
Algoritmos de estimação de distribuição para predição *ab initio* de estruturas de proteínas

Daniel Rodrigo Ferraz Bonetti

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

**Algoritmos de estimação de distribuição para predição
ab initio de estruturas de proteínas**

Daniel Rodrigo Ferraz Bonetti

Orientador: Prof. Dr. Alexandre Cláudio Botazzo Delbem

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, como parte dos requisitos para obtenção do título de Doutor em Ciências - Ciências de Computação e Matemática Computacional. *EXEMPLAR DE DEFESA.*

USP – São Carlos
Dezembro de 2014

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados fornecidos pelo(a) autor(a)

FBona Ferraz Bonetti, Daniel Rodrigo
Algoritmos de Estimação de Distribuição para
Predição Ab Initio de Estruturas de Proteínas /
Daniel Rodrigo Ferraz Bonetti; orientador Alexandre
Cláudio Botazzo Delbem. -- São Carlos, 2014.
228 p.

Tese (Doutorado - Programa de Pós-Graduação em
Ciências de Computação e Matemática Computacional) --
Instituto de Ciências Matemáticas e de Computação,
Universidade de São Paulo, 2014.

1. Algoritmos de Estimação de Distribuição. 2.
Predição de Estruturas de Proteínas. 3. Ab initio. 4.
Energia de van der Waals. 5. Modelo Probabilístico.
I. Botazzo Delbem, Alexandre Cláudio, orient. II.
Título.

Dedico este trabalho aos meus familiares.

Agradecimentos

A minha esposa, Vanessa, pelo incentivo e recorrente apoio durante todas as etapas do desenvolvimento deste trabalho, apoiando-me tanto no aspecto profissional quanto no aspecto efetivo, que certamente contribuíram para que este trabalho fosse desenvolvido de forma prazerosa e com qualidade.

Ao meu orientador Alexandre Delbem, pelo apoio, amizade e enorme dedicação à este trabalho, sempre disposto a discutir e ouvir ideias que, certamente contribuíram de forma muito significativa para meu aprendizado pessoal e profissional, e também para que fosse possível a realização de um trabalho com alta qualidade.

A todos os meus familiares, que sempre acreditaram nas minhas escolhas e me apoiaram da melhor forma possível, em especial meus pais e minha irmã.

Ao meu supervisor no exterior, Jochen Einbeck, pela enorme dedicação a este trabalho, pela amizade e por todo o conhecimento cultural transmitido. Também pela recepção tanto na Inglaterra quanto na Alemanha. E também por ter contribuído para que fosse possível a apresentação e divulgação deste trabalho na Ludwig-Maximilians-Universität München, juntamente com Andreas Bender.

Ao professor Dorival Leão, pela sabedoria, amizade e pelo recorrente acompanhamento do desenvolvimento das metodologias estatísticas utilizadas neste trabalho de doutorado.

Ao professor Richard Garratt, pela sabedoria e transmissão dos conhecimentos em estruturas de proteínas, que foram fundamentais para o melhor entendimento do problema.

Ao professor Horácio Perez-Sanches, pela iniciativa e otimismo para a realização do trabalho conjunto e também pela recepção durante o congresso IWBBIO. A velha amiga Camila, pelos anos de amizade e também por ter oferecido sua casa durante a passagem por Madrid, durante a realização do congresso IWBBIO.

Aos familiares da minha esposa, em especial o Berto e a Ângela, pela amizade e auxílio em todos os sentidos.

À minha cunhada preferida, Larissa, pela amizade e pelos momentos “Hey Jude”, Paralamas e muitos outros momentos divertidos que passamos juntos. Também pelo apoio profissional e pessoal. E também ao cunhado Paulo, pelo apoio e pelos momentos de distração que passamos juntos.

Aos meus grandes amigos de Piracicaba e suas respectivas esposas, Libardi, Danilo, Bruno e Adriano, que me forneceram momentos de alegria em suas companhias.

Aos amigos de São Carlos, Rubens, Cabeça, Hiro, Rafael, Coletta, Xinêis e Marcy, pelos muitos momentos divertidos que passamos juntos. Em especial o Hiro e o Cabeça por todo o conhecimento estatístico transmitido. Ao pessoal do LCR e ao Felipe, pela manutenção no cluster do LCR.

A todos que tornam possível o estágio no exterior, em especial a Melissa Schuessler, do escritório de relações internacionais da Durham University. E também à Sue, por ter oferecido um local para morarmos durante o estágio no exterior. À Durham University, ao departamento de Ciências Matemáticas e ao Ustinov College, pela acolhida durante o estágio no exterior

Às amizades feitas durante o estágio no exterior, Ankit, James, Gabriel, Dan, Gina, professores Mathias e Paul. Em especial à Thomai Tsiftsi, pelos conhecimentos estatísticos, paciência e amizade. E também à Nathália, Carla e Bruna, pela grande amizade.

Ao departamento de Sistemas de Computação e ao programa de pós-graduação do ICMC, pela oportunidade de participar deste doutorado. Aos funcionários do ICMC, pelo empenho em suas funções.

À professora Graça Pimentel e a Cláudia, pela acolhida durante a transição entre o mestrado e doutorado e também aos colegas do laboratório Intermídia.

À comunidade e aos colaboradores do StackExchange.com, pelas discussões de assuntos mais específico.

Ao CNPq, pelo fundamental suporte financeiro durante os primeiros meses de pesquisa.

E um agradecimento especial à FAPESP, pelo suporte financeiro investido durante o desenvolvimento deste trabalho bem como o suporte financeiro oferecido para a realização do estágio no exterior.

Resumo

As proteínas são moléculas presentes nos seres vivos e essenciais para a vida. Para entender a função de uma proteína é preciso conhecer sua estrutura tridimensional. No entanto, encontrar a estrutura da proteína pode ser um processo caro e demorado, exigindo profissionais altamente qualificados. Neste sentido, métodos computacionais têm sido investigados buscando predizer a estrutura de uma proteína a partir de uma sequência de aminoácidos. Em geral, tais métodos computacionais utilizam conhecimentos de estruturas de proteínas já determinadas por métodos experimentais, para tentar predizer proteínas com estrutura desconhecida. Embora métodos computacionais como, por exemplo, o Rosetta, I-Tasser e Quark tenham apresentado sucesso em suas previsões, são apenas capazes de produzir estruturas significativamente semelhantes às já determinadas experimentalmente. Com isso, por utilizarem conhecimento *a priori* de outras estruturas pode haver certa tendência em suas previsões. Buscando elaborar um algoritmo eficiente para Predição de Estruturas de Proteínas livre de tendência foi desenvolvido um Algoritmo de Estimação de Distribuição (EDA) específico para esse problema, com modelagens *full-atom* e *ab initio*. O fato do algoritmo proposto ser *ab initio* é mais interessante para aplicação envolvendo proteínas com baixa similaridade, com relação às estruturas já conhecidas. Três tipos de modelos probabilísticos foram: univariado, bivariado e hierárquico. O univariado trata o aspecto de multi-modalidade de uma variável, o bivariado trata os ângulos diedrais (ϕ, ψ) de um mesmo aminoácido como variáveis correlacionadas. O hierárquico divide o problema em subproblemas e tenta tratá-los separadamente. Os resultados desta pesquisa mostraram que é possível obter melhores resultados quando considerado a relação bivariada (ϕ, ψ). O hierárquico também mostrou melhorias nos resultados obtidos, principalmente para proteínas com mais de 50 resíduos. Além disso, foi realizada uma comparação com algumas heurísticas da literatura, como: Busca Aleatória, Monte Carlo, Algoritmo Genético e Evolução Diferencial. Os resultados mostraram que mesmo uma metaheurística pouco eficiente, como a Busca Aleatória, pode encontrar a solução correta, porém utilizando muito conhecimento *a priori* (previsão que pode ser tendenciosa). Por outro lado, o algoritmo proposto neste trabalho foi capaz de obter a estrutura correta sem utilizar conhecimento *a priori*, caracterizando uma previsão puramente *ab initio* (livre de tendência).

Abstract

Proteins are molecules present in the living organism and essential for their life. To understand the function of a protein, its 3D structure should be known. However, to find the protein structure is an expensive and a time-consuming task, requiring highly skilled professionals. Aiming to overcome such a limitation, computational methods for Protein Structure Prediction (PSP) has been investigated, in order to predict the protein structure from its amino acid sequence. Most of computational method require knowledge from already determined structures from experimental methods in order to predict an unknown protein. Although computational methods such as Rosetta, I-Tasser and Quark have showed success in their predictions, they are only capable to predict quite similar structures to already known proteins obtained experimentally. The use of such a prior knowledge in the predictions of Rosetta, I-Tasser and Quark may lead to biased predictions. In order to develop a computational algorithm for PSP free of bias, we developed an Estimation of Distribution Algorithm applied to PSP with full-atom and *ab initio* model. A computational algorithm with *ab initio* model is mainly interesting when dealing with proteins with low similarity with the known proteins. In this work, we developed an Estimation of Distribution Algorithm with three probabilistic models: univariate, bivariate and hierarchical. The univariate deals with multi-modality of the distribution of the data of a single variable. The bivariate treats the dihedral angles (ϕ, ψ) within an amino acid as correlated variables. The hierarchical splits the original problem into subproblems and attempt to treat these problems in a separated manner. The experiments show that, indeed, it is possible to achieve better results when modeling the correlation (ϕ, ψ). The hierarchical also showed that is possible to improve the quality of results, mainly for proteins above 50 residues. Besides, we compared our proposed techniques among other metaheuristics from literatures such as: Random Walk, Monte Carlo, Genetic Algorithm and Differential Evolution. The results show that even a less efficient metaheuristic such as Random Walk managed to find the correct structure, however using many prior knowledge (prediction that may be biased). On the other hand, our proposed EDA for PSP was able to find the correct structure with no prior knowledge at all, so we can call this prediction as pure *ab initio* (biased-free).

Lista de Figuras

1.1	Tamanhos de proteínas e tecnologias (CRX e RMN)	2
1.2	Gráfico do crescimento de sequências e estruturas de proteínas a partir de 1990	3
2.1	Exemplo de um aminoácido e uma ligação peptídica	10
2.2	Níveis de estruturas de proteínas	12
2.3	Quantidade de estruturas de proteínas determinadas	15
2.4	Determinação de estruturas de proteínas com a CRX	16
2.5	Determinação de estruturas de proteínas com a RMN	17
2.6	Identificação dos ângulos ϕ e ψ	24
2.7	Identificação dos ângulos quirais das cadeias laterais	25
2.8	Diagrama de Ramachandran	26
2.9	Energia potencial de Lennard-Jones modificada para PSP	30
2.10	Representação da SASA e do soluto	31
2.11	Cálculo da superfície da molécula	32
3.1	Etapas de um EA típico	36
3.2	Métodos de substituição da população pelos filhos	37
3.3	Método de MC aplicado para aproximar π	40
3.4	Representação de uma população e geração de novo indivíduo para o GA	42
3.5	Esquema do funcionamento de um EDA	45
3.6	Exemplos de EDAs para o problema <i>OneMax</i>	47
3.7	Distribuição dos valores da variável x_1	54
4.1	Exemplo do processo evolutivo para a RW	61
4.2	Geração de nova conformação utilizando o método de MC para PSP	62
4.3	Exemplo de recombinação e mutação do GA para PSP	64
4.4	Comparação entre a densidade real e estimada utilizando modelo probabilístico unimodal	67
4.5	Comparação entre o GA, DE e UMDA _c para 200 mil avaliações no problema de PSP	68
4.6	Métodos de amostragem univariados	70
4.7	Estimação e amostragem com o KDE2D	73
4.8	Estimação e amostragem com o FGM	77
4.9	Agrupamento hierárquico dividido em dois grupos	79
4.10	Exemplo da representação de união de dois subproblemas	79
4.11	Exemplo de sobreposição de subproblemas	80

4.12	Esquema de funcionamento do EDA hierárquico	81
4.13	Ilustração entre o ProtPred-EDA e MURCIA	83
4.14	Métodos de otimização propostos para PSP	85
5.1	Proteínas nativas utilizadas nos experimentos	90
5.2	Calibração dos parâmetros	92
5.3	Calibração do número de misturas para a FGMO	93
5.4	EDA para a proteína 1R8T	99
5.5	EDA para a proteína 2LLR	100
5.6	EDA para a proteína 1A11	101
5.7	EDA para a proteína 2LX0	102
5.8	EDA para a proteína 2LVG	103
5.9	EDA para a proteína 2KK7	104
5.10	EDA para a proteína 2X43	105
5.11	EDA para a proteína 2A3D	106
5.12	EDA para a proteína 2ZGG	107
5.13	Síntese dos resultados com base nas 10% das melhores soluções, minimizando a energia de van der Waals.	108
5.14	Estruturas de proteína preditas	109
5.15	Dez porcento melhores soluções obtidas para a energia de van der Waals por cada método	112
5.16	Relação entre a quantidade de ligações de hidrogênio e o RMSD.	113
5.17	Relação entre a porcentagem de estruturas secundárias e o RMSD.	114
5.18	Energia de van der Waals pelo RMSD e pelo tamanho da proteína.	115
5.19	Energia de van der Waals pelo RMSD e pelo tamanho da proteína.	117
5.20	Conformação de proteína com melhor RMSD considerando o efeito da energia de solvatação	118
5.21	Estruturas de proteína preditas	119
5.22	Experimentos comparando previsões com e sem o uso do ADB para gerar a população inicial	120
5.23	Comparação das energias de van der Waals, eletrostática, de solvatação e de ligações de hidrogênio para a proteína com menor energia de van der Waals predita . .	121
5.24	Calibração do parâmetro α para os EDAs hierárquicos	124
5.25	Comparação dos EDAs com os EDAs hierárquicos para a proteína 2LVG	125
5.26	Comparação da energia de van der Waals entre os EDAs propostos e suas extensões hierárquicas com $m = 2$ e $\alpha = 2$	126
5.27	Comparação do RMSD entre os EDAs propostos e suas extensões hierárquicas com $m = 2$ e $\alpha = 2$	127
5.28	Síntese das 10% das melhores soluções obtidas pelo critério energia de van der Waals para os EDAs propostos e suas extensões hierárquicas com $m = 2$	129
5.29	Calibração do tamanho da população para os métodos de referência	131
5.30	Calibração dos parâmetros	132
5.31	ProtPred-EDA para a proteína 1R8T	135
5.32	ProtPred-EDA para a proteína 2LLR	136
5.33	ProtPred-EDA para a proteína 1A11	137
5.34	ProtPred-EDA para a proteína 2LX0	138
5.35	ProtPred-EDA para a proteína 2LVG	139
5.36	ProtPred-EDA para a proteína 2KK7	140

5.37	ProtPred-EDA para a proteína 2X43	141
5.38	ProtPred-EDA para a proteína 2A3D	142
5.39	ProtPred-EDA para a proteína 2ZGG	143
5.40	Comparação entre a quantidade de pontes de hidrogênio de cada conformação com o RMSD	144
5.41	Comparação entre a porcentagem de estruturas secundárias de cada conformação com o RMSD	145
5.42	Tamanho da proteína pelo hipervolume	146
5.43	Síntese das 10% melhores soluções em relação a energia de van der Waals obtidas pelos métodos de referência e pelos FGMO e hFGMO	151
5.44	Estruturas das melhores proteínas preditas entre os métodos de referência e os EDAs propostos	152
5.45	Experimento com/sem ADB para gerar a população inicial, mostrando a diferença entre energia de van der Waals	153
5.46	Experimento com/sem ADB para gerar a população inicial, mostrando a diferença entre RMSD	154
5.47	Nível de conhecimento <i>a priori</i> necessário para predizer corretamente a estrutura da proteína 1A11 de acordo com a metaheurística utilizada	155
A.1	Grafo da interação de variáveis encontrado pelo rBOA na última geração para a proteína 1A11	172
B.1	Quantidade de memória necessária para o ProtPred-EDA	174
B.2	Versões do ProtPred	175
B.3	Orientações L e D dos aminoácidos	177
C.1	Energia de solvatação com diferentes pesos, mostrando a distribuição do RMSD de três proteínas	181
C.2	Energia de solvatação com diferentes pesos, mostrando a distribuição do RMSD de três proteínas	182
C.3	Energia de solvatação com diferentes pesos, mostrando a distribuição do RMSD de três proteínas	183
C.4	Energia de solvatação com diferentes pesos, mostrando a distribuição do RMSD de três proteínas	184
C.5	Comparação do tempo de execução da energia de solvatação em CPU e em GPU .	186
C.6	Energia de solvatação com diferentes pesos, mostrando a distribuição do RMSD de três proteínas	188
C.7	Estrutura da proteína 1A11 predita utilizando energia de van der Waals e solvatação	188
C.8	Estrutura da proteína 1MZT predita utilizando energia de van der Waals e solvatação	189
C.9	Estrutura da proteína 4L5M predita utilizando energia de van der Waals e solvatação	189
D.1	Tela inicial do Servidor Galaxy do ICMC-USP	194
D.2	Exibição inicial da ferramenta ProtPred-EDA	195
D.3	Exemplo de ajuste de parâmetros para a proteína 1A11	195

Lista de Tabelas

2.1	Tabela dos 20 aminoácidos	13
4.1	Combinação de subproblemas	81
5.1	Proteínas utilizadas nos experimentos	89
5.2	Parâmetros utilizados para calibração	91
5.3	Tabela dos parâmetros para as oito melhores execuções para os três modelos probabilísticos com a proteína 2LVG.	93
5.4	Soluções da Fronteira de Pareto obtidas pelos critérios energia de van der Waals e RMSD	110
5.5	Tabela de parâmetros utilizados na calibração dos algoritmos de referência	130
5.6	Parâmetros das oito melhores execuções (baseando-se na energia de van der Waals) da RW, MC, GA e DE para a proteína 2LVG	133
5.7	Tabela da Fronteira de Pareto das metaheurísticas avaliadas	146
5.10	Hipervolume considerando os critérios energia de van der Waals e RMSD, para as nove proteínas	156
5.8	Melhores e piores soluções para cada proteína considerando os fatores energia de van der Waals, RMSD e tempo de execução	161
5.9	Continuação da Tabela 5.8	162
C.1	Pesos escolhidos por proteína	180
C.2	Proteínas utilizadas nos experimentos	185
E.1	P-valores para a proteína 1R8T da comparação dois-a-dois do teste de Wilcoxon para os valores de energia de van der Waals utilizando o teste de Wilcoxon	198
E.2	P-valores para a proteína 2LLR da comparação dois-a-dois do teste de Wilcoxon para os valores de energia de van der Waals utilizando o teste de Wilcoxon	198
E.3	P-valores para a proteína 1A11 da comparação dois-a-dois do teste de Wilcoxon para os valores de energia de van der Waals utilizando o teste de Wilcoxon	198
E.4	P-valores para a proteína 2LX0 da comparação dois-a-dois do teste de Wilcoxon para os valores de energia de van der Waals utilizando o teste de Wilcoxon	199
E.5	P-valores para a proteína 2LVG da comparação dois-a-dois do teste de Wilcoxon para os valores de energia de van der Waals utilizando o teste de Wilcoxon	199
E.6	P-valores para a proteína 2KK7 da comparação dois-a-dois do teste de Wilcoxon para os valores de energia de van der Waals utilizando o teste de Wilcoxon	199

