que os ajude a realizar o seu trabalho. Querem também que o software não os retarde, confunda, facilite a prática de erros ou dificulte a finalização de seus trabalhos (CONSTANTINE, 1995).

Existe um processo para desenvolvimento de projeto de interface e análise no qual algumas etapas precisam ser seguidas visando um melhor resultado. São quatro atividades, quanto a interface, que foram definidas por (MANDEL, 1997): análise e modelagem, projeto, construção e validação. A Figura 28 mostra um espiral desse processo, onde cada atividade pode ser executada mais de uma vez, de acordo com as linhas, indicando que novos recursos foram desenvolvidos.

Neste projeto de mestrado, as três primeiras etapas foram realizadas para cada ferramenta desenvolvida. Entretanto, a última, validação de interfaces, não será realizada a priori, por não estar no escopo deste projeto.

Existem várias soluções para a criação de interface amigável para processamento e análise de dados na área da Biologia Computacional, como citado no Capítulo 4. Neste

Figura 28: Processo de desenvolvimento de projeto de interfaces (Adaptado de (PRESSMAN, 2011)).

trabalho, para o desenvolvimento das interfaces amigáveis, foi utilizado o *framework* Galaxy.

5.2 Métodos para avaliação de resultados

Nos módulos de análise, algumas métricas são utilizadas a partir dos resultados obtidos com a execução das ferramentas. Esses métodos avaliam tanto em nível de solução dos algoritmos, indicando qual é a melhor entre um conjunto de soluções, quanto em relação aos métodos utilizados, avaliando os seus desempenhos.

5.2.1 *Ranking* de dominância

Ranking de dominância é uma técnica muito utilizada para classificação e ordenação de métodos (KUNCHEVA; RODRIGUEZ, 2007). Para realizar esse *ranking*, cada método é comparado com todos os outros em um conjunto de métodos, um-a-um, e são calculados os ganhos e perdas, em cada sequência. A Equação 1 demonstra esse processo, onde é verificado se o valor de x_i é maior do que o valor de x_j , e caso seja, a quantidade de *vitórias* (w_i) é incrementada, e caso não, a quantidade de *derrotas* (l_i) é incrementada (HEMELRIJK; WANTIA; GYGAX, 2005).

$$w_i = \begin{cases} 1 & x_i > x_j \\ 0 & \text{senão} \end{cases} \quad l_i = \begin{cases} 1 & x_i < x_j \\ 0 & \text{senão} \end{cases} \quad (1)$$

Esses números referente a perdas e ganhos são usados para determinar a dominância de um método, que é o resultado da diferença entre o número de ganhos e o número de perdas em todas as comparações realizadas. Assim, quanto maior a dominância de um método, melhor será seu posicionamento no *ranking* em relação a todos os outros (HEMELRIJK; WANTIA; GYGAX, 2005). A Equação 2 demonstra esse processo, na qual é definido que a dominância de um método (d_i) é o resultado da subtração da somatória da quantidade

de vitórias pela somatória da quantidade de derrotas (KUNCHEVA; RODRIGUEZ, 2007).

$$d_i = \sum w_i - \sum l_i \quad (2)$$

Nesse trabalho, foi desenvolvido o algoritmo para calcular o *ranking* de dominância e é utilizado para ranquear vários métodos aplicados no PSP pela ferramenta *Dominance Ranking* (Seção 5.5.3.2) disponível no servidor.

5.2.2 Fronteira de pareto e dominância

Nos algoritmos evolutivos multiobjetivos várias funções-objetivos precisam ser calculadas de forma a maximizar ou minimizar as soluções, simultaneamente. Esses objetivos, por sua vez, podem ser conflitantes, uma vez que um precisa ser maximizado, o outro minimizado, ou um é mais relevante do que o outro. Portanto, o conjunto de soluções possui mais do que uma solução ótima e a fronteira de Pareto serve para indicar quais são as melhores, ou seja, é mais interessante estimar um conjunto de soluções eficientes.

Pode-se usar como exemplo o problema de tomada de decisão apresentado por Ticona (2003), que apresenta a questão de ter que comprar um carro, o qual possua o melhor preço e conforto (TICONA, 2003). A Figura 29 ilustra as opções encontradas. O objetivo nesse caso é maximizar o conforto e minimizar o preço. Assim, as soluções 1 e 2 podem ser descartadas, pois a solução 5 oferece melhores condições de preço e conforto. Já as soluções 3, 4 e 5 podem ser consideradas as melhores opções de compra, e demonstram o confronto dos objetivos, pois quanto maior o conforto maior o preço, e vice-versa (TICONA, 2003).

A dominância de Pareto é utilizada para comparar e ordenar as soluções, gerando um *ranking* com as melhores encontradas. Essa dominância obedece a regra: Uma solução A domina a solução B se, e somente se, $f(A) \leq f(B)$ e $f(A) \neq f(B)$, ou seja, a desigualdade é atendida em pelo menos um dos objetivos. A relação de dominância pode ser descrita como: $f(A) \prec f(B)$ (DEB, 2001).

No exemplo de Ticona (2003), as soluções 1 e 2 são dominadas pelas soluções 3, 4 e 5. Assim, as opções 1 e 2 compõem o *conjunto de soluções dominadas*, enquanto que as opções 3, 4 e 5 compõem o *conjunto de soluções não-dominadas*, também conhecido como *conjunto Pareto ótimo*, o qual pode ser representado em um espaço cartesiano formando a *Fronteira de Pareto* (TICONA, 2003).

No contexto de proteínas, as propriedades físicas, como energia potencial, interações de hidrogênio, entre outras, podem ser uma função-objetivo diferente, mas empregadas simultaneamente.

Uma vez que as soluções são separadas por fronteiras, é verificado qual a solução domina mais nessa fronteira. As soluções são ordenadas de acordo com a quantidade de soluções dominadas. Assim, quem está na posição 1 da fronteira 0 é quem mais domina e não é dominado por nenhuma outra solução.

Figura 29: Exemplo que ilustra várias opções de compra de carro, considerando o seu custo e conforto (TICONA, 2003).

A ferramenta desenvolvida nesse trabalho *Sort by Front Dominance* (Seção 5.5.3.3) utiliza o algoritmo disponível no 2PG que aplica essa técnica.

5.3 Métricas para comparação de estruturas

Os métodos de predição geram conformações tridimensionais da proteína, com a descrição completa com os resíduos e átomos, e as relativas posições de cada um no espaço 3D. Existem alguns métodos que verificam a similaridade de uma conformação comparando-a com uma estrutura tridimensional de uma proteína modelo (ou de referência). Os principais métodos utilizados pelo CASP são RMSD, TM-Score e GDT-TS.

No Koala, também foram criadas ferramentas de análises que realizam a comparação dos modelos preditos pelos algoritmos disponíveis no sistema com estruturas de referência. Essas três métricas RMSD, TM-Score e GDT-TS estão implementadas no servidor criado.

5.3.1 RMSD

O RMSD (do inglês, *Root-Mean-Square Deviation*) é uma medida que demonstra a distância média entre os átomos, normalmente da estrutura principal, de duas proteínas alinhadas. Na comparação, são utilizadas dois modelos tridimensionais, um modelo predito e outro que demonstra a estrutura nativa da proteína. A semelhança entre os modelos é feita pelo RMSD com base nos valores atômicos do $C\alpha$ das duas proteínas sobrepostas (RAO; ROSSMANN, 1973) (MAIOROV; CRIPPEN, 1994).

O RMSD gera um *score* que quantifica a similaridade entre dois modelos usando como base as coordenadas atômicas do $C\alpha$ da cadeia principal. O valor alto do RMSD significa que as estruturas não são similares ou possuem pequenas partes iguais, e quanto mais

próximo de 0 significa que as estruturas são similares (MAIOROV; CRIPPEN, 1994). O *score* do RMSD é dado em *ångström*(Å)¹.

A ferramenta *Calculate RMSD* (Seção 5.5.3.4) implementa essa técnica e utiliza um programa disponível no *framework* GROMACS (HESS et al., 2008) para realizar o cálculo do RMSD.

5.3.2 TM-Score

O TM-Score (do inglês, *Template Modeling Score*) é um método para calcular a similaridade entre duas estruturas de proteínas com estruturas terciárias diferentes. Esse método estende o que é realizado nos métodos de GDT-TS (ZEMLA et al., 1999) e MaxSub (SIEW et al., 2000) para gerar uma nova fórmula de calcular a similaridade entre proteínas, de forma que a comparação seja *full-length* (estrutura completa da proteína) e sem a dependência do tamanho da proteína (ZHANG; SKOLNICK, 2004).

O *score* calculado pelo TM-Score varia de $[0, 1]$ e demonstra a diferença entre duas estruturas de proteínas. Um *score* igual a 1 significa uma igualdade completa entre as estruturas. Valores menores do que 0.20 demonstram que são estruturas proteicas não relacionadas. Já valores maiores do que 0.5 demonstram que as estruturas podem assumir o mesmo dobramento (*fold*) (ZHANG; SKOLNICK, 2004).

Estudos demonstram que proteínas com *score* igual a 0.5 tem uma probabilidade de 37% de ser da mesma topologia e 13% de ter o mesmo dobramento. Essa probabilidade aumenta caso o valor do TM-Score seja maior do que 0.5 (XU; ZHANG, 2010). O TM-Score também é utilizado no CASP como uma métrica para validar os modelos preditos pelos métodos participantes.

A ferramenta *Calculate TM-Score* (Seção 5.5.3.5) implementa essa técnica e utiliza o programa disponível por Zhang (2004) para realizar o cálculo do TM-Score.

5.3.3 GDT-TS

O GDT-TS (do inglês, *Global Distance Test - Total Score*) é um método para identificar a similaridade entre duas estruturas de proteínas, desde que possuam uma sequência de aminoácidos idêntica e estruturas terciárias diferentes (ZEMLA et al., 1999). O GDT é considerado mais eficiente do que outros métodos que analisam a similaridade entre proteínas por não ser sensível a pequenas perturbações nas estruturas, como *loops* (voltas) ou regiões mal formadas após uma modelagem ruim (ZEMLA, 2003).

Essa métrica é normalmente usada para validar a predição de estruturas de proteínas e comparar com estruturas determinadas experimentalmente por raio-x ou ressonância

---

¹O *ångström* é uma unidade de medida que corresponde a 10^{-10} m, ou 0,1 nanômetro. É muito usada na física para lidar com grandezas da ordem de átomos ou distância entre planos cristalinos.

magnética, e por isso é a principal métrica utilizada no CASP para verificar os modelos gerados pelos preditores.

Para realizar o cálculo do GDT, são definidos valores de corte (*cutoff*), isto é, só são considerados os átomos que possuírem a distância menor ou igual ao valor de corte definido. Assim, quanto maior o valor de corte maior pode ser o *score* do GDT (ZEMLA et al., 1999) (ZEMLA, 2003).

A ferramenta *Calculate GDT-TS* (Seção 5.5.3.6) implementa essa técnica e utiliza o programa disponível por Zhang (2004) para realizar o cálculo do GDT-TS.

5.4 Aspectos técnicos de instalação e desenvolvimento

Para desenvolver uma infraestrutura computacional em nuvens com diversas ferramentas disponíveis para pesquisas com predição de proteínas, foi criada uma instância do *framework* Galaxy e configurada com novos módulos, recebendo o nome de Koala. A Seção 5.4.1 aborda a instalação dessa instância na infraestrutura de *cloud computing* da USP. Na Seção 5.5 são mostradas as ferramentas que foram desenvolvidas para a execução de alguns algoritmos aplicados no PSP e também ferramentas de análises dos resultados.

5.4.1 Instalação do Koala no Cloud USP

A instalação do Galaxy e o desenvolvimento de ferramentas não é uma tarefa trivial dependendo do ambiente computacional no qual será utilizado esse sistema. Portanto, o ambiente de *cloud* da USP e aspectos de rede e internet exigiram um certo conhecimento técnico em computação para a instalação e configuração do Koala.

O Cloud USP (USP Nuvem) é um projeto iniciado em 2010 pela Superintendência de Tecnologia de Informação (STI) da USP, e tem como objetivo prover uma infraestrutura com servidores virtuais de forma a suprir as necessidades da instituição e de seus usuários (STI, 2010).

O paradigma Computação nas Nuvens ("*Cloud Computing*") é um modelo que possibilita um acesso conveniente e por demanda (*on-demand*) a recursos computacionais compartilhados que podem ser, rapidamente, disponibilizados com um nível de esforço pequeno. O modelo de *cloud* é composto pelos modelos de serviços SaaS (*Software as a Service*), PaaS (*Platform as a Service*) e IaaS (*Infrastructure as a Service*) (MELL; GRANCE, 2011). O Cloud USP pode ser considerado um IaaS, pois provê processamento, armazenamento, recursos de rede e todos os outros componentes para que seja executada qualquer aplicação. A Figura 30 demonstra como é feita a separação da infraestrutura em nuvem e os clientes que a utilizam.

O Koala só pode ser instalado em sistemas operacionais com base Unix (Linux e Mac). No servidor disponível pelo Cloud USP, está instalado um sistema operacional Ubuntu

Figura 30: Diagrama da *cloud computing*. Esse modelo tem três camadas bem definidas, com diversos recursos em cada uma delas. Isso fica invisível para o usuário, que pode acessar todos os recursos de vários dispositivos (Adaptado de (TECHONOMY, 2012))

12.04. Abaixo são descritas, resumidamente, as instalações e configurações que foram realizadas neste servidor para instalação do Koala:

- Instalação de pacotes e suas dependência: diversos pacotes, bibliotecas e aplicações precisaram ser instaladas, como o *git*, *nginx*, *python-dev*, *openssl* e outras;
- Galaxy: A partir do *Git*, o código-fonte foi baixado e instalado. O *script* de execução do Galaxy faz a instalação das bibliotecas necessárias na primeira vez que é executado;
- Servidor *web nginx*: é um servidor *proxy HTTP*, o qual foi configurado para permitir o acesso ao sistema através da *internet*;
- PostgreSQL: banco de dados relacional usado pelo Koala. É nele que são armazenados os usuários, os históricos e todos os itens do servidor;
- Pymol: biblioteca utilizada para a criação de imagens moleculares a partir do arquivo PDB da proteína;
- Highcharts: uma *API* em *JavaScript* utilizada para a geração de gráficos *on-the-fly* em páginas *HTML* usadas como saída de alguns módulos desenvolvidos;

- JSmol: suíte de *softwares HTML5* para visualização 3D de estruturas moleculares em tempo real direto no navegador;
- Pulsar: sistema distribuído que cria um serviço em diversas máquinas clientes para execução de *jobs* em vários computadores.

Como o Koala é escrito em Python, para o servidor ser executado é preciso a criação de um ambiente virtual chamado de *virtualenv*, no qual vários pacotes foram instalados e configurados. Dessa forma, tudo o que é instalado nesse ambiente virtual não interfere no que estiver no sistema operacional, possibilitando que diversos servidores diferentes com dependências específicas sejam executados no mesmo computador.

O código-fonte do Koala está disponível através do Github, pelo link <https://github.com/adeleicibus/koala-server>; a documentação está sendo desenvolvida e pode ser acessada pelo link <http://koala-server.rtfid.org>.

Figura 31: Servidor Koala. O painel à esquerda mostra as seções de ferramentas desenvolvidas. Além disso, exibe alguns *workflows* gerados com essas ferramentas e compartilhados no servidor, ficando acessíveis para todos os usuários. O painel no centro é a página inicial do Koala e onde as ferramentas são exibidas.

5.4.2 Pulsar

O Koala possui diversas ferramentas de predição de estruturas de proteínas e análises, as quais podem exigir um poder computacional muito grande. A infraestrutura onde o Koala foi instalado inicialmente fornecia uma qualidade de processamento e memória razoáveis, compartilhada por todas as ferramentas disponíveis no sistema.

Entretanto, esse único computador onde o Koala foi instalado não possuía recursos computacionais para atender muitos usuários ou executar um número muito grande de *jobs*. Isso se torna ainda mais evidente quando alguns dos *softwares* usados no Koala não foram desenvolvidos para processamento paralelo, consumindo ainda mais tempo de CPU. Nesse computador, era possível executar 5 *jobs* por usuário, e disponibilizar uma cota de 2GB para cada usuário.

Um estudo com o serviço público do Galaxy mostrou que mesmo oferecendo uma infraestrutura computacional robusta, os *jobs* criados nessa instância ficavam em média 400 minutos ($\sim 6H$) em espera para ser executado, com um total de 300 mil iniciados pelos usuários (AFGAN et al., 2015). Isso evidenciou a necessidade de distribuir essas requisições em vários computadores menores, diminuindo custos com a compra e manutenção de *clusters* e servidores de grande porte. Assim, foi criado o Pulsar², um servidor dedicado que pode ser executado em computadores na *cloud* para execução de *jobs* remotamente. A Figura 32 mostra como funciona o mecanismo de compartilhamento de *jobs* entre o servidor público do Galaxy e servidores Pulsar, chamado de *cloud bursting* (AFGAN et al., 2015).

Figura 32: Configuração do Pulsar na *cloud*. Através de um arquivo chamado *mapper*, regras são definidas para selecionar qual servidor Pulsar vai receber o *job*. Assim, é possível conectar diversos computadores para a execução das ferramentas no servidor (Adaptado de (AFGAN et al., 2015)).

Para aumentar a capacidade de processamento do Koala e permitir que as ferramentas disponíveis sejam executadas em um ambiente mais robusto, foi utilizado o Pulsar para compartilhar a execução dos *jobs* com outras máquinas também disponibilizadas pela Cloud USP. Assim, a infraestrutura do Koala passou de um único computador, com uma CPU de 8 núcleos, 8GB de RAM e 1TB de armazenamento, para 12 computadores com a mesma configuração, um total de 96 núcleos, 96GB de RAM e 12TB de armazenamento. Com essa infraestrutura, cada usuário tem uma cota de 50GB e pode executar até 15 *jobs*

²Disponível em: <https://pulsar.readthedocs.org/>

em paralelo, três vezes mais do que a configuração anterior. Isso vai permitir aos usuários maior agilidade na execução dos *jobs* e menos tempo de espera para receber os resultados. A Figura 33 mostra como é feita essa divisão em apenas quatro servidores, apenas para exemplificar.

Figura 33: Processo de compartilhamento de *jobs*. São 12 computadores disponíveis para executar todas as ferramentas do servidor, e estão em um ambiente de *cloud* executando o servidor Pulsar. Cada computador tem uma fila de *jobs* de algumas ferramentas.

O compartilhamento de *jobs* das ferramentas é feito através de regras definidas e que são verificadas no momento da execução da ferramenta. Isso significa que, de acordo com o uso, um *job* de uma ferramenta pode ser executado no computador A ou no computador B. Para isso, basta definir regras diferentes para as ferramentas (em arquivos chamados de *mapper*), que vai desde a seleção da ferramenta X para executar no computador A, até a verificação de carga e o envio do novo *job* para o computador mais ocioso. Esse processo é dinâmico, gerando um balanceamento de carga entre os nós.

Para os usuários, esse processo de compartilhamento de *jobs* é totalmente transparente, uma vez que toda a divisão acontece através de arquivos de configuração e regras estabelecidas no Koala, sem a intervenção do usuário nesse processo. Assim, o usuário envia o *job* e o servidor se encarrega de encontrar o local de execução e apresentar os resultados.

5.5 Módulos desenvolvidos

Com o servidor Koala instalado e configurado, algumas ferramentas (módulos) foram desenvolvidas para que a integração de diversos algoritmos aplicados ao PSP estivesse acessível através de uma interface amigável e em uma infraestrutura robusta e com alta capacidade de processamento.

Outras ferramentas que implementam uma interface de execução para vários algoritmos do Gromacs foram desenvolvidas pelo aluno de iniciação científica Lucas Exposto, do Prof. Rodrigo Faccioli, e integradas no servidor Koala. Essas ferramentas estão disponíveis na seção "Gromacs" no painel de ferramentas.

A Seção 5.5.1 descreve as ferramentas que podem ser utilizadas para *download*, *upload* e manipulação de arquivos para serem usados nos demais módulos disponíveis. A Seção 5.5.2 apresenta as ferramentas disponíveis para execução de algoritmos de predição de estrutura de proteína. Já a Seção 5.5.3 detalha os módulos disponíveis para a execução de métodos de análises e comparação de resultados gerados pelas outras ferramentas.

5.5.1 Pré-processamento dos dados

Alguns tipos de arquivos precisam ser pré-processados de alguma forma para que fiquem disponíveis de acordo com os requisitos de um algoritmo ou ferramenta. No Koala existem várias ferramentas para realizar esse pré-processamento, que podem estar relacionados com o *download* de arquivo de referência para uma comparação de predição de vários preditores, validação e correção dos dados a respeito da conformação da proteína, e manipulação dos modelos descritos nos arquivos PDBs.

A Seção 5.5.1.1 demonstra o funcionamento da ferramenta *Download PDB* a partir do *Protein Data Bank*. As Seções 5.5.1.2 e 5.5.1.3 descrevem os módulos que validam e corrigem informações conformacionais no arquivo PDB. As Seções 5.5.1.4 e 5.5.1.5 manipulam os *models* existentes em um arquivo PDB, podendo separá-los em diversos outros arquivos ou unir vários arquivos PDBs em um único arquivo com várias conformações.

5.5.1.1 Download PDB

A ferramenta *Download PDB* realiza o *download* de um arquivo PDB do *site* do *Protein Data Bank* (RCSB) a partir de um *ID*, que é um código que identifica a proteína no banco de dados do RCSB. O usuário pode inserir diversos IDs de uma só vez na ferramenta, agilizando o processo de pesquisa e *download*. Com esse módulo, o sistema permite que o usuário receba arquivos de outros sites sem sair do ambiente ao qual ele é mais familiarizado e onde a sua pesquisa está sendo realizada.