E.7	P-valores para a proteína 2X43 da comparação dois-a-dois do teste de Wilcoxon para os valores de energia de van der Waals utilizando o teste de Wilcoxon	200
E.8	P-valores para a proteína 2A3D da comparação dois-a-dois do teste de Wilcoxon para os valores de energia de van der Waals utilizando o teste de Wilcoxon	200
E.9	P-valores para a proteína 2ZGG da comparação dois-a-dois do teste de Wilcoxon para os valores de energia de van der Waals utilizando o teste de Wilcoxon	200
E.10	P-valores para a proteína 1R8T da comparação dois-a-dois do teste de Wilcoxon para os valores de RMSD utilizando o teste de Wilcoxon	201
E.11	P-valores para a proteína 2LLR da comparação dois-a-dois do teste de Wilcoxon para os valores de RMSD utilizando o teste de Wilcoxon	201
E.12	P-valores para a proteína 1A11 da comparação dois-a-dois do teste de Wilcoxon para os valores de RMSD utilizando o teste de Wilcoxon	201
E.13	P-valores para a proteína 2LX0 da comparação dois-a-dois do teste de Wilcoxon para os valores de RMSD utilizando o teste de Wilcoxon	202
E.14	P-valores para a proteína 2LVG da comparação dois-a-dois do teste de Wilcoxon para os valores de RMSD utilizando o teste de Wilcoxon	202
E.15	P-valores para a proteína 2KK7 da comparação dois-a-dois do teste de Wilcoxon para os valores de RMSD utilizando o teste de Wilcoxon	202
E.16	P-valores para a proteína 2X43 da comparação dois-a-dois do teste de Wilcoxon para os valores de RMSD utilizando o teste de Wilcoxon	203
E.17	P-valores para a proteína 2A3D da comparação dois-a-dois do teste de Wilcoxon para os valores de RMSD utilizando o teste de Wilcoxon	203
E.18	P-valores para a proteína 2ZGG da comparação dois-a-dois do teste de Wilcoxon para os valores de RMSD utilizando o teste de Wilcoxon	203
E.19	P-valores para a proteína 1R8T da comparação dois-a-dois do teste de Wilcoxon para os valores de tempo de execuçãp utilizando o teste de Wilcoxon	204
E.20	P-valores para a proteína 2LLR da comparação dois-a-dois do teste de Wilcoxon para os valores de tempo de execuçãp utilizando o teste de Wilcoxon	204
E.21	P-valores para a proteína 1A11 da comparação dois-a-dois do teste de Wilcoxon para os valores de tempo de execuçãp utilizando o teste de Wilcoxon	204
E.22	P-valores para a proteína 2LX0 da comparação dois-a-dois do teste de Wilcoxon para os valores de tempo de execuçãp utilizando o teste de Wilcoxon	205
E.23	P-valores para a proteína 2LVG da comparação dois-a-dois do teste de Wilcoxon para os valores de tempo de execuçãp utilizando o teste de Wilcoxon	205
E.24	P-valores para a proteína 2KK7 da comparação dois-a-dois do teste de Wilcoxon para os valores de tempo de execuçãp utilizando o teste de Wilcoxon	205
E.25	P-valores para a proteína 2X43 da comparação dois-a-dois do teste de Wilcoxon para os valores de tempo de execuçãp utilizando o teste de Wilcoxon	206
E.26	P-valores para a proteína 2A3D da comparação dois-a-dois do teste de Wilcoxon para os valores de tempo de execuçãp utilizando o teste de Wilcoxon	206
E.27	P-valores para a proteína 2ZGG da comparação dois-a-dois do teste de Wilcoxon para os valores de tempo de execuçãp utilizando o teste de Wilcoxon	206
E.28	P-valores do teste de Wilcoxon para o EDA hierárquico com o modelo probabilístico UNI, avaliando três valores para α , considerando o aspecto energia de van der Waals para a proteína 2LVG.	207
E.29	P-valores do teste de Wilcoxon para o EDA hierárquico com o modelo probabilístico KDE2D, avaliando três valores para α , considerando o aspecto energia de van der Waals para a proteína 2LVG.	207

E.30 P-valores do teste de Wilcoxon para o EDA hierárquico com o modelo probabilístico FGM, avaliando três valores para α , considerando o aspecto energia de van der Waals para a proteína 2LVG.	207
E.31 P-valores do teste de Wilcoxon entre os EDAs sem hierarquia e com hierarquia em que $m = 2, 3$, avaliando a energia de van der Waals, para a proteína 2LVG. Os valores acima de 0,05 estão destacados em negritos	207
E.32 P-valores do teste de Wilcoxon entre os EDAs sem hierarquia e com hierarquia em que $m = 2, 3$, avaliando o RMSD, para a proteína 2LVG. Os valores acima de 0,05 estão destacados em negritos	208
E.33 P-valores do teste de Wilcoxon entre os EDAs sem hierarquia e com hierarquia em que $m = 2, 3$, avaliando o tempo de execução, para a proteína 2LVG. Os valores acima de 0,05 estão destacados em negritos	208
E.34 P-valores do teste de Wilcoxon para os EDAs UNIO, KDEO e FGMO, com e sem ADB, para nove proteínas, considerando o aspecto energia de van der Waals	209
E.35 P-valores do teste de Wilcoxon para os EDAs UNIO, KDEO e FGMO, com e sem ADB, para nove proteínas, considerando o aspecto RMSD.	209
E.36 P-valores do teste de Wilcoxon para os métodos RW, MC, GA, DE, FGMO e hFGMO com e sem ADB, para nove proteínas, considerando o aspecto energia de van der Waals.	209
E.37 P-valores do teste de Wilcoxon para os métodos RW, MC, GA, DE, FGMO e hFGMO com e sem ADB, para nove proteínas, considerando o aspecto RMSD.	210
E.38 P-valores do teste de Wilcoxon, com e sem energia de solvatação, para nove proteínas, considerando o aspecto RMSD.	210

Lista de Abreviaturas e Siglas

ADB	<i>Angle DataBase</i>
ASP	<i>Atomic Solvation Parameters</i>
BB	<i>Building Block</i>
BIC	<i>Bayesian Informatin Criteria</i>
BMDA	<i>Bivariate Marginal Distribution Algorithm</i>
BOA	<i>Bayesian Optimiztion Algorithm</i>
CASP	<i>Critical Assessment of protein Structure Prediction</i>
CC	<i>Complexidade Combinada</i>
CGA	<i>compact Genetic Algorithm</i>
CORN	Carbono-Oxigênio-Radical-Nitrogênio
CRX	Cristalografia de Raio-X
DE	<i>Differential Evolution</i>
DM	Dinâmica Molecular
EA	<i>Evolutionary Algorithm</i>
EcGA	<i>Extended compact Genetic Algorithm</i>
EDA	<i>Estimation of Distribution Algorithm</i>
EGNA	<i>Estimation of Gaussian Networks Algorithm</i>
EM	<i>Expectation-Maximization</i>
FDP	Função Densidade Probabilidade
FGM	Modelo probabilístico bivariado que utiliza <i>Finite Gaussian Mixture</i>
FGMO	<i>Finite Gaussian Mixture model-based Optimization</i>
GA	<i>Genetic Algorithm</i>
GPU	<i>Graphic Processing Unit</i>
hBOA	<i>hierarchical Bayesian Optimization Algorithm</i>
hFGMO	<i>hierarchical Finite Gaussian Mixture model-based Optimization</i>
hKDEO	<i>hierarchical Kernel Density Estimation model-based Optimization</i>
HM	<i>Hierarchy Model</i>
hUNIO	<i>hierarchical Univariate model-based Optimization</i>

KDE	<i>Kernel Density Estimation</i>
KDE2D	<i>Modelo probabilístico bivariado com KDE</i>
KDEO	<i>Kernel Density Estimation model-based Optimization</i>
LCR	Laboratório de Computação Reconfigurável
LNCC	Laboratório Nacional de Computação Científica
MC	Monte Carlo
MIC	Microscopia Eletrônica
MM	Mecânica Molecular
MPI	<i>Message Passing Interface</i>
MPM	Modelo Produto Marginal
MQ	Mecânica Quântica
NFS	<i>Network File System</i>
PBIL	<i>Population-Based Incremental Learning</i>
PDB	<i>Protein Data Bank</i>
PSP	<i>Protein Structure Prediction</i>
RB	<i>Bayesian Network</i>
rBOA	<i>real-value Bayesian Optimization Algorithm</i>
rEcGA	<i>real-coded Extended Compact Genetic Algorithm</i>
RMN	Ressonância Magnética Nuclear
RMSD	<i>Root mean square deviation</i>
RTR	<i>Restricted Tournament Replacement</i>
RW	<i>Random Walk</i>
SASA	<i>Solvent Accessible Surface Area</i>
SHCLVND	<i>Stochastic Hill Climbing with Learning by Vectors of Normal Distribution</i>
SoD	<i>Split-on-Demand</i>
UMDA	<i>Univariate Marginal Distribution Algorithm</i>
UNI	Modelo probabilístico univariado proposto
UNIO	<i>Univariate model-based Optimization</i>

List of Symbols

α	Número de variáveis sobrepostas no EDA hierárquico
$\hat{\mu}_k$	Vetor da estimativa das médias dos componentes de mistura (FGM)
$\hat{\pi}$	Vetor da estimativa dos pesos dos componentes de mistura (FGM)
$\hat{\Sigma}$	Estimativa da matriz de covariância dos componentes de mistura (FGM)
$\hat{\theta}$	Modelo probabilístico
τ	Truncamento da população pelos τ melhores indivíduos
a	Quantidade de átomos
A	Matriz dos C_α s de uma proteína
$A_{cubeside}$	Superfície do cubo (energia de solvatação)
$A_{intercept}$	Área do cubo interceptada (energia de solvatação)
A_{vdw}	Constante do fator de repulsão utilizada na energia de van der Waals
a_c	Número de C_α s em certa proteína
B	Ponto inicial (utilizado na RW)
b	Tamanho do bloco
B_{vdw}	Constante do fator de atração utilizada na energia de van der Waals
c	Cardinalidade das variáveis de um problema
C	Constante utilizada caso $q_{i,j} \leq 0,8$
c_r	Taxa de recombinação
c_s	Vetor que armazena a soma acumulada dos pesos de componentes de misturas
d	Dimensões do problema (número de variáveis)
D	Dendrograma gerado pelo agrupamento hierárquico
$d_{E_{ij}}$	Distância Euclidiana entre dois átomos
E_{chg}	Energia eletrostática
E_{comp}	Energia de comprimento de ligação
E_{hbond}	Energia de pontes de hidrogênio
E_{imp}	Energia imprória
E_{lig}	Energia de ligação
E_{sol}	Energia de solvatação
E_{tor}	Energia de torção

E_{total}	Energia potencial da proteína
E_{ub}	Energia Urey-Bradley
E_{vdw}	Energia de van der Waals
f	Número de filhos a serem gerados
F	Amplitude da diferença da recombinação no DE
f_{mc}	<i>Fitness</i> de um indivíduo no método de MC
f_{sasa}	Fator de cada tipo de átomo (energia de solvatação)
g	Metros percorridos pela RW
h	Janela utilizada na distribuição kernel
h_{mc}	Contador de pontos dentro do círculo (exemplo de MC)
K	Quantidade de componentes de mistura utilizado no FGM
l	Lado do quadrado (exemplo de MC)
L	População mesclada na forma $L = U \cup W$
M	População mesclada na forma $M = P \cup O$
m_f	Fator de mutação
m_r	Taxa de mutação
n	Número de indivíduos (tamanho da população)
n_p	Pontos gerados no exemplo de MC
N_c	Número de cubos que interceptam a superfície da molécula (energia de solvatação)
n_x	Número de pontos para criação do KDE
O	Conjunto dos indivíduos Filhos
o	Novo indivíduo filho gerado
O_j^k	$k = 1 \dots f$ e $j = 1 \dots d$
P	População
P_j^i	$i = 1 \dots n$ e $j = 1 \dots d$
Pr_j	Probabilidade de S_j ser igual a 1
q	Distância relativa
r	Número de resíduos
r_{rw}	Distância entre o ponto B até o ponto final na RW
R_i	Raio de van der Waals para o átomo i
r_k	Utilizado para calcular a janela h
s	Quantidade de indivíduos selecionados
S	Conjunto dos indivíduos selecionados
S_j^l	$l = 1 \dots s$ e $j = 1 \dots d$
s_{mc}	Tamanho da mutação do MC
$SASA$	Área interceptada média (energia de solvatação)
t	Tamanho do torneio
u	Número aleatório uniforme $[0; 1]$
U	Utilizado para indicar que os indivíduos desta população são da população inicial
V	Conjunto dos Subproblemas
v	Número de indivíduos da RTR

- V_k^p p é identificação do subproblema e k contem os valores do subproblema p
- W Utilizado para indicar que os indivíduos desta população são gerados aleatoriamente
- w Matriz de probabilidades (FGM)
- w_1 Peso da energia de van der Waals
- w_2 Peso da energia de solvatação
- x Variáveis de um problema
- x_i Variáveis de um problema de otimização

Sumário

Resumo	i
Abstract	iii
Lista de Abreviaturas e Siglas	xiii
Lista de Símbolos	xv
1 Introdução	1
1.1 Objetivos	7
1.2 Relevância da pesquisa	8
1.3 Organização	8
2 Predição de Estruturas de Proteínas	9
2.1 Estrutura de Proteínas	9
2.2 O Problema da Predição da Estrutura de Proteínas	11
2.3 Técnicas existentes de PSP	14
2.3.1 Técnicas experimentais	14
2.3.2 Técnicas <i>in silico</i>	18
2.4 <i>Ab initio</i>	21
2.4.1 Modelo <i>lattice</i>	23
2.4.2 Modelo <i>off-lattice</i>	23
2.4.3 Modelo <i>full-atom</i>	24
2.5 Energia potencial da proteína	27
2.5.1 Energia de van der Waals	28
2.5.2 Energia de solvatação	30
2.6 Considerações finais	33
3 Algoritmos Evolutivos e Metaheurísticas	35
3.1 Algoritmos Evolutivos	35
3.2 Métodos de busca de referência	38
3.2.1 Busca Aleatória	38
3.2.2 Monte Carlo	39
3.2.3 Algoritmo Genético	41
3.2.4 Evolução Diferencial	42

3.3	Algoritmos de Estimação de Distribuição	44
3.3.1	Exemplo de um EDA	46
3.3.2	EDAs no domínio discreto	49
3.3.3	EDAs no domínio contínuo	51
3.4	EDAs para problemas hierárquico	55
3.5	Considerações finais	56
4	Algoritmos de Estimação de Distribuição para Predição de Estruturas de Proteínas	59
4.1	Algoritmos de referência	60
4.1.1	Busca Aleatória	60
4.1.2	Monte Carlo	60
4.1.3	Algoritmo Genético	62
4.1.4	Evolução Diferencial	63
4.2	Algoritmos de Estimação de Distribuição	65
4.2.1	Univariado	68
4.2.2	2-D Kernel	70
4.2.3	Misturas Finitas Gaussianas	73
4.3	Hierárquico	77
4.4	Energia de Solvatação em GPU	82
4.5	Considerações Finais	84
5	Resultados	87
5.1	Modelos probabilísticos propostos	90
5.1.1	Calibração dos parâmetros dos EDAs	90
5.1.2	Experimentos e análises iniciais	91
5.1.3	Análises complementares	111
5.2	Hierárquico	123
5.2.1	Calibração	123
5.2.2	Resultados com EDAs hierárquicos	125
5.3	Comparação com outros métodos	130
5.3.1	Calibração	130
5.3.2	Comparação entre FGMO, hFGMO e métodos de referência	131
5.3.3	Metaheurísticas e a quantidade de conhecimento <i>a priori</i>	147
5.4	Síntese dos resultados	156
5.5	Considerações finais	158
6	Considerações Finais	163
6.1	Trabalhos futuros	167
Apêndice A	Experimento preliminar com o rBOA	171
Apêndice B	Etapas preliminares ao desenvolvimento do ProtPred-EDA	173
Apêndice C	Energia de solvatação	179
C.1	Influência da energia de solvatação no RMSD	179
C.2	Energia de solvatação em GPU	183

Apêndice D Executando o ProtPred-EDA	191
D.1 ProtPred-EDA local	191
D.1.1 Código-fonte	191
D.1.2 Binários	192
D.1.3 Executando	193
D.2 ProtPred-EDA servidor	193
Apêndice E Análises estatísticas	197

Introdução

Desde a descoberta de uma nova doença até o desenvolvimento de sua cura pode requerer 10 anos de pesquisa e custar cerca de cinco bilhões de dólares (Herper, 2013). Uma das principais razões que contribuem para o alto custo e tempo é o problema em encontrar a estrutura terciária da proteína responsável pela doença. A estrutura terciária de certa proteína é a forma como é encontrada na natureza, ou seja, na forma capaz de exercer sua função, pois, sabe-se que a função que cada proteína exerce depende de sua conformação tridimensional. Assim, conhecendo tal conformação para uma proteína relacionada à causa de uma doença é possível desenvolver fármacos que poderão inibir as ações da doença no organismo. No entanto, a determinação dessa proteína é um processo lento e complexo, exigindo profissionais altamente qualificados e método sofisticados.

A maioria dos métodos existentes utilizados para encontrar a conformação da proteína são experimentais. Tais métodos tentam revelar a estrutura da proteína assim como ela se manifesta na natureza. A determinação da conformação da proteína por Cristalografia de Raio-X (CRX) é um dos métodos mais utilizados. Esse método consiste em disparar um feixe de raio-x sobre um cristal de proteína. A partir disso é possível criar um mapa de difração dos átomos de hidrogênio. Combinando o mapa de difração a sua sequência de aminoácidos da proteína é possível construir a conformação da proteína.

No entanto, as vezes não é possível obter um cristal de proteína, inviabilizando o uso do método CRX. Uma alternativa para esses casos é a Ressonância Magnética Nuclear (RMN) que utiliza apenas uma solução altamente concentrada com a proteína alvo. Essa solução é submetida a um processo que excita os *spins* dos átomos e, dependendo de quanto o átomo se desloca, é possível determinar a conformação da proteína. Entretanto, a aplicação de RMN ainda é restrita a proteínas relativamente pequenas.

Isso significa que nos casos em que a proteína for pequena e que seja possível obter um cristal a proteína poderá ser determinada por ambos os métodos CRX e RMN. Para as proteínas não pequenas e que seja possível obter um cristal é possível utilizar apenas um dos métodos, a CRX. Nos casos em que não seja possível obter um cristal mas a proteína seja pequena então pode-se usar a RMN, a princípio. No entanto, quando não é possível obter um cristal para uma proteína não pequena, é esperado que nenhum dos métodos tenha sucesso. A Figura 1.1 mostra um gráfico hipotético da região em que ambos os métodos experimentais não poderiam ser utilizados. É importante mencionar que há outros métodos experimentais propostos mais recentemente (conforme mostra a Seção 2.3), porém a aplicação deles tem sido relativamente restrita.

Na verdade, em uma análise realizada por Slabinski et al. (2007) é discutido alguns fatores que tornam difícil a determinação de estruturas de proteínas. Um dos fatores considerados é o tamanho da proteína. Embora os métodos experimentais para determinação de estruturas de proteínas funcionem para proteínas pequenas, existe uma classe de proteínas muito pequenas (até 100 resíduos) em que a taxa de sucesso da obtenção de cristais é baixa, inviabilizando o uso da CRX.

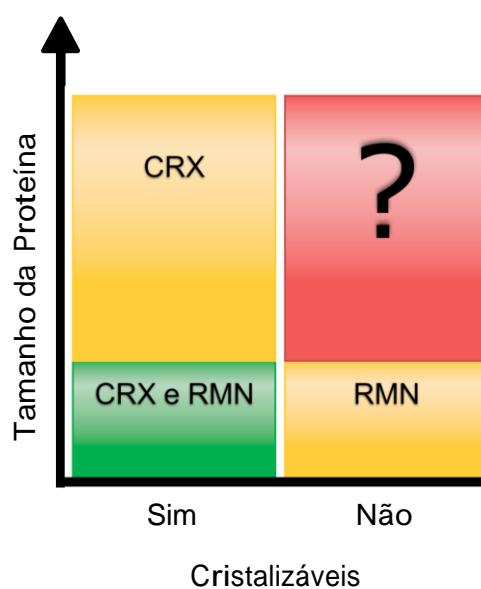


Figura 1.1: Relação entre o tamanho da proteína, a possibilidade de se obter cristais dela e a utilização dos métodos CRX e RMN, destacando a inadequação desses métodos para várias proteínas não pequenas.

A estrutura primária da proteína (sua sequência de aminoácidos) é um identificador para a estrutura terciária da proteína, isto é, para cada estrutura primária existe uma estrutura terciária que corresponde a sua conformação tridimensional mais estável. Ao contrário dos métodos utilizados para encontrar a estrutura terciária da proteína, os métodos utilizados para encontrar a estrutura primária estão mais consolidados. A maior dificuldade em determinar estruturas terciárias em relação as primárias pode ser evidenciada com base em um banco de dados de sequências público, o UniProtKB (ExPASy Proteomics Server, 2009), que possui uma taxa de crescimento relativamente alta em relação a taxa de crescimento do banco de dados de estruturas terciárias, o PDB (Berman

et al., 2000). A Figura 1.2 mostra uma comparação do crescimento entre sequências e estruturas terciárias.

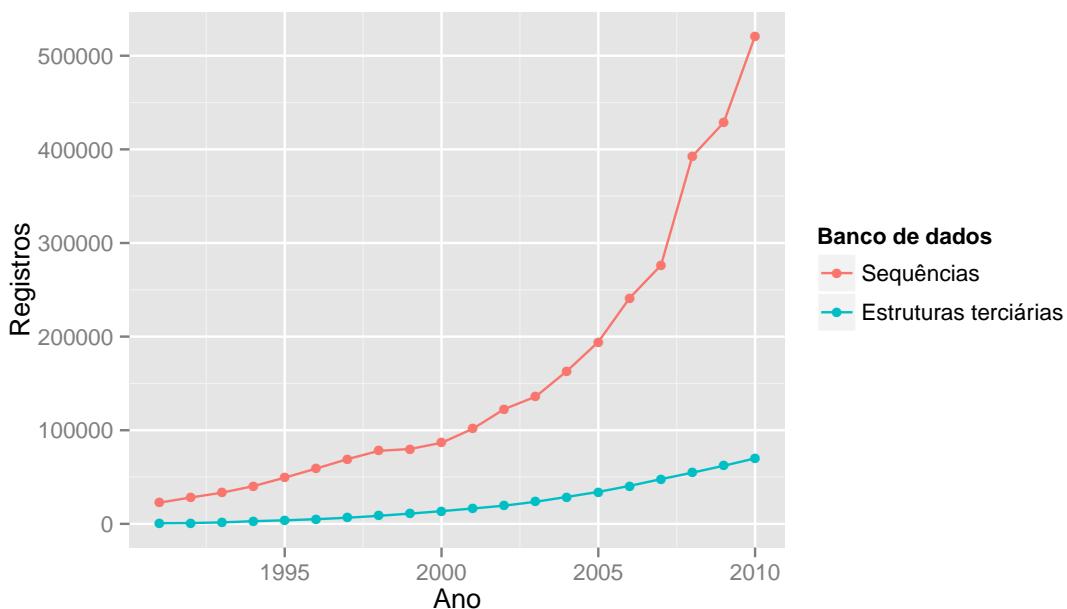


Figura 1.2: Gráfico do crescimento de sequências e estruturas terciárias de proteínas a partir de 1990.

Dessa forma, para reduzir o alto custo e tempo necessário pelos métodos experimentais, métodos computacionais (conhecidos como *in silico*) têm sido investigados. Tais métodos tentam encontrar a estrutura terciária da proteína a partir da sequência de aminoácidos, procurando entre diversas conformações tridimensionais possíveis, as que são mais adequadas segundo cálculos energéticos e fatores estequiométricos (Narayanan & Lakshmikutty, 2006).

Sabe-se que as proteínas estabilizam-se no estado de menor energia (Anfinsen, 1972) e, por conta disso, os métodos computacionais procuram por estruturas de proteínas com menor energia. Essa busca em determinar a estrutura da proteína de forma indireta ou não experimental é conhecida como Predição de Estrutura de Proteína (PSP, do inglês, *Protein Structure Prediction*, Capítulo 2). Basicamente, os métodos *in silico* para isso podem ser classificados em baseados em conhecimento *a priori* ou métodos que não se baseiam em conhecimento *a priori*. Estes, utilizam o cálculo de energia para avaliar uma conformação tridimensional.

Os métodos que se baseiam em conhecimento *a priori* têm sido mais utilizados, priorizando conformações de uma proteína que se assemelham mais a conformações de proteínas obtidas por meio de métodos experimentais como a CRX e a RMN. Embora esses métodos tenham produzido resultados relevantes, a qualidade dos resultados obtidos pode ser baixa para regiões da sequência de aminoácidos da proteína com baixa similaridade, ou mesmo quando há poucas proteínas homólogas no banco de dados de sequências. Dessa forma, o método *in silico* com alta dependência das técnicas CRX e RMN podem não ser adequadas para vários tipos de proteínas.

Métodos que não utilizam conhecimento *a priori* ou utilizam relativamente pouco podem ser considerados *ab initio*. Tais métodos utilizam informações da sequência de aminoácidos e, em geral, do mapa de Ramachandran (que restringe os ângulos diedrais aos valores que correspondem a dobramentos factíveis para estruturas secundárias¹ (Berg et al., 2002)), além do uso de propriedades físico-químicas que descrevem as interações entre átomos da proteína e desses com o solvente. Sabe-se que a estrutura de uma proteína ativa (*in vivo*) corresponde à estrutura da molécula com a menor energia (Seção 2.1). Com base nessa propriedade, métodos *ab initio* realizam cálculos para determinar a disposição no espaço físico dos aminoácidos (e dos átomos de cada aminoácido) de uma estrutura primária de forma que a conformação tenha menor energia. Assim, o problema de PSP pode ser entendido como um problema de minimização de energia e, dessa forma, não depende de técnicas como CRX e RMN ou de conhecimento *a priori* obtido das estruturas determinadas por essas técnicas.

Por outro lado, o cálculo da energia requer o uso de modelos aproximados das interações entre os átomos da proteína, uma vez que os modelos mais elaborados (segundo cálculos da Mecânica Quântica (Greiner, 2001)) requerem tempo de computação significativo, mesmo para conjuntos pequenos de átomos, tornando-os inviáveis para avaliar conformações de proteínas (Webster, 2000; Bujnicki, 2009). Assim, a avaliação de cada conformação pode ser vista como um processo aproximado, que requer investigação para se determinar um equilíbrio efetivo entre a precisão do cálculo da energia e o custo computacional desse cálculo.

Existem algumas maneiras de representar as proteínas em uma estrutura de dados no computador. A primeira, mais simples, utiliza variáveis discretas, conhecida como modelo *lattice* (Seção 2.4.1), mas não tem sido capaz de representar a proteína para propósitos práticos. A segunda é conhecida como *full-atom* (Seção 2.4.3), em que todos os átomos da proteína são considerados, mais completa, capaz de representar melhor as estruturas de proteínas. Dessa forma, é utilizado a representação *full-atom* neste trabalho no qual todos os átomos da proteína são considerados. O valor da energia de uma proteína com representação *full-atom* está diretamente relacionado com o posicionamento que cada átomo se encontra no espaço tridimensional.

No entanto, as coordenadas Cartesianas dos átomos de uma conformação de uma proteína podem ser alteradas livremente até se obter o posicionamento que reflete a menor energia da conformação. Isso não é possível porque a proteína possui ligações covalentes entre os átomos de um mesmo aminoácido e entre aminoácidos. Isso em geral impede que átomos isolados da proteína possam ser deslocados sem movimentar outros. Assim, para que novas conformações de proteínas possam ser geradas com posicionamentos diferentes entre os átomos, é necessário produzir alterações nos ângulos diedrais (Seção 2.4).

Baseado nisso, a Predição de Estruturas de Proteínas pode ser tratada como um problema de otimização no qual se espera obter uma conformação de proteína (variável resposta) alterando os ângulos diedrais (variáveis do problema). No entanto, sabe-se que o espaço de busca das possíveis

¹Segmentos da proteína que formam dobramentos frequentes, ver Seção 2.1.

conformações (todas as possíveis combinações de ângulos diedrais) de uma proteína é combinatório. Dessa forma, a utilização de algoritmos de busca exatos (Woeginger, 2003) (que garantam o ótimo global) em geral é inviável. Por exemplo, para uma função objetivo (energia) multimodal (com vários ótimos locais e/ou globais) e com pouco menos que uma centena de variáveis (ângulos diedrais), em geral não é possível garantir a solução ótima em um tempo computacional aceitável. Em PSP, o tamanho das proteínas varia de algumas dezenas de aminoácidos (cerca de 2.000 átomos) a milhares de aminoácidos (centenas de milhares de átomos). Buscando lidar com a complexidade desse problema, existem várias pesquisas em andamento dedicadas à predição *ab initio* de estruturas de proteínas (Bonneau, 2001; Zhang, 2008; Mijajlovic et al., 2010). Nesse contexto, a escolha da técnica de otimização para orientar a exploração de conformações em espaço de busca combinatório, bem como a modelagem das interações físico-químicas buscando um compromisso entre qualidade das estimativas e custo computacional, são etapas importantes no desenvolvimento de um algoritmo *ab initio* eficiente para PSP.

Apesar dos esforços no desenvolvimento de métodos para PSP *ab initio*, ainda não há um método definitivo para o problema, que consiga lidar com um grande diversidade de proteínas, apesar dos avanços nessa área evidenciados pelo CASP1 a 11. Dentre esses, têm sido investigados os Algoritmos Evolutivos (EA, do inglês, *Evolutionary Algorithm*) para o problema de PSP na literatura (Unger, 2004; Berenboym & Avigal, 2008; Tantar et al., 2010; Bonetti et al., 2013), que em geral se destacam em problemas de otimização global relativamente complexos (multimodais), de larga-escala e/ou multi-objetivos (Deb, 2001).

Quanto melhor for o método de otimização, maiores são as chances de encontrar a solução ou as aproximações dessa com um mesmo esforço computacional. Porém, o desenvolvimento de melhores algoritmos de otimização para o problema de PSP não é trivial. Os ângulos diedrais nas conformações de proteínas relacionam-se e pelo fato de haver vários ótimos locais, um algoritmo de otimização adequado deve ser capaz de tratar distribuições multivariadas e não-paramétricas de forma eficiente.

O ProtPred, desenvolvido no próprio grupo de pesquisa² consiste em um conjunto de Algoritmos Evolutivos que têm sido desenvolvidos para o problema de PSP, iniciado com o Algoritmo Genético (GA, do inglês, *Genetic Algorithm*) (Lima, 2006; de Lima et al., 2007, 2008). Desde então, tem recebido melhorias em relação aos potenciais de energia, em termos de eficiência computacional e refinamentos desses potenciais para uso com métodos de otimização global, além de novos algoritmos como o algoritmo multi-objetivo proposto por Brasil et al. (2013).

Neste cenário de Algoritmos Evolutivos, os Algoritmos de Estimação de Distribuição (EDA, do inglês, *Estimation of Distribution Algorithm*) têm ganhado cada vez mais atenção entre os pesquisadores devido ao sucesso em conseguir explorar regiões promissoras do espaço de busca, em geral, com menos avaliações de soluções do espaço. Os EDAs têm sido aplicados nas mais diversas áreas. Para o problema de PSP foi encontrado um EDA que utiliza o modelo HP (Santana

²Laboratório de Computação Reconfigurável - LCR, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo.

et al., 2004, 2008), o que torna o processo de estimação dos valores das variáveis mais simples, em comparação com o modelo *full-atom*. Por outro lado, não é possível estender a ideia de um EDA para PSP com modelo HP para o modelo *full-atom*, pois não seria possível mapear todas as combinações dos valores das variáveis, visto que as variáveis neste caso são representadas por ângulos diedrais. Dessa forma, considerando as características *full-atom* do problema de PSP, é proposto neste trabalho um novo EDA, específico para tal problema, que substitui o GA do ProtPred por um EDA.

Os principais tipos de EDA para variáveis contínuas que poderiam ser apropriados ao problema de PSP puramente *ab initio* são enumerados a seguir. O *Univariate Marginal Distribution Algorithm* (UMDA) é um dos EDAs mais simples da literatura para variáveis contínuas, porém, pode não ser apropriado ao problema de PSP, por não ser capaz de tratar distribuições não-paramétricas e nem relacionamentos de variáveis. Observe que os ângulos diedrais de um mesmo aminoácido tendem a estar correlacionados. Por outro lado, o *Bivariate Marginal Distribution Algorithm* (BMDA) poderia ser capaz de modelar somente os relacionamentos de variáveis (ϕ, ψ) de um mesmo aminoácido, mas não seria capaz de tratar distribuições não-paramétricas (Pelikan & Muhlenbein, 1999), pois, em geral, as novas soluções são amostradas utilizando uma distribuição normal no BMDA. O *real-valued Bayesian Optimization Algorithm* (rBOA) é capaz de modelar correlação de variáveis e distribuições não-paramétricas (Ahn et al., 2004), entretanto, não seria explorado de forma eficiente, pois pode gastar tempo computacional adicional para encontrar os relacionamentos (ϕ, ψ) que já se sabe de antemão (principalmente os pares (ϕ, ψ) de um mesmo aminoácido). Esse efeito torna-se mais crítico conforme aumenta o tamanho da molécula (o número de variáveis). Na verdade, esse fato foi também verificado em experimentos preliminares desta pesquisa, os quais estão sumarizados no Apêndice A.

Um dos principais fatores que caracterizam um EDA é sua capacidade de estimar os valores das variáveis a partir de um conjunto de soluções promissoras e utilizar essa estimativa para amostrar novas soluções. Esse processo de estimação pode ser caracterizado como o modelo probabilístico do EDA. Dessa forma, o sucesso dos EDAs está diretamente relacionado com a qualidade do modelo probabilístico e a eficiência do método de construção do modelo.

Neste trabalho foram desenvolvidos um total de quatro modelos probabilísticos dedicados ao contexto de PSP puramente *ab initio*. O mais simples, chamado UNI, simula o processo de uma distribuição kernel univariada (Seção 4.2.1). Aproveitando o conhecimento do problema sobre relação de ângulos (ϕ, ψ) do mesmo aminoácido, foram propostos dois modelos que consideram correlações bivariadas: 1) utilizando uma distribuição kernel bi-dimensional (KDE2D, do inglês, *two-dimensional Kernel Density Estimation*) (Seção 4.2.2) e 2) misturas de Gaussianas finitas (FGM, do inglês, *Finite Gaussian Mixtures*) (Seção 4.2.3). Por fim, foi desenvolvido um modelo hierárquico (HM, do inglês, *Hierarchy Model*) que pode ser combinado com qualquer um dos outros três modelos probabilísticos desenvolvidos. O HM divide o problema em subproblemas menores e tenta tratá-los de forma independente juntando-os por uma estrutura de árvore para compor a solução final.

Diferentes aspectos foram considerados para avaliar a qualidade do EDA para PSP proposto e os modelos probabilísticos desenvolvidos. Foi considerado a qualidade das proteínas preditas em relação à proteína nativa (RMSD, do inglês, *Root Mean Square Deviation*, (Bergeron, 2002)), o tempo de cada execução e o valor das energias obtidas. Verificou-se que, de fato, os modelos probabilísticos mais sofisticados como o KDE2D, FGM e HM produziram os melhores resultados (considerando energia e RMSD). Além disso, foi realizada uma comparação entre o EDA proposto com outras metaheurísticas da literatura como a Busca Aleatória (RW, do inglês, *Random Walk*) (Pearson, 1905), Monte Carlo (MC (Metropolis & Ulam, 1949)), GA e a Evolução Diferencial (DE, do inglês, *Differential Evolution*) (Storn & Price, 1997). Verificou-se que todas as metaheurísticas podem encontrar a estrutura correta da proteína, para um caso específico, se o espaço de busca for reduzido a uma região suficientemente próxima da solução correspondente à proteína nativa. Em outras palavras, essas metaheurísticas requerem conhecimento *a priori* para ter sucesso, isto é, não podem ser consideradas métodos puramente *ab initio*. Por exemplo, a RW é relativamente limitada, assim, para encontrar a solução correta foi necessário reduzir significativamente o espaço de busca. Por outro lado, o EDA pode, em geral, encontrar a solução correta sem reduzir o espaço de busca, isto é, sem tendência fornecida *a priori* para a busca se concentrar em uma ou outra região do espaço de busca. Isso é uma vantagem, pois pode beneficiar a predição de estruturas com regiões de sua sequência que têm baixa similaridade.

Verificou-se também que há um aumento na qualidade das metaheurísticas seguindo a ordem RW, MC, GA, DE e EDA. Uma das principais diferenças entre tais metaheurísticas é o número de indivíduos que são utilizados para compor cada novo indivíduo da população da geração seguinte. Isso indica que há uma relação entre quantidade de indivíduos utilizados para construir cada nova solução e a qualidade da metaheurística. A RW não utiliza informação de sua população atual para construir as novas soluções, isto é, basicamente não há herança de características ocorrendo, pois uma nova solução não é modificada para gerar uma nova. O MC utiliza informação de um indivíduo da população para construir uma nova solução. Para o GA, basicamente, informações de dois indivíduos (recombinação, Seção 3.2.3) são herdados na construção de uma nova solução. O DE utiliza informação de três indivíduos para construir uma nova. Por outro lado, os EDAs utilizam um conjunto de indivíduos promissores, chamados de indivíduos selecionados. Assim, ao se gerar uma nova solução a partir do modelo probabilístico, a informação de vários indivíduos está sendo considerada. Dessa forma, o número de indivíduos utilizados para amostrar os indivíduos da próxima geração compõe um dos fatores principais no desenvolvimento de um EA eficiente.