A ferramenta recebe como parâmetro de entrada somente o ID da proteína que o usuário deseja receber o arquivo PDB, o qual pode ser usado como referência em várias

ferramentas de análise disponível no Koala. A Figura 34 mostra a interface *web* desse módulo.

Esta ferramenta tem como saída (*output*) um arquivo PDB para cada ID informado pelo usuário, gerando um item no histórico para cada arquivo. Assim, o usuário tem liberdade de usar os arquivos em experimentos diferentes.

Figura 34: Módulo *Download PDB*, a partir do repositório do PDB.org (RCSB). Informando apenas o *PDBID*, a ferramenta realiza o *download* do arquivo PDB do *Protein Data Bank*.

5.5.1.2 Rename Atoms

A ferramenta *Rename Atoms* verifica a existência de átomos descritos no arquivo PDB que não estão com o mesmo código definido no arquivo que descreve o campo de força utilizado pelo GROMACS para o cálculo dos objetivos (energias).

Arquivos PDBs frequentemente usam nomenclatura atômica inconsistente e fora dos padrões usados por um campo de força escolhido. Normalmente, a nomenclatura de resíduos é encontrada em arquivos *.rtp* no banco de dados do GROMACS, então a ferramenta pode verificar quais átomos precisam ser trocados no arquivo PDB de entrada. A Figura 35 exibe os parâmetros dessa ferramenta.

Figura 35: Módulo *Rename atoms*, verifica a existência de átomos no arquivo PDB que não estão de acordo com o banco de dados de nomenclatura utilizado pelo GROMACS, e faz a alteração desses átomos para o nome correto.

Essa ferramenta recebe como parâmetro de entrada o arquivo PDB que deverá ser analisado e o campo de força que será usado como banco de dados de nomenclatura atômica.

Atualmente, os campos de força disponíveis na ferramenta são:

- CHARMM27
- AMBER99

e para cada campo de força existe um arquivo *.rtp* associado, o qual é usado para checar os átomos do arquivo.

O resultado da ferramenta é um novo arquivo PDB com todos os átomos verificados e, se necessário, renomeados de acordo com o banco de dados de topologia do GROMACS, podendo ser utilizado em outras ferramentas sem a possibilidade de erros.

5.5.1.3 Check PDB

A ferramenta *Check PDB* verifica se a estrutura atômica do arquivo PDB está de acordo com os programas do GROMACS que criam um arquivo de topologia a partir das coordenadas do arquivo PDB.

Pode ocorrer em um arquivo PDB a ausência de algum átomo, causando algum erro na criação do arquivo de topologia usado pelo GROMACS. Nesse sentido, a ferramenta verifica se todos os resíduos do arquivo PDB possuem todos os átomos, e caso isso não aconteça, o arquivo é rejeitado para o cálculo dos objetivos pelo GROMACS. A Figura 36 mostra a interface da ferramenta e seus parâmetros.

Figura 36: Módulo *Check PDB*, verifica a estrutura atômica de todos os resíduos que compõe o arquivo PDB de acordo com o campo de força escolhido. Caso o PDB não tenha a estrutura aceita pelo GROMACS, é descartado da análise.

A ferramenta recebe como parâmetro de entrada um arquivo PDB ou um arquivo comprimido contendo diversos arquivos PDBs, e o campo de força que será utilizado como banco de dados de resíduos. Atualmente, os campos de força disponíveis na ferramenta são:

- CHARMM27
- AMBER99

O resultado dessa ferramenta é um arquivo comprimido contendo todos os arquivos aceitos pelo GROMACS, ou seja, que possuem todos os átomos de todos os resíduos. Também é exibido um log com os possíveis erros, caso o arquivo esteja incorreto.

5.5.1.4 Split models

A ferramenta *Split models* tem como função separar as diversas conformações (*models*) contidas em um arquivo PDB em arquivos PDBs com apenas um *model*.

Existem algumas ferramentas no Koala que geram um único arquivo PDB contendo diversas conformações para uma proteína. Essas conformações são descritas como modelos (*models*). Alguns programas aceitam o arquivo PDB de entrada contendo vários modelos conformacionais, mas outros não. Essa ferramenta então existe para permitir ao usuário manipular os arquivos PDBs no ambiente do Koala, sem ter que fazer isso manualmente. A Figura 37 exibe a interface da ferramenta.

Figura 37: Módulo *Split Models*, o qual realiza a divisão dos modelos descritos em um arquivo PDB em vários arquivos PDBs separados.

Essa ferramenta recebe como parâmetro apenas o arquivo PDB para ser dividido. O resultado pode ser um arquivo comprimido com todos os arquivos PDBs gerados, ou seja, um arquivo PDB para cada modelo contido no arquivo de origem, ou vários itens no histórico do usuário contendo um arquivo PDB cada.

5.5.1.5 Merge PDB

A ferramenta *Merge PDB* realiza a união de vários arquivos PDBs em um único arquivo PDB, contendo o número de modelos igual ao número de arquivos PDBs selecionados.

Como no Koala existem ferramentas que recebem como parâmetro de entrada um arquivo PDB com vários modelos, essa ferramenta existe para unir os resultados de outros algoritmos, que geram um arquivo PDB para cada conformação predita, em um único arquivo PDB. Nesse processo, a ferramenta copia apenas o cabeçalho de um arquivo PDB para o arquivo resultante, o que exige que todos os arquivos inseridos pertençam à mesma proteína. A Figura 38 exibe os parâmetros da ferramenta.

Figura 38: Módulo *Merge PDB*, realiza a união de vários arquivos PDBs em um único arquivo PDB resultante, onde cada arquivo é descrito como modelo dentro do arquivo final.

A ferramenta recebe como entrada um conjunto de arquivos PDBs separados em itens no histórico ou um arquivo compactado com vários arquivos. O arquivo resultante é um arquivo PDB contendo a quantidade de modelos igual ao número de arquivos PDBs selecionados.

5.5.2 Algoritmos para predição de estruturas

O servidor *web* Koala possui diversas ferramentas que executam algoritmos aplicados ao problema de predição de estrutura de proteínas. Como descrito no Capítulo 3, os *frameworks* 2PG, ProtPred-EDA e MEAMT possuem várias ferramentas para execução de vários algoritmos e podem ser interligado (*workflow*) entre si e com outras ferramentas de análises.

As Seções 5.5.2.1, 5.5.2.2, 5.5.2.3, 5.5.2.4 e 5.5.2.5 apresentam as ferramentas criadas para os algoritmos disponíveis no *framework* 2PG. Já as Seções 5.5.2.6, 5.5.2.7, 5.5.2.8, 5.5.2.9, 5.5.2.10 e 5.5.2.11 demonstram as ferramentas para os algoritmos disponíveis no *framework* ProtPred-EDA. E as Seções 5.5.2.12 e 5.5.2.13 apresentam as ferramentas do algoritmo MEAMT.

5.5.2.1 2PG *Build Conformation*

Esta ferramenta executa um programa disponível dentro do *framework* 2PG. A partir de um arquivo FASTA ou de uma sequência de aminoácidos, o que corresponde a sequência primária da proteína, o programa "*protpred-Gromacs_pop_initial*" constrói a conformação da proteína e salva em formato PDB. Esses são os indivíduos que serão utilizados como população inicial nos algoritmos de otimização multiobjetivos também disponíveis no *framework*.

Como parâmetros de entrada, o programa recebe o arquivo FASTA, contendo a sequência de aminoácidos, o tamanho da população, o qual irá definir a quantidade de *models*³ que o programa irá gerar, alguns parâmetros para o GROMACS e o campo de força utilizado, que, caso seja o AMBER, a ferramenta disponibilizará opções sobre cargas no *C* e no *N* terminal. A Figura 39 mostra a execução dela no servidor criado.

Figura 39: Ferramenta 2PG *Build Conformation*, executa um programa disponível no 2PG para a criação de conformações aleatórias para a estrutura primária da proteína inserida.

Não existe nenhuma validação de dados de entrada para essa ferramenta. Como *output*, a ferramenta gera um arquivo PDB contendo vários modelos de conformações com base no arquivo FASTA inserido. A quantidade de *models* é controlada através do parâmetro de entrada *Size of population*.

³Um *model* corresponde a uma conformação da proteína. É a descrição das coordenadas de todos os átomos que compõe a proteína.

5.5.2.2 2PG Mono

Esta ferramenta executa o algoritmo mono-objetivo disponível no *framework* 2PG, o qual está definido pelo programa "*protpred-Gromacs-Mono*". Este programa realiza a predição da estrutura da proteína considerando apenas um objetivo. O cálculo do objetivo também é feito pelo GROMACS.

Como parâmetro de entrada, a ferramenta recebe um arquivo FASTA e um arquivo PDB, o qual corresponde à população inicial para o algoritmo. Além disso, é possível definir o campo de força utilizado, a quantidade de gerações que o algoritmo irá executar e quantas vezes o programa poderá mexer no ângulos torcionais da proteína a fim de melhorar a predição. A Figura 40 apresenta a tela com os parâmetros dessa ferramenta.

The screenshot displays the '2PG_Mono Execution of 2PG Framework (Galaxy Tool Version 1.0.0)' interface. It includes several input fields and buttons:

- Source of the aminoacids sequence:** A dropdown menu set to 'An upload file'.
- Sequence File:** A section with file selection icons and a dropdown menu currently showing 'No fasta dataset available.'
- Rotamer Library:** A dropdown menu set to 'None'.
- Force Field:** A dropdown menu set to 'Charmm27'.
- Population File:** A section with file selection icons and a dropdown menu showing '6: T0763TS006_5-D1.pdb'.
- Number of Generation:** A text input field containing the value '1'.
- How many rotation:** A text input field containing the value '1'.
- Objective:** A dropdown menu set to 'Potential'.
- Create a compressed file with all the PDB files?:** Two radio buttons, 'Yes' and 'No', both of which are currently unselected.
- Use Jmol visualization?:** Two radio buttons, 'Yes' and 'No', both of which are currently unselected.
- Execute:** A blue button with a checkmark icon and the text 'Execute'.

Below the 'Use Jmol visualization?' section, there is a note: 'Jmol offers a 3D visualization of the molecule. You must have JavaScript enabled on your browser.'

Figura 40: Ferramenta 2PG *Mono*, executa o programa "*protpred-Gromacs-Mono*" do *framework* 2PG para predição da estrutura de proteína.

Os arquivos de entrada, FASTA e PDB, são validados no momento da execução da ferramenta. Como saída, essa ferramenta apresenta uma página HTML com a sequência primária enviada, o arquivo PDB e uma imagem da estrutura das 10 melhores soluções encontradas pelo algoritmo e um arquivo compactado com todos os arquivos gerados na execução do programa, todos disponíveis para *download*. O usuário pode escolher também se quer exibir as estruturas tridimensionais usando o *software* JSmol.

5.5.2.3 2PG *Random*

A ferramenta criada executa o programa "*protpred-Gromacs-Random_Algorithm*" disponível no *framework* 2PG. Este programa realiza uma busca aleatória, ou seja, as soluções são geradas e aceitas, não existe nenhum critério de seleção. A partir de um arquivo PDB contendo um *model*, ou seja, um indivíduo, o programa gera uma quantidade de soluções (*models*) para o usuário.

Como parâmetros de entrada, a ferramenta recebe um arquivo FASTA, um arquivo PDB identificado como sendo a população inicial sempre com um indivíduo, alguns parâmetros para o GROMACS e o campo de força a ser utilizado. Um parâmetro (*Step Number*) defini a quantidade de execuções do algoritmo. A Figura 41 mostra a execução da ferramenta no servidor.

2PG Random Algorithm, an evolutionary framework applied to PSP (Galaxy Tool Version 1.0.0) Options

Source of the amino acids sequence

An upload file

You can use a file in your history as an input or you can insert your own amino acid sequence.

Sequence File

No fasta dataset available.

Population File

6: T0763TS006_5-D1.pdb

Step Number

3000

How many rotation

1

Rotamer Library

None

Force Field

Charmm27

Create a compressed file with all the PDB files?

Yes No

Use Jmol visualization?

Yes No

Jmol offers a 3D visualization of the molecule. You must have JavaScript enabled on your browser.

Execute

Figura 41: Ferramenta 2PG *Random Algorithm to PSP*, executa o algoritmo *Random* disponível no 2PG para predição da estrutura de proteína.

As validações para essa ferramenta estão relacionadas à quantidade de *models* no PDB, pois o programa aceita um PDB com apenas um *model*. Como saída, essa ferramenta apresenta uma página HTML com a sequência primária enviada, o arquivo PDB e uma imagem da estrutura das 10 melhores soluções encontradas pelo algoritmo e um

arquivo compactado com todos os arquivos gerados na execução do programa, todos disponíveis para *download*. O usuário pode escolher também se quer exibir as estrutura tridimensionais usando o *software* JSmol.

5.5.2.4 2PG Monte Carlo Metropolis

Esta ferramenta executa um programa disponível no *framework* 2PG chamado "*protpred-Gromacs-MC_Metropolis*", que implementa o algoritmo Monte Carlo Metropolis no 2PG. Usando como base um arquivo PDB inserido pelo usuário como a população inicial, o programa gera várias soluções.

Em relação aos parâmetros de entrada, a ferramenta recebe um arquivo FASTA, um arquivo PDB como sendo a população inicial e o campo de força. A quantidade de execuções do algoritmo é definida pelo parâmetro *Monte Carlo Steps*. A Figura 42 mostra a execução desta ferramenta no servidor.

Figura 42: Ferramenta 2PG *Monte Carlo Metropolis*, executa um programa disponível no 2PG que implementa o algoritmo Monte Carlo para predição de estrutura de proteínas.

A ferramenta valida se o arquivo PDB inserido contem somente um *model*. Como saída, essa ferramenta apresenta uma página HTML com a sequência primária enviada, o arquivo PDB e uma imagem da estrutura das 10 melhores soluções encontradas pelo algoritmo e um arquivo compactado com todos os arquivos gerados na execução do programa, todos

disponíveis para *download*. O usuário pode escolher também se quer exibir as estrutura tridimensionais usando o *software* JSmol.

5.5.2.5 2PG NSGA-II

A ferramenta 2PG NSGA-II executa o programa "*protpred-Gromacs-NSGA2*", um dos algoritmos disponíveis no *framework* 2PG. Esse programa é uma implementação do algoritmo evolutivo multiobjetivo NSGA-II.

A partir de um arquivo PDB com vários *models*, o qual é a população inicial de indivíduos para o algoritmo, o programa calcula a melhor solução de acordo com os objetivos selecionados e retorna para o usuário as melhores conformações preditas. A Figura 43 mostra a execução da ferramenta no servidor instalado.

2PG_NSQA2 Execution of 2PG Framework (Galaxy Tool Version 1.0.0) Options

Source of the aminoacids sequence

An upload file

You can use a file in your history as an input or you can insert your own aminoacid sequence.

Sequence File

No fasta dataset available.

Population File

6: T0763TS006_5-D1.pdb

Number of Generation

1

Rotamer Library

None

How many rotation

1

Objectives

☐ Select/Unselect all

☐ Potential

☐ Gyrate

☐ Hydrophobic

☐ Hydrophilic

☐ Total Area

☐ H Bond Main

☐ Electrostatic

☐ GBSA_Solvation

☐ Van der Waals

Create a compressed file with all the PDB files?

Yes No

Use Jmol visualization?

Yes No

Jmol offers a 3D visualization of the molecule. You must have JavaScript enabled on your browser.

Execute

Figura 43: Ferramenta 2PG NSGA-II, executa o programa "*protpred-Gromacs-NSGA2*" disponível no 2PG que implementa um algoritmo multiobjetivo para predição de estrutura de proteína.

A ferramenta recebe, como parâmetros de entrada, um arquivo FASTA com a sequência de aminoácidos, um arquivo PDB com os modelos, no qual cada modelo indica um indivíduo para o algoritmo, parâmetros para o GROMACS e quais os objetivos serão calculados pelo programa. O número de gerações do algoritmo, ou seja, quantas vezes ele vai executar, pode ser definido pelo usuário através do parâmetro *Number of Generation*.

A ferramenta valida o arquivo PDB de entrada com o valor setado no parâmetro *Size of Population*, que devem ser iguais. A quantidade de objetivos, que tem que ser mais do que dois, também é validada.

Como saída, essa ferramenta apresenta uma página HTML com a sequência primária enviada, o arquivo PDB e uma imagem da estrutura das 10 melhores soluções encontradas pelo algoritmo e um arquivo compactado com todos os arquivos gerados na execução do programa, todos disponíveis para *download*. O usuário pode escolher também se quer exibir as estruturas tridimensionais usando o pacote JSmol.

5.5.2.6 ProtPred-EDA RW

A ferramenta ProtPred-EDA RW (do inglês, *Random Walk*) executa um dos algoritmos de otimização disponível no *framework* ProtPred-EDA. A partir de uma sequência de aminoácidos, o sistema gera a população aleatória de indivíduos para realizar a predição da estrutura da proteína. A Figura 44 apresenta os parâmetros da ferramenta.

The screenshot displays the 'ProtPred-EDA Random Walk' web interface. At the top, it identifies the tool as an 'evolutionary algorithm applied to PSP (Galaxy Tool Version 1.0.0)' with an 'Options' dropdown. The 'Source of amino acid sequence' section offers a dropdown for 'An upload file' and a text area for a custom sequence. Below this, the 'Sequence File' section includes file upload icons and a note that no FASTA dataset is currently available. The 'Population size' is set to 500, and 'Evaluations' is set to 1,000,000. A note explains that convergence criteria 2 is based on reaching a maximum number of evaluations. The 'Threshold' is set to 0.0001, with a note that convergence criteria 1 is based on the standard deviation of fitness. The 'Van der Waals energy weight' is 1.0, 'Electrostatic energy weight' is 0.0, 'Solvation energy weight' is 0.0, 'Hydrogen Bond energy weight' is 0.0, and 'Torsion energy weight' is 0.0. The 'Use Amino Acid Database (ADB)' section has 'Yes' and 'No' buttons. The 'Use L amino acids' section also has 'Yes' and 'No' buttons, with a note that 'L is common in nature'. An 'Execute' button is at the bottom.

Figura 44: Módulo ProtPred-EDA RW, executa o algoritmo *Random Walk* presente no ProtPred-EDA para predição de estrutura de proteínas.

Como parâmetro de entrada, além da sequência de aminoácidos, o usuário pode controlar o tamanho da população do algoritmo e quantas execuções ele vai ter, como sendo

um dos critérios de parada. O usuário também pode definir o peso que cada objetivo (energia) vai ter na predição.

Todos os parâmetros de entrada são validados pelo programa relacionado com a ferramenta antes de executar o algoritmo de predição. Como saída, a ferramenta gera um arquivo HTML contendo todos os arquivos gerados na execução, além de imagem 3D das estruturas preditas. A ferramenta também pode gerar um arquivo compactado com todos os arquivos gerados na predição, o que inclui arquivos de log e arquivos PDBs com as melhores conformações encontradas.

5.5.2.7 ProtPred-EDA MC

A ferramenta ProtPred-EDA MC realiza a predição da estrutura terciária da proteína executando o algoritmo Monte Carlo, algoritmo implementado no ProtPred-EDA. Sem usar estrutura de referência, apenas com a sequência primária da proteína, o ProtPred-EDA MC gera conformações de acordo com os parâmetros escolhidos. A Figura 45 demonstra a interface da ferramenta.

Como parâmetro de entrada, a ferramenta recebe o tamanho da população que será gerada no algoritmo, valores de perturbação e critério de rejeição para os indivíduos, como também a quantidade de vezes que o algoritmo irá executar. É possível também definir qual objetivo será considerado na execução do programa.

A saída desta ferramenta é um arquivo HTML descrevendo todos os arquivos gerados na execução e imagens das estruturas preditas, além de um arquivo compactado com vários arquivos gerados pelo algoritmo ao término da execução. Dentre esses arquivos estão arquivos de *log* e os arquivos PDBs representando a solução do algoritmo.

5.5.2.8 ProtPred-EDA EDA

A ferramenta ProtPred-EDA EDA executa o algoritmo de otimização disponível no ProtPred-EDA chamado de Algoritmo de Estimação de Distribuição (do inglês, *Estimation Distribution Algorithm*) para a predição da estrutura terciária da proteína. A Figura 46 apresenta a tela inicial da ferramenta.

Para os parâmetros de entrada, o usuário pode definir os tamanhos da população, filhos que serão gerados em cada geração e a quantidade de selecionados. Também é possível escolher o modelo probabilístico que será usado na execução, e o método de torneio que será usado para escolher os melhores indivíduos. Além desses, o usuário também pode definir qual o objetivo terá peso maior na execução.

Esta ferramenta também tem como saída um arquivo HTML com os arquivos gerados e imagens 3D, além de um arquivo compactado (*zip*) contendo os arquivos PDBs e arquivos de *log*.

ProtPred-EDA Monte Carlo evolutionary algorithm applied to PSP (Galaxy Tool Version 1.0.0) Options

Source of amino acid sequence

An upload file

You can use a file in your history as an input or you can insert your own aminoacid sequence.

Sequence File

No fasta dataset available.

Population size

500

Mutation factor (s_{mc})

0.02

+ and - pertubation for each variable

Boltzmann

0.259

Aceptation/rejection criterion

Evaluations

1000000

Convergence criteria 2: convergence occurs when a maximum number of evolution is reached.

Threshold

0.0001

Convergence criteria 1: convergence occurs when the standard-deviation of fitness of population is below the specified value.

Van der Waals energy weight

1.0

Electrostatic energy weight

0.0

Solvation energy weight

0.0

Hydrogen Bond energy weight

0.0

Torsion energy weight

0.0

Use Amino Acid Database (ADB)

Use L amino acids

L is common in nature

Figura 45: Módulo ProtPred-EDA MC, executa o algoritmo Monte Carlo implementado no ProtPred-EDA para predição de estrutura de proteína.