1.1 Objetivos

O objetivo global deste trabalho é o desenvolvimento de um EDA específico para o problema de PSP puramente *ab initio* e *full-atom*, capaz de prever estruturas de proteínas. O objetivo secundário é o desenvolvimento de modelos probabilísticos adequados que fossem capazes de extrair informações relevantes de um conjunto de soluções promissoras e, assim, guiar o processo de

busca do EDA em direção a regiões promissoras do espaço de busca sem precisar de conhecimento *a priori* de quais são essas regiões.

1.2 Relevância da pesquisa

Este trabalho de doutorado contribuiu para a área de computação por meio do desenvolvimento de novas metaheurísticas para variáveis contínuas em problemas com múltiplos ótimos locais e correlação de variáveis. Cada metaheurística desenvolvida possui seu próprio modelos probabilísticos, que são inéditos no contexto de EDAs para PSP como o KDE2D e FGM. Para isso, os modelos probabilísticos foram implementados de forma eficiente a fim de predizer estruturas de proteínas rapidamente, permitindo que as técnicas estatísticas utilizadas como o KDE2D e FGM possam ser utilizadas em outros algoritmos ou então para outros problemas (*benchmarks* ou mesmo outros problemas do mundo real).

Os EDAs propostos foram projetados exclusivamente para o problema de PSP. Pode-se dizer que a contribuição deste trabalho com a área da biologia ainda é pequena, pois, embora os EDAs propostos possam garantir a menor energia na maioria das vezes, ainda há fatores do problema que precisão ser considerados. Por exemplo, espera-se obter um ganho considerável para o problema de PSP após a implementação de uma versão multi-objetivo bem como o refinamento das funções de energia. Dessa forma, utilizando uma metaheurística moderna e modelos probabilísticos eficientes, novas pesquisas podem surgir tendo este trabalho como base.

1.3 Organização

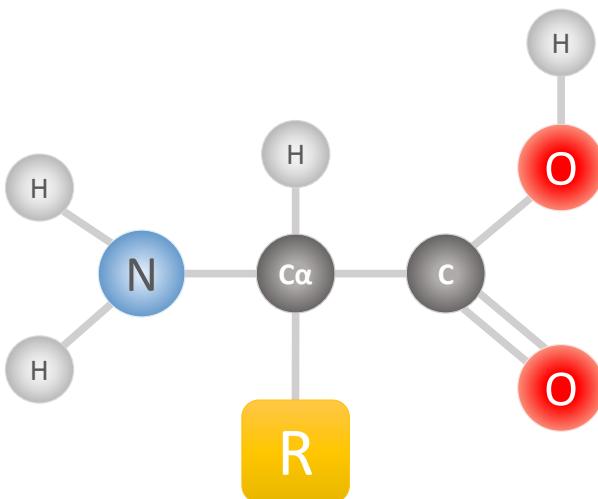
Este trabalho está organizado da seguinte maneira. No Capítulo 2 é apresentado o problema de Predição de Estruturas de Proteínas. No Capítulo 3 é mostrado as técnicas de otimização computacionais utilizadas para otimizar uma determina função. O Capítulo 4 discute as técnicas que foram desenvolvidas neste trabalho aplicadas ao problema de Predição de Estruturas de Proteínas. Os resultados obtidos durante a realização deste trabalho são apresentados no Capítulo 5. Por fim, no Capítulo 6 é apresentado as considerações finais sobre este trabalho de doutorado.

Predição de Estruturas de Proteínas

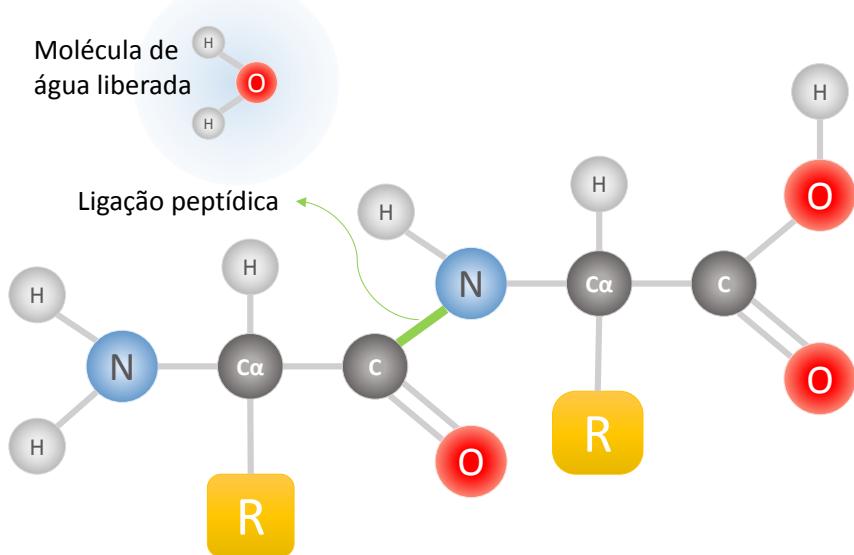
Este capítulo situa a abordagem investigada para PSP em relação a várias técnicas experimentais e *in silico* existentes. A Seção 2.1 descreve alguns conceitos sobre as estruturas das proteínas. A Seção 2.2 trata o problema de predição de estruturas de proteínas. A Seção 2.3 apresenta as técnicas experimentais e *in silico* para determinar estruturas de proteínas. A Seção 2.4 introduz métodos computacionais para PSP *ab initio*. A Seção 2.5 descreve as duas funções de energia utilizadas neste trabalho. Por fim, a Seção 2.6 apresenta as considerações finais deste capítulo.

2.1 Estrutura de Proteínas

As proteínas são compostos orgânicos constituídos por uma sequência de até 20 tipos principais de aminoácidos conectados por meio de ligações peptídicas (ver Tabela 2.1). Tais ligações conectam dois aminoácidos por meio dos átomos *C* e *N* (Figura 2.1(b)). Os aminoácidos têm como base estrutural o carbono alfa ($C\alpha$) que se liga a um grupo amina (NH), um grupo carboxila (CO) e um átomo de hidrogênio (Figura 2.1(a)). Uma ligação peptídica é o resultado da reação entre o grupo carboxila e o grupo amino. As ligações peptídicas, ao reagirem, liberam uma molécula de água (H_2O) como produto secundário e o aminoácido passa a ser chamado de resíduo. As ligações entre os átomos $N - C\alpha - C$ de aminoácidos encadeados caracterizam a cadeia principal da proteína. Cada tipo de aminoácido tem uma cadeia lateral própria, o que diferencia os tipos de aminoácidos. Os encadeamentos de aminoácidos por meio de ligações peptídicas geram sequências (também chamadas de estrutura primárias) que se enovelam formando diferentes tipos de proteínas. Comparando a uma simples molécula de água com apenas três átomos (H_2O), uma proteína pode possuir várias dezenas de milhares de átomos (Marzzoco & Torres, 1999).



(a) Estrutura básica de um aminoácido.



(b) Encadeamento de dois aminoácidos destacando a ligação peptídica.

Figura 2.1: Exemplo de um aminoácido e uma ligação peptídica.

O processo pelo qual as proteínas são formadas é chamado biossíntese, que se desenvolve a partir de informação codificada nos genes, o DNA. Entretanto, as proteínas não são formadas diretamente a partir do DNA. Primeiramente, o DNA é transcrito em pré-RNA mensageiro (pré-mRNA) e depois em mRNA, que será utilizado como molde pelo ribossomo para produzir a proteína, processo chamado tradução. Após a formação da sequência de aminoácidos, esta rapidamente dobra-se e assume seu formato natural. A forma como a proteína é dobrada depende da estrutura primária. Existem certos tipos de proteínas, conhecidas como chaperonas (Laskey et al., 1978), que auxiliam no processo de dobramento. Para que uma proteína desnaturada (que ainda não foi dobrada ou sem forma estável), assuma a forma natural (dobrada e estável), os ângulos diédrais (Seção 2.4) são alterados até a proteína estabilizar-se em um estado de baixa energia (Berg et al., 2002).

Uma conformação de uma proteína é uma representação tridimensional de um dobramento geometricamente possível da proteína. Sabe-se que as proteínas encontradas na natureza estão em uma conformação (estado) que represente a menor energia. Para que as proteínas desempenhem suas funções é necessário que estejam no estado de menor energia, pois este é um estado estável (de baixa variabilidade em sua conformação). Assim, a função de cada proteína é determinada pela sua estrutura tridimensional. Diferentes sequências de aminoácidos resultam, em geral, em conformações diferentes que correspondem a proteínas com diferentes funções. As proteínas podem ter função: enzimática (que facilitam reações bioquímicas), transportadora (que transportam substâncias como, por exemplo, a glicose, através de membranas celulares), estrutural (dando suporte e resistência), sinalizadora (controlando e coordenando atividades no organismo), motora (transportam estruturas de um lugar da célula para outro) e reguladora (que regulam atividades metabólicas do organismo) (Lodish et al., 2003).

As estruturas das proteínas são classificadas em uma hierarquia de quatro níveis. A estrutura primária da proteína é a própria sequência de aminoácidos. A estrutura secundária refere-se a segmentos da estrutura primária que formam dobramentos isolados (com um padrão), como a α -hélice, folha- β e as *voltas* (ver Figura 2.2). Há também estruturas super-secundárias, também chamadas de motivos, que formam dobramentos com certos padrões presentes em várias estruturas de proteínas (Chiang et al., 2007). A combinação de todas as estruturas secundárias de uma proteína forma uma estrutura terciária, que corresponde a forma final que a proteína irá assumir. Existem também várias estruturas terciárias que se ligam a outras estruturas terciárias para formar uma estrutura mais complexa, conhecidas como estruturas quaternárias. Por exemplo, a hemoglobina (Figura 2.2(d)), que é formada pela ligação de quatro estruturas terciárias.

2.2 O Problema da Predição da Estrutura de Proteínas

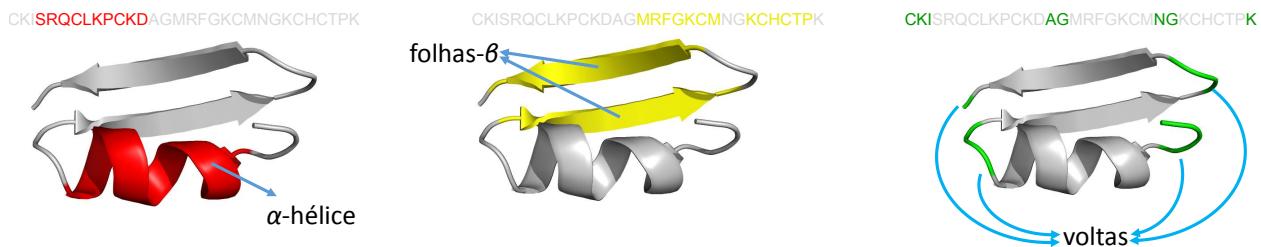
Várias pesquisas estão em andamento para tentar explicar como efetivamente ocorrem os detalhes do processo de dobramento das proteínas, porém ainda não é conhecido nenhuma expliação que tenha sido completamente aceita. Devido à complexidade de entender esse processo e mimetizá-lo, por exemplo, *in silico*, várias pesquisas têm buscado um caminho alternativo: a predição da estrutura da proteína “dobrada”.

As ligações entre os átomos de carbono da cadeia principal são simples, permitindo que os átomos rotacionem com certa liberdade, gerando uma grande variedade de conformações, cujo número aumenta quanto maior a sequência de aminoácidos. A estabilidade de uma estrutura depende da energia associada às interações das ligações não-covalentes¹, pois essa energia varia significativamente com mudanças nas conformações. Assim, as conformações com menor energia são mais estáveis e provavelmente correspondem ou aproximam-se da estrutura da proteína nativa. A estru-

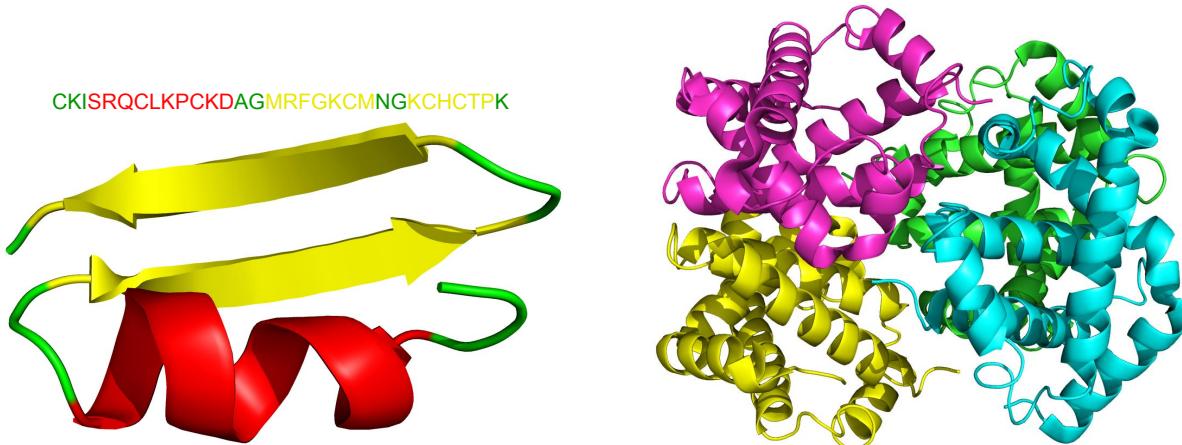
¹Interação que ocorre entre átomos próximos que não compartilham elétrons.

CKISRQCLKPCKDAGMRFKGKCMNGKCHCTPK

- (a) Estrutura primária ou sequência de aminoácidos para a proteína 2CK5. Cada letra corresponde à sigla de um aminoácido, mostrado na Tabela 2.1. Assim, o encadeamento das siglas (aminoácidos) na forma CKISRQCLKPCKDAGMRFKGKCMNGKCHCTPK corresponde a estrutura primária da proteína 2CK5.



- (b) Estrutura da proteína 2CK5, destacando as estruturas secundárias correspondentes à estrutura primária.



(c) Estrutura terciária da proteína 2CK5.

(d) Estrutura quaternária da proteína 4HHB. Estrutura formada por quatro cadeias de estruturas terciárias representadas em cores diferentes.

Figura 2.2: Níveis de estruturas de proteínas. As estruturas foram desenhadas utilizando o software de modelagem molecular PyMol (Schrödinger, LLC, 2010).

tura nativa de uma proteína refere-se a conformação da proteína da maneira com que é encontrada na natureza, ou seja, na conformação que permite desempenhar sua função.

Em termodinâmica, a energia livre de um sistema é a porção de energia disponível que, a temperatura e pressão constantes, é capaz de realizar trabalho (força necessária para produzir uma modificação no sistema). Um sistema é a matéria representada por uma região definida e ambiente pela região ao redor. No contexto de PSP, o sistema pode ser definido como a molécula da proteína. Assim, é esperado que as proteínas encontradas na natureza tenham um valor de energia negativo, que corresponde ao nível energético característico de uma reação espontânea. Nessa situação, a proteína deve estar em um estado termodinâmico estável. Para sair desse estado, seria necessária alguma modificação no meio em que a proteína se encontra, como o aumento da temperatura, que

Tabela 2.1: Tabela dos 20 aminoácidos. É mostrado também o número de átomos e a quantidade de ângulos quirais (Seção 2.4.3) para cada aminoácido.

Aminoácido	Abreviação	Sigla	Átomos	Ângulos quirais
Glicina	Gly	G	10	0
Alanina	Ala	A	13	1
Cisteína	Cys	C	14	1
Serina	Ser	S	14	1
Treonina	Thr	T	17	1
Valina	Val	V	19	1
Asparagina	Asn	N	17	2
Aspartato	Asp	D	16	2
Fenilalanina	Phe	F	23	2
Histidina	His	H	20	2
Isoleucina	Ile	I	22	2
Leucina	Leu	L	22	2
Prolina	Pro	P	17	2
Tirosina	Tyr	Y	24	2
Triptofano	Trp	W	27	2
Glutamato	Glu	E	19	3
Glutamina	Gln	Q	20	3
Metionina	Met	M	20	3
Arginina	Arg	R	26	4
Lisina	Lys	K	24	4

pode aumentar a energia livre da molécula e consequentemente alterar sua conformação (Anfinsen, 1972; Nelson & Cox, 2004).

Métodos experimentais como a Cristalografia de Raio-X e Ressonância Magnética Nuclear (Seção 2.3.1) têm sido os métodos mais utilizados hoje para a determinação da estrutura da proteína nativa. Entretanto, pesquisas na área de PSP *in silico* estão recebendo grande atenção entre os pesquisadores (Gibas & Jambeck, 2001). Os métodos baseados em conhecimento, como os que baseiam em Homologia de sequências (Hilbert et al., 1993) ou similaridade a nível estrutural ou energético, também chamados de *Threading* (Baxevanis & Ouellette, 2001), dependem fortemente de resultados experimentais relativos a outras proteínas para produzir predições adequadas.

Por outro lado, o método de predição *ab initio* busca encontrar a estrutura da proteína baseando-se principalmente nas interações físico-químicas entre os átomos da proteína. No entanto, esse é um problema combinatório NP-completo (Setubal & Meidanis, 1997; Khimasia & Coveney, 1997) que necessita de métodos computacionais eficientes para encontrar aproximações da solução desse problema que sejam adequadas. O problema de PSP puramente *ab initio* pode ser formulado como um problema de minimização, o qual pretende-se encontrar o mínimo global de uma função que estima a energia livre de uma conformação de proteína. É assumido que o mínimo global

da função é o ponto de menor energia livre, correspondendo à conformação da proteína nativa (Bergeron, 2002), Seção 2.5.

Neste trabalho, o termo “puramente *ab initio*” é utilizado para distinguir de métodos de PSP também chamados de *ab initio* que se caracterizam por não só minimizar a energia livre, mas também por utilizar outras informações como o tipo de estrutura secundária que provavelmente cada aminoácido da sequência pertence

2.3 Técnicas existentes de PSP

A primeira estrutura de proteína foi determinada em 1958 por Kendrew et al. (1958). Desde então, têm sido desenvolvidas várias técnicas para encontrar estruturas de proteínas. As técnicas mais comuns são as experimentais (*in vitro*), responsáveis pela maioria das estruturas já determinadas. Outro tipo são os métodos *in silico*, que simulam e avaliam os estados que as proteínas podem assumir, utilizando representações e cálculos computacionais. As Seções 2.3.1 e 2.3.2 apresentam algumas das principais técnicas *in vitro* e *in silico*.

2.3.1 Técnicas experimentais

Há várias técnicas capazes de determinar estruturas de proteínas *in vitro*, dentre elas destacam-se: Cristalografia de Raio-X (CRX) (Glasser, 1934), Ressonância Magnética Nuclear (RMN) (Rabi et al., 1938) e Microscopia Eletrônica (MIC) (Ruska, 1986). Ao contrário de técnicas *in vivo* em que os experimentos são realizados no próprio organismo vivo, as técnicas *in vitro* são caracterizadas por permitirem que os experimentos sejam realizados em laboratório, a partir de amostras de proteínas (Setubal & Meidanis, 1997). A Figura 2.3 mostra a quantidade de proteínas que já foram determinadas usando essas três técnicas *in vitro*. Na sequência, as características básicas dessas técnicas são apresentadas.