5.5.2.9 ProtPred-EDA rBOA

A ferramenta ProtPred-EDA rBOA executa o algoritmo rBOA disponível no *framework* ProtPred-EDA para a realização da predição da estrutura terciária da proteína. Somente com a sequência de aminoácidos, o algoritmo prediz conformações para a proteína inserida.

Como parâmetros de entrada, o usuário pode definir o tamanho da população, selecionados na geração seguinte e dos pais dos indivíduos. Além disso, é possível definir dois critérios de parada (convergência), que são o número de execuções e um *threshold*. O usuário define também os pesos de cada objetivo disponível para o algoritmo. A Figura

ProtPred-EDA EDA evolutionary algorithm applied to PSP (Galaxy Tool Version 1.0.0) Options

Source of amino acid sequence
 An upload file
 You can use a file in your history as an input or you can insert your own aminoacid sequence.

Sequence File
 No fasta dataset available.

Population size
 500

Offspring size
 500

Probabilistic Model
 UNI - Univariate

Mutation
 2
 Perturbation criteria used in univariate

Selection Method
 Tournament

Size of selected
 500

Tournament size
 2

Evaluations
 1000000
 Convergence criteria 2: convergence occurs when a maximum number of evolution is reached.

Threshold
 0.0001
 Convergence criteria 1: convergence occurs when the standard-deviation of fitness of population is below the specified value.

Van der Waals energy weight
 1.0

Electrostatic energy weight
 0.0

Solvation energy weight
 0.0

Hydrogen Bond energy weight
 0.0

Torsion energy weight
 0.0

Execute

Figura 46: Módulo ProtPred-EDA EDA, realiza a predição da estrutura de proteínas executando o algoritmo EDA implementado no *framework* ProtPred-EDA.

47 demonstra os parâmetros da ferramenta.

Esta ferramenta tem como saída (*output*) um arquivo HTML descrevendo os resultados obtidos, além de um arquivo comprimido (*zip*) contendo todos os arquivos de *log* e arquivos PDBs que representam as soluções do algoritmo.

5.5.2.10 ProtPred-EDA GA

A ferramenta ProtPred-EDA GA executa um dos algoritmos disponíveis no *framework*, o Algoritmo Genético (do inglês, *Genetic Algorithm*). Como os outros algoritmos, esse

ProtPred-EDA rBOA evolutionary algorithm applied to PSP (Galaxy Tool Version 1.0.0) Options

Source of amino acid sequence

An upload file

You can use a file in your history as an input or you can insert your own aminoacid sequence.

Sequence File

No fasta dataset available.

Population size

5000

Size of selected

2500

Parents size

1

Number of mixture components

2

Evaluations

1000000

Convergence criteria 2: convergence occurs when a maximum number of evolution is reached.

Threshold

0.0001

Convergence criteria 1: convergence occurs when the standard-deviation of fitness of population is below the specified value.

Van der Waals energy weight

1.0

Electrostatic energy weight

0.0

Solvation energy weight

0.0

Hydrogen Bond energy weight

0.0

Torsion energy weight

0.0

Execute

Figura 47: Módulo ProtPred-EDA rBOA, realiza a predição da estrutura de proteína executando o algoritmo rBOA disponível no ProtPred-EDA.

também é puramente *ab initio*, e realiza a predição da estrutura da proteína apenas recebendo a estrutura primária. A Figura 48 mostra a interface da ferramenta.

Os parâmetros de entrada podem definir o tamanho da população de indivíduos, dos filhos gerados a partir da combinação dos pais, dos selecionados para reprodução e a quantidade de indivíduos que participam do torneio para reproduzir. Também é possível refinar a busca definindo valores para os operadores de recombinação e mutação. Além disso, pode-se definir os pesos de cada objetivo que será usado na predição.

O resultado da ferramenta é um item no histórico do usuário que corresponde a um arquivo HTML com os arquivos gerados na execução, além de um arquivo compactado com os arquivos de *log* e PDBs com as coordenadas atômicas da proteína.

ProtPred-EDA Genetic Algorithm an evolutionary algorithm applied to PSP (Galaxy Tool Version 1.0.0) Options

Source of amino acid sequence

An upload file

You can use a file in your history as an input or you can insert your own aminoacid sequence.

Sequence File

No fasta dataset available.

Population size

500

Offspring size

500

Size of selected

5000

Tournament size

2

Crossover Rate

0.3

Mutation Rate

0.05

Selection Method

Tournament

Evaluations

1000000

Convergence criteria 2: convergence occurs when a maximum number of evolution is reached.

Threshold

0.0001

Convergence criteria 1: convergence occurs when the standard-deviation of fitness of population is below the specified value.

Van der Waals energy weight

1.0

Electrostatic energy weight

0.0

Solvation energy weight

0.0

Hydrogen Bond energy weight

0.0

Torsion energy weight

0.0

Execute

Figura 48: Módulo ProtPred-EDA GA, executa o algoritmo genético disponível no framework ProtPred-EDA para predição da estrutura de proteína.

5.5.2.11 ProtPred-EDA DE

A ferramenta ProtPred-EDA DE executa o algoritmo *Diferencial Estimation*, implementado no *framework*. O usuário insere o arquivo FASTA contendo a sequência de aminoácidos da proteína, a única informação necessária para o algoritmo realizar a predição. A Figura 49 mostra a tela inicial da ferramenta.

Como parâmetro de entrada, é possível definir o tamanho da população que será usada na execução, a quantidade de filhos gerados, a quantidade de selecionados para mutação e reprodução e o tipo de seleção dos indivíduos. Além disso, o usuário também define o

ProtPred-EDA DE an evolutionary algorithm applied to PSP (Galaxy Tool Version 1.0.0) Options

Source of amino acid sequence

An upload file

You can use a file in your history as an input or you can insert your own aminoacid sequence.

Sequence File

No fasta dataset available.

Population size

3000

Offspring size

3000

Size of selected

5000

Tournament size

2

Crossover Rate

0.4

Selection Method

Tournament

Evaluations

1000000

Convergence criteria 2: convergence occurs when a maximum number of evolution is reached.

Threshold

0.0001

Convergence criteria 1: convergence occurs when the standard-deviation of fitness of population is below the specified value.

Van der Waals energy weight

1.0

Electrostatic energy weight

0.0

Solvation energy weight

0.0

Hydrogen Bond energy weight

0.0

Torsion energy weight

0.0

Execute

Figura 49: Módulo ProtPred-EDA DE, executa o algoritmo *Diferencial Estimation* disponível no *framework* ProtPred-EDA.

limiar de recombinação e mutação, e o tipo de seleção. Os pesos dos objetivos definindo quais serão usados também está disponível na ferramenta.

Esta ferramenta tem como resultado uma página HTML contendo as melhores soluções encontradas, bem como uma imagem 3D da molécula da melhor solução e um arquivo compactado com todos os arquivos gerados na execução.

5.5.2.12 MEAMT Build Population

A ferramenta *MEAMT Build Population* gera conformações aleatórias a partir da sequência de aminoácidos que formam a proteína. Essas conformações serão usadas no algoritmo de predição da estrutura. A Figura 50 mostra a tela inicial dessa ferramenta.

MEAMT Build Population as random conformation (Galaxy Tool Version 1.0.0) Options

Source of the amino acids sequence

An upload file

You can use a file in your history as an input or you can insert your own amino acid sequence.

Sequence File

No fasta dataset available.

Size of Population

400

Objectives

☒ Select/Unselect all

☒ Hydrogen Bond

☒ Electrostatic

☒ Solvatation

☒ Van der Waals

Van der Waals Weight

1

Hydrogen Bond Weight

0.5

Solvatation Weight

0.5

Electrostatic Weight

0.5

Define a name to the output file?

Yes No

Execute

Figura 50: Módulo MEAMT *Build Population*, realiza a geração de conformações aleatórias para a sequência de aminoácidos informada.

Como parâmetros, a ferramenta recebe o tamanho da população que deverá ser gerada. Como esse algoritmo é multi-tabelas, ou seja, utiliza várias sub-populações, o tamanho de cada sub-população equivale a $\frac{1}{8}$ do tamanho definido pelo usuário. Isso vai depender também dos objetivos selecionados. Para cada objetivo, duas sub-populações são criadas.

O resultado dessa ferramenta é um item no histórico do usuário que corresponde a um arquivo texto (*txt*) contendo todas os valores conformacionais gerados.

5.5.2.13 MEAMT

O módulo desenvolvido MEAMT executa o programa MEAMT, um algoritmo multiobjetivo para predições *ab initio* de estrutura de proteínas. Sendo assim, a ferramenta recebe apenas a sequência de aminoácidos da proteína para realizar a predição. A Figura 51 mostra a tela inicial da ferramenta.

A ferramenta recebe como parâmetros de entrada, além do arquivo FASTA, um arquivo de texto (*txt*) contendo a população aleatória gerada pelo módulo *MEAMT Build Population*. Além disso, o usuário pode determinar os pesos de cada um dos objetivos disponíveis no algoritmo e a quantidade de gerações para reprodução e geração de novos indivíduos.

MEAMT algorithm applied to PSP (Galaxy Tool Version 1.0.0) Options

Source of the amino acids sequence

An upload file

You can use a file in your history as an input or you can insert your own amino acid sequence.

Sequence File

No fasta dataset available.

Population File

6: T0763TS006_5-D1.pdb

Number of Generation

1

Objectives

☒ Select/Unselect all

☒ Hydrogen Bond

☒ Electrostatic

☒ Solvation

☒ Van der Waals

Van der Waals Weight

1

Hydrogen Bond Weight

0.5

Solvation Weight

0.5

Electrostatic Weight

0.5

Create a compressed file with all the PDB files?

Figura 51: Módulo MEAMT, executa o algoritmo evolutivo multiobjetivo MEAMT para predição *ab initio* de proteína.

Como resultado, a ferramenta gera um item no histórico do usuário que corresponde a um arquivo HTML formatado com *links* para *download* de todos os arquivos gerados na predição, além de possibilitar a visualização da melhor solução em 3D usando o pacote JSmol. O usuário também pode escolher a geração de um arquivo compactado somente com os arquivos PDBs, que poderá ser usado em outra ferramenta.

5.5.3 Módulos de Análises

O Koala possui diversas ferramentas desenvolvidas para a realização de análises e comparação de resultados, os quais podem ter sido gerados por outras ferramentas no servidor ou gerados por outros programas fora do sistema e importado para o ambiente do Koala. Essas análises estão relacionadas a comparação de métodos e resultados, guiando o usuário em relação ao melhor método utilizado.

As Seções 5.5.3.2, 5.5.3.1 e 5.5.3.3 apresentam os módulos que realizam a comparação de métodos utilizados para predição. Já as Seções 5.5.3.4, 5.5.3.5 e 5.5.3.6 abordam

as ferramentas disponíveis para verificação de similaridades das estruturas geradas pelos algoritmos no Koala.

5.5.3.1 2PG *Sort by Front Dominance*

Esta ferramenta avalia diversas soluções em relação à Fronteira de Pareto, separando-as em fronteiras, e classificando-as de acordo com um *ranking* de dominância. A Figura 52 mostra a execução dessa ferramenta no servidor.

Sort by Front Dominance to PSP methods (Galaxy Tool Version 1.0.0) Options

Source of the PDB Files

PDB Files

PDB Files

6: T0763TS006_5-D1.pdb
5: T0763TS006_4-D1.pdb
4: T0763TS006_3-D1.pdb
3: T0763TS006_2-D1.pdb
2: T0763TS006_1-D1.pdb

Objective

☐ Select/Unselect all

☐ Potential
☐ Gyrate
☐ Hydrophobic
☐ Hydrophilic
☐ Total Area
☐ H Bond Main
☐ Electrostatic
☐ GBSA_Solvatation
☐ Van der Waals

You must select only two objectives.

Check the PDB structure with Gromacs?

Yes No

Check if the Gromacs program PDB2GMX accepts the PDB file.

Rename the missing atoms on the PDB file according to Gromacs?

Yes No

Check if the PDB file has missing atoms according Gromacs.

Use Jmol visualization?

Yes No

Jmol offers a 3D visualization of the molecule. You must have JavaScript enabled on your browser.

Execute

Figura 52: Ferramenta *Sort by Front Dominance*, ordena as melhores soluções de acordo com o *ranking* gerado pela Fronteira de Pareto.

Como parâmetros de entrada, a ferramenta recebe vários arquivos PDBs contendo uma conformação da proteína gerada por algum método de predição e os objetivos selecionados para o cálculo multiobjetivo. Na execução, um *script* para verificar se o PDB é válido e

outro para renomear átomos de acordo com a topologia utilizada são executados antes de calcular as fronteiras e dominâncias.

A contribuição desse método é a possibilidade de encontrar a melhor solução dentre várias soluções não importando qual algoritmo a gerou, qual o tamanho da proteína ou qual o seu tipo, e nem o uso de estrutura de referência, diferentemente de outras métricas que possuem limitações e a necessidade de comparação com uma proteína de referência.

É verificado se o usuário selecionou apenas dois objetivos por vez na execução. A combinação entre eles é livre. No histórico do usuário é salvo um arquivo HTML como sendo o *output* desta ferramenta. Este arquivo contém o arquivo *.front*, no qual são registradas todas as soluções ordenadas pelo *ranking*, os arquivos *.xvg*, os quais podem ser usados para a geração de gráficos das fronteiras e um gráfico das duas primeiras fronteiras. E, por fim, os arquivos *.png* referente as imagens das cinco melhores moléculas preditas, geradas com o Pymol, e podendo ser visualizadas pelo JSmol.

5.5.3.2 Dominance Ranking

Esta ferramenta realiza um *ranking* dos arquivos PDBs gerados por vários preditores (algoritmos) de acordo com os valores obtidos no cálculo de uma combinação de objetivos. O arquivo PDB que obter os melhores *rankings* em várias combinações de objetivos terá uma dominância maior sobre os demais arquivos, e conseqüentemente, representará a melhor solução entre todas as analisadas nas combinações criadas.

Para executar a ferramenta, o usuário seleciona diversos arquivos PDBs, normalmente soluções geradas por diversos algoritmos para uma mesma proteína alvo, a fim de comparar os métodos em relação a uma ou mais combinação de objetivos. Esses arquivos podem estar também compactados em arquivo *zip* ou *gz*. Depois disso, o usuário insere várias combinações de dois objetivos dentre os nove disponíveis no servidor. O processo de cálculo dos objetivos é o mesmo descrito na Seção 5.5.3.1. O usuário também seleciona se deseja checar a estrutura dos arquivos (Seção 5.5.1.3) e/ou renomear os átomos que não estão presentes para o uso do GROMACS (Seção 5.5.1.2). A Figura 53 exibe a interface da ferramenta.

Este método também não possui limitações ou a necessidade de estrutura de referência, podendo ser usado para qualquer tipo ou tamanho de proteína, mesmo não conhecendo a estrutura nativa.

O resultado dessa ferramenta é um arquivo HTML contendo um gráfico para cada combinação de objetivos e a disposição dos arquivos PDBs; uma tabela com o *ranking* criado, com as vitórias, derrotas e a dominância, e conseqüentemente, a melhor solução; imagens em 3D das moléculas referente aos PDBs com os melhores *ranking*, com a possibilidade de visualização com o pacote JSmol; e um *link* para *download* de todos os arquivos gerados na análise.

Dominance Ranking of PSP methods (Galaxy Tool Version 1.0.0) Options

Source of the PDB Files

PDB Files

PDB Files

6: T0763TS006_5-D1.pdb
5: T0763TS006_4-D1.pdb
4: T0763TS006_3-D1.pdb
3: T0763TS006_2-D1.pdb
2: T0763TS006_1-D1.pdb

Objective Combination

1: Objective Combination

Objectives

☐ Select/Unselect all

☐ Potential
☐ Gyrate
☐ Hydrophobic
☐ Hydrophilic
☐ Total Area
☐ H Bond Main
☐ Electrostatic
☐ GBSA_Solvatation
☐ Van der Waals

+ Insert Objective Combination

Check the PDB structure with Gromacs?
Yes No
Check if the Gromacs program PDB2GMX accepts the PDB file.

Rename the missing atoms on the PDB file according to Gromacs?
Yes No
Check if the PDB file has missing atoms according Gromacs.

Use Jmol visualization?
Yes No
Jmol offers a 3D visualization of the molecule. You must have JavaScript enabled on your browser.

✓ Execute

Figura 53: Ferramenta *Dominance Ranking*, cria um *ranking* com todos os arquivos PDBs de acordo com os valores dos objetivos calculados em combinações de dois objetivos.

5.5.3.3 Sort Method by Front

Este módulo cria um *ranking* para vários arquivos PDBs gerados por diversos algoritmos para uma proteína alvo de acordo com a dominância de Pareto. Entretanto, nenhum cálculo de objetivos é realizado pela ferramenta.

O único parâmetro dessa ferramenta é um arquivo texto (tabular) contendo os valores calculados de um combinação de dois objetivos para cada arquivo PDB, os quais podem ter sido gerados por qualquer algoritmo. A partir disso, a ferramenta lê o arquivo, calcula

a fronteira de Pareto e gera um *ranking* com as soluções. A Figura 54 mostra a tela inicial da ferramenta.

Figura 54: Ferramenta *Sort Method by Front Dominance*, realiza um *ranking* com a fronteira de Pareto gerada a partir de um arquivo texto contendo os valores de uma combinação de dois objetivos para cada arquivo PDB.

O resultado dessa ferramenta é um item no histórico do usuário que corresponde a um arquivo HTML com um gráfico mostrando a fronteira de Pareto, quais soluções dominam e quais são dominadas.

5.5.3.4 Calculate RMSD

A ferramenta *Calculate RMSD* realiza a comparação de estruturas com uma estrutura de referência a fim de encontrar similaridades. Uma estrutura similar a uma outra de referência indica que o modelo predito está bem próximo da estrutura nativa. Essa métrica é muito utilizada na literatura e uma das utilizadas na comparação de resultados do CASP.

O usuário insere um arquivo PDB como estrutura de referência, a qual será usada para comparação dos demais arquivos PDBs preditos para a mesma proteína da estrutura de referência. É possível incluir somente um arquivo PDB ou selecionar vários e até mesmo realizar a comparação com um arquivo compactado com vários PDBs. A Figura 55 mostra a tela inicial dessa ferramenta.

O resultado do módulo é uma página HTML com uma tabela com todos os *scores* calculados para cada um dos PDBs inseridos. Também é possível realizar o *download* de todos os arquivos gerados na execução, o que inclui imagens tridimensionais dos arquivos PDBs, podendo ser visualizados também no *software* JSmol.

5.5.3.5 Calculate TM-Score

Esse módulo executa o algoritmo TM-Score que realiza a comparação entre duas estruturas de proteínas a fim de encontrar similaridades, porém esse método é para comparar proteínas com estruturas terciárias diferentes. Esse método pode indicar que duas estruturas podem ter a mesma topologia e/ou o mesmo dobramento (*fold*).

Como parâmetro, o usuário insere um arquivo PDB como referência para a comparação, e seleciona um ou vários arquivos PDBs para serem comparados. Essa ferramenta executa o método disponibilizado em (ZHANG; SKOLNICK, 2004). A Figura 56 exhibe os parâmetros dessa ferramenta.

The screenshot shows the 'Calculate RMSD from a set of PDB files (Galaxy Tool Version 1.0.0)' interface. It includes a 'PDB Reference' section with a file selection button and a dropdown menu showing '6: T0763TS006_5-D1.pdb'. Below this is the 'Source of the PDB Files' dropdown set to 'PDB Files'. The 'PDB Files to be compared to the reference' section features a list of files: '6: T0763TS006_5-D1.pdb', '5: T0763TS006_4-D1.pdb', '4: T0763TS006_3-D1.pdb', '3: T0763TS006_2-D1.pdb', and '2: T0763TS006_1-D1.pdb'. There are three checkboxes: 'Check the PDB structure with Gromacs?' (Yes/No), 'Rename the missing atoms on the PDB file according to Gromacs?' (Yes/No), and 'Use Jmol visualization?' (Yes/No). Each checkbox has a brief description. At the bottom is an 'Execute' button.

Figura 55: Ferramenta *Calculate RMSD*, implementa a métrica RMSD para comparação de estruturas de proteínas a fim de encontrar similaridades entre elas.

Como resultado, o módulo cria um item no histórico do usuário que corresponde a um arquivo HTML com uma tabela com os *scores* calculados para todos os arquivos PDBs. Também está disponível imagens das moléculas em 3D, podendo ser visualizadas no pacote JSmol. Todos os arquivos podem ser baixados pelo usuário.

5.5.3.6 Calculate GDT-TS

A ferramenta *Calculate GDT-TS* compara várias estruturas com uma estrutura de referência para encontrar similaridades entre proteínas que possuem a mesma sequência de aminoácidos mas dobramento diferentes. Essa métrica é a principal métrica para quantificar o sucesso das predições no CASP.

Em relação aos parâmetros de entrada, o usuário seleciona um arquivo PDB como referência e vários outros que serão comparados com o de referência. É possível também selecionar um arquivo compactado com vários arquivos PDBs. Essa ferramenta executa o mesmo programa implementado e disponibilizado em (ZHANG; SKOLNICK, 2004). A Figura 57 mostra a tela inicial da ferramenta.

O resultado dessa ferramenta é um arquivo HTML contendo uma tabela com os valores calculados dos *scores* de cada arquivo PDB inserido, uma imagem para as melhores soluções (PDB) e *link* para todos os arquivos gerados na execução, ficando disponível para

Calculate TM-Score from a set of PDB files (Galaxy Tool Version 1.0.0) Options

PDB Reference
6: T0763TS006_5-D1.pdb

Source of the PDB Files
PDB Files

PDB Files
6: T0763TS006_5-D1.pdb
5: T0763TS006_4-D1.pdb
4: T0763TS006_3-D1.pdb
3: T0763TS006_2-D1.pdb
2: T0763TS006_1-D1.pdb

Use Jmol visualization?
Yes No
Jmol offers a 3D visualization of the molecule. You must have JavaScript enabled on your browser.

Execute

Figura 56: Ferramenta *Calculate TM-Score*, realiza a comparação entre duas proteínas com estruturas terciárias diferentes.

Calculate GDT-TS from a set of PDB files (Galaxy Tool Version 1.0.0) Options

PDB Reference
6: T0763TS006_5-D1.pdb

Source of the PDB Files
PDB Files

PDB Files
6: T0763TS006_5-D1.pdb
5: T0763TS006_4-D1.pdb
4: T0763TS006_3-D1.pdb
3: T0763TS006_2-D1.pdb
2: T0763TS006_1-D1.pdb

Use Jmol visualization?
Yes No
Jmol offers a 3D visualization of the molecule. You must have JavaScript enabled on your browser.

Execute

Figura 57: Ferramenta *Calculate GDT-TS*.

download.

Resultados

Este capítulo apresenta os experimentos e resultados relativos a algumas ferramentas desenvolvidas no servidor Koala. Para demonstrar a contribuição deste trabalho, foram realizadas predições de estruturas de proteínas que já possuem estrutura terciária conhecida e com isso comparar as predições. Para a predição foram usados todos os algoritmos disponíveis no Koala: 2PG, ProtPred-EDA e MEAMT (ver Capítulo 3). Para demonstrar as análises que podem ser realizadas no Koala, os resultados obtidos das predições foram analisados a partir de duas maneiras diferentes: com e sem estrutura de referência. Usando uma estrutura de referência, as estruturas preditas foram comparadas usando a métrica RMSD. Sem usar uma estrutura de referência, as predições foram comparadas a fim de encontrar a melhor estrutura predita usando a ferramenta *Sort By Front Dominance* (ver Capítulo 5).

Em cada experimento foi analisada a diferença de parâmetros, para que eles ficassem com parâmetros parecidos. Entretanto, não é a intenção deste trabalho encontrar qual o melhor algoritmo dentre os disponíveis no servidor, mas disponibilizar um serviço com várias opções de predição e análise, em um único ambiente, integrado, e que garanta o compartilhamento e reprodutibilidade das pesquisas, além de disponibilizar métodos de comparação e análises já conhecidos na literatura (RMSD, por exemplo) e outros dois novos métodos de *ranking* de estruturas.

Para os testes de predição, foram usadas quatro proteínas disponíveis no PDB (*Protein Data Bank*) que possuem a sua estrutura terciária conhecida e determinada por método de RNM (Ressonância Magnética) ou Cristalografia de Raio X (CRX). Optou-se também por proteínas pequenas (menos de 100 aminoácidos) e com apenas um ou duas estruturas secundárias, devido limitações de objetivos nos algoritmos *ab initio* (BRASIL; DELBEM; SILVA, 2013). Além disso, essas proteínas já foram utilizadas em experimentos realizados pelos algoritmos disponíveis no Koala. As proteínas são apresentadas na Tabela 4 e a Figura 58 mostra as estruturas terciárias de cada uma das proteínas escolhidas.

Nos experimentos, foi definido o número de avaliações (gerações) para todos os algoritmos como sendo o único critério de convergência. Não foi avaliada a diferença de

Tabela 4: Proteínas retiradas do RCSB PDB e utilizadas nos experimentos de predição de estrutura terciária com os algoritmos disponíveis no Koala e análises das predições.

Proteína	Resíduos	Átomos	Método	Nome
2EVQ	12	198	RNM	<i>HP7</i>
1A11	36	391	RNM	<i>ACETYLCHOLINE RECEPTOR M2</i>
1VII	36	596	RNM	<i>Villin</i>
1ERC	40	567	RNM	<i>PHEROMONE ER-1</i>

(a) Estrutura da proteína 2EVQ nativa

(b) Estrutura da proteína 1A11 nativa

(c) Estrutura da proteína 1VII nativa

(d) Estrutura da proteína 1ERC nativa

Figura 58: Estruturas terciárias das proteínas nativas usadas nos experimentos. Foram retiradas do site do *Protein Data Bank* (<http://www.rcsb.org/>). Estão sendo exibidas no padrão *ribbon*. As imagens foram geradas pelo *software* PyMOL (Schrödinger, LLC, 2010)

tempo de execução de cada um dos algoritmos, uma vez que esse não era o objetivo deste trabalho. Entretanto, nas páginas dos resultados obtidos apresentam o tempo gasto em cada predição.

Antes de realizar as predições, foi necessário enviar alguns arquivos para os algoritmos realizarem a predição, quem possuem a sequência primária das proteínas usadas nos experimentos, o qual corresponde ao arquivo FASTA. Todos eles foram baixados do site do RCSB *Protein Data Bank* (pdb.org). No Koala, existe uma ferramenta localizada na opção *Get Data* do menu lateral de ferramentas com o nome de *Upload File*. A partir dessa ferramenta, é possível realizar o *upload* de arquivos, tanto do computador ou usando uma URL¹, para dentro do ambiente do Koala, ficando disponível para utilizá-lo em outras ferramentas. Os arquivos FASTA das proteínas 2EVQ, 1A11, 1VII e 1ERC foram baixados para um computador e depois enviados para o Koala para serem usados

¹URL é um endereço eletrônico usado como identificação na internet.

nas ferramentas de predição.

Na Seção 6.1 são mostradas as predições realizadas com todos os algoritmos, destacando os valores de parâmetros definidos para cada algoritmo e execução das ferramentas. Nas Seções 6.1.1 e 6.1.2 são elucidadas as aplicações das métricas de comparação de estruturas a partir de duas formas diferentes: com e sem referência em quatro proteínas com estruturas terciárias conhecidas. Na Seção 6.2 é mostrado como pode ser feito um fluxo de trabalho automático no Koala com as ferramentas do servidor. E, por fim, a Seção 6.3 apresenta uma das formas de compartilhar uma pesquisa e seus resultados no servidor Koala, tornando-a reproduzível.

6.1 Experimentos

Para a realização dos experimentos, foram definidos os valores dos parâmetros de cada algoritmo disponível no 2PG, ProtPred-EDA e MEAMT com base nos testes realizados por (FACCIOLI et al., 2011) (FACCIOLI et al., 2012) (FACCIOLI; BORTOT; DELBEM, 2014), (BONETTI, 2015) e (BRASIL, 2012) (BRASIL; DELBEM; SILVA, 2013), respectivamente, e adaptados para as proteínas utilizadas neste trabalho. Todos os valores foram inseridos diretamente nas ferramentas desenvolvidas para cada algoritmo, e os *jobs* criados estão salvos e disponíveis no Koala para reprodução.

As execuções das predições foram realizadas na infraestrutura onde o Koala está instalado, no Cloud USP, onde tem 12 computadores disponíveis. A configuração de cada máquina corresponde a um processador Intel Xeon 2.4Ghz com 8 núcleos, 16GB de RAM e 1TB HD. A execução dos algoritmos foi distribuída entre todos os doze computadores através da configuração realizada no Pulsar. Por hora, essa distribuição é dinâmica, ou seja, novos *jobs* serão direcionados para o computador mais ocioso.

As ferramentas relacionadas com os algoritmos de predição estão em uma seção chamada de *Ab initio predictors*, e nela estão contidas duas subseções, uma para as ferramentas que criam conformações aleatórias (*Build conformations*) e uma para os algoritmos de predição (*Predictors*). Todas as ferramentas usadas nos experimentos estão localizadas nessas duas seções. Ao clicar em uma das seções, ela se expande mostrando todas as ferramentas relacionadas com a seção e assim pode ser escolhida para execução. Ao ser escolhida, os parâmetros da ferramenta aparecem no painel central para serem preenchidos e, no final da página, um botão *Execute* pode ser pressionado para executar a ferramenta com os parâmetros escolhidos.

A Tabela 5 apresenta os valores dos principais parâmetros definidos para os algoritmos do 2PG, mostra também uma breve descrição de cada um. Por sua vez, a Tabela 6 apresenta os parâmetros de execução e os valores definidos para os algoritmos disponíveis no ProtPred-EDA. E, por fim, a Tabela 7 mostra os parâmetros relacionados ao algoritmo MEAMT e uma breve descrição de cada um.

Tabela 5: Parâmetros definidos para os algoritmos do 2PG para predição das quatro proteínas usadas nos experimentos.

Parâmetro	Valor	Descrição
População (Mono, NSGA2)	1000	Tamanho de população utilizado
População (MC, Random)	1	Tamanho de população utilizado
Avaliações	200	Número de gerações
Campo de força	Amber99	Campo de força para o arquivo de topologia
Objetivos	Potential GBSA Solvation	Objetivos utilizados nas predições
<i>How many rotation</i>	1	Quantidade de rotações realizadas nos ângulos torcionais
<i>MC Steps</i>	10000	Quantidade de avaliações para o algoritmo Monte Carlo

Tabela 6: Parâmetros definidos para os algoritmos do ProtPred-EDA para predição das quatro proteínas usadas nos experimentos.

Parâmetro	Valor	Descrição
População	500	Tamanhos de população utilizados
População	5000	Tamanho de população utilizado
Avaliações	1000000	Critério de convergência, número de avaliações
<i>Threshold</i>	1.0	Segundo critério de convergência (desvio padrão)
Van der Waals	1.0	Peso para o objetivo Van der Waals
<i>Solvation</i>	0.0	Peso para o objetivo de Solvatação
<i>Electrostatic</i>	0.0	Peso para o objetivo de Eletrostática
<i>Hydrogen Bond</i>	0.0	Peso para o objetivo de Ligação de hidrogênio
<i>Torsion</i>	0.0	Peso para o objetivo de Torção
Taxa de mutação (MCM)	1.0	Taxa de mutação para o algoritmo MCM
Taxa de mutação (GA)	0.9	Taxa de mutação para o algoritmo GA
Taxa de recombinação (GA)	0.3	Taxa de recombinação para o algoritmo GA
Taxa de recombinação (DE)	0.2	Taxa de recombinação para o algoritmo DE
Modelo probabilístico (EDA)	KDE2D	Modelo probabilístico usado no algoritmo EDA
Pressão de seleção (EDA)	1	Quantidade de indivíduos selecionado no torneio

Tabela 7: Parâmetros definidos para os algoritmos do MEAMT para predição das quatro proteínas usadas nos experimentos.

Parâmetro	Valor	Descrição
População	640	Tamanhos de população utilizados
Avaliações	20000	Critério de convergência, número de gerações
Van der Waals	1.0	Peso para o objetivo Van der Waals
<i>Solvation</i>	0.5	Peso para o objetivo de Solvatação
<i>Electrostatic</i>	0.5	Peso para o objetivo de Eletrostática
<i>Hydrogen Bond</i>	0.5	Peso para o objetivo de Ligação de Hidrogênio

Antes da execução dos experimentos, foram criados os arquivos de população inicial para os algoritmos do 2PG e MEAMT, os quais possuem um parâmetro de entrada que corresponde a população inicial. Para isso, foram usadas as ferramentas disponíveis no

Koala que geram conformações aleatórias com o uso da sequência de aminoácidos (Seção 5.5.2.1 e 5.5.2.12), localizadas na seção *Build Conformations*.

Foi realizada uma execução de todos os algoritmos para cada proteína de acordo com os parâmetros apresentados em cada configuração de algoritmo. Assim, a Tabela 8 sumariza as execuções realizadas.

Tabela 8: Lista dos parâmetros principais e comuns dos algoritmos preditores executados no Koala.

Algoritmo	População	Avaliações	Objetivo(s)
2PG Mono	1000	200	Potential
2PG NSGA-II	1000	200	Potential
2PG Random	1	10000	GBSA Solvation
2PG Monte Carlo	1	10000	Potential
ProtPred-EDA RW	500	1000000	Potential
ProtPred-EDA MC	500	1000000	Potential
ProtPred-EDA EDA	500	1000000	Potential
ProtPred-EDA rBOA	5000	1000000	Potential
ProtPred-EDA GA	500	1000000	Potential
ProtPred-EDA DE	500	1000000	Potential
MEAMT	640	20000	Van der Waals Solvation Electrostatic Hydrogen Bond

Ao final das execuções, cada ferramenta gerou dois *outputs*, um corresponde a um arquivo compactado contendo todas as soluções geradas na predição e o outro corresponde a um arquivo HTML com as melhores estruturas preditas, links para *download* dos arquivos e demais arquivos gerados por cada algoritmo. A página também permite visualizar a molécula em 3D com o uso do JSMol. As Tabelas 9, 10, 11 e 12 apresentam a quantidade de arquivos PDBs gerados por cada algoritmo e um valor total. Cada algoritmo tem a sua especificidade no que tange a geração de soluções, por isso os valores são diferentes entre eles.

Um dos resultados exibidos na página HTML é uma imagem 3D das melhores estruturas preditas pelo algoritmo. As Figuras 59, 60, 61 e 62 mostram a melhor estrutura tridimensional predita (verde) de cada algoritmo para as proteínas 2EVQ, 1A11, 1VII e 1ERC, alinhadas com a estrutura nativa (azul).

Todos os *datasets* gerados na execução dos algoritmos estão disponíveis publicamente no servidor, podendo ser copiados para a área de um outro usuário e reproduzido. Assim, a predição se torna transparente para toda a comunidade científica. Os históricos gerados com as predições dos algoritmos estão disponíveis publicamente no servidor Koala através dos links:

Tabela 9: Quantidade de soluções geradas por todos os algoritmos para a proteína 2EVQ

Proteína	Algoritmo	Quantidade
2EVQ	ProtPredEDA_DE	251
2EVQ	ProtPredEDA_EDA	76
2EVQ	ProtPredEDA_GA	1377
2EVQ	ProtPredEDA_MCM	402
2EVQ	ProtPredEDA_rBOA	269
2EVQ	ProtPredEDA_RW	6
2EVQ	2PG-Mono	20
2EVQ	2PG-MC	10
2EVQ	2PG-NSGA2	20
2EVQ	2PG-Random	20
2EVQ	MEAMT	1
Quantidade Total		2452

Tabela 10: Quantidade de soluções geradas por todos os algoritmos para a proteína 1A11

Proteína	Algoritmo	Quantidade
1A11	ProtPredEDA_DE	404
1A11	ProtPredEDA_EDA	109
1A11	ProtPredEDA_GA	1639
1A11	ProtPredEDA_MCM	168
1A11	ProtPredEDA_rBOA	157
1A11	ProtPredEDA_RW	8
1A11	2PG-Mono	20
1A11	2PG-MC	10
1A11	2PG-NSGA2	20
1A11	2PG-Random	20
1A11	MEAMT	1
Quantidade Total		2556

Tabela 11: Quantidade de soluções geradas por todos os algoritmos para a proteína 1VII

Proteína	Algoritmo	Quantidade
1VII	ProtPredEDA_DE	725
1VII	ProtPredEDA_EDA	127
1VII	ProtPredEDA_GA	1495
1VII	ProtPredEDA_MCM	200
1VII	ProtPredEDA_rBOA	109
1VII	ProtPredEDA_RW	9
1VII	2PG-Mono	10
1VII	2PG-MC	10
1VII	2PG-NSGA2	20
1VII	2PG-Random	10
1VII	MEAMT	1
Quantidade Total		2716

Figura 59: Melhor estrutura terciária predita de cada preditor para a proteína 2EVQ. A estrutura em azul representa a estrutura nativa, e a em verde representa a estrutura predita. As estruturas preditas estão alinhadas com a estrutura nativa, e as imagens foram geradas pelo programa Pymol (Schrödinger, LLC, 2010).

Figura 60: Melhor estrutura terciária predita de cada preditor para a proteína 1A11. A estrutura em azul representa a estrutura nativa, e a em verde representa a estrutura predita. As estruturas preditas estão alinhadas com a estrutura nativa, e as imagens foram geradas pelo programa Pymol (Schrödinger, LLC, 2010).

Figura 61: Melhor estrutura terciária predita de cada preditor para a proteína 1VII. A estrutura em azul representa a estrutura nativa, e a em verde representa a estrutura predita. As estruturas preditas estão alinhadas com a estrutura nativa, e as imagens foram geradas pelo programa Pymol (Schrödinger, LLC, 2010).

Figura 62: Melhor estrutura terciária predita de cada preditor para a proteína 1ERC. A estrutura em azul representa a estrutura nativa, e a em verde representa a estrutura predita. As estruturas preditas estão alinhadas com a estrutura nativa, e as imagens foram geradas pelo programa Pymol (Schrödinger, LLC, 2010).

Tabela 12: Quantidade de soluções geradas por todos os algoritmos para a proteína 1ERC

Proteína	Algoritmo	Quantidade
1ERC	ProtPredEDA_DE	644
1ERC	ProtPredEDA_EDA	129
1ERC	ProtPredEDA_GA	1352
1ERC	ProtPredEDA_MCM	183
1ERC	ProtPredEDA_rBOA	183
1ERC	ProtPredEDA_RW	9
1ERC	2PG-Mono	20
1ERC	2PG-MC	10
1ERC	2PG-NSGA2	20
1ERC	2PG-Random	10
1ERC	MEAMT	1
Quantidade Total		2561

- 2PG: <http://200.144.255.35:8084/u/ade felicibus-15/h/2pg-1>;
- ProtPredEDA: <http://200.144.255.35:8084/u/ade felicibus-15/h/protpred-eda>;
- MEAMT: <http://200.144.255.35:8084/u/ade felicibus-15/h/meamt>.

Com todas as predições finalizadas, foi possível realizar várias análises com os resultados. Nas próximas seções será mostrado como pode ser feita a comparação das estruturas preditas pelos algoritmos disponíveis no Koala (ou geradas por outros preditores fora do Koala e importados no servidor) usando tanto uma das métricas mais usadas na literatura para identificar similaridades de estruturas proteicas quanto um método desenvolvido neste trabalho para encontrar a melhor solução entre um conjunto de soluções, sem usar conhecimento prévio da proteína.

6.1.1 Comparação com estrutura de referência

Esta seção tem como objetivo mostrar como pode ser feito um dos tipos de análises disponíveis no Koala, que corresponde a comparação de estruturas tridimensionais usando uma estrutura como referência, chamada de RMSD. Ao final da análise, é possível visualizar as melhores estruturas preditas com melhor *score* para a métrica utilizada.

Os métodos de comparação usando uma estrutura de referência são os mais usados para validar uma predição, porém os *scores* calculados por eles fornecem informações limitadas de quão similares as proteínas são (XU; ZHANG, 2010) e possuem limitações quanto ao tamanho e tipo da proteína, e a necessidade de ter uma estrutura de referência determinada por métodos experimentais (Cristalografia de Raio-X e Ressonância Magnética), o que nem sempre é possível (ZHANG; SKOLNICK, 2004).

As ferramentas que fazem as análises com estrutura de referência estão localizadas na seção do painel esquerdo chamada de *Solution Comparison*. Ao clicar nesta seção, ela se expande e exibe as três ferramentas disponíveis para comparação de estruturas terciária. Uma vez que a ferramenta é selecionada, as suas opções são carregadas no painel central junto com um botão para executá-la.

Para realizar essa análise no Koala, foi executada somente a ferramenta que calcula o RMSD, por ser a métrica mais utilizada na literatura. A ferramenta *Calculate RMSD* recebeu como parâmetro de entrada um arquivo PDB de cada proteína como referência. Esse arquivo foi retirado do repositório do *Protein Data Bank* para as proteínas usadas nos experimentos usando a ferramenta de *Download PDB*, inserindo somente o *PDBID* da proteína. Além desse arquivo PDB, a ferramenta recebeu todas as estruturas preditas por todos os algoritmos, os quais são compactados em arquivo *zip*. Uma a uma essas estruturas preditas são comparadas com a estrutura de referência e é gerado um *score*, o qual indica o quanto a estrutura predita é similar a estrutura de referência.

Como resultado, a ferramenta gerou uma página HTML com uma tabela ordenada com base nos valores do *score* calculado de cada PDB (solução). Além disso é possível visualizar uma figura em 3D das estruturas com melhor pontuação. Todos os arquivos gerados na execução da ferramenta ficam disponíveis para o usuário realizar o *download*.

As Tabelas 13, 14, 15 e 16 mostram os 10 melhores RMSD *scores* calculados para as predições das proteínas 2EVQ, 1A11, 1VII e 1ERC, respectivamente, realizadas com todos os algoritmos. Todos os valores estão em ângström (Å).

Tabela 13: *Ranking* dos 10 melhores RMSD calculados para a proteína 2EVQ.

Nº	Proteína	Algoritmo	Estrutura predita	RMSD (Å)
1	2EVQ	2PG-Mono	monoSolution-M3.pdb	3.653
2	2EVQ	2PG-Mono	monoSolution-M17.pdb	3.696
3	2EVQ	2PG-Mono	monoSolution-M2.pdb	3.704
4	2EVQ	2PG-Mono	monoSolution-M9.pdb	3.762
5	2EVQ	2PG-Mono	monoSolution-M15.pdb	3.767
6	2EVQ	2PG-Mono	monoSolution-M18.pdb	3.776
7	2EVQ	2PG-Mono	monoSolution-M10.pdb	3.776
8	2EVQ	2PG-Mono	monoSolution-M7.pdb	3.778
9	2EVQ	2PG-Mono	monoSolution-M6.pdb	3.786
10	2EVQ	2PG-Mono	monoSolution-M1.pdb	3.792

Os resultados para os algoritmos na abordagem totalmente *ab initio* são promissores. A partir das imagens 3D das soluções alinhadas com a nativa, é possível visualizar algumas soluções mais próximas da nativa, como é o caso do 2PG-NSGAI para a proteína 1VII e o ProtPredEDA-EDA para a proteína 1ERC. Mesmo que os valores de RMSD sejam altos, como nas proteínas 1VII e 1ERC, eles podem ser melhorados realizando refinamento e dinâmica molecular com as melhores soluções apresentadas. É possível observar também

Tabela 14: *Ranking* dos 10 melhores RMSD calculados para a proteína 1A11.