Cristalografia de Raio-X

Se fosse possível colocar a proteína em um microscópio e observar sua estrutura, o processo de determinar a estrutura da proteína seria relativamente simples. No entanto, mesmo os microscópios mais eficientes disponíveis hoje não são capazes de revelar detalhes de objetos em nível atômico (Schmolze et al., 2011). Por outro lado, o comprimento de onda do raio-x é de aproximadamente 1 Å (Angstrom), semelhante ao diâmetro de um átomo de hidrogênio, tornado viável seu uso na determinação de estruturas de proteínas.

O primeiro passo para determinar a estrutura de proteína com a CRX é a obtenção de um cristal da proteína. A cristalização da proteína pode ser considerada a etapa mais complicada e trabalhosa da CRX, pois a obtenção de cristais de proteínas com qualidade pode requerer até um

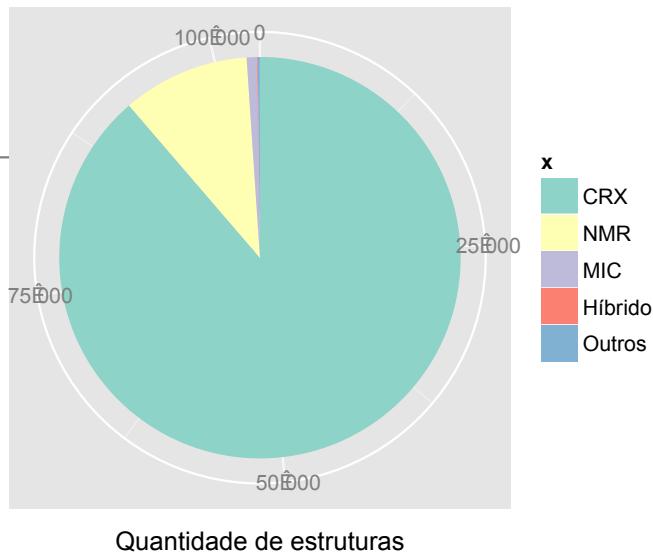


Figura 2.3: Quantidade de estruturas de proteínas determinadas por diferentes técnicas *in vitro*.

ano de investigação. Por razões desconhecidas, não é possível obter cristais de algumas proteínas, inviabilizando a técnica de CRX para tais proteínas (Webster, 2000).

Após a obtenção de um cristal de proteína, um feixe de raio-x é incidido sobre o cristal. Uma parte do feixe emitido passa direto pelo cristal de proteína atingindo diretamente um filme. Outra parte do feixe é desviada de acordo com o tipo de átomo presente no cristal. Um átomo de carbono é capaz de desviar o feixe de raio-x seis vezes mais que um átomo de hidrogênio, pois o carbono tem seis vezes mais elétrons que o hidrogênio. Com isso, o espalhamento dos pontos do filme (simbolizados pelos feixes difratados na Figura 2.4) segue um padrão dependente da posição dos átomos de C e H no cristal. A Figura 2.4 mostra o esquema de funcionamento da CRX.

Até mesmo uma proteína pequena gera muitos pontos no filme de raio-x. Pode-se reconstruir a geometria da proteína analisando os pontos obtidos em várias camadas do filme com base também na sequência de aminoácidos da proteína e com auxílio do computador. Dessa forma, a determinação da estrutura corresponde em resolver o problema inverso de calcular as posições relativas no espaço tridimensional dos átomos C e H, com base nesses dados. Naturalmente, o resultado final depende da qualidade do cristal utilizado e da quantidade dos pontos obtidos (Berg et al., 2002).

Todo esse processo em geral requer especialistas, tempo e recursos financeiros significativos. Além disso, a forma cristalizada da proteína pode esconder aspectos fundamentais da estrutura da proteína, visto que a proteína cristalizada (estática) pode ter sua geometria com certa diferença em relação a mesma proteína *in vivo*. Além disso, as proteínas cristalizadas não vibram da mesma forma que no estado *in vivo*. Em geral, isso pode ser contornado realizando uma simulação de Dinâmica Molecular (DM) a partir da estrutura obtida por CRX para estimar as conformações mais prováveis da proteína em organismos *in vivo*. Apesar dessas restrições, a CRX é o método que mais tem determinado estruturas de proteínas (Berg et al., 2002).

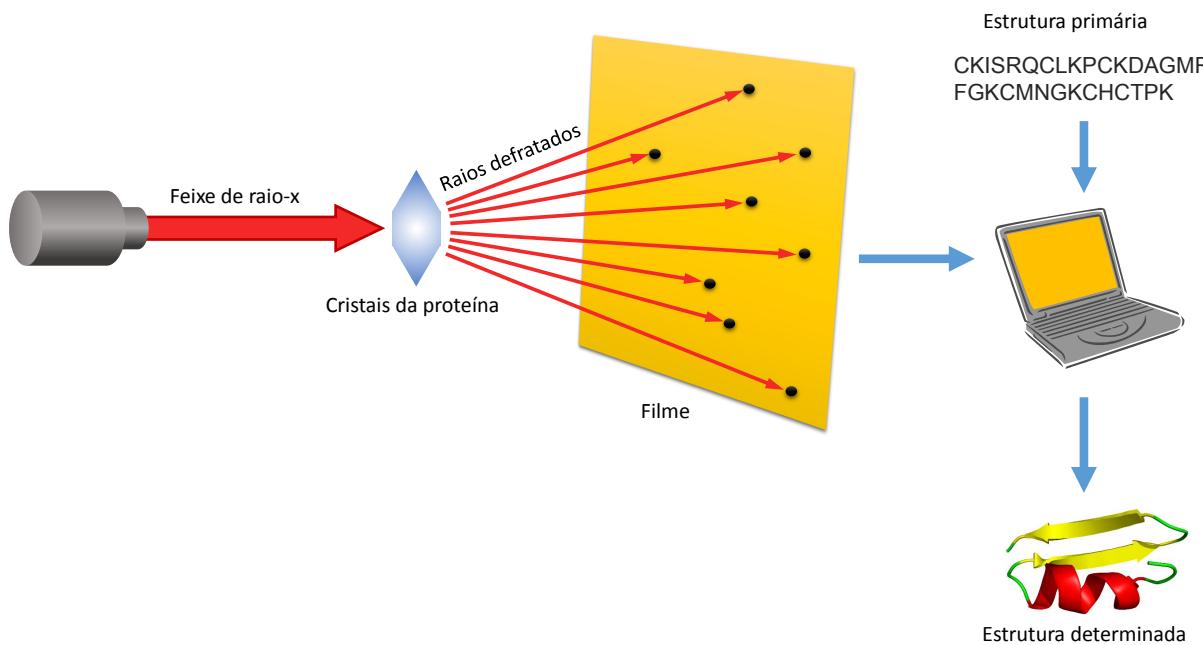


Figura 2.4: Determinação de estruturas de proteínas com a CRX.

Ressonância Magnética Nuclear

A vantagem da Ressonância Magnética Nuclear (RMN) é que ela trabalha diretamente com uma solução altamente concentrada de proteínas, sem a necessidade de obter cristais de proteínas (uma das etapas mais críticas da CRX). Consequentemente, o processo de RMN é em geral mais rápido que o da CRX, além de ser viável para algumas proteínas que não se consegue obter cristais. No entanto, a RMN somente é aplicável a proteínas com até cerca de 100 aminoácidos, isto é, para proteínas relativamente pequenas (Webster, 2000). Em alguns casos, é possível também quebrar a proteína em subdomínios e tratá-los de forma independente, juntando os subdomínios para reconstruir a estrutura da proteína completa (Alberts et al., 2007).

A RMN baseia-se no fato de que o núcleo de alguns átomos é magnético, conhecido como *spin*. Por exemplo, o núcleo do hidrogênio tem apenas um próton e nenhum nêutron, por isso produz um estado magnético. Esse estado (*spin*) pode assumir duas orientações quando um campo magnético externo é introduzido, produzindo uma variação na energia, proporcional a força gerada pelo campo magnético. A frequência da variação dos dois estados magnéticos pode ser obtida por meio da ressonância com sinais emitidos com frequência controlada. Variando o campo magnético é possível obter um espectro de ressonância. Núcleos de diferentes átomos inseridos em meios diferentes também emitem frequência de radiação um pouco diferente. Em outras palavras, utilizando uma certa frequência, a variação da energia (entre os dois estados) é absorvida. Ao utilizar diferentes frequências é possível mapear a distância entre elementos químicos e criar um espectro bidimensional das distâncias desses elementos. Assim, utilizando o mapa de distância mais a sequência de aminoácidos é possível construir a estrutura terciária da proteína (Berg et al., 2002). A Figura 2.5 mostra um esquema da determinação de estruturas de proteínas com a RMN.

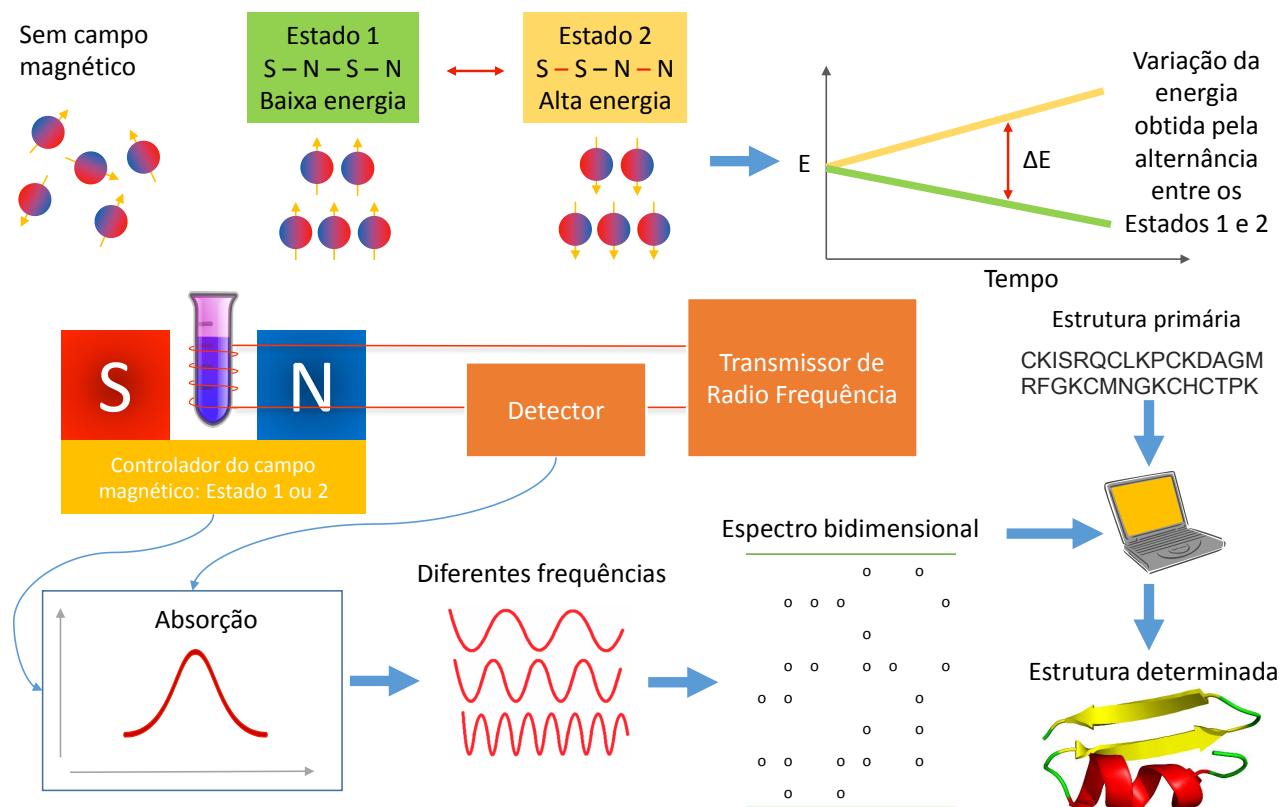


Figura 2.5: Determinação de estruturas de proteínas com a RMN. A partir de uma solução altamente concentrada da proteína um campo magnético externo é gerado, alterando o estado magnético dos *spins* dos átomos. Conforme varia-se a frequência do campo magnético, possibilita-se localizar as posições relativas dos prótons em um mapa bidimensional. Nesta figura, as Siglas *N* significa (polaridade) Norte e *S* significa (polaridade) Sul.

Microscopia Eletrônica

Outro método de determinação de estrutura de proteína que não necessita da obtenção de cristais é a Microscopia Eletrônica (MIC), pois trabalha diretamente com a molécula. Em geral, é utilizada para determinar grandes complexos de macromoléculas. Se uma determinada estrutura de proteína possui subestruturas simétricas é possível utilizar métodos como, por exemplo, a difração de raio-x para gerar mapas de densidade 3D da proteína, assim como feito para os cristais em CRX. Considerando que uma proteína possui certa simetria, várias imagens de MIC da proteína são tiradas de vários ângulos. Ao combinar tais imagens por meio de um alinhamento é possível reconstruir uma representação 3D da proteína. Apesar da MIC não revelar o posicionamento de cada átomo, pode ser combinada com CRX ou RMN para se obter tal posicionamento. Pelo fato da MIC ser mais eficiente para estruturas multi-moleculares, tem sido mais utilizada para determinação de complexos de ribossomos, RNA transportador e cápside de vírus (RCSB, 2014).

2.3.2 Técnicas *in silico*

Os métodos experimentais *in vitro* são os responsáveis pela maioria das estruturas de proteínas encontradas. Certas limitações deles têm motivado pesquisas que utilizam métodos computacionais, *in silico*, para PSP (Gibas & Jambeck, 2001). Em geral, os métodos computacionais utilizam algoritmos que buscam encontrar estruturas de proteínas com a menor energia (Seção 2.5). Existem duas principais abordagens para PSP *in silico*: métodos baseados em conhecimento *a priori* e *ab initio*.

Também conhecidos como *template-based*, os métodos baseados em conhecimento *a priori* utilizam informação de estruturas de proteínas determinadas por métodos *in vitro* para predizer novas proteínas *in silico*. Em geral, esse conhecimento é organizado em estatísticas sobre as estruturas e subestruturas já determinadas, conforme apresentado no Capítulo 1. Observe que, de acordo com que as Figuras 1.1 e 1.2 mostram, podem existir aspectos de estruturas que não são conhecidos ou aspectos cuja frequência deles na natureza possa ser diferente da verificada por meio das estruturas, até então determinadas. Essas restrições estatísticas (das proteínas obtidas por métodos experimentais) podem introduzir certa tendência nas previsões que as utilizam. Apesar disso, esse tipo de previsão tem crescido significativamente, devido ao sucesso obtido em vários casos (Sobha et al., 2008).

Os métodos *ab initio*, também conhecidos como *template-free*, não utilizam o princípio conhecimento de outras estruturas em suas previsões. Em geral, têm se mostrado capazes de predizer alguns pedaços pequenos de proteínas ou estruturas secundárias como, por exemplo, hélices e voltas. Com o avanço em relação aos modelos de campos de força (MacKerell Jr et al., 2000; Case et al., 2004; Scott et al., 1999) e à capacidade computacional, verifica-se uma tendência de que os métodos *ab initio* aumentem seu espaço entre os algoritmos de PSP (Zhou et al., 2011b).

Os métodos *ab initio* podem ser divididos em: 1) semi *ab initio*, que utilizam estatísticas sobre subsequências relativamente pequenas de aminoácidos (de até 20 aminoácidos) em suas previsões como, por exemplo, o Rosetta e o Quark; 2) puramente *ab initio*, que não utilizam informação estatística sobre subsequências ou estimativa do tipo de estrutura secundária de cada aminoácido na proteína, apenas utilizam a sequência de aminoácidos da proteína e eventualmente informações sobre as regiões factíveis do diagrama de Ramachandran (Ramachandran et al., 1963). A vantagem de se utilizar (1) em relação a (2) é que o espaço de busca é restrito a suas regiões promissoras (obtidas de acordo com o conhecimento *a priori*). Entretanto, a busca somente nessas regiões pode gerar tendência, guiando o processo de busca na direção de estruturas já conhecidas, que podem não ser modelos adequados. Por outro lado, o método (2) pode analisar qualquer região do espaço de busca e, então, encontrar conformações sem similares dentre as já determinadas pelos métodos experimentais.

Na sequência, são apresentadas duas abordagens que usam conhecimento *a priori*. Os métodos computacionais *ab initio* são apresentados na Seção 2.4

Homologia

A modelagem por Homologia, também conhecida como Modelo Comparativo, primeiramente procura por sequências de proteínas disponíveis no PDB (do inglês, *Protein Data Bank*, (Berman et al., 2000)) que sejam semelhantes à sequência da proteína alvo (proteína com estrutura desconhecida) (Orengo et al., 2003). A partir de um conjunto de proteínas semelhantes é realizado um alinhamento com a sequência da proteína desconhecida, com certa tolerância na similaridade das sequências. Sabe-se que, em geral, proteínas com sequências semelhantes possuem estruturas semelhantes.

As proteínas homólogas são caracterizadas por possuírem um ancestral comum, isto é, são proteínas que possuem estruturas e sequências semelhantes. Nesse caso, podem ser consideradas da mesma família e, como consequência, possuem alta Homologia. Por exemplo, a família do barril TIM possui estruturas similares, as quais possuem oito α -hélices e oito folhas- β paralelas (Branden & Tooze, 1999). No entanto, existem certas sequências de proteínas com alta similaridade (cerca de 75%) que possuem estruturas significativamente diferentes, inviabilizando o método de predição por Homologia nesses casos (Kosloff & Kolodny, 2008). As proteínas homólogas são conhecidas por possuírem sequências e estruturas semelhantes. Lembrando que a função da proteína é determinada pela estrutura que a mesma assume, e proteínas semelhantes em geral desempenham funções semelhantes, pode agrupar proteínas semelhantes em famílias, enfatizando a provável origem comum. Apesar de existirem várias estruturas de proteínas, muitas delas são semelhantes. Em geral, para cada proteína, existem aproximadamente 40% de proteínas semelhantes que podem ser utilizadas por métodos baseados em Homologia (Becker et al., 2001). Com o aumento do depósito de estruturas de proteínas no PDB, favorece também a utilidade desses métodos.

Basicamente, a Homologia consiste em quatro passos: 1) Identificação das estruturas de proteínas conhecidas que se relacionam com a sequência da proteína alvo. Pode-se utilizar o banco de dados de sequências de proteínas como o UniProtKB (ExPASy Proteomics Server, 2009) e o PDB para estruturas; 2) Alinhamento da sequência da proteína alvo com as proteínas de modelo. Este é o passo mais importante pois a proteína alvo vai basear-se nas proteínas obtidas para se tornarem os modelos. Quando a diferença de similaridade da proteína alvo com os modelos é superior a 70% o alinhamento pode ser simples. No entanto, quando a similaridade é menor que 40% é necessário utilizar várias sequências de proteínas (Orengo et al., 2003); 3) Construção do modelo para a proteína alvo. Nos casos das regiões da sequência com baixa similaridade (menor que 40%), pedaços de modelos tridimensionais são gerados e avaliados para todas as possibilidades de modelos (Becker et al., 2001); 4) Avaliação e refinamento do modelo construído. Esses passos podem ser repetidos até que um modelo adequado seja encontrado (Martí-Renom et al., 2000). Programas computacionais como, por exemplo, o Modeller (Sali & Blundell, 1993) e o Swiss-Model (Arnold et al., 2006) têm se destacado na modelagem de proteínas por Homologia.