Nº	Proteína	Algoritmo	Estrutura predita	RMSD (Å)
1	1A11	ProtPredEDA_EDA	fasta.txt-108000.pdb	3.054
2	1A11	ProtPredEDA_EDA	fasta.txt-79500.pdb	3.150
3	1A11	ProtPredEDA_EDA	fasta.txt-124000.pdb	3.232
4	1A11	ProtPredEDA_EDA	fasta.txt-101000.pdb	3.240
5	1A11	ProtPredEDA_EDA	fasta.txt-81500.pdb	3.281
6	1A11	ProtPredEDA_EDA	fasta.txt-93500.pdb	3.285
7	1A11	ProtPredEDA_EDA	fasta.txt-146500.pdb	3.290
8	1A11	ProtPredEDA_EDA	fasta.txt-90000.pdb	3.298
9	1A11	ProtPredEDA_EDA	fasta.txt-92500.pdb	3.361
10	1A11	ProtPredEDA_EDA	fasta.txt-158000.pdb	3.364

Tabela 15: *Ranking* dos 10 melhores RMSD calculados para a proteína 1VII.

Nº	Proteína	Algoritmo	Estrutura predita	RMSD (Å)
1	1VII	ProtPredEDA_EDA	fasta.txt-601500.pdb	6.809
2	1VII	ProtPredEDA_rBOA	fasta.txt-800000.pdb	6.911
3	1VII	ProtPredEDA_rBOA	fasta.txt-687500.pdb	6.964
4	1VII	ProtPredEDA_EDA	fasta.txt-474500.pdb	7.045
5	1VII	ProtPredEDA_EDA	fasta.txt-983000.pdb	7.177
6	1VII	ProtPredEDA_EDA	fasta.txt-228000.pdb	7.188
7	1VII	ProtPredEDA_rBOA	fasta.txt-672500.pdb	7.226
8	1VII	ProtPredEDA_EDA	fasta.txt-686000.pdb	7.275
9	1VII	ProtPredEDA_EDA	fasta.txt-461000.pdb	7.296
10	1VII	ProtPredEDA_EDA	fasta.txt-401000.pdb	7.299

Tabela 16: *Ranking* dos 10 melhores RMSD calculados para a proteína 1ERC.

Nº	Proteína	Algoritmo	Estrutura predita	RMSD (Å)
1	1ERC	ProtPredEDA_DE	fasta.txt-95500.pdb	6.763
2	1ERC	ProtPredEDA_MCM	fasta.txt-34500.pdb	7.354
3	1ERC	ProtPredEDA_MCM	fasta.txt-31000.pdb	7.379
4	1ERC	ProtPredEDA_GA	fasta.txt-75000.pdb	7.379
5	1ERC	ProtPredEDA_GA	fasta.txt-68000.pdb	7.405
6	1ERC	ProtPredEDA_GA	fasta.txt-139500.pdb	7.407
7	1ERC	ProtPredEDA_GA	fasta.txt-147000.pdb	7.408
8	1ERC	ProtPredEDA_GA	fasta.txt-113000.pdb	7.409
9	1ERC	ProtPredEDA_GA	fasta.txt-140000.pdb	7.411
10	1ERC	ProtPredEDA_GA	fasta.txt-135000.pdb	7.412

que os algoritmos do ProtPredEDA apresentarem os menores RMSD em relação aos outros algoritmos. Vale ressaltar que o ProtPredEDA utiliza a representação da proteína em diedral, enquanto que os algoritmos do 2PG utiliza a representação em cartesiano. A Tabela 17 apresenta os valores de RMSD calculados pelos algoritmos disponíveis no Koala para as proteínas utilizadas nos experimentos e disponíveis na literatura. É importante ressaltar que nem todos os algoritmos já realizaram predições com todas essas proteínas.

Tabela 17: Valores de RMSD disponíveis na literatura para as proteínas utilizadas nos experimentos. Os valores com * receberam refinamento ou dinâmica molecular.

Proteína	Algoritmo	RMSD (Å)
2EVQ	MEAMT	1.783
1A11	ProtPredEDA ²	2.280
1VII	2PG-NSGAI ³	4.06*
	MEAMT ⁴	5.130
1ERC	MEAMT	5.536

Todo esse processo de análise pode ser realizado automaticamente com o uso de fluxos de trabalho (*workflow*) no Koala, facilitando a realização dos experimentos no servidor. A Seção 6.2 mostra como podem ser criados esses fluxos e como eles funcionam.

6.1.2 Comparação sem estrutura de referência

O objetivo desta seção é apresentar uma outra forma de análise de predição que pode ser realizada no ambiente do Koala. Através da ferramenta *Sort by Front Dominance*, é possível selecionar as possíveis melhores soluções dentre um conjunto de soluções geradas por vários preditores para uma mesma proteína alvo. Ao final da análise, o sistema gera um *ranking* com todas as estruturas ordenado pela dominância que cada solução exerce sobre as demais.

Para cada solução (arquivo PDB) de uma proteína alvo, são calculados dois *scores* para uma combinação de dois objetivos. Utilizando a técnica de Fronteira de Pareto, uma solução domina ou é dominada por outras. As soluções que mais dominam outras e não são dominadas são classificadas na fronteira zero, indicando que são as possíveis melhores soluções. As demais são agrupadas em outras fronteiras. Depois de agrupadas em fronteiras, as soluções são ordenadas pela dominância. Assim, a solução que não é dominada por nenhuma outra domina todas as outras e é classificada na posição 1, seguida das demais que dominam menos e/ou são dominadas por outras soluções.

As ferramentas que fazem as análises sem estrutura de referência estão localizadas na seção do painel esquerdo chamada de *Solution Ranking*. Ao clicar nesta seção, ela se

²(BONETTI, 2015)

³(FACCIOLI, 2012)

⁴(BRASIL; DELBEM; SILVA, 2013)

expande e exibe as duas ferramentas disponíveis para *ranking* de soluções de preditores. Uma vez que a ferramenta é selecionada, as suas opções são carregadas no painel central junto com um botão para executá-la.

A análise realizada no Koala recebeu como parâmetro de entrada vários arquivos compactados contendo os resultados de todos os algoritmos nas predições descritas na Seção 6.1 para cada proteína. Além disso, foi definido a combinação de objetivos *Potential* e *GBSA Solvation*, os mesmos que foram usados nas predições.

É importante ressaltar que para realizar a análise não foi inserido nenhuma informação da proteína predita, tão pouco sobre os algoritmos que geraram as soluções, mostrando que o método é totalmente independente e funciona para qualquer proteína que esteja descrita em um arquivo PDB.

No final da execução, como resultado, a ferramenta criou um gráfico interativo com as quatro primeiras Fronteiras de Pareto, ordenou as melhores soluções dentre todas as preditas pelos algoritmos em uma tabela e gerou uma imagem 3D das 10 melhores.

As Tabelas 18, 19, 20 e 21 apresentam as 10 melhores soluções indicadas pela ferramenta *Sort by Front Dominance*, enquanto que as Figuras 63, 64, 65 e 66 apresentam o gráfico das Fronteiras de Pareto para as proteínas 2EVQ, 1A11, 1VII e 1ERC, respectivamente. Os gráficos foram gerados pela biblioteca Highcharts e adaptado (HIGHSOFT, 2009).

Tabela 18: *Ranking* com as 10 melhores soluções de acordo com o método *Sort by Front Dominance* para a proteína 2EVQ.

Proteína	Algoritmo	Estrutura predita	<i>Ranking</i> Dominância	Fronteira	Potential GBSA	RMSD
2EVQ	2PG-Mono	NSGAIIISolutions-M13.pdb	1 2420	0	-1.662 -2.285	8.289
2EVQ	2PG-Mono	NSGAIIISolutions-M18.pdb	2 2416	0	-1.659 -2.295	8.242
2EVQ	2PG-Mono	NSGAIIISolutions-M19.pdb	3 2416	0	-1.659 -2.295	8.242
2EVQ	2PG-Mono	NSGAIIISolutions-M16.pdb	4 2415	0	-1.660 -2.290	8.267
2EVQ	2PG-NSGA2	NSGAIIISolutions-M12.pdb	5 2414	0	-1.660 -2.287	8.273
2EVQ	2PG-NSGA2	NSGAIIISolutions-M20.pdb	6 2411	0	-1.658 -2.297	8.244
2EVQ	2PG-NSGA2	NSGAIIISolutions-M1.pdb	7 2409	0	-1.662 -2.260	8.414
2EVQ	2PG-NSGA2	monoSolution-M15.pdb	8 1382	0	-1.713 -1.479	3.767
2EVQ	2PG-NSGA2	monoSolution-M9.pdb	9 1381	0	-1.713 -1.469	3.762
2EVQ	2PG-NSGA2	monoSolution-M16.pdb	10 1379	0	-1.713 -1.466	3.800

Com esse experimento, foi possível observar a diferença entre os resultados apresentados pela métrica RMSD e *Sort by front dominance*, uma vez que a melhor solução

Figura 63: Gráfico das quatro primeiras Fronteira de Pareto da proteína 2EVQ, com um total de 44 soluções classificadas nessas fronteiras usando a combinação de objetivos *Potential* e *GBSA Solvation*. Os números próximos a alguns pontos indicam o grau de dominância da solução.

Tabela 19: *Ranking* com as 10 melhores soluções de acordo com o método *Sort by Front Dominance* para a proteína 1A11.

Proteína	Algoritmo	Estrutura predita	<i>Ranking</i> Dominância	Fronteira	Potential GBSA	RMSD
1A11	ProtPredEDA-rBOA	fasta.txt-427500.pdb	1	0	-2.869	7.981
			2278		-1.982	
1A11	ProtPredEDA-rBOA	fasta.txt-320000.pdb	2	0	-2.856	5.803
			2271		-2.009	
1A11	ProtPredEDA-rBOA	fasta.txt-312500.pdb	3	0	-2.825	7.708
			2250		-2.010	
1A11	ProtPredEDA-DE	fasta.txt-79500.pdb	4	0	-2.807	9.619
			2223		-2.062	
1A11	ProtPredEDA-rBOA	fasta.txt-292500.pdb	5	0	-2.756	9.064
			1912		-2.109	
1A11	ProtPredEDA-DE	fasta.txt-28000.pdb	6	0	-2.632	7.194
			1871		-2.138	
1A11	ProtPredEDA-rBOA	fasta.txt-435000.pdb	7	0	-2.891	9.675
			1026		-1.937	
1A11	ProtPredEDA-EDA	fasta.txt-73000.pdb	8	0	-2.922	5.100
			1020		-1.932	
1A11	ProtPredEDA-EDA	fasta.txt-80500.pdb	9	0	-2.944	4.341
			942		-1.917	
1A11	2PG-NSGAI	NSGAIISolutions-M16.pdb	10	0	-3.434	6.898
			789		-1.830	

Figura 64: Gráfico das quatro primeiras Fronteira de Pareto da proteína 1A11, com um total de 79 soluções classificadas nessas fronteiras usando a combinação de objetivos *Potential* e *GBSA Solvation*. Os números próximos a alguns pontos indicam o grau de dominância da solução.

Tabela 20: *Ranking* com as 10 melhores soluções de acordo com o método *Sort by Front Dominance* para a proteína 1VII.

Proteína	Algoritmo	Estrutura predita	<i>Ranking</i> Dominância	Fronteira	Potential GBSA	RMSD
1VII	2PG-Mono	monoSolution-M10.pdb	1	0	-4.783	9.365
			2589		-4.051	
1VII	2PG-Mono	monoSolution-M3.pdb	2	0	-4.785	9.300
			2587		-4.045	
1VII	2PG-Mono	monoSolution-M7.pdb	3	0	-4.786	9.365
			2585		-4.036	
1VII	2PG-Mono	monoSolution-M6.pdb	4	0	-4.785	9.384
			2583		-4.036	
1VII	2PG-NSGA2	NSGAIIISolutions-M20.pdb	5	0	-4.857	7.585
			2447		-3.858	
1VII	2PG-NSGA2	NSGAIIISolutions-M17.pdb	6	0	-4.859	7.681
			2419		-3.828	
1VII	2PG-NSGA2	NSGAIIISolutions-M18.pdb	7	0	-4.859	7.681
			2419		-3.828	
1VII	2PG-NSGA2	NSGAIIISolutions-M19.pdb	8	0	-4.859	7.681
			2419		-3.828	
1VII	2PG-NSGA2	NSGAIIISolutions-M10.pdb	9	0	-4.860	7.628
			2405		-3.820	
1VII	2PG-NSGA2	NSGAIIISolutions-M11.pdb	10	0	-4.860	7.628
			2405		-3.820	

Pareto Fronts

A graph with Potential and GBSA_Solvation objectives showing at most four fronts, but the first as line and the others as scatter

Figura 65: Gráfico das quatro primeiras Fronteira de Pareto da proteína 1VII, com um total de 68 soluções classificadas nessas fronteiras usando a combinação de objetivos *Potential* e *GBSA Solvation*. Os números próximos a alguns pontos indicam o grau de dominância da solução.

Tabela 21: *Ranking* com as 10 melhores soluções de acordo com o método *Sort by Front Dominance* para a proteína 1ERC.

Proteína	Algoritmo	Estrutura predita	<i>Ranking</i> Dominância	Fronteira	Potential GBSA	RMSD
1ERC	2PG-NSGAI	NSGAIISolutions-M12.pdb	1 2365	0	-4.961 -5.466	9.346
1ERC	2PG-NSGAI	NSGAIISolutions-M11.pdb	2 2365	0	-4.961 -5.466	9.346
1ERC	2PG-NSGAI	NSGAIISolutions-M19.pdb	3 2359	0	-4.957 -5.575	9.287
1ERC	2PG-NSGAI	NSGAIISolutions-M20.pdb	4 2359	0	-4.956 -5.580	9.279
1ERC	2PG-NSGAI	NSGAIISolutions-M16.pdb	5 2357	0	-4.958 -5.544	9.387
1ERC	2PG-NSGAI	NSGAIISolutions-M2.pdb	6 2355	0	-4.962 -5.396	9.502
1ERC	2PG-NSGAI	NSGAIISolutions-M1.pdb	7 2355	0	-4.962 -5.396	9.502
1ERC	ProtPredEDA_DE	fasta.txt-28000.pdb	8 19	0	-1.536 -5.978	11.365
1ERC	2PG-NSGAI	NSGAIISolutions-M8.pdb	9 2359	1	-4.960 -5.450	9.397
1ERC	2PG-NSGAI	NSGAIISolutions-M13.pdb	10 2359	1	-4.959 -5.465	9.373

Figura 66: Gráfico das quatro primeiras Fronteira de Pareto da proteína 1ERC, com um total de 39 soluções classificadas nessas fronteiras usando a combinação de objetivos *Potential* e *GBSA Solvation*. Os números próximos a alguns pontos indicam o grau de dominância da solução.

encontrada por ambas são diferentes. São métodos diferentes que usam técnicas distintas para analisar a melhor estrutura predita. Enquanto o RMSD verifica o posicionamento dos átomos com base em uma estrutura de referência, o *Sort by front dominance* avalia a estrutura através de uma combinação de dois objetivos.

Estudos apontam que o RMSD possui certas limitações e problemas ao calcular a qualidade de estrutura proteína em relação a estrutura nativa de uma proteína alvo. Portanto, o valor do RMSD pode não apresentar o quão iguais duas estruturas são (XU; ZHANG, 2010). O método *Sort by front dominance* é uma alternativa para avaliar uma outra informação a fim de encontrar uma melhor solução em um conjunto de soluções, sem dependências de estrutura de referência. A Tabela 22 apresenta a diferença entre os *rankings* gerados pelos dois métodos para as melhores soluções apresentadas por cada um.

Uma forma de avaliar a qualidade de uma estrutura de proteína é utilizando o gráfico de Ramachandran, o qual descreve as regiões favorecidas, permitidas e não permitidas para todos os aminoácidos que compõem a proteína. Assim, se o gráfico apontar muitos aminoácidos em regiões não permitida, pode indicar um problema com a estrutura. Por

Tabela 22: Relação das melhores soluções apresentadas pelos métodos *Sort by Front Dominance* e pelo RMSD, indicando onde a melhor solução de um método está posicionada no outro.

Proteína	Algoritmo	Estrutura predita	<i>Ranking Sort</i>	<i>Ranking RMSD</i>
2EVQ	2PG-NSGAI	NSGAIISolutions-M13.pdb	1	2435
2EVQ	2PG-Mono	monoSolution-M3.pdb	40	1
1A11	ProtPredEDA-rBOA	fasta.txt-427500.pdb	1	1872
1A11	ProtPredEDA-EDA	fasta.txt-108000.pdb	514	1
1VII	2PG-Mono	monoSolution-M10.pdb	1	787
1VII	ProtPredEDA-EDA	fasta.txt-601500.pdb	128	1
1ERC	2PG-NSGAI	NSGAIISolutions-M12.pdb	1	1410
1ERC	ProtPredEDA-DE	fasta.txt-95500.pdb	1434	1

conta disso, o gráfico de Ramachandran é muito utilizado para determinar a qualidade de estruturas experimentais (RAMACHANDRAN; RAMAKRISHNAN; SASISEKHARAN, 1963).

Neste trabalho, o gráfico de Ramachandran, o qual exibe pontos em preto e amarelo para indicar os aminoácidos que estão dentro da região favorecida ou permitida, e os pontos em vermelho para mostrar os aminoácidos em região atípica, foi empregado para comparar as primeiras estruturas dos *rankings* do RMSD e do *Sort*, com o intuito de verificar qual das métricas apresentou a melhor distribuição de aminoácidos nas regiões permitidas do gráfico de Ramachandran. Além disso, as estruturas com menor RMSD e primeira posição do *ranking* do *Sort by front dominance* foram alinhadas com a estrutura nativa.

Para a proteína 2EVQ, tanto as soluções apresentadas pelo *Sort by front dominance* e pelo RMSD apresentaram uma ótima distribuição dos aminoácidos no gráfico de Ramachandran, com 100% dos aminoácidos na região favorecida e 0% na atípica. No alinhamento com a nativa, a solução com menor RMSD se aproximou mais da estrutura de referência, enquanto que a melhor solução apresentada pelo *Sort by front dominance* apresentou uma estrutura que contém uma hélice- α .

A Figura 67 apresenta o gráfico de Ramachandran para a estrutura gerada pelo algoritmo 2PG-NSGAI e melhor posicionada no *ranking* do *Sort by front dominance*, e a Figura 68 apresenta o gráfico para a solução gerada pelo algoritmo 2PG-Mono com menor RMSD. Ademais, a Figura 69 mostra o alinhamento das duas estruturas com a estrutura nativa da proteína 2EVQ.

Para a proteína 1A11, ambas as soluções apresentadas pelo *Sort by front dominance* e pelo RMSD apresentaram uma ótima distribuição dos aminoácidos no gráfico de Ramachandran, com 100% dos aminoácidos na região favorecida e 0% na atípica. No alinhamento com a nativa, a solução com menor RMSD se aproximou mais da estrutura de referência, enquanto que a melhor solução apresentada pelo *Sort by front dominance* apresentou uma estrutura mais aberta, porém poderia se fechar caso aumentasse a quantidade

Figura 67: Gráfico de Ramachandran para a solução do *Sort by front dominance* para a proteína 2EVQ, gerada pelo algoritmo 2PG-NSGAIL. Gerado e adaptado de (LOVELL et al., 2003).

de gerações.

A Figura 70 apresenta o gráfico de Ramachandran para a estrutura gerada pelo algoritmo ProtPredEDA-rBOA e melhor posicionada no *ranking* do *Sort by front dominance*, e a Figura 71 apresenta o gráfico para a solução gerada pelo algoritmo ProtPredEDA-EDA com menor RMSD. Ademais, a Figura 72 mostra o alinhamento das duas estruturas com a estrutura nativa da proteína 1A11.

Para a proteína 1VII, a solução apresentada pelo método *Sort* possui mais de 90% dos aminoácidos na região favorecida, enquanto que a solução com menor RMSD possui pouco mais de 80%, e ambas apresentam dois aminoácidos (6.1%) na região atípica. No alinhamento, é possível ver que a estrutura com menor RMSD está mais próxima da estrutura nativa, mas a solução apresentada pelo *Sort by Front Dominance* também está

Figura 68: Gráfico de Ramachandran para a solução com menor RMSD para a proteína 2EVQ, gerada pelo algoritmo 2PG-Mono. Gerado e adaptado de (LOVELL et al., 2003).

(a) Solução do *ranking* do *Sort*.

(b) Solução com menor RMSD.

Figura 69: Comparação entre as estruturas de melhor *ranking* pelo método *Sort by front dominance* (a) e com menor RMSD (b) para a proteína 2EVQ. As estruturas foram alinhadas e as imagens geradas pelo programa Pymol (Schrödinger, LLC, 2010).

Figura 70: Gráfico de Ramachandran para a solução do *Sort by front dominance* para a proteína 1A11, gerada pelo algoritmo ProtPredEDA-rBOA. Gerado e adaptado de (LOVELL et al., 2003).

com uma conformação parecida.

A Figura 73 apresenta o gráfico de Ramachandran para a estrutura gerada pelo algoritmo 2PG-Mono e melhor posicionada no *ranking* do *Sort by front dominance*, e a Figura 74 apresenta o gráfico para a solução gerada pelo algoritmo ProtPredEDA-EDA com menor RMSD. Ademais, a Figura 75 mostra o alinhamento das duas estruturas com a estrutura nativa da proteína 1VII.

Para a proteína 1ERC, a solução melhor colocada no *ranking* do *Sort* apresenta 97.3% dos aminoácidos na região favorecida e 0% na região atípica, enquanto que a solução de menor RMSD três aminoácidos (8.1%) na região atípica. No alinhamento é possível visualizar que a estrutura com menor RMSD está bem mais aberta do que a melhor posicionada no *ranking* do *Sort by Front Dominance*, indicando que o algoritmo poderia

Figura 71: Gráfico de Ramachandran para a solução com menor RMSD para a proteína 1A11, gerada pelo algoritmo ProtPredEDA-EDA. Gerado e adaptado de (LOVELL et al., 2003).

(a) Solução do *ranking* do *Sort*.

(b) Solução com menor RMSD.

Figura 72: Comparação entre as estruturas de melhor *ranking* pelo método *Sort by front dominance* (a) e com menor RMSD (b) para a proteína 1A11. As estruturas foram alinhadas e as imagens geradas pelo programa Pymol (Schrödinger, LLC, 2010).

Figura 73: Gráfico de Ramachandran para a solução do *Sort by front dominance* para a proteína 1VII, gerada pelo algoritmo 2PG-Mono. Gerado e adaptado de (LOVELL et al., 2003).

convergir para uma solução melhor.

A Figura 76 apresenta o gráfico de Ramachandran para a estrutura gerada pelo algoritmo 2PG-NSGAI e melhor posicionada no *ranking* do *Sort by front dominance*, e a Figura 77 apresenta o gráfico para a solução gerada pelo algoritmo ProtPredEDA-DE com menor RMSD. Ademais, a Figura 78 mostra o alinhamento das duas estruturas com a estrutura nativa da proteína 1ERC.

Os resultados finais mostraram que as soluções apresentadas pelo método *Sort by front dominance* podem ser mais promissoras do que as com menor RMSD. Assim, essas estruturas podem ser usadas como entrada para um processo de dinâmica molecular, a fim de melhorar o modelo predito. Essa técnica foi aplicada em 2014 no CASP11 em dois *branches*: um apenas com os algoritmos do 2PG e o outro em cooperação com o projeto

Figura 74: Gráfico de Ramachandran para a solução com menor RMSD para a proteína 1VII, gerada pelo algoritmo ProtPredEDA-EDA. Gerado e adaptado de (LOVELL et al., 2003).

(a) Solução do *ranking* do *Sort*.

(b) Solução com menor RMSD.

Figura 75: Comparação entre as estruturas de melhor *ranking* pelo método *Sort by front dominance* (a) e com menor RMSD (b) para a proteína 1VII. As estruturas foram alinhadas e as imagens geradas pelo programa Pymol (Schrödinger, LLC, 2010).

Figura 76: Gráfico de Ramachandran para a solução do *Sort by front dominance* para a proteína 1ERC, gerada pelo algoritmo 2PG-NSGA2. Gerado e adaptado de (LOVELL et al., 2003).

WeFold, e apresentou bons resultados em alguns casos, mostrando que em um cenário onde não existe informação *a priori* da proteína, este método pode indicar boas soluções promissoras (FACCIOLI; BORTOT; DELBEM, 2014) (FACCIOLI et al., 2014).

Apenas como referência, foi avaliada a acurácia do método *Sort by front dominance* em classificar, até a fronteira quatro, as melhores soluções apresentadas pelo *ranking* do RMSD. O método classificou, para a proteína 2EVQ, 80% das soluções, enquanto que usando os dados do CASP11 (Seção a seguir), o método conseguiu classificar, para um dos alvos, 100% das soluções até a fronteira quatro.

No CASP, a quantidade de preditores é bem maior do que os algoritmos que foram usados nos experimentos, além de não serem totalmente *ab initio*, o que justifica as melhores soluções encontradas. Ademais, a combinação de objetivos que foi utilizada pode

Figura 77: Gráfico de Ramachandran para a solução com menor RMSD para a proteína 1ERC, gerada pelo algoritmo ProtPredEDA-DE. É possível visualizar os aminoácidos glicina (GLY) e prolina (PRO) em regiões não permitida. Gerado e adaptado de (LOVELL et al., 2003).

(a) Solução do *ranking* do *Sort*.

(b) Solução com menor RMSD.

Figura 78: Comparação entre as estruturas de melhor *ranking* pelo método *Sort by front dominance* (a) e com menor RMSD (b) para a proteína 1ERC. As estruturas foram alinhadas e as imagens geradas pelo programa Pymol (Schrödinger, LLC, 2010).

não ser a melhor para todos os casos. A Tabela 23 apresenta as taxas de classificação do método para as proteínas utilizadas nos experimentos.

Tabela 23: Taxa de sucesso do método *Sort by front by dominance* em classificar as 10 melhores soluções, de acordo com o *ranking* do RMSD, nas 4 primeiras fronteiras, para as quatro proteínas utilizadas nos experimentos.

Alvo	Resíduos	Qtde modelos	Menor RMSD (Å)	Taxa de classificação
2EVQ	12	2452	3.653	80%
1A11	25	2556	3.054	20%
1VII	36	2716	6.809	0%
1ERC	40	2561	6.763	0%

Experimentos com dados do CASP11

Para demonstrar a independência e eficiência do método *Sort by Front Dominance*, uma outra análise foi realizada utilizando os dados do CASP11, na qual foram selecionados 10 *targets* usados na versão 11 do CASP, que foi realizada em 2014 (CASP11, 2014). É importante ressaltar que os alvos determinados pelo CASP não tinham a estrutura terciária conhecida e participaram todos os tipos de preditores, como totalmente *ab initio* ou os que realizam predição por homologia.

Para cada alvo, os preditores submeteram os 5 melhores modelos preditos. No término da competição, as estruturas dos alvos foram determinadas experimentalmente e assim foi possível verificar quais as melhores predições realizadas. Assim, todos os modelos gerados foram comparados com uma estrutura de referência usando várias métricas, dentre elas o RMSD e o GDT-TS, esta última é a utilizada para gerar o *ranking*. A Tabela 24 sumariza os resultados apresentados pelo CASP para os 5 melhores modelos dos 10 alvos selecionados: T0760-D1, T0762-D1, T0763-D1, T0764-D1, T0765-D1, T0766-D1, T0768-D1, T0770-D1, T0771-D1, T0772-D1. Os alvos foram escolhidos aleatoriamente, mas com o critério de possuir somente um domínio. (Disponível em <http://www.predictioncenter.org/casp11/results.cgi>).

No Koala, a ferramenta *Sort by Front Dominance* recebeu como entrada todos os modelos gerados pelos participantes em cada um dos 10 alvos escolhidos, e foi definida a combinação de objetivos *Potential* e *GBSA Solvation*. A Tabela 25 sumariza os resultados apresentados pela ferramenta, mostrando a acurácia do método em classificar as 10 melhores soluções nas 4 primeiras fronteiras para cada alvo. A Figura 79 mostra os histogramas com as frequências das 10 melhores soluções classificadas entre as 4 (0, 1, 2 e 3) primeiras fronteiras para cada alvo, e a Figura 80 mostra os histogramas com as frequências de todas as soluções em todas as fronteiras para cada alvo.

Tabela 24: Resultados apresentados pelo CASP11 para os 10 *targets* utilizados no experimento com a ferramenta *Sort by front dominance* no Koala.

Alvo	Grupo	Modelo	Ranking	RMSD(Å)	GDT-TS
T0760-D1	MULTICOM-NOVEL	T0760TS041_1-D1	1	2.5	75.12
T0760-D1	MULTICOM-NOVEL	T0760TS041_2-D1	2	2.7	74.25
T0760-D1	MULTICOM-CLUSTER	T0760TS420_4-D1	3	2.6	74.13
T0760-D1	FALCON_MANUAL	T0760TS381_5-D1	4	3.1	73.88
T0760-D1	HHPredA	T0760TS346_1-D1	4	2.8	73.88
T0762-D1	myprotein-me	T0762TS216_3-D1	1	1.9	86.38
T0762-D1	MULTICOM-CLUSTER	T0762TS420_3-D1	1	1.9	86.38
T0762-D1	MULTICOM-CONSTRUCT	T0762TS008_3-D1	3	1.9	86.19
T0762-D1	MULTICOM-CONSTRUCT	T0762TS008_5-D1	4	1.9	86.09
T0762-D1	MULTICOM-CONSTRUCT	T0762TS008_1-D1	4	1.6	86.09
T0763-D1	CNIO	T0763TS067_1-D1	1	4.2	39.04
T0763-D1	Seok-refine	T0763TS425_1-D1	2	3.0	24.61
T0763-D1	Seok	T0763TS296_2-D1	3	3.1	24.42
T0763-D1	nns	T0763TS038_4-D1	4	4.5	23.08
T0763-D1	MULTICOM-CONSTRUCT	T0763TS008_1-D1	5	3.1	22.69
T0764-D1	Seok-server	T0764TS011_1-D1	1	2.3	78.50
T0764-D1	Seok-server	T0764TS011_2-D1	1	2.3	78.50
T0764-D1	QUARK	T0764TS499_1-D1	3	2.5	78.11
T0764-D1	QUARK	T0764TS499_2-D1	4	2.4	78.04
T0764-D1	Zhang-Server	T0764TS277_1-D1	5	2.3	77.80
T0765-D1	Wallner	T0765TS347_1-D1	1	2.2	83.22
T0765-D1	wfKsrFdit-BW-Sk-McG	T0765TS120_2-D1	1	1.9	83.22
T0765-D1	wfAll-Cheng	T0765TS403_3-D1	3	2.4	82.57
T0765-D1	QA-RecombineIt_WFH	T0765TS364_1-D1	3	2.3	82.57
T0765-D1	QA-RecombineIt_H2	T0765TS434_1-D1	3	1.9	82.57
T0766-D1	FLOUDAS_SERVER	T0766TS448_1-D1	1	1.1	94.91
T0766-D1	FLOUDAS_SERVER	T0766TS448_4-D1	1	1.1	94.91
T0766-D1	myprotein-me	T0766TS216_1-D1	3	1.1	94.68
T0766-D1	MULTICOM-REFINE	T0766TS268_3-D1	3	1.1	94.68
T0766-D1	Zhang-Server	T0766TS277_4-D1	5	1.1	94.44
T0768-D1	myprotein-me	T0768TS216_1-D1	1	1.5	86.89
T0768-D1	BAKER-ROSETTASERVER	T0768TS184_3-D1	2	1.6	86.71
T0768-D1	raghavagps-tsppred	T0768TS117_1-D1	3	1.8	82.34
T0768-D1	raghavagps-tsppred	T0768TS117_4-D1	4	1.9	81.29
T0768-D1	RaptorX	T0768TS050_1-D1	5	2.2	80.77
T0770-D1	HHPredA	T0770TS346_1-D1	1	2.4	68.59
T0770-D1	slbio	T0770TS492_1-D1	2	2.4	68.09
T0770-D1	MULTICOM-NOVEL	T0770TS041_2-D1	3	2.5	66.56
T0770-D1	RBO_Aleph	T0770TS479_2-D1	4	2.5	66.50
T0770-D1	RBO_Aleph	T0770TS479_1-D1	5	2.6	66.45
T0771-D1	NEFILIM	T0771TS054_3-D1	1	3.5	31.79
T0771-D1	BAKER	T0771TS064_3-D1	2	3.3	30.46
T0771-D1	SHORTLE	T0771TS241_1-D1	3	4.5	28.97
T0771-D1	MULTICOM	T0771TS290_2-D1	4	5.6	28.31
T0771-D1	BAKER-ROSETTASERVER	T0771TS184_5-D1	5	2.3	28.15
T0772-D1	nns	T0772TS038_1-D1	1	3.4	66.59
T0772-D1	nns	T0772TS038_2-D1	2	2.6	66.24
T0772-D1	RaptorX	T0772TS050_1-D1	3	3.1	66.12
T0772-D1	FALCON_MANUAL	T0772TS381_4-D1	3	2.7	66.12
T0772-D1	nns	T0772TS038_5-D1	5	2.8	65.77

Histograma das 10 melhores soluções

(a) Alvo T0760-D1

Histograma das 10 melhores soluções

(b) Alvo T0762-D1

Histograma das 10 melhores soluções

(c) Alvo T0763-D1

Histograma das 10 melhores soluções

(d) Alvo T0764-D1

Histograma das 10 melhores soluções

(e) Alvo T0765-D1

Histograma com as 10 melhores soluções

(f) Alvo T0766-D1

Histograma das 10 melhores soluções

(g) Alvo T0768-D1

Histograma das 10 melhores soluções

(h) Alvo T0770-D1

Histograma das 10 melhores soluções

(i) Alvo T0771-D1

Histograma das 10 melhores soluções

(j) Alvo T0772-D1

Figura 79: Histograma das 10 melhores soluções do CASP11 com a distribuição entre as fronteiras, mostrando uma concentração das soluções nas primeiras fronteiras.

(a) Alvo T0760-D1

(b) Alvo T0762-D1

(c) Alvo T0763-D1

(d) Alvo T0764-D1

(e) Alvo T0765-D1

(f) Alvo T0766-D1

(g) Alvo T0768-D1

(h) Alvo T0770-D1

(i) Alvo T0771-D1

(j) Alvo T0772-D1

Figura 80: Histograma com todas as soluções de todos os alvos classificados pelo método *Sort* para todas as fronteiras geradas, demonstrando uma concentração nas primeiras fronteiras.

Tabela 25: Taxa de sucesso do método *Sort by front by dominance* em classificar as 10 melhores soluções nas 4 primeiras fronteiras.

Alvo	Resíduos	Qtde modelos	Maior GDT	Taxa de classificação
T0760-D1	201	175	75.12	90%
T0762-D1	257	178	86.38	80%
T0763-D1	130	501	39.04	50%
T0764-D1	312	182	78.50	100%
T0765-D1	76	477	83.22	60%
T0766-D1	108	181	94.91	50%
T0768-D1	143	186	86.89	50%
T0770-D1	456	181	68.59	60%
T0771-D1	151	436	31.79	60%
T0772-D1	214	190	66.59	60%

Os resultados da ferramenta *Sort by front dominance* com os dados do CASP11 mostraram um potencial muito grande para esse método, o qual pode ser melhorado, estudando a melhor combinação de objetivos para avaliar as soluções. Em destaque os resultados obtidos com os alvos T0760-D1, T0762-D1 e T0764-D1, onde a taxa de classificação foi alta, chegando a 100% em um dos casos. Isso indica que quase 70% das 10 melhores soluções apareceram até a fronteira quatro, o que é mostrado nos histogramas, tanto em relação a todas as fronteiras como também em relação às 10 melhores soluções. Alguns casos aparecem uma classificação bem ruim, como foi o caso dos alvos T0760-D1 e T0770-D1, que apareceu a fronteira 10 e 11 entre as 10 melhores soluções.

Esse método apresenta-se como uma alternativa para os métodos de comparação de solução de preditores. Em um cenário onde as proteínas não possuem a sua estrutura tridimensional determinada, os resultados desse método podem indicar quais as soluções mais promissoras, sem usar conhecimento prévio da proteína alvo.

6.2 Workflow

Esta seção tem como finalidade demonstrar o funcionamento de fluxos de trabalho (*workflow*) dentro do ambiente do Koala, utilizando as diversas ferramentas disponíveis. Com esse recurso, é possível realizar uma pesquisa que depende de vários passos (ferramentas) de forma automática.

Para demonstrar esse recurso, foram criados dois *workflows* para realizarem as análises descritas nas Seções 6.1.1 e 6.1.2, com e sem o uso de estruturas de referências, os quais foram chamados de *Comparison of predictions with reference* e *Ranking of solutions by Sort by Front Dominance*, respectivamente, e estão públicos para qualquer usuário importá-los para seu ambiente. As Figuras 81 e 82 apresentam um esquema geral dos *workflows* criados, com o intuito de ilustrar o processo de execução dos passos de um

fluxo genérico.

Figura 81: Esquema geral de execução dos passos de um *workflow* de predição e análise no Koala. Nesse caso, são mostrados os passos do *workflow Comparison of predictions with reference* como exemplo.

Figura 82: Esquema geral de execução dos passos de um *workflow* de predição e análise no Koala. Nesse caso, são mostrados os passos do *workflow Ranking of solutions by Sort by Front Dominance* como exemplo.

O recurso de *workflow*, como outros no Koala, só está disponível para usuários que realizaram o cadastro de usuário no servidor. A criação da conta é livre e simples, basta informar um nome e um endereço de email. O servidor só permite a criação de uma conta com o mesmo endereço de email.

Para criar um *workflow* no Koala, basta clicar na opção que existe no menu superior com o nome de *Workflow* e, em seguida, usar a opção *Create new workflow*. Ao invés de criar um novo, pode-se também importar um *workflow* de duas formas: (a) importando um arquivo salvo referente ao *workflow*; e (b) selecionando um *workflow* que esteja público no Koala. A partir disso, um *workbench* é exibido onde o usuário pode incluir, excluir e editar as ferramentas que fazem parte do *workflow* e como elas estão conectadas. No menu direito, estão disponíveis todos os parâmetros da ferramenta selecionada, os quais podem

ser definidos diretamente nesse ambiente ou serem solicitados no momento da execução dos passos. As Figuras 83 e 84 mostram o *workbench* com as ferramentas usadas nos dois *workflows* criados.

Figura 83: Workbench do workflow *Comparison of predictions with reference* para predição e análise com estrutura de referência.

Após a criação dos *workflows*, a execução (*run*) pode ser realizada. Usando a mesma opção do menu superior *Workflow*, uma lista com todos os *workflows* é exibida. Em cada um existe um sub-menu vinculado com várias opções. Selecionando a opção *Run*, todos os

Figura 84: Workbench do workflow *Ranking of solutions by Sort by Front Dominance* para predição e análise sem estrutura de referência.

passos (*steps*) são exibidos no painel central. Cada passo corresponde a uma ferramenta e contém os campos para todos os seus parâmetros, e podem receber valores personalizados para cada execução.

A sequência de execução dos passos é definida de acordo com a dependência gerada pela conexão entre as ferramentas, assim as que não dependem de nenhuma outra são executadas primeiro, obedecendo a execução de um fluxo de trabalho sequencial. Os

resultados do *workflow* podem ser enviados para um novo item do histórico (*history*) do usuário, organizando as informações em um novo item pro usuário. As Figuras 85 e 86 mostram a execução dos dois *workflows* que foram criados nesse trabalho.

Figura 85: Passos de execução do *workflow Comparison of predictions with reference* ao ser executado.

O resultado de um *workflow* está condicionado a uma marcação no momento da sua criação. Ao inserir uma ferramenta no *workbench* do *workflow*, seu *output* fica desmarcado. Isso significa que o resultado da execução dessa ferramenta ficará oculto no painel de histórico do usuário. É preciso selecionar qual ferramenta terá o seu resultado exibido no *history*. Para os *workflows* criados nesse trabalho, os resultados das ferramentas de predição e análises foram marcados para exibição, os demais ficaram ocultos.

As ferramentas do Koala foram desenvolvidas para que pudessem ser integradas umas com as outras, favorecendo o uso de *workflows*. Assim, esse sistema facilita o uso dos vários módulos disponíveis, principalmente para usuários que não possuem experiência em computação, pois oferece uma maneira de automatizar e personalizar a execução de várias ferramentas através de um ambiente gráfico.

Outros *workflows* foram criados com diversas combinações de ferramentas e estão

Running workflow "Ranking of solutions by Sort by Front Dominance"

Generic workflow to perform predictions and analysis using the Sort by front dominance metric.

Step 1: Input dataset
Step 2: 2PG Build Population (version 1.0.0)
Step 3: 2PG Build Population (version 1.0.0)
Step 4: ProtPred-EDA Monte Carlo (version 1.0.0)
Step 5: ProtPred-EDA RW (version 1.0.0)
Step 6: ProtPred-EDA EDA (version 1.0.0)
Step 7: ProtPred-EDA rBOA (version 1.0.0)
Step 8: ProtPred-EDA Genetic Algorithm (version 1.0.0)
Step 9: ProtPred-EDA DE (version 1.0.0)
Step 10: MEAMT Build Population (version 1.0.0)
Step 11: 2PG Mono (version 1.0.0)
Step 12: 2PG NSGA2 (version 1.0.0)
Step 13: 2PG Random Algorithm, (version 1.0.0)
Step 14: 2PG Monte Carlo Metropolis (version 1.0.0)
Step 15: MEAMT algorithm (version 1.0.0)
Step 16: Sort by Front Dominance (version 1.0.0)

☒ Send results to a new history named:

Figura 86: Passos de execução do *workflow Ranking of solutions by Sort by Front Dominance* ao ser executado.

disponíveis no servidor Koala para todos os usuários através do link:

http://200.144.255.35:8084/workflow/list_published.

6.3 Reprodutibilidade

Esta seção tem como objetivo demonstrar uma das técnicas de reprodutibilidade que está disponível no servidor Koala, chamada de *Page*, a qual consiste em uma página *web* descrevendo alguma pesquisa realizada no servidor, podendo relacionar os itens gerados (*history*, *datasets*, *inputs* e *outputs*) na execução de ferramentas e *workflows* de forma intuitiva e transparente.

Com esse recurso, é possível criar uma descrição completa dos passos realizados no Koala para chegar a um resultado. A página aceita também a inserção de *links* para objetos do Koala, como um arquivo usada como entrada para uma ferramenta (*input*), e embutir esses objetos diretamente na página. Um objeto embutido significa que ele está

público e acessível a todos. Na própria *Page*, o usuário pode selecionar um determinado objeto e importá-lo para o seu ambiente para reproduzir o que está sendo descrito na página.