A predição de estruturas de proteínas utilizando Homologia tem sido uma estratégia interessante no processo de desenvolvimento de fármacos, pois estruturas com qualidade adequada para

esse fim podem ser determinadas com um custo relativamente baixo. Em um estudo realizado por Sánchez & Sali (1998) foi predito mais de 1.000 estruturas de proteínas para uma espécie de fermento, em que apenas 40 delas tiveram suas estruturas determinadas por métodos experimentais. A Homologia pode também auxiliar no entendimento da ligação entre duas proteínas sem haver suas estruturas terciárias determinadas (Ogawa & Toyoshima, 2002).

Embora a Homologia tenha produzido resultados significativos, ainda é necessário desenvolver métodos mais eficientes para construção de modelos, para alinhamento de múltiplas sequências e para refinamentos, para que erros gerados nessas fases não impeçam a geração de modelos úteis (Cavasotto & Phatak, 2009). Naturalmente, há também a limitação de uso quando não se tem um conjunto de proteínas homólogas suficientemente grande. Em um estudo realizado por Peng & Xu (2010), utilizando dados obtidos por Pieper et al. (2006), mostra que mais de 70% dos modelos do banco de dados de sequências Modbase (Karchin et al., 2005) possuem menos de 30% de similaridade.

Threading

Outro método que utiliza conhecimento *a priori* para prever estruturas de proteínas é o de *Threading* ou *fold-recognition* (Jones et al., 1992). Em geral, o *Threading* é utilizado para complementar a Homologia, isto é, quando não se encontra estruturas homólogas no PDB ou quando a similaridade da proteína alvo é baixa (Jones, 1997). Diferentemente da Homologia, em que a sequência da proteína alvo é comparada com sequências de proteínas, o *Threading* utiliza mecanismos para comparar trechos da estrutura da proteína alvo com outros trechos de estruturas que sejam similares em algum aspecto. Por exemplo, considerando que os dois primeiros aminoácidos da proteína alvo tem certa energia, é feita uma comparação dessa energia com as possíveis energias das estruturas já existentes. Essas energias podem ser calculadas utilizando os mesmos potenciais utilizados nas técnicas *ab initio* (Shao et al., 2011). O pedaço de estrutura que possuir maior similaridade (de acordo com esse fator energético) será o candidato a assumir os dois primeiros aminoácidos, com a nova estrutura. Isso é repetido até que a sequência da proteína alvo seja inteiramente percorrida. Conforme novos trechos da proteína alvo são compostos, os novos modelos de proteína alvo já obtidos são avaliados até que se encontre o que melhor satisfaça as restrições energéticas.

Além do fator energético, existem outros fatores que podem ser utilizados para avaliar a qualidade das estruturas modeladas: 1) a similaridade de sequência da proteína alvo e das sequências das demais proteínas utilizadas; 2) o ambiente em que a proteína se encontra, que mede a viabilidade de alinhar um resíduo da proteína alvo no mesmo ambiente do resíduo específico da proteína alvo; 3) a consistência da estrutura, que mede a compatibilidade de regiões de estrutura local (estruturas secundárias) e global; e 4) o fator de penalização de lacunas em posições em que não se consegue informação das proteínas utilizadas para modelar, que evita a criação de espaços, possibi-

litando que resíduos menos favoráveis sejam alocados nessas tais posições. Pode-se ainda atribuir um peso para cada fator, dependendo do propósito de uso do método (Zaki & Bystroff, 2008).

A cada dois anos há uma competição chamada CASP (*Critical Assessment of protein Structure Prediction*) que envolve pessoas e métodos computacionais que tentam predizer estruturas de proteínas que ainda não foram depositadas no PDB. O I-TASSER (Wu et al., 2007) e o Quark (Xu & Zhang, 2012a) estão entre os métodos de *Threading* de maior destaque nesse contexto devido ao sucesso que têm obtido na competição CASP.

2.4 *Ab initio*

O método para PSP *ab initio* é caracterizado por não utilizar conhecimento *a priori* para prever estruturas de proteínas. A princípio, a partir da sequência de aminoácidos são calculados potenciais de campo de força (Seção 2.5) para avaliar conformações candidatas para uma proteína (estruturas tridimensionais) buscando identificar as mais plausíveis (de menor energia) e usar essas para se propor conformações melhores. O número de pesquisas sobre *ab initio* para PSP tem sido inferior aos métodos baseados em conhecimento *a priori*, pois, de fato, estes têm, em geral, apresentado resultados mais efetivos. De acordo com a avaliação do CASP10 (realizado em 2012), a quantidade de grupos de pesquisas relacionados à predição com métodos baseados em conhecimento *a priori* têm sido significativamente maior em relação à predição *ab initio*. Isso refletiu na quantidade de proteínas preditas no CASP10, em que mais de 40.000 estruturas de proteínas preditas foram submetidas utilizando métodos baseados em conhecimento *a priori*, enquanto que apenas algumas dezenas de proteínas preditas foram submetidas utilizando *ab initio* (Huang et al., 2014; Tai et al., 2014).

Os métodos *ab initio* mostram-se mais importantes quando a Homologia ou a similaridade (em termos das sequências ou outro fator de comparação) é baixa, restringindo o uso ou a qualidade das predições das abordagens que utilizam conhecimento *a priori*. Como a principal base de conhecimento *a priori* depende de cristais de proteínas (CRX, Seção 2.3.1), podem existir aspectos que sejam característicos das proteínas para as quais não se conseguiu cristais e que, por isso, ainda não foram observados ou têm sido estimados com baixa frequência. Em outras palavras, o universo de conformações de proteínas que os métodos baseados em conhecimento *a priori* utilizam pode não conter conformações ou subestruturas de proteínas com relevância estatística não desprezível.

Em geral, os métodos *ab initio* utilizam certa quantidade de conhecimento *a priori*, tais como: estatísticas do diagrama de Ramachandran e estatísticas de tóplas de aminoácidos como, por exemplo, o Rosetta, que utiliza 3 ou 9 aminoácidos (Rohl et al., 2004) ou o Quark, que utiliza de 1 a 20 aminoácidos para compor suas estatísticas (Xu & Zhang, 2012a). Alguns algoritmos utilizam bibliotecas de ângulos diedrais para restringir os ângulos diedrais ϕ , ψ e χ 's em regiões factíveis do espaço de busca. Neste trabalho, foi utilizado uma biblioteca de ângulos diedrais definida por

Tuffery (2003). Para essa biblioteca de ângulos diedrais utilizada neste trabalho, deu-se o nome de ADB (do inglês, *Angle DataBase*), utilizada em parte dos experimentos realizados.

Os algoritmos que utilizam potenciais de energia (Seção 2.5) e algum conhecimento *a priori*, também têm sido chamados de métodos semi-*ab initio* (como é o caso do Rosetta, I-TASSER, e Quark, que tem apresentado resultados relevantes no CASP (Wolff et al., 2010)), enquanto que os métodos baseados apenas nos potenciais de energia podem ser melhor caracterizados pelo termo puramente *ab initio* (Lee et al., 2009; Brasil et al., 2013).

O desenvolvimento de um método *ab initio* é um problema computacionalmente complexo (Setubal & Meidanis, 1997), sendo conhecido como um dos maiores problemas da biologia computacional ainda não resolvido (Gibas & Jambeck, 2001). A descrição das interações entre os átomos da proteína por meio dos Hamiltonianos (Friedman, 1981) mais utilizados podem não ser adequada para todos os tipos de proteínas, ou mesmo certos Hamiltonianos podem não modelar adequadamente interações presentes em conformações não usuais que podem ser geradas por métodos de otimização global (Brasil et al., 2013).

A avaliação de moléculas por Hamiltonianos tem se mostrado um problema complexo. Por exemplo, os vencedores do prêmio Nobel de Química de 2013 têm tentando compreender e avaliar as moléculas por meio de equações matemáticas (Karplus et al., 2013) que sintetizam suas principais interações há várias décadas. No trabalho de Brasil et al. (2013) foi realizado uma extensão para a energia de ligações de hidrogênio, favorecendo a formação de folhas- β em um algoritmo de otimização global *ab initio*.

Por outro lado, modelos mais simples, com menos graus de liberdade, possibilitam a convergência mais rápida que a de modelos relativamente complexos. Por exemplo, utilizando a representação *full-atom*, em que todos os átomos da proteína são considerados pelos potenciais de energia, requer um tempo de computação maior que o dos algoritmos que utilizam modelagem *lattice* (Wong et al., 2010), o qual representa cada aminoácido como um ponto no espaço, ao invés de cada átomo. No entanto, deve se observar que modelos mais simples, como o caso do *lattice* não representam adequadamente interações relevantes em certos tipos de proteínas (Chivian et al., 2003). Os modelos de água, que podem ser utilizados para simular o comportamento da proteína imersa em um solvente, também podem aumentar significativamente o tempo de computação do algoritmo de predição. Por exemplo, modelos de água explícito, em que é considerado a interação de todas as moléculas de água que circundam a proteína, exigem mais recursos computacionais em relação aos modelos implícitos (Seção 2.5.2), em que a água é considerada como um espaço contínuo (Zhou, 2003). A combinação de certos modelos em um algoritmo de predição pode inabilitar o seu uso devido ao tempo de computação requerido ser intratável. Por outro lado, uma combinação de modelos mais simples e computacionalmente menos custosos podem resultar em previsões ruins. Assim, o desenvolvimento de algoritmos de predição *ab initio* enfrentam o desafio de encontrar um compromisso na relação na relação conflitante entre qualidade dos modelos e eficiência computacional dos cálculos que tais modelos demandam.

2.4.1 Modelo *lattice*

O modelo *lattice* foi inicialmente proposto Shakhnovich et al. (1991) e em seguida estendido por Unger & Moult (1993). O termo *lattice* pode ser entendido como uma malha quadricular que possui quadrados regulares (2D) ou cubos regulares (3D). O modelo *lattice* é uma simplificação da representação da proteína do problema de PSP para um algoritmo que busca. No entanto, mesmo com essa simplificação, o problema de PSP permanece intratável (Mansour et al., 2010). No modelo *lattice*, os resíduos da proteína são representados por pontos no vértice da malha quadricular, sendo que os resíduos vizinhos na sequência de aminoácidos são vizinhos na malha quadricular com comprimento de ligação único (distância entre os vértices na malha). Algumas características de cada resíduo (Seção 2.1) podem ser utilizadas para avaliar uma conformação na *lattice* como, por exemplo, as interações polares. Nesse caso, a estrutura interna (cadeia lateral) dos resíduos não é considerada (Khimasia & Coveney, 1997).

Os EAs têm sido aplicados ao problema de PSP com modelagem *lattice* com representação discreta das variáveis, em que a posição de cada resíduo na *lattice* depende da posição do resíduo anterior, em um deslocamento na estrutura primária no sentido $N - C_{\text{terminal}}$, em que N corresponde ao átomo de hidrogênio do primeiro resíduo da estrutura primária e C_{terminal} refere-se ao átomo de carbono do último resíduo da estrutura primária. Existem também representações contínuas, em que os resíduos são representados pelas coordenadas Cartesianas absolutas. No entanto, mesmo utilizando representação discreta (relativamente simples) o problema ainda é NP-completo (Setubal & Meidanis, 1997).

O modelo HP (Lau & Dill, 1989) é um exemplo que utiliza a representação *lattice* para avaliar a proteína com base na característica de cada resíduo, em que os resíduos hidrofóbicos são representados por H e os polares (ou hidrofílicos) representados por P. Neste modelo, conformações de proteínas são geradas de forma a maximizar o contato de aminoácidos hidrofóbicos. Assim, vários algoritmos têm sido aplicados para tentar maximizar a quantidade de interações de aminoácidos vizinhos H-H (Hart, 1997). A qualidade dos resultados pode depender de outros fatores, além da quantidade de contatos. Em Gabriel et al. (2012) foi mostrado que é possível obter resultados mais relevantes se, ao invés de considerar somente os contatos H-H para avaliação de uma conformação, for considerada também a distância Euclidiana entre os aminoácidos hidrofóbicos.

2.4.2 Modelo *off-lattice*

O modelo *off-lattice* é capaz de representar as proteínas com maior grau de detalhamento (com relação ao modelo *lattice*), permitindo que os ângulos diedrais ϕ e ψ assumam valores do mapa de Ramachandran. Ao invés de representar a proteína como um conjunto de pontos por resíduo, o modelo *off-lattice* pode representar o C_α , tanto quanto com ou sem os átomos C e N . A cadeia lateral pode ou não ser incluída no modelo. Caso seja incluída, pode ser representada por apenas um átomo unificado ou todos os átomos (Bujnicki, 2009). Além disso, a cadeia lateral pode ser

definida como: 1) flexível, em que pode-se movimentar livremente; 2) semi-flexível, quando é utilizado conhecimento da cadeia lateral de outras estruturas e; 3) rígidas, é adotado o valor com maior frequência, baseado em cadeias laterais já observadas (Krane & Raymer, 2003).

Esse modelo possibilita certo compromisso entre sua qualidade e seu custo computacional. Porém, é importante observar que a complexidade dele é maior do que a do modelo *lattice* (que resulta na formulação de um problema de otimização NP-completo). Como exemplo, considere a representação *off-lattice* e que os ângulos diedrais ϕ e ψ possam apenas assumir quatro valores do mapa de Ramachandran. Considere também uma proteína com 100 resíduos. Então a quantidade de conformações possíveis será 4^{100} . A avaliação de todas essas conformações é impossível na prática. No entanto, se considerar que cada conjunto de 10 resíduos pode assumir apenas quatro valores, será necessário avaliar apenas 4^{10} conformações, isto é, cerca de um milhão, tornando o método baseado em modelo *off-lattice* adequado apenas para proteínas pequenas (Webster, 2000).

2.4.3 Modelo *full-atom*

Modelos *ab initio full-atom* geralmente representam a conformação da proteína utilizando ângulos diedrais, representados por um conjunto de quatro átomos conectados. Os ângulos diedrais da cadeia principal das proteínas são definidos como ϕ , o ângulo de torção entre os átomos $N - C_\alpha$; ψ , o ângulo entre $C_\alpha - C$ e ω , ângulo entre $C - N$. Os ângulos diedrais ϕ e ψ podem girar livremente e podem assumir valores de -180 a $+180$ graus, porém o ângulo diedral ω é, em geral, fixado em 180 graus, restringindo a distância entre $C_\alpha - C_\alpha$ em aproximadamente $3,8$ Å. A Figura 2.6 mostra um exemplo de cadeia principal contendo quatro aminoácidos e destaca os ângulos diedrais ϕ e ψ do segundo aminoácido, no sentido $N - C_{\text{terminal}}$.

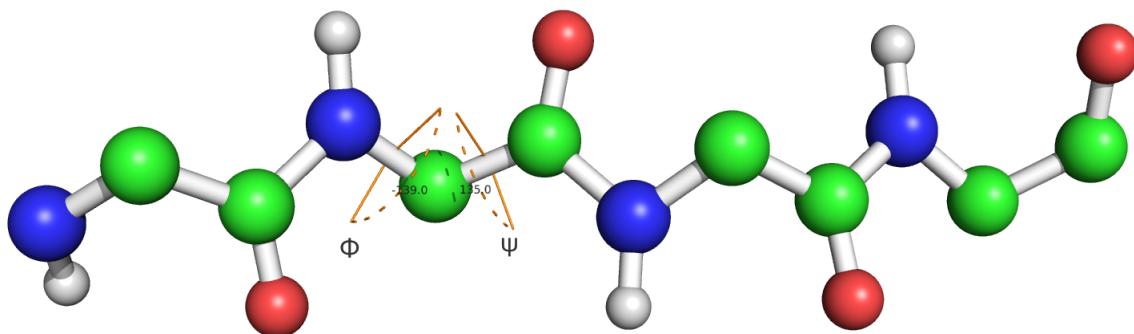


Figura 2.6: Identificação dos ângulos ϕ e ψ em uma cadeia principal com quatro aminoácidos.

Os ângulos diedrais da cadeia lateral são conhecidos como ângulos quirais e são denotados como $\chi_1 - \chi_4$, pois o número de ângulos varia de acordo com o tipo de aminoácido. Por exemplo, a cadeia lateral do aminoácido Lisina possui quatro ângulos diedrais, enquanto que o aminoácido Glicina não possui nenhum (ver Figura 2.7). O ângulo quiral χ_1 é definido pelo ângulo entre os átomos $C_\alpha - C_\beta$, χ_2 por $C_\beta - C_\gamma$, χ_3 por $C_\gamma - C_\delta$ e χ_4 por $C_\delta - C_\epsilon$. A Figura 2.7 mostra os

ângulos diedrais das cadeias laterais para os 20 aminoácidos e a Tabela 2.1 mostra a quantidade de ângulos diedrais por aminoácido.

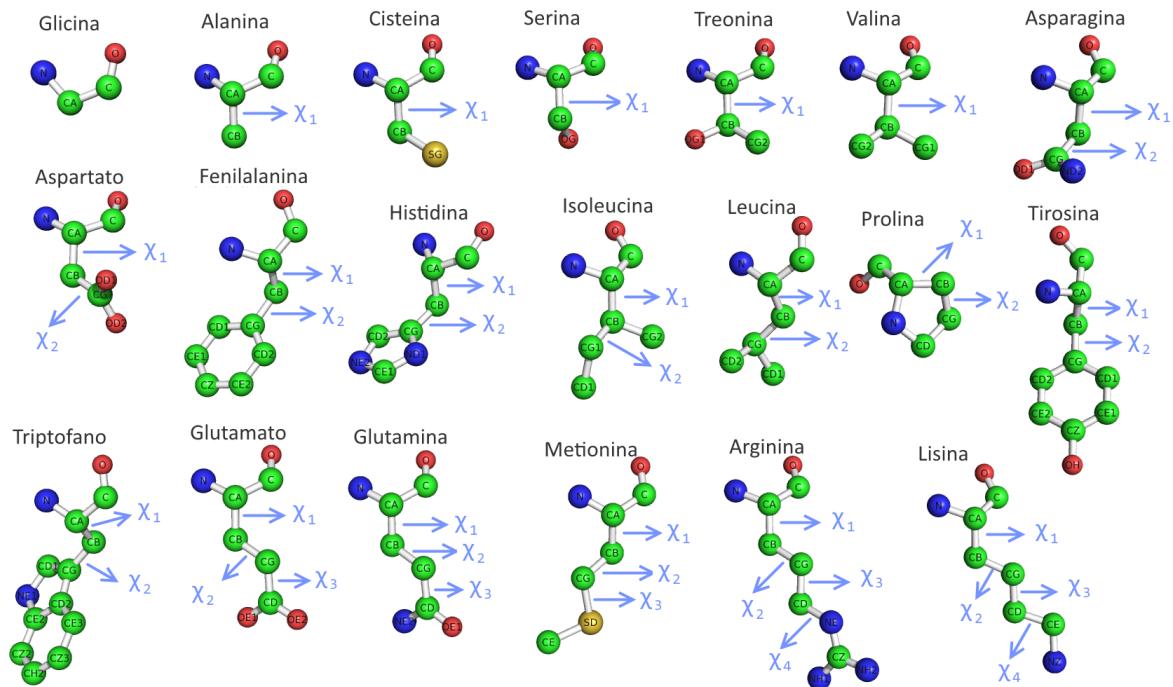


Figura 2.7: Identificação dos ângulos quirais das cadeias laterais para os 20 aminoácidos da Tabela 2.1. Considere $C_\alpha, C_\beta, C_\gamma, C_\delta, C_\epsilon$ como CA, CB, CD, CG e CE, respectivamente.

O conjunto dos valores de todos os ângulos diedrais de uma proteína reflete a estrutura terciária. Pequenas alterações nos valores desses ângulos podem alterar significativamente a conformação da proteína. O mapa de Ramachandran (Figura 2.8) mostra a região de pares de ângulos (ϕ, ψ) que são mais comuns de ocorrer. A Figura 2.8 também destaca as regiões onde há ocorrência de α -hélice, folha- β e a α -hélice invertida (menos comum na natureza).

O número de todas as conformações (possíveis estruturas) que uma proteína, mesmo relativamente pequena (com 80 ou menos aminoácidos), pode assumir com base apenas na estrutura primária e nos valores possíveis de ϕ e ψ pode não ser computacionalmente viável de ser avaliado. A complexidade computacional é de ordem $O(g^{2r})$, em que g é a quantidade de graus de liberdade dos ângulos diedrais e r é a quantidade de resíduos da estrutura primária. O valor $2r$ no expoente refere-se às duas variáveis relativas aos ângulos diedrais ϕ e ψ de cada resíduo, desconsiderando assim os ângulos quirais da cadeia lateral. Por exemplo, para uma proteína pequena com 200 aminoácidos e supondo por simplicidade que os ângulos ϕ e ψ possam assumir somente três valores (cada um relativo a uma das seguintes estruturas secundárias: α -hélice, folha- β e volta), o número de conformações existentes é 3^{400} , tornando impraticável o uso de uma busca exaustiva (Woeginger, 2003). Por essa razão, algoritmos aplicados ao problema de PSP *ab initio* utilizam algum tipo de estratégia que oriente o processo de busca em direção a regiões promissoras do espaço de busca.

Além da complexidade computacional alta devido a um espaço de busca de tamanho combinatorio, esse problema é altamente multimodal, uma vez que possui grande número de ótimos locais

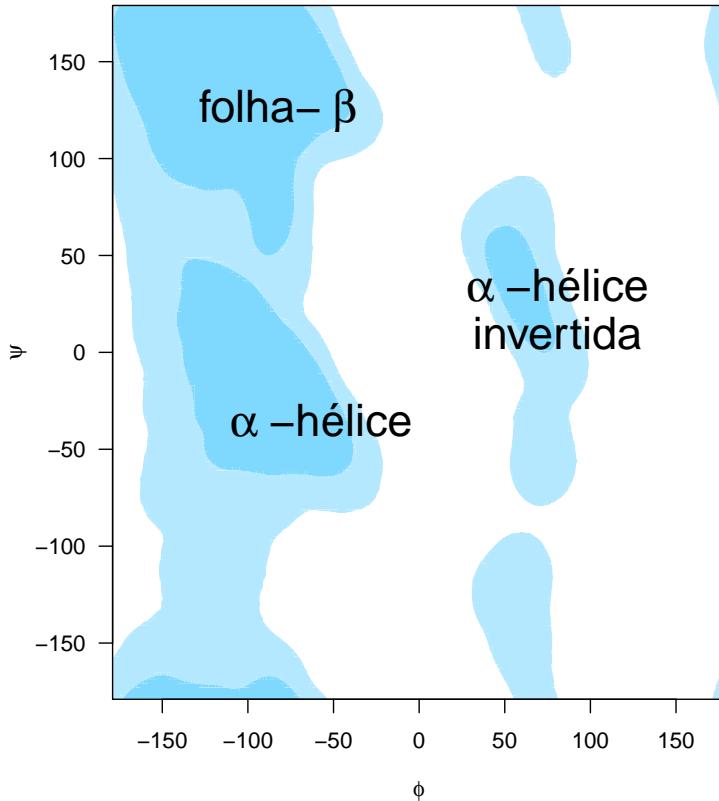


Figura 2.8: Diagrama de Ramachandran, mostrando a densidade de valores do par (ϕ, ψ) .

devido a existir, em geral, pontos de mínima energia associados aos potenciais envolvendo cada par de elementos da molécula. Esse aspecto multimodal e a larga escala do problema (r grande) em geral têm restringido o sucesso tanto dos métodos exatos de otimização global quanto das heurísticas quando aplicados à PSP *ab initio* com modelo *full-atom*.

Dessa forma, o sucesso dos métodos *ab initio* dependem da capacidade de heurísticas (Russell & Norvig, 2003) utilizadas para se explorar o espaço de busca, com o objetivo de encontrar a conformação com menor energia em um tempo de computação aceitável (Lee et al., 2009). Além de heurísticas, metaheurísticas têm sido abordagens investigadas para PSP. Nesse contexto, os Algoritmos Genéticos (Holland, 1975) foram uma das principais metaheurísticas investigadas para PSP (outras metaheurísticas são apresentadas no Capítulo 3). Todas as técnicas requerem uma estimativa da energia da proteína, geralmente utilizada como função objetivo (*fitness*) para um algoritmo de busca ou otimização. Com base nos valores de energia de cada conformação avaliada, o algoritmo pode distinguir as soluções mais plausíveis e orientar o processo de busca a regiões promissoras do espaço de busca. Assim, nota-se que tais funções desempenham um papel crucial no sucesso desses algoritmos. Os potenciais de energia utilizados como função objetivo neste trabalho são descritos na Seção 2.5.

2.5 Energia potencial da proteína

Encontrar a conformação da proteína (ou uma aproximação desta) com a menor energia livre requer o uso de modelos bem representativos das interações entre os átomos da proteína. Com modelos computacionais mais coerentes da representação das proteínas, os algoritmos de otimização global podem obter melhores estimativas para encontrar regiões promissoras no espaço de busca. Por exemplo, utilizando campos de força inadequados ou pouco representativos, os algoritmos de busca poderão minimizar a energia, no entanto, as conformações obtidas pelo algoritmo de otimização poderão ser diferentes da proteína que existe na natureza, conduzindo o processo de busca na direção errada. Por outro lado, algoritmos de otimização que utilizam modelos capazes de representar adequadamente o comportamento das proteínas irá guiar, em geral, o processo de busca na direção de soluções melhores.

Modelos baseados em Mecânica Quântica (MQ) são os que melhor descrevem o comportamento da interação de átomos (Greiner, 2001), pois seus modelos descrevem a energia em nível subatômico. A energia calculada por meio de tais modelos aplicados em proteínas deve corresponder a energia potencial próxima da energia da proteína nativa. No entanto, a MQ exige muitos recursos computacionais, sendo utilizada em geral para moléculas pequenas como a H_2O , entre outras.

O cálculo da energia potencial da proteína pode ser realizado utilizando a Mecânica Molecular (MM), que exige menos recursos computacionais por modelar a energia em nível molecular e também é capaz de representar a energia da proteína com certa precisão (Mackerell Jr, 2004). Há também modelos híbridos chamados MQ/MM em que ambos os aspectos de interesse em relação as abordagens MQ (precisão) e MM (eficiência computacional) são combinadas. A abordagem híbrida foi inicialmente proposta por (Warshel & Levitt, 1976), que juntamente com Martin Karplus ganharam o prêmio Nobel em Química em 2013 (Karplus et al., 2013).

Neste trabalho, é utilizado a abordagem da MM, em que potenciais de energia em nível molecular são combinados para obter uma estimativa da energia potencial da proteína, isto é, a energia total da proteína originada por meio da soma das energias relativa a cada potencial. Cada tipo de energia possui uma certa contribuição para a energia potencial da proteína.

As energias envolvidas em uma proteína podem ser classificadas em ligação covalentes e não-covalentes. As energias de ligação covalente entre átomos são calculadas para cada átomo da molécula e os átomos a uma vizinhança envolvida, de dois até quatro átomos. Essas energias podem ser: energia imprópria E_{imp} , energia de comprimento E_{comp} , energia de ligação E_{lig} , energia de torção E_{tor} e Urey-Bradley E_{ub} (Friesner, 2002). As energias para átomos que não possuem ligação covalente são calculadas para todas as combinações de pares de átomos da molécula e, por conta disso, exigem maior tempo computacional. São elas: energia de van der Waals E_{vdw} , energia eletrostática E_{chg} , energia de solvatação E_{sol} e energia de pontes de hidrogênio E_{hbond} (Nelson & Cox, 2004). A energia potencial da proteína pode ser obtida pela soma das energias covalentes e não-covalentes, como mostra a Equação 2.1:

$$E_{total} = E_{imp} + E_{comp} + E_{lig} + E_{tor} + E_{ub} + E_{vdw} + E_{sol} + E_{chg} + E_{hbond}. \quad (2.1)$$

As distâncias entre átomos vizinhos ligados de forma covalente varia relativamente pouco para diferentes conformações de uma proteína. Assim, energias de ligação covalente têm contribuição similar para a energia total da proteína em conformações significativamente diferentes. Por outro lado, as energias de ligação não-covalentes possuem maior contribuição para a estabilidade da proteína. Por exemplo, sabe-se que a energia de van der Waals (Seção 2.5.1) pode contribuir com até 65% da energia total da proteína. A energia de solvatação (Seção 2.5.2) é importante para modelar a interação da proteína com o solvente, pois sabe-se que a maioria das proteínas estão imersas em meio aquoso.

É necessário que, além da escolha dos potenciais de energia, seja também utilizado um conjunto de parâmetros, que em geral são obtidos por MQ e dados experimentais. Esses parâmetros podem ser relativamente complicados de estimar. Por exemplo, é realizado um estudo das características da interação entre dois átomos de carbono e logo em seguida é feito um ajuste de parâmetros de uma função que busca representar um tipo de interação relevante entre eles. No entanto, esses parâmetros variam não somente pelas características desses dois átomos. A presença de outros átomos ao redor deles também afeta o modelo ajustado, conduzindo a erros que podem não ter sido tratados adequadamente nessa modelagem.

Uma outra abordagem utiliza parâmetros específicos para cada tipo de átomo, conduzindo a um número maior de parâmetros de forma a representar apropriadamente suas características e também vários tipos de interação entre os átomos. Os conjuntos de parâmetros mais comuns são: CHARMM (Brooks et al., 1983; MacKerell Jr et al., 2000), AMBER (Cornell et al., 1995), GROMOS (Scott et al., 1999) e OPLS (Jorgensen & Tirado-Rives, 1988). O uso desses conjuntos de parâmetros depende da precisão exigida e também da aplicação em questão. Neste trabalho é utilizado o conjunto de parâmetros CHARMM, apropriado para proteínas e que tem sido empregado em outros métodos desenvolvidos no laboratório do grupo de pesquisa no ICMC-USP. É importante destacar que o uso de diferentes conjuntos de parâmetros utilizados no problema de minimização da energia da proteína podem produzir resultados significativamente diferentes (Mijajlovic et al., 2010).

2.5.1 Energia de van der Waals

A energia de van der Waals modela a atração e repulsão entre átomos. Em geral, é utilizado o potencial de Lennard-Jones (também conhecido como Lennard-Jones 12 – 6) para calcular a energia de van der Waals. O tipo do átomo e a distância entre eles é um dos fatores que influenciam no valor da energia. A Equação 2.2 descreve a relação entre dois átomos i e j dado a distância Euclidiana $d_{E_{ij}}$ e as constantes do raio de van der Waals R , conforme a Equação 2.2. O raio de van der Waals é a representação de um átomo por uma esfera com um raio, em que o raio depende de cada tipo do átomo (Bondi, 1964).

$$q_{ij} = \frac{d_{E_{ij}}}{R_i + R_j}. \quad (2.2)$$

A pequenas distâncias, a energia de van der Waals é bem repulsiva, pois a nuvem de elétrons (região ao redor do núcleo do átomo em que os elétrons circulam) entre o par de átomos começa a se sobrepor. Quando a distância entre dois pares de átomos é pequena, o potencial cresce rapidamente, tendendo a infinito. Por outro lado, quando átomos de um determinado par estão bem afastados um do outro, não há praticamente nenhum tipo de interação. Nesse caso, o potencial tende a zero. Para aumentar a eficiência computacional em geral é definido um raio de corte, chamado *cutoff*, em 8 Å (Cui et al., 1998), que evita a computação desnecessária para pares de átomos que não possuem interação significativa. É importante destacar que, diferentes raios de cortes têm sido propostos na literatura (Lagüe et al., 2004; Klauda et al., 2007). Para evitar lidar com números grandes que podem resultar em dificuldades de representação numérica em computadores, foi definido um valor de *tapering-off*, que limita o potencial a uma constante $C > 0$, se $r_{ij} \leq 0,8$. O potencial de Lennard-Jones possui um ponto de equilíbrio, conhecido como contato de van der Waals, que corresponde a um valor mínimo.

O potencial de Lennard-Jones (Jones, 1924) utilizado em PSP pode ser descrito pela Equação 2.3.

$$f_{LJ}(q_{ij}) = \begin{cases} A_{vdw}q_{ij}^{-12} - B_{vdw}q_{ij}^{-6} & \text{if } q_{ij} > 0,8, \\ C & \text{if } q_{ij} \leq 0,8, \end{cases} \quad (2.3)$$

em que A_{vdw} e B_{vdw} são os fatores multiplicativos do termo de repulsão e atração do potencial, definidos em 1 e 2, respectivamente, e C é dado por $A_{vdw}q_{ij}^{-12} - B_{vdw}q_{ij}^{-6}$ com $q_{ij} = 0,8$. A Figura 2.9 mostra o potencial de Lennard-Jones ajustada para estimar a energia de van der Waals em PSP.

A energia de van der Waals pode ser definida como a soma dos potenciais de Lennard-Jones entre todos os pares de átomos da molécula. Isso resulta em $\frac{a^2-a}{2}$ interações, em que a é o número de átomos, conforme mostra a Equação 2.4.

$$E_{vdw} = \sum_{i=1}^{a-1} \sum_{j=i+1}^a f_{LJ}(q_{ij}). \quad (2.4)$$

Assim, para encontrar a menor energia de van der Waals de uma molécula é necessário balancear as distâncias entre todos os pares de átomos de forma a atingir a menor energia global. O tempo computacional de um algoritmo que calcula a energia de pares de átomos com ligação não-covalentes pode ser reduzido com base no *cutoff*. Ao invés de utilizar um *loop* duplo em um algoritmo (devido aos dois somatórios na Equação 2.4) para a soma das energias parciais, é possível utilizar uma estrutura eficiente para o cálculo da energia de van der Waals, reduzindo a complexidade computacional de $O(a^2)$ para $O(a)$ conforme proposto em Bonetti et al. (2013).

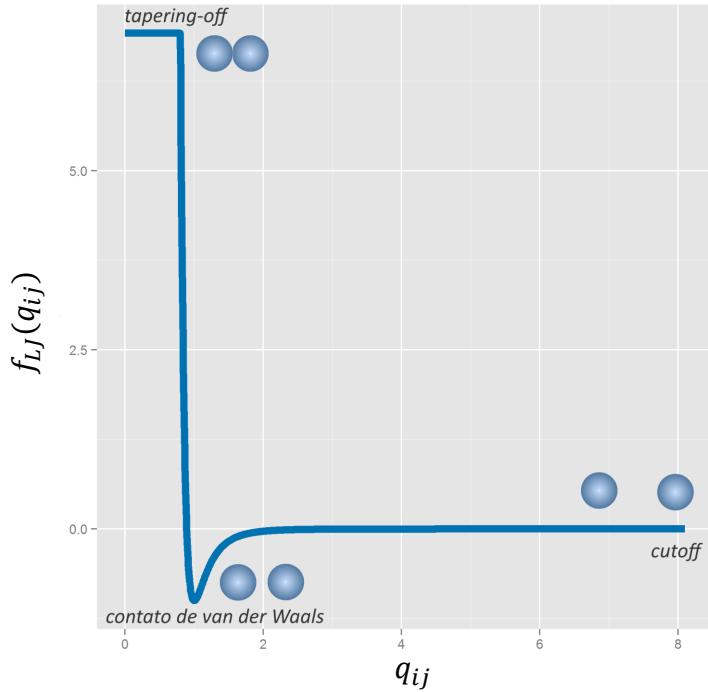


Figura 2.9: Energia potencial de Lennard-Jones modificada para PSP.

Para isso, a proteína que se deseja calcular a energia é colocada em uma estrutura de grade tridimensional dividida em células cúbicas. Se a distância máxima entre dois átomos dentro de uma célula for menor que o *cutoff* da energia de van der Waals (8 \AA) é possível garantir que um átomo dentro de uma célula apenas irá interagir com os átomos da própria célula e das células diretamente vizinhas. Assim, calculando as energias parciais apenas para os átomos que pertencem a mesma célula e as células vizinhas é possível reduzir significativamente o tempo de processamento, evitando o cálculo de interações de células distantes e, assim, reduzindo efetivamente a complexidade computacional para $O(a)$.

2.5.2 Energia de solvatação

Em geral as proteínas (sólido) na natureza estão envolvidas por moléculas de água (solvente). Este meio também contribui para a estabilização da proteína. Essa contribuição pode ser contabilizada por meio da energia de solvatação, que estima a energia da interação sólido-solvente. Um modelo de solvente que requer menos cálculos supõe que o solvente é um meio contínuo e uniforme. Essa modelagem do solvente é chamada de implícita, contrapondo a explícita, em que a interação com cada molécula de água é individualmente calculada. A energia de solvatação do modelo implícito pode ser determinada com base no cálculo da área de superfície de acessibilidade (SASA, do inglês, *Solvent Accessible Surface Area*), que é a área que o solvente utiliza para interagir com o sólido. Considere o sólido como um conjunto de esferas cujos centros coincidem com os centros de cada átomo que compõe a molécula e que o raio de cada átomo seja igual ao raio de van der Waals (Seção 2.5.1) do tipo de átomo que a esfera representa, conforme mostra a

Figura 2.10. O solvente corresponde a uma esfera (de água), que é rolada por todo o perímetro do soluto (conforme o sentido das setas na Figura 2.10). A superfície da área determinada pelo percurso do centro da esfera rolada é chamada de SASA (Hermann, 1972) no caso bidimensional. A extensão para três dimensões é esclarecida na apresentação da proposta de Gaudio & Takahata (1992).