Além dos objetos do Koala, é possível também inserir outros tipos, como imagem, links para sites da internet e tabelas. Assim, esse recurso pode ser usado como uma extensão de um artigo, por exemplo, com conteúdo extra e a descrição completa dos resultados obtidos e dos arquivos e parâmetros utilizados. Como esses recursos são públicos, um revisor ou um leitor pode reproduzir tudo o que foi descrito, fazendo com que a pesquisa se torne mais transparente e reprodutível.

Para demonstrar esse recurso no Koala, foi criada a página *Predições e análises no Koala*, a qual descreve os experimentos realizados neste trabalho de mestrado, desde a predição até a análise, usando as ferramentas e *workflows* disponíveis. Tudo o que foi feito pode ser reproduzido por qualquer usuário.

Para criar uma *Page*, o usuário precisa ter uma conta no Koala na qual essa página será vinculada. Quando o usuário entrar no Koala, no menu superior *User*, terá uma opção chamada de *Saved Pages*, onde estão localizadas todas as páginas criadas por ele. Nesta tela tem um botão chamado de *Add new page*, que é usado para criar uma nova página. As informações iniciais são relacionadas ao título e descrição da página, após isso é mostrado um editor de texto comum e outras opções para inserir os objetos do Koala.

Na página *Predições e análise no Koala*, os objetos relacionados às execuções das ferramentas de predição foram separados em *histories*, um *history* para cada *framework* e seus resultados para as proteínas. As análises também foram organizadas em vários históricos, um para cada técnica de comparação e *ranking* de estruturas. Dentro de cada *history* estão todos os resultados obtidos com as execuções e os arquivos utilizados, além dos parâmetros definidos para cada ferramenta.

Todos esses *histories* foram embutidos na página criada, seguidos de uma descrição sobre o que representa cada um dos itens. Todos esses itens estão públicos no Koala e podem ser reproduzidos por outros usuários. A Figura 87 mostra a página criada.

Galaxy / Koala Analyze Data Workflow Shared Data Visualization Admin Help User Using 11.2 GB

Private Page | Predição e análise no Koala

Predições e análises no ambiente Koala: dados complementares

DEFELICIBUS, A¹; DELBEM, A. C. B^{1,2}; FACCIOLO, R³.

¹Programa de Pós-Graduação Interunidades Bioengenharia - Universidade de São Paulo
²Instituto de Ciências Matemáticas e da Computação - Universidade de São Paulo
³Centro Universitário Barão de Mauá - Ribeirão Preto

Como usar este documento

Este documento é uma cópia dos experimentos realizados neste trabalho de mestrado e contem informações suplementares em relação aos dados que foram apresentados na dissertação. Este disponibiliza acesso a todos os arquivos de entrada para todas as ferramentas usadas nos experimentos, como o acesso aos resultados gerados pelas ferramentas, então você pode reexecutá-los, mudar os parâmetros, ou até mesmo aplicá-los com o seu conjunto de dados próprios. Essencialmente, nesta página estão listados todos os itens gerados nos experimentos, que são cinco *histories* e dois *workflows*. Ao clicar em algum desses itens, seja pelo nome ou pelo ícone + no canto esquerdo, você o expande, exibindo todos os *datasets* (nos *histories*) ou *steps* (nos *workflows*). Você também pode importá-los para o seu ambiente e usá-los em seu ambiente do Koala; basta clicar no ícone + verde no canto direito para importar um item. Para importar ou salvar um *workflow* você precisa criar uma conta no Koala, caso você ainda não possua uma. O processo é bem simples e solicita somente um nome de usuário e uma senha.

Este é o *history* detalhando as predições realizadas com os algoritmos do *framework* 2PG para as proteínas 1VII, 1ERC, 1ZDD e 2X43.

Galaxy History | 2PG
 Resultados das predições com os algoritmos do 2PG para as proteínas 1VII, 1ERC, 1ZDD e 2X43.

Este é o *history* detalhando as predições realizadas com os algoritmos do *framework* ProtPred-EDA para as proteínas 1VII, 1ERC, 1ZDD e 2X43

Galaxy History | ProtPred-EDA
 Resultados das predições com os algoritmos do ProtPred-EDA para as proteínas 1VII, 1ERC, 1ZDD e 2X43.

Este é o *history* detalhando as predições realizadas com o algoritmo MEAMT para as proteínas 1VII, 1ERC, 1ZDD e 2X43

Galaxy History | MEAMT
 Resultados das predições com o algoritmo MEAMT para as proteínas 1VII, 1ERC, 1ZDD e 2X43.

Este *history* detalha as análises realizadas com as soluções geradas pelos preditores usando a métrica de comparação de estruturas RMSD, com base em uma estrutura de referência retirada do *Protein Data Bank*.

Galaxy History | RMSD
 Análise das predições usando a métrica RMSD.

Este *history* é o detalhamento das análises realizadas com as soluções geradas pelos preditores usando o método *Sort by Front Dominance*, o qual não utiliza uma estrutura como referência.

Galaxy History | Sort by front dominance
 Análise e ranking das soluções geradas pelos preditores usando o método *Sort by front dominance*.

Este é um *workflow* genérico para realizar a predição de estrutura de proteína usando todos os algoritmos disponíveis no Koala, e a análise dos resultados usando as métricas de comparação de estrutura RMSD, GDT-TS e TM-Score, com base em uma estrutura de referência.

Galaxy Workflow | Comparison of predictions with reference
 Workflow genérico para predição e análise usando as métricas RMSD, GDT-TS e TM-Score.

Este *workflow* genérico executa todos os algoritmos preditores disponíveis no Koala e a análise dos resultados usando o método *Sort by Front Dominance*, o qual não utiliza estrutura de referência.

Galaxy Workflow | Comparison of predictions without reference by Sort by Front Dominance
 Workflow genérico para predição e análise usando o método *Sort by Front Dominance*.

Figura 87: Page "Predições e análises no Koala" criada no servidor Koala para descrever os experimentos realizados neste trabalho.

Para que a página se torne acessível por todos no servidor Koala, é preciso publicá-la. Todos os itens no Koala, ao ser criado, se mantém privado para o usuário que o criou, e só depois de publicado é que ele pode ser acessado por outros usuários. A página *Prediction and analysis on Koala* foi publicada e pode ser acessada pelo link: <http://200.144.255.35:8084/u/adeFelicibus-15/p/predicao-e-analise-no-koala>.

6.4 Considerações Finais

Este capítulo mostrou os resultados obtidos dos experimentos realizados na plataforma *web* proposta nesse trabalho. Inicialmente, as proteínas utilizadas nos experimentos foram apresentadas, bem como o processo de *upload* dos arquivos necessários para o Koala. O objetivo dos experimentos era mostrar as funcionalidades da plataforma criada, detalhando as ferramentas e a forma de apresentação dos resultados de cada uma, além de outros recursos, como *workflow* e *Page*. Ademais, um novo método para avaliar soluções de preditores foi apresentado. Antes dos experimentos, foram definidos os parâmetros para cada método de predição.

Os primeiros experimentos executados mostram o uso dos algoritmos de predição que estão disponíveis no servidor Koala. Foram usadas quatro proteínas para os experimentos: 2EVQ, 1A11, 1VII e 1ERC, com diferentes tamanhos e com uma ou duas hélices- α em todos os preditores. Cada um foi calibrado com seus parâmetros individuais de acordo com experimentos prévios, com o intuito de utilizar os valores de parâmetros que apresentaram os melhores resultados. Todas as predições foram realizadas na infraestrutura do Koala. Mesmo que o objetivo não era comparar qual algoritmo preditor era melhor, com os resultados obtidos mostraram que todos possuem contribuições interessantes em determinado tipo de proteína.

Em seguida, foi realizado um experimento de análises dos resultados gerados pelo primeiro experimento, que foi a predição das estruturas de quatro proteínas. Para essa análise, foi usada uma ferramenta do Koala que executa métrica RMSD para comparação de estruturas utilizando uma estrutura de referência, a fim de encontrar a similaridade entre a estrutura predita e a nativa. Assim, os resultados obtidos com as predições pelos algoritmos disponíveis no Koala foram usados nesse experimento, com o intuito de encontrar o melhor modelo predito, entre todos gerados pelos preditores, usando a métrica RMSD. Foi possível observar que os algoritmos EDA, DE, MCM e rBOA do ProtPred-EDA obtiveram melhores resultados do que os outros algoritmos para as quatro proteínas analisadas. Porém, o algoritmo 2PG-NSGAII obteve um bom *score* para a proteína 1VII.

Foi também realizado um outro experimento de análise usando os dados gerados pelos preditores, agora usando uma métrica que não precisa de estrutura de referência: *Sort by front dominance*. Essa ferramenta comparou todos os modelos gerados pelos preditores e criou um *ranking* de acordo com a dominância que uma solução exercia sobre as demais,

usando como base a técnica de Fronteira de Pareto. Esse método apresentou resultados parecidos com os apresentados pelas métricas que utilizam uma estrutura de referência, mostrando que é possível encontrar as melhores soluções dentre um conjunto de soluções sem usar conhecimento anterior sobre a proteína. Foi observado que os resultados desse método apontaram soluções bem diferentes do que o *ranking* do RMSD. O algoritmo 2PG-NSGAI foi o que apresentou o maior número de soluções entre as 10 melhores. Com o uso do gráfico de Ramachandran, foi observado que a melhor solução apresentada pelo método *Sort* apresentou menos aminoácidos nas áreas não favorecidas do que a solução com menor RMSD, indicando que essas soluções estão mais estáveis energeticamente.

A partir dos resultados obtidos com os modelos gerados pelos preditores disponíveis no Koala, foi realizado um outro experimento usando a ferramenta *Sort by front dominance*, mas agora com dados gerados pelo CASP11. Foram utilizados 10 *targets* do CASP, escolhidos aleatoriamente, como entrada para o método. Como resultado, foi observado que o método conseguiu classificar as 10 melhores soluções, em alguns alvos, mais de 90% das soluções, indicando uma acurácia muito boa para o método. Entretanto, ainda é preciso avaliar qual a melhor combinação de objetivos seria melhor empregado para uma determinada análise.

Foi também realizado experimentos para demonstrar os recursos de *workflow* e *Page* disponíveis no Koala e como eles podem ser úteis para pesquisas biomédicas. Para o *workflow*, foi mostrado como seria a realização dos experimentos apresentados nesse trabalho usando esse recurso, indicando que todo o processo pode ser feito de forma automática e reproduzível. Para o *Page*, foi demonstrado como uma pesquisa completa realizada no Koala pode ser compartilhada publicamente no servidor, contendo todos os arquivos e parâmetros utilizados, permitindo a sua reprodução por qualquer usuário. Por fim, as considerações finais deste trabalho serão apresentadas no Capítulo 7.

Conclusões

Este trabalho propõe o desenvolvimento do Koala, um sistema para integração de métodos de predição e análises de estruturas de proteínas, o qual consiste de várias ferramentas disponíveis para realizar predições e também análises dos resultados gerados por qualquer preditor, utilizando arquivos PDBs.

O Koala possui 11 algoritmos evolutivos aplicados ao problema de PSP. Com isso, uma diversidade grande de resultados podem ser gerados para análise. Todos esses algoritmos que fazem parte de três *frameworks* para PSP: 2PG, ProtPredEDA e MEAMT, foram desenvolvidos pelo grupo de pesquisa LCR ICMC-USP e foram integrados nessa plataforma. Além disso, vários métodos de análises e comparação de estruturas foram desenvolvidos neste trabalho e disponibilizados por meio do Koala.

Para o desenvolvimento do Koala, foi utilizado como base o *framework* Galaxy, um sistema computacional para pesquisas biomédicas e bioinformática muito difundido e utilizado na comunidade científica. O Koala foi instalado e configurado para ser executado em um ambiente de *cloud computing*, atualmente utiliza a infraestrutura do *Cloud USP*. No total, o Koala possui uma infraestrutura computacional disponível com 96 núcleos, 150GB de RAM e 12TB de disco rígido.

O sistema também inclui novos módulos, que foram desenvolvidos para a execução dos algoritmos preditores e métodos de análises. Para o 2PG, foram desenvolvidos os módulos: 2PG *Build Conformation*, 2PG Mono, 2PG NSGAI e 2PG Monte Carlo (Seção 3.1); para o *framework* ProtPredEDA: RW, EDA, MCM, DE, GA e rBOA (Seção 3.2; para o MEAMT: MEAMT *Build population* e MEAMT (Seção 3.3). Além dos métodos de predição, foram desenvolvidos módulos para análises dos resultados, os quais foram divididos em duas categorias: *Solution Ranking* e *Solution Comparison*. O primeiro possui dois métodos desenvolvidos para realizar um *ranking* com as soluções geradas por vários preditores, que aplica o conceito de dominância de Pareto para encontrar as soluções mais promissoras (Seção 5.2). O segundo possui métricas para comparação de estruturas de proteínas com base em uma estrutura de referência (nativa): RMSD, GDT-TS e TM-Score (Seção 5.3).

Todos os módulos foram desenvolvidos com o objetivo de garantir a integração entre si e de criação de fluxos de trabalhos (*workflows*). Assim, é possível criar uma sequência de passos necessários para a realização de uma pesquisa, conectando várias ferramentas, de forma automática. No Koala já existem vários *workflows* desenvolvidos, com várias combinações de módulos e métodos de análises. Com isso, o usuário iniciante é capaz de realizar experimentos envolvendo vários métodos de predição e análise.

Para avaliar a funcionalidade dos módulos desenvolvidos, foram realizados experimentos de predição e análises no servidor Koala. Primeiramente, os algoritmos preditores foram executados para a predição de quatro proteínas, cada um com os seus parâmetros específicos. Os resultados gerados pelos preditores foram analisados, inicialmente, pela métrica RMSD, mostrando que os algoritmos do ProtPredEDA obtiveram os menores valores. Em seguida, os resultados foram analisados pelo método *Sort by front dominance*, mostrando que os algoritmos do 2PG apareceram nas primeiras posições do *ranking*, com diferença significativa de resultados em relação à métrica RMSD. Essa diferença foi confrontada utilizando o gráfico de Ramachandran, o qual mostrou para duas das quatro proteínas analisadas que as melhores soluções indicadas pelo método *Sort by front dominance* apresentaram uma melhor distribuição dos aminoácidos nas Figuras 73 e 76, enquanto nos outros dois casos ambas as soluções estavam com distribuições iguais dos aminoácidos.

O método *Sort by front dominance* também foi testado utilizando dados do CASP11. Foram escolhidos 10 alvos aleatórios com apenas um domínio para avaliar a acurácia do método em classificar as 10 melhores soluções, segundo o *ranking* do CASP, nas primeiras fronteiras. O *Sort by front dominance* obteve, em alguns alvos, uma taxa de 90% a 100% na classificação dessas soluções, o que é um resultado muito bom, já que não foi utilizado nenhum conhecimento *a priori* das proteínas alvo.

Os resultados obtidos pelos preditores apresentados nos experimentos desse trabalho podem ser ainda melhores, caso seja aplicada uma dinâmica molecular nas soluções mais promissoras encontradas. Além disso, outros aspectos podem ser avaliados, como o uso de mais objetivos no processo de predição para serem avaliados ao mesmo tempo, ao invés de utilizar apenas um ou dois.

Todos os experimentos realizados no Koala estão disponíveis publicamente para qualquer usuário acessar, visualizar e reproduzir em seu ambiente, com os mesmos dados ou utilizando os próprios dados. Isso garante a reprodutibilidade de qualquer análise realizada na plataforma, fazendo com que a pesquisa fique mais transparente e acessível. Para descrever os experimentos realizados nesse trabalho e disponibilizá-lo no servidor, foi criada uma página com todos os *histories* e *workflows* criados durante o projeto.

De forma geral, o objetivo principal deste trabalho, isto é, o desenvolvimento de um sistema com diversas ferramentas de predição e análises de estruturas de proteína, em um ambiente integrado, de fácil acesso e que garante a reprodutibilidade dos experimentos,

foi alcançado com o Koala. Assim, foram geradas contribuições para a área da Bioinformática, disponibilizando uma ferramenta web de fácil acesso e com uma interface gráfica amigável. Para a área de Computação Evolutiva, mostrou-se que é possível aplicar a dominância de Pareto no contexto do PSP e selecionar melhores soluções sem conhecimento *a priori* da proteína, e para área de Computação Distribuída, utilizando servidores HTTP para distribuir processos em diversas máquinas virtuais, diminuindo o gargalo de utilização de memória e CPU.

Por fim, espera-se que o sistema proposto para integração de ferramentas e análises seja utilizado pela comunidade científica para a realização de testes de predição de estruturas, como o que é feito no CASP e pelos participantes do projeto *WeFold*, utilizando diversos algoritmos evolutivos que têm sido disponibilizados. Além disso, espera-se que o Koala seja uma base para novas implementações e beneficie o desenvolvimento de novos trabalhos. A seguir, são apresentados os possíveis trabalhos futuros gerados nessa pesquisa.

7.1 Trabalhos Futuros

Este projeto permitiu também que novos trabalhos sejam realizados na área de bioinformática e computação, contribuindo ainda mais para problemas como o PSP. Os trabalhos futuros podem ser divididos nos itens a seguir:

- Melhoria da infraestrutura computacional: na infraestrutura atual do Koala, os recursos computacionais podem não ser suficiente caso a demanda de uso seja alta. Os algoritmos preditores exigem um consumo de CPU e memória muito grande, exigindo que novos recursos sejam alocados. Uma alternativa para essa questão é a instalação em um *cluster*, o qual possui recursos computacionais menos limitados e gerenciamento de processos mais eficiente;
- Estimativa de tempo de conclusão do *job*: em um ambiente computacional complexo que executa programas que exigem um alto poder computacional, poder prever o tempo que um *job* levará até a sua conclusão passa a ser um recurso interessante, uma vez que isso pode ser usado para informar o usuário e ser usado como balanceamento de carga do servidor. Existem pesquisas sendo realizadas, inclusive no ICMC-USP, com essa finalidade;
- Integração com bases de dados de proteínas: utilizar os conhecimentos depositados nos bancos de dados de proteínas pode melhorar as respostas apresentadas pelos algoritmos e métodos de análises. Isso é possível através de conexões usando *web services* e ontologias;

- Apresentação de resultado preliminar: durante a execução da predição, algumas soluções já são criadas. Essas soluções poderiam ser apresentadas ao usuário como uma forma de resultado preliminar, para que assim o usuário pudesse efetivamente acompanhar o processo de predição;
- Desenvolvimento de uma API do Koala: com uma API seria possível a integração do Koala com outros sistemas, expandindo as possibilidades desse programa;
- Desenvolvimento de ferramentas para dinâmica molecular: com o uso da dinâmica molecular, mesmo sendo um processo custoso, as predições podem ser melhoradas. Assim, seria importante fornecer ferramentas para o usuário executar mais esse passo nas melhores soluções apresentadas pelas métricas de análises já disponíveis. Ademais, isso possibilitaria a criação de mais um *workflow* no Koala;
- Documentação de uso e tutoriais: mesmo que o Koala possua uma interface amigável, o seu uso pode não ser tão intuitivo para todos os usuários. O desenvolvimento de uma documentação de uso mais detalhada, com tutoriais ensinando a utilizar os vários recursos do sistema, como *workflows*, e melhorias na usabilidade do sistema, possibilitará que os usuários não tenham tanta dificuldade em utilizar o *software*;
- Criação de ferramentas de análises dos arquivos de fronteiras: usando os resultados da métrica *Sort by front dominance*, análises mais detalhadas podem ser feitas para extrair mais informações desses resultados, auxiliando os usuários na tomada de decisão a respeito do que foi apresentado;
- Testes estatísticos relacionando as métricas de análises: criar opções avançadas nas ferramentas com o intuito de realizar mais análises e testes estatísticos correlacionando os resultados dos métodos disponíveis no sistema, melhorando a resposta final para o usuário;
- Disponibilização de novos preditores: oferecer novas ferramentas para outros preditores, de outros laboratórios do Brasil ou de outros países, a fim de aumentar a diversidade de métodos e resultados;
- Aplicar o *ranking* de dominância: uma limitação do método *Sort by front dominance* é a combinação de objetivos, uma restrição do método que pode influenciar o resultado. O *Ranking* de Dominância pode considerar várias combinações e assim apontar uma solução que tem uma boa colocação no *ranking* em várias combinações diferentes.

Referências

AFGAN, E. et al. Enabling cloud bursting for life sciences within Galaxy. **Concurrency and Computation: Practice and Experience**, 2015. v. 22, n. 6, p. 14, jun. 2015. ISSN 15320626. Disponível em: <<http://doi.wiley.com/10.1002/cpe.3536>>.

ALTSCHUL, S. F. et al. Basic local alignment search tool. **Journal of molecular biology**, 1990. v. 215, n. 3, p. 403–10, out. 1990. ISSN 0022-2836. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/2231712>>.

BAKER, J.; CHAN, F. The location of transition states: A comparison of Cartesian, Z-matrix, and natural internal coordinates. **Journal of Computational Chemistry**, 1996. John Wiley & Sons, Inc., v. 17, n. 7, p. 888–904, 1996. ISSN 1096-987X. Disponível em: <[http://dx.doi.org/10.1002/\(SICI\)1096-987X\(199605\)17:7<888::AID-JCC12>3.0.CO;2-7](http://dx.doi.org/10.1002/(SICI)1096-987X(199605)17:7<888::AID-JCC12>3.0.CO;2-7)>.

BARBOSA, S. D. J.; SILVA, B. S. D. **Interação Humano-Computador**. [S.l.]: Editora Campus - Elsevier, 2010.

BASKIN, D. et al. GeneSifter: Next Generation Data Management and Analysis for Next Generation Sequencing. **geospiza.com**, 2009. p. 1–9, 2009. Disponível em: <http://www.geospiza.com/Products/WhitePaper_06102009.pdf>.

BAXEVANIS, A. D.; OUELLETTE, B. F. F. **Bioinformatics: A Pratical Guide to the Analysis of Genes and Proteins**. Second edition. [S.l.]: John Wiley and Sons, 2001. ISBN 0471383902.