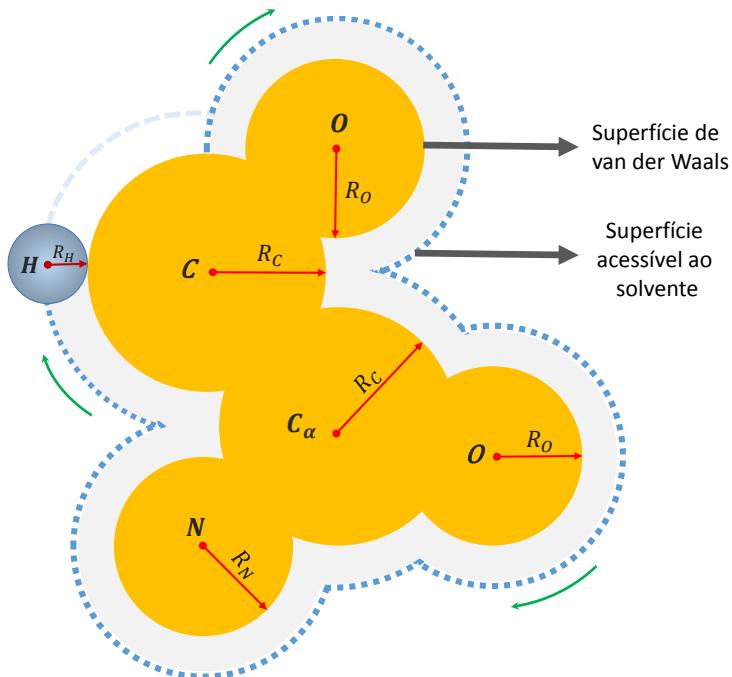


Figura 2.10: Representação da SASA e do soluto utilizando o raio de van der Waals. A esfera representa o solvente, que percorre a superfície do soluto calculando a SASA. No exemplo desta figura, a esfera ainda precisa percorrer um quarto do último átomo da proteína para completar o percurso.

Na proposta de Gaudio & Takahata (1992), primeiramente, acomoda-se o soluto em uma espécie de grade tridimensional dividindo-a em cubos com dimensões múltiplas de 2 Å. Tal divisão é chamada de nível 1. Em seguida, cada cubo é classificado em interno, externo ou com superfície para molécula. Os cubos do tipo interno ou externo são desconsiderados. Cada uma das dimensões dos demais cubos é dividida ao meio, resultando em 8 novos cubos de lado 1 Å no lugar desses cubos de lado 2 Å. Os novos cubos pertencem ao chamado nível 2. O mesmo processo de divisão é repetido produzindo 8 novos cubos de lado 0.5 Å, chamados de nível 3. O processo de divisão pode continuar até atingir uma precisão satisfatória. Assim, a precisão do cálculo da área da superfície molecular aumenta conforme o número de níveis aumenta, pois será menor a superfície dos cubos, portanto, aproximando mais da superfície da molécula de proteína que cada cubo representa (Gaudio & Takahata, 1992). A Figura 2.11 mostra um exemplo em duas dimensões do procedimento para o cálculo da superfície da molécula utilizando os níveis 1, 2, 3 e 4.

Para manter um equilíbrio entre precisão e tempo computacional os cálculos em geral limitam-se até o nível 5. Sabe-se que a área interceptada média $SASA$ para o átomo i é linearmente proporcional a uma única face da superfície de um cubo $A_{cubeside}$, variando de acordo com nível

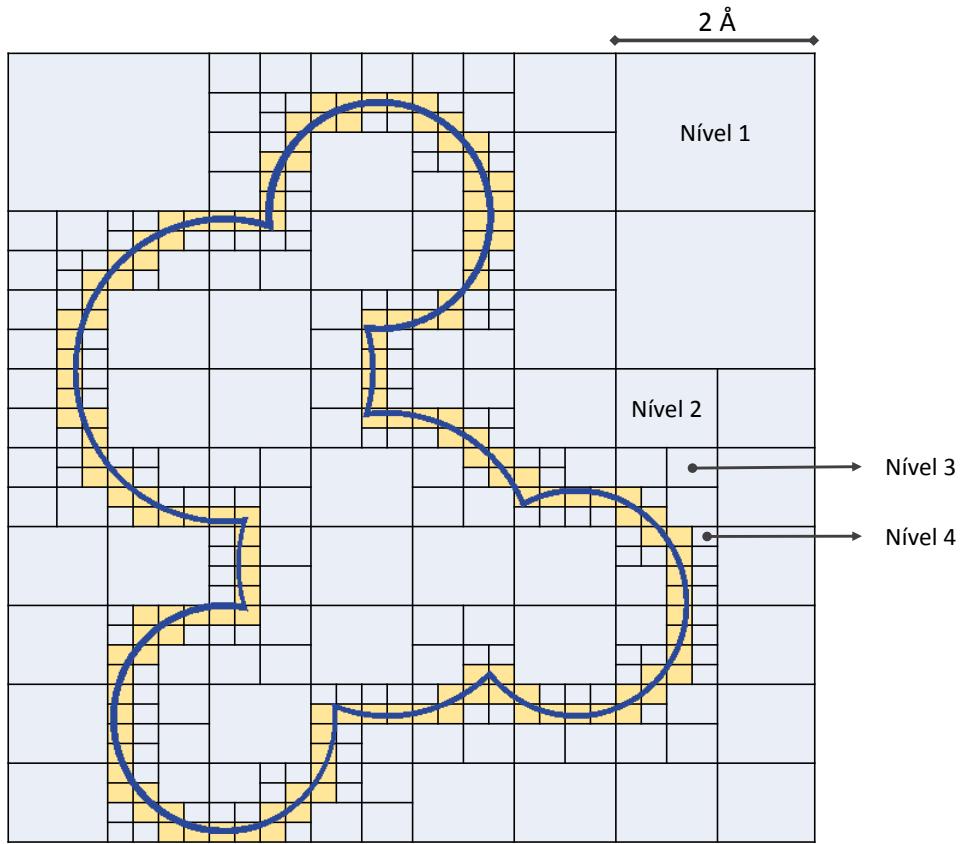


Figura 2.11: Cálculo da superfície da molécula. A molécula é colocada em um reticulado composto por cubos de lado 2 Å no nível 1. Cada cubo que contém a superfície da molécula tem seus lados divididos ao meio. O processo de divisão continua até atingir o nível desejado, no caso, até o nível 4.

e o tipo do átomo. Dessa forma, Gaudio & Takahata (1992) determinaram fatores numéricos que são específicos de cada átomo e de cada nível. Chamados f_{sasa} esses fatores são utilizados para calcular a área do cubo interceptada $A_{intercept}$ pela molécula, conforme descrito pela Equação 2.5.

$$A_{intercept} = f_{sasa} A_{cubeside}. \quad (2.5)$$

Assim, a área total da superfície $SASA$, pode ser calculada por todas as frações computadas $A_{intercept}$, conforme a Equação 2.6.

$$SASA = \sum_{i=1}^{N_c} A_{intercept}(i), \quad (2.6)$$

em que $A_{intercept}(i)$ é a área interceptada pelo átomo i e N_c é o número de cubos que interceptam a superfície da molécula.

A partir dos valores de $SASA$ é possível calcular a energia de solvatação E_{sol} da molécula. Um conjunto de parâmetros de solvatação (ASP, do inglês, *Atomic Solvation Parameters*) é utilizado para determinar os parâmetros de cada átomo denotado por $ASP(x)$, em que x é o tipo do átomo.

A energia de solvatação pode então ser estimada multiplicando-se os parâmetros relacionados a cada átomo na superfície e com a área correspondente *SASA*. Considerando somente os átomos da cadeia principal (C, N e O) pode-se utilizar os valores definidos por Eisenberg & McLachlan (1986), em que $ASP(C) = 16$ e $ASP(N) = ASP(O) = -6$. Por fim, a Equação 2.7 determina como é calculado a energia de solvatação, considerando a o número de átomos.

$$E_{sol} = \sum_{i=1}^a ASP(i)SASA_i, \quad (2.7)$$

2.6 Considerações finais

Neste capítulo foi apresentado os conceitos básicos da composição das estruturas de proteínas e algumas das dificuldades principais presentes na PSP de forma puramente *ab initio*, isto é, a partir somente da sequência de aminoácidos. Vários métodos têm sido investigados para se predizer estruturas de forma mais eficiente. No entanto, o número de estruturas que se consegue predizer com esse tipo de abordagem é relativamente pequeno se for considerado a taxa de crescimento entre sequências e estruturas, principalmente no caso puramente *ab initio*.

Considerando os aspectos discutidos neste capítulo sobre os métodos computacionais para PSP, verifica-se que a abordagem *ab initio* pode contribuir para um melhor entendimento do problema de PSP, em geral, para proteínas com regiões de baixa similaridade.

A melhoria dos métodos de PSP puramente *ab initio* para se obter algoritmos mais eficientes e eficazes pode requerer diversas linhas de investigação: (1) melhoria das técnicas experimentais, (2) determinação de campos de força da energia que possibilitem menos cálculos e sejam mais representativos, (3) desenvolvimento de heurísticas e metaheurísticas que sejam capazes de investigar melhor o espaço de busca de conformações de proteínas, bem como do espaço de objetivos (Brasil et al., 2013) relativo às energias consideradas como a energia de van der Waals (Seção 2.5.1) e energia de solvatação (Seção 2.5.2).

Neste trabalho, foca-se em avançar em relação à contribuição que heurística e metaheurística podem gerar em PSP puramente *ab initio*, conforme apresentado no Capítulo 3.

Algoritmos Evolutivos e Metaheurísticas

Este capítulo apresenta uma introdução sobre Algoritmos Evolutivos e metaheurísticas com destaque para Algoritmos de Estimação de Distribuição (EDAs) para variáveis contínuas. Além disso, busca-se avaliar os aspectos qualitativos deles que podem beneficiar a otimização utilizando EDAs em relação ao uso de outras metaheurísticas. A Seção 3.1 mostra as principais características presentes em Algoritmos Evolutivos. A Seção 3.2 descreve algumas metaheurísticas baseadas em princípios significativamente diferentes. As Seções 3.3.2 e 3.3.3 descrevem EDAs que têm se destacado para problemas de variáveis discretas e contínuas. A Seção 3.4 apresenta EDAs para problemas hierárquicos. Por fim, a Seção 3.5 é dedicada às considerações finais deste capítulo.

3.1 Algoritmos Evolutivos

Na natureza, o processo de evolução das espécies ocorre por meio da seleção e de algumas mudanças aleatórias. Os Algoritmos Evolutivos (EAs, do inglês, *Evolutionary Algorithms*) (Gaspar-Cunha et al., 2012) são algoritmos de otimização global inspirados na evolução natural das espécies. Basicamente um EA consiste das seguintes etapas: (1) criação da população inicial, (2) avaliação, (3) seleção, (4) geração de novas soluções e (5) substituição de indivíduos ruins. Todas essas etapas são explicadas na sequência. Em EA, um indivíduo representa uma solução candidata para o problema e um conjunto de indivíduos representa uma população. A Figura 3.1 ilustra o funcionamento de um EA.

A população inicial (Etapa 1) é geralmente iniciada aleatoriamente utilizando uma distribuição uniforme com intervalo definido pela faixa de valores que o problema pode assumir. Considere uma população definida pela matriz P_j^i ($j = 1 \dots d$, d o número de dimensões do espaço de busca de um problema) e i um indivíduo (solução candidata) com $i = 1 \dots n$, em que n é o número de indivíduos. Cada indivíduo i é avaliado por uma função chamada *fitness* na Etapa 2, que é utilizada para medir a qualidade de cada indivíduo (quanto maior o *fitness* melhor é o indivíduo). A complexidade e a quantidade de recursos necessários para calcular o *fitness* de um indivíduo depende de cada problema e pode ser o gargalo em termos de tempo de computação do algoritmo.

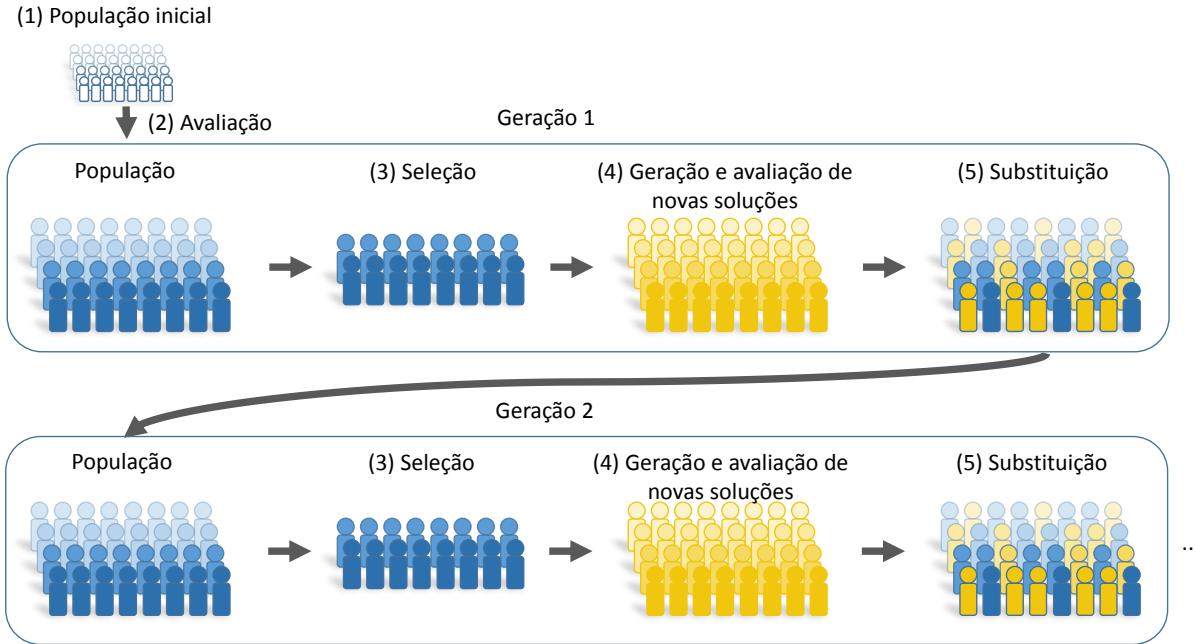


Figura 3.1: Etapas de um EA típico.

Existem vários operadores de seleção (Etapa 3) que podem ser utilizados em um EA para determinar um conjunto dos indivíduos selecionados (Goldberg, 2002). Um dos operadores de seleção é o torneio, em que cada indivíduo da população de selecionados é o vencedor de uma competição, que consiste em sortear t indivíduos ($t \geq 2$) da população atual e escolher o de melhor *fitness*. Outro operador comum é a seleção por truncamento que ordena a população pelo *fitness* e utiliza os τ melhores indivíduos como selecionados. Assim, considerando $\tau = 0,5$ somente os 50% melhores indivíduos serão selecionados. Outro operador é a seleção proporcional, como o método da roleta, em que cada indivíduo recebe uma probabilidade diretamente proporcional ao seu valor do *fitness* (Bäck, 1996).

O procedimento de geração de novas soluções (Etapa 4) é um dos principais diferenciais entre os vários EAs. Nesta etapa, os indivíduos que foram selecionados podem ser modificados de forma a produzir novos indivíduos. As estratégias para geração de novos indivíduos estão diretamente relacionadas com a capacidade do EA em procurar por regiões promissoras do espaço de busca. Alguns EAs utilizam operadores de recombinação e mutação (Seção 3.2.3), enquanto outros utilizam modelos mais sofisticados para geração de novos indivíduos. Por exemplo, existem

Algoritmos Genéticos que apenas fazem recombinação de um ponto (Seção 3.2.3), enquanto que outros podem utilizar recombinação envolvendo mais pontos, além de sofisticados esquemas de mutação. Os indivíduos gerados (também chamado de filhos) podem ser representados pela matriz O (do inglês, *Offspring*), que possui f linhas (número de indivíduos a serem gerados) e d colunas (número de variáveis do problema). Assim, o valor da variável j do novo indivíduo gerado i pode ser escrito na forma O_j^i .

O último passo de um EA é a substituição de indivíduos da população pelos novos indivíduos (Etapa 5). Há diversas estratégias para realizar essa substituição. O truncamento, mescla a população P e os filhos O na forma $M = P \cup O$. Os indivíduos de M são ordenados de acordo com o *fitness* e os n melhores indivíduos de M (Figura 3.2(a)) compõe a nova população. Outra maneira de substituição de indivíduos da população é definir $f = n$, trocando a população inteira pelos filhos gerados, de forma que a nova população é igual a O (Figura 3.2(b)). No entanto, esse tipo de substituição pode resultar em perda dos melhores indivíduos encontrados até então. O conjunto das Etapas 1-5 é conhecido como uma geração, ou iteração de um EA, e pode ser sintetizado como mostra o Algoritmo 1.

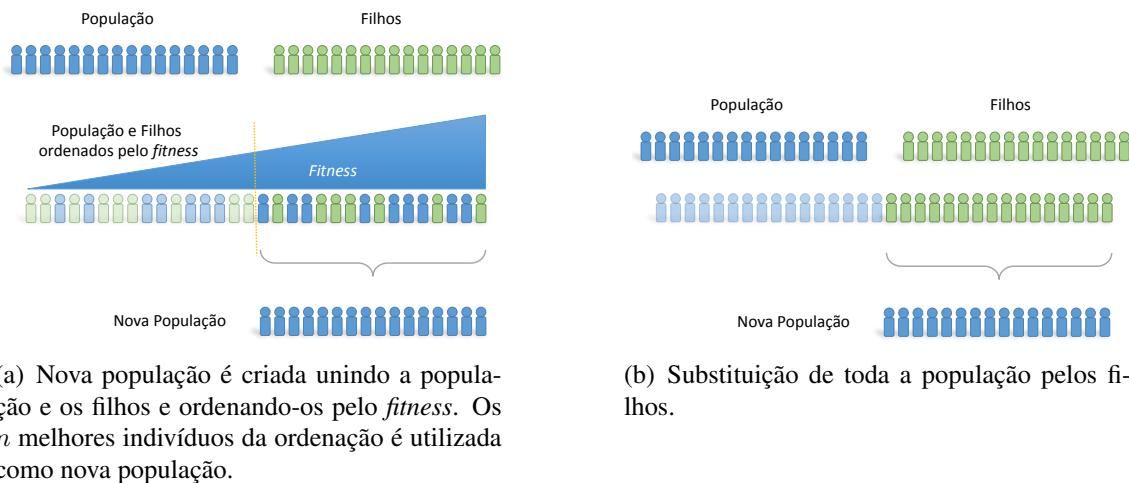


Figura 3.2: Métodos de substituição da população pelos filhos.

A ideia básica é que em cada geração de um EA as soluções na população sejam cada vez melhores. Considerando que a população inicial seja aleatória, o número de gerações necessárias para encontrar soluções ótimas (de alta qualidade) depende da complexidade do problema, além de parâmetros do EA, como o tamanho da população, pressão de seleção realizada pelo torneio (ou outro operador de seleção), e dos tipos de operadores de reprodução utilizados na geração das soluções (Gaspar-Cunha et al., 2012). Apesar de o tempo total de execução de um EA estar relacionado com o número de gerações, uma das medidas de desempenho e comparação utilizadas em EAs é o número de chamadas à função de avaliação (*fitness*) que, em geral, é a etapa que requer maior tempo de computação. Esse número indica também quantos pontos (soluções