BLANKENBERG, D. et al. Galaxy: a web-based genome analysis tool for experimentalists. **Current protocols in molecular biology / edited by Frederick M. Ausubel ... [et al.]**, 2010. Chapter 19, n. January, p. Unit 19.10.1–21, jan. 2010. ISSN 1934-3647. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/20069535>>.

BLUNDELL, L. N. J. T. L. **Protein crystallography**. [S.l.]: London: Academic Press, 1976.

BONETTI, D. R. F. **Algoritmos de Estimiação de Distribuição para Predição Ab Initio de Estruturas de Proteínas**. Tese (Doutorado) — Instituto de Ciências Matemáticas e da Computação - Universidade de São Paulo, 2015.

BRANDEN, C.; TOOZE, J. **Introduction to Protein Structure**. [S.l.]: Garland Publishing, 1991.

- BRASIL, C. R. S. **Algoritmo Evolutivo de muitos objetivos para predição ab initio de estrutura de proteínas**. Tese (Doutorado) — Instituto de Ciências Matemáticas e da Computação - Universidade de São Paulo, 2012.
- BRASIL, C. R. S.; DELBEM, A. C. B.; BONETTI, D. R. F. Investigating relevant aspects of moeas for protein structures prediction. In: **Proceedings of the 13th annual conference on Genetic and evolutionary computation**. New York, NY, USA: ACM, 2011. (GECCO '11, v. 1), p. 705–712.
- BRASIL, C. R. S.; DELBEM, A. C. B.; SILVA, F. L. B. da. Multiobjective evolutionary algorithm with many tables for purely ab initio protein structure prediction. **Journal of computational chemistry**, 2013. v. 34, n. 20, p. 1719–34, jul. 2013. ISSN 1096-987X. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/23666867>>.
- BROOKS, F. P. No silver bullet: Essence and accidents of software engineering. In: **Computer**. [S.l.: s.n.], 1987. v. 20, n. 4, p. 1–15.
- BURDICK, D. B. et al. SEQADAPT: an adaptable system for the tracking, storage and analysis of high throughput sequencing experiments. **BMC bioinformatics**, 2010. v. 11, p. 377, jan 2010. ISSN 1471-2105. Disponível em: <<http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2916924>>.
- CAMPBELL, C. S. I. J.; CHEN., F. Gromacsimum: A web-based application for molecular dynamics simulations. In: . [S.l.: s.n.], 2014.
- CAMPBELL, P. N.; SMITH, A. D.; PETERS TIMOTHY, J. **Bioquímica Ilustrada - Bioquímica y biología molecular en la era posgenómica**. 5 edition. ed. [S.l.]: Elsevier, 2006.
- CASADEVALL, A.; FANG, F. C. Reproducible science. **Infection and immunity**, 2010. v. 78, n. 12, p. 4972–5, dec 2010. ISSN 1098-5522.
- CASP11. CRITICAL ASSESSMENT OF TECHNIQUES FOR PROTEIN STRUCTURE PREDICTION. 2014. n. 1, p. 291, 2014. Disponível em: <http://www.predictioncenter.org/casp11/doc/CASP11_Abstracts.pdf>.
- CONSTANTINE, L. What do users want? engineering usability into software. **Windows tech journal**, 1995. v. 4, n. 12, p. 30–39, 1995.
- CYBOK, D. A Grid workflow infrastructure. **Concurrency and Computation: Practice and Experience**, 2006. v. 18, n. 10, p. 1243–1254, 2006. ISSN 1532-0626. Disponível em: <<http://doi.wiley.com/10.1002/cpe.998>>.
- DAVIDSON, S.; OVERTON, C.; BUNEMAN, P. Challenges in Integrating Biological Data Sources. **Journal of Computational Biology**, 1995. v. 2, n. 4, p. 557–572, jan. 1995. ISSN 1066-5277. Disponível em: <<http://www.liebertonline.com/doi/abs/10.1089/cmb.1995.2.557>>.
- DEB, K. **Multi-Objective Optimization Using Evolutionary Algorithms**. [S.l.]: Wiley, 2001.
- DEB, K. et al. A fast and elitist multiobjective genetic algorithm: Nsga-ii. **Evolutionary Computation, IEEE Transactions on**, 2002. v. 6, n. 2, p. 182–197, Apr 2002. ISSN 1089-778X.

- DILL, K. a.; MACCALLUM, J. L. The protein-folding problem, 50 years on. **Science (New York, N.Y.)**, 2012. v. 338, n. 6110, p. 1042–6, nov. 2012. ISSN 1095-9203. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/23180855>>.
- DONAHUE, G. M.; WEINSCHENK, S.; NOWICKI, J. Usability is good business. **Compuware Corp.**, 1999. 1999. Disponível em: <<http://half-tide.net/UsabilityCost-BenefitPaper.pdf>>.
- DRAKE, N. How to catch a cloud. **Nature**, 2015. v. 522, n. 7554, p. 115–116, jun. 2015. ISSN 0028-0836. Disponível em: <<http://www.nature.com/doifinder/10.1038/522115a>>.
- EMBL, E. **Biomacromolecular structures: an introduction to the EBI resources**. [S.l.], maio 2015. Disponível em: <<http://www.ebi.ac.uk/training/online/courseogs/biomacromolecular-structures-introduction-ebi-reso>>.
- FACCIOLI, R.; BORTOT, L.; DELBEM, A. Application of molecular dynamics for generating models and ranked by dominance criteria. **CASP11 Meeting**, 2014. v. 1, p. 13–14, 2014.
- FACCIOLI, R. et al. Application of dominance criteria for ranking models. **CASP11 Meeting**, 2014. v. 1, p. 237–238, 2014.
- _____. A mono-objective evolutionary algorithm with gromacs for protein structure prediction in structural and energetic contexts. **IEEE World Congress on Computational Intelligence**, 2012. p. 1–8, 2012.
- FACCIOLI, R. A. **Implementação de um Framework de Computação Evolutiva Multi-Objetivo para Predição ab initio da Estrutura Terciária de Proteínas**. Tese (Doutorado) — Escola de Engenharia de São Carlos - Universidade de São Paulo, 2012.
- FACCIOLI, R. A.; BORTOT, L. O.; DELBEM, A. C. B. Multi-Objective Evolutionary Algorithm NSGA-II for Protein Structure Prediction using Structural and Energetic Properties. **International Journal of Natural Computing Research**, 2014. v. 4, n. 1, p. 1–12, 2014.
- FACCIOLI, R. A. et al. Protpred-gromacs: Evolutionary algorithm with gromacs for protein structure prediction. **BIOMAT 2011 International Symposium on Mathematical and Computational Biology**, 2011. p. 1–12, 2011.
- FELLER, S. E.; MACKERELL, A. D. An Improved Empirical Potential Energy Function for Molecular Simulations of Phospholipids. **The Journal of Physical Chemistry B**, 2000. v. 104, n. 31, p. 7510–7515, ago. 2000. ISSN 1520-6106. Disponível em: <<http://pubs.acs.org/doi/abs/10.1021/jp0007843>>.
- FIELDS, S. Proteomics in genomeland. **Science**, 2001. v. 291, n. 5507, p. 1221–1224, 2001. Disponível em: <<http://www.sciencemag.org/content/291/5507/1221.short>>.
- FIELDS, S.; JOHNSTON, M. A crisis in postgenomic nomenclature. **Science**, 2002. v. 296, n. 5568, p. 671–672, 2002. Disponível em: <<http://www.sciencemag.org/content/296/5568/671.short>>.
- GALAXY. **Galaxy Project Statistics**. [S.l.], Junho 2015. <https://wiki.galaxyproject.org/GalaxyProject/Statistics> - Acessado em: 14/06/2015.

GENOMATIX. **Genomatix Genome Analyzer**. [S.l.], Maio 2014. Disponível em: <<http://www.genomatix.de/solutions/genomatix-genome-analyzer.html>>.

GIANNOPOULOU, E. G.; ELEMENTO, O. An integrated ChIP-seq analysis platform with customizable workflows. **BMC bioinformatics**, 2011. BioMed Central Ltd, v. 12, n. 1, p. 277, jan. 2011. ISSN 1471-2105. Disponível em: <<http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3145611>>.

GOECKS, J.; NEKRUTENKO, A.; TAYLOR, J. Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. **Genome Biology**, 2010. p. 1–13, 2010.

HASTINGS, W. K. Monte carlo sampling methods using markov chains and their applications. **Biometrika**, 1970. v. 57, n. 1, p. 97–109, 1970. Disponível em: <<http://biomet.oxfordjournals.org/content/57/1/97.abstract>>.

HEMELRIJK, C. K.; WANTIA, J.; GYGAX, L. The construction of dominance order: comparing performance of five methods using an individual-based model. **Behaviour**, 2005. v. 142, n. 8, p. 1037–1058, ago. 2005. ISSN 0005-7959. Disponível em: <<http://booksandjournals.brillonline.com/content/10.1163/156853905774405290>>.

HESS, B. et al. Gromacs 4: Algorithms for highly efficient, load-balanced, and scalable molecular simulation. **Journal of Chemical Theory and Computation**, 2008. Stockholm Center for Biomembrane Research, Stockholm University, SE-10691 Stockholm, Sweden, v. 4, n. 3, p. 435–447, March 2008.

HIGHSOFT. Highcharts. 2009. Disponível em: <<http://www.highcharts.com/>>.

HOLLAND, J. H. **Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence**. [S.l.]: University of Michigan Press, 1975.

HRONKOVIC, J. **Algorithmics for Hard Problems: Introduction to Combinatorial Optimization, Randomization, Approximation, and Heuristics**. Berlin, Heidelberg: Springer-Verlag, 2010. ISBN 3642079091, 9783642079092.

HU, Z. et al. VisANT: data-integrating visual framework for biological networks and modules. **Nucleic acids research**, 2005. v. 33, n. Web Server issue, p. W352–7, jul. 2005. ISSN 1362-4962. Disponível em: <<http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1160192>>.

JOHNSON, M. et al. NCBI BLAST: a better web interface. **Nucleic acids research**, 2008. v. 36, n. Web Server issue, p. W5–9, jul. 2008. ISSN 1362-4962. Disponível em: <<http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2447716>>.

JUNIOR, W. E. F.; FRANCISCO, W. Proteínas: Hidrólise, precipitação e um tema para o ensino de química. **Química Nova Escola**, 2006. n. 24, p. 1–5, November 2006.

KARP, G. **Biologia Celular e Molecular - Conceitos e Experimentos**. Third edition. [S.l.]: John Wiley and Sons, 2002.

KHOURY, G. a. et al. WeFold: A competition for protein structure prediction. **Proteins**, 2014. n. August 2013, p. 1–19, mar. 2014. ISSN 1097-0134. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/24677212>>.

KIM, D. E.; CHIVIAN, D.; BAKER, D. Protein structure prediction and analysis using the Robetta server. **Nucleic acids research**, 2004. v. 32, n. Web Server issue, p. W526–31, jul. 2004. ISSN 1362-4962. Disponível em: <<http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=441606>>.

KLAUDA, J. B. et al. An ab initio study on the torsional surface of alkanes and its effect on molecular simulations of alkanes and a DPPC bilayer. **The journal of physical chemistry. B**, 2005. v. 109, n. 11, p. 5300–11, mar. 2005. ISSN 1520-6106. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/16863197>>.

KONG, L. et al. ABrowse—a customizable next-generation genome browser framework. **BMC bioinformatics**, 2012. BioMed Central Ltd, v. 13, n. 2, p. 8, jan. 2012. ISSN 1471-2105. Disponível em: <<http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3265404>>.

KUNCHEVA, L.; RODRIGUEZ, J. An experimental study on rotation forest ensembles. **Multiple Classifier Systems**, 2007. p. 459–468, 2007.

LIMA, T. W. D. et al. Evolutionary algorithm to ab initio protein structure prediction with hydrophobic interactions. **Evolutionary Computation, 2007. CEC 2007. IEEE Congress on**, 2007. p. 612–619, 2007.

LODISH, H. et al. **Biologia Celular e Molecular**. [S.l.]: Artmed, 2004.

LOVELL, S. C. et al. Structure validation by C alpha geometry: phi,psi and C beta deviation. **Proteins-Structure Function and Genetics**, 2003. v. 50, n. 3, p. 437–450, 2003. ISSN 1097-0134.

LUND, M.; TRULSSON, M.; PERSSON, B. Faunus: An object oriented framework for molecular simulation. **Source code for biology and medicine**, 2008. v. 3, n. 1, fev. 2008. ISSN 1751-0473.

MAIOROV, V. N.; CRIPPEN, G. M. Significance of Root-Mean-Square Deviation in Comparing Three-dimensional Structures of Globular Proteins. **Journal of Molecular Biology**, 1994. v. 235, n. 2, p. 625–634, 1994. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0022283684710175>>.

MANDEL, T. **The elements of user interface design**. [S.l.]: Wiley New York, 1997.

MAOJO, V. et al. Nanoinformatics and DNA-based computing: catalyzing nanomedicine. **Pediatric research**, 2010. v. 67, n. 5, p. 481–9, maio 2010. ISSN 1530-0447. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/20118825>>.

MAYER, G. et al. Controlled vocabularies and ontologies in proteomics: Overview, principles and practice. **Biochimica et Biophysica Acta (BBA) - Proteins and Proteomics**, 2014. Elsevier B.V., v. 1844, n. 1, p. 98–107, fev. 2014. ISSN 15709639. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S1570963913000800>>.

MELL, P.; GRANCE, T. The NIST definition of cloud computing. **National Institute of Standards and Technology**, 2011. p. 7, 2011. Disponível em: <<http://faculty.winthrop.edu/domanm/csci411/Handouts/NIST.pdf>>.

METROPOLIS, N. et al. Equation of State Calculations by Fast Computing Machines. **The Journal of Chemical Physics**, 1953. v. 21, n. 6, 1953.

- METROPOLIS, N.; ULAM, S. The Monte Carlo Method. **Journal of the American Statistical Association**, 1949. v. 44, n. 247, p. 335–341, 1949. Disponível em: <<http://amstat.tandfonline.com/doi/abs/10.1080/01621459.1949.10483310>>.
- MORAN, T. The command language grammar: A representation for the user interface of interactive computer systems. **International journal of man-machine studies**, 1981. p. 3–50, 1981. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0020737381800223>>.
- MOSIMANN, S.; MELESHKO, R.; JAMES, M. N. A critical assessment of comparative molecular modeling of tertiary structures of proteins. **Proteins**, 1995. v. 23, n. 3, p. 301–17, nov. 1995. ISSN 0887-3585. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/8710824>>.
- MURRAY, R. K. et al. **Bioquímica Ilustrada de Harper**. 29 edition. ed. [S.l.]: Artmed, 2014.
- NELSON, E. K. et al. LabKey Server: an open source platform for scientific data integration, analysis and collaboration. **BMC bioinformatics**, 2011. BioMed Central Ltd, v. 12, n. 1, p. 71, jan. 2011. ISSN 1471-2105. Disponível em: <<http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3062597>>.
- NIXON, C. et al. Nuclear magnetic resonance. **Anaesthesia**, 1986. Blackwell Publishing Ltd, v. 41, n. 2, p. 131–137, 1986. ISSN 1365-2044. Disponível em: <<http://dx.doi.org/10.1111/j.1365-2044.1986.tb13166.x>>.
- NORMAN, D. A. **The design of everyday things**. [S.l.]: London: MIT Press (UK edition), 1988.
- PEARSON, K. The random walk. **Nature**, 1905. v. 72, p. 294; 318; 342, 1905.
- PERKINELMER. **Geospiza**. [S.l.], Maio 2014. Disponível em: <<http://www.geospiza.com/>>.
- PREECE, J. et al. **Human-Computer Interaction**. Essex, UK, UK: Addison-Wesley Longman Ltd., 1994. ISBN 0201627698.
- PRESSMAN, R. S. **Software Engineering: a Practitioner's Approach**. 7. ed. [S.l.]: McGraw-Hill, NY, EUA, 2011. ISBN 0073375977.
- RAMACHANDRAN, G. N.; RAMAKRISHNAN, C.; SASISEKHARAN, V. Stereochemistry of polypeptide chain configurations. **Journal of molecular biology**, 1963. Academic Press Inc. (London) Ltd., v. 7, n. 1, p. 95–99, 1963. ISSN 00222836. Disponível em: <[http://dx.doi.org/10.1016/S0022-2836\(63\)80023-6](http://dx.doi.org/10.1016/S0022-2836(63)80023-6)>.
- RAO, S. T.; ROSSMANN, M. G. Comparison of super-secondary structures in proteins. **Journal of molecular biology**, 1973. v. 76, n. 2, p. 241–56, maio 1973. ISSN 0022-2836. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/4737475>>.
- RITTER, O. et al. Prototype Implementation of the Integrated Genomic Database. **Computers and Biomedical Research**, 1994. v. 27, n. 2, p. 97–115, abr. 1994. ISSN 00104809. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S0010480984710111>>.

RUSSEL, P. J. **iGenetics: A Molecular Approach**. 3. ed. [S.l.]: Benjamin Cummings, 2009.

SANDVE, G. K. et al. Ten simple rules for reproducible computational research. **PLoS computational biology**, 2013. v. 9, n. 10, p. 1–4, out. 2013. ISSN 1553-7358. Disponível em: <<http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3812051>>.

SCHATZ, M. C. The missing graphical user interface for genomics. **Genome Biology**, 2010. p. 1–3, 2010.

Schrödinger, LLC. The PyMOL molecular graphics system, version 1.7.1.3. August 2010.

SERVER, L. **LabKey Software Foundation**. [S.l.], Junho 2014. Disponível em: <<https://www.labkey.org/project/home/begin.view>>.

SETUBAL, J. M. J. **Introduction to Computational Molecular Biology**. 1. ed. [S.l.]: Hardcover, 1997.

SIEW, N. et al. MaxSub: an automated measure for the assessment of protein structure prediction quality. **Bioinformatics (Oxford, England)**, 2000. v. 16, n. 9, p. 776–785, 2000. ISSN 1367-4803.

SIMONS, K. T. et al. Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions. **Journal of molecular biology**, 1997. v. 268, n. 1, p. 209–25, abr. 1997. ISSN 0022-2836. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/9149153>>.

SPOEL, D. van der et al. **Gromacs User Manual version 4.0**. [S.l.], 2009.

STI, S. d. T. d. I. **Cloud USP**. [S.l.], Setembro 2010. Disponível em: <<http://www.sti.usp.br/?q=node/5370>>.

STORN, R.; PRICE, K. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. **Journal of Global Optimization**, 1997. Springer Netherlands, v. 11, p. 341–359, 1997. ISSN 0925-5001. 10.1023/A:1008202821328. Disponível em: <<http://dx.doi.org/10.1023/A:1008202821328>>.

TAYLOR, I. J. et al. **Workflows for e-Science: Scientific Workflows for Grids**. [S.l.]: Springer Publishing Company, Incorporated, 2014. ISBN 1849966192, 9781849966191.

TEAM, G. **Genomatix Genome Analyzer Quickstart Guide**. [S.l.], 2011. Disponível em: <<http://www.genomatix.de/download/flyer/docs/QuickstartGGA.pdf>>.

TECHONOMY. **Do You Get the Cloud, Or Do You Fake it?** [S.l.], 08 2012. Disponível em: <<http://techonomy.com/2012/08/do-you-get-the-cloud-or-do-you-fake-it>>.

TICONA, W. G. C. **Aplicação de algoritmos genéticos multiobjetivos para alimento de seqüências biológicas**. Dissertação (Mestrado) — Instituto de Ciências Matemáticas e de Computação - ICMC-USP, 2003.

WANG, J.; CIEPLAK, P.; KOLLMAN, P. a. How well does a restrained electrostatic potential (RESP) model perform in calculating conformational energies of organic and biological molecules? **Journal of Computational Chemistry**, 2000. v. 21, n. 12, p. 1049–1074, set. 2000. ISSN 0192-8651.

XU, D.; ZHANG, Y. Ab initio protein structure assembly using continuous structure fragments and optimized knowledge-based force field. **Proteins**, 2012. v. 80, n. 7, p. 1715–35, jul. 2012. ISSN 1097-0134. Disponível em: <<http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3370074>>.

XU, J.; ZHANG, Y. How significant is a protein structure similarity with TM-score = 0.5? **Bioinformatics (Oxford, England)**, 2010. v. 26, n. 7, p. 889–95, abr. 2010. ISSN 1367-4811. Disponível em: <<http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2913670>>.

ZEMLA, a. LGA: a method for finding 3D similarities in protein structures. **Nucleic Acids Research**, 2003. v. 31, n. 13, p. 3370–3374, jul. 2003. ISSN 1362-4962. Disponível em: <<http://nar.oxfordjournals.org/lookup/doi/10.1093/nar/gkg571>>.

ZEMLA, A. et al. Processing and analysis of CASP3 protein structure predictions. **Proteins: Structure, Function, and Genetics**, 1999. v. 37, n. S3, p. 22–29, 1999. ISSN 0887-3585.

ZHANG, Y. I-TASSER server for protein 3D structure prediction. **BMC bioinformatics**, 2008. v. 9, p. 40, jan. 2008. ISSN 1471-2105. Disponível em: <<http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2245901>>.

ZHANG, Y.; SKOLNICK, J. Scoring function for automated assessment of protein structure template quality. **Proteins**, 2004. v. 57, n. 4, p. 702–10, dez. 2004. ISSN 1097-0134. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/15476259>>.

ZHANG, Z.; CHEUNG, K.-H.; TOWNSEND, J. P. Bringing Web 2.0 to bioinformatics. **Briefings in bioinformatics**, 2009. v. 10, n. 1, p. 1–10, jan. 2009. ISSN 1477-4054. Disponível em: <<http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2638627>>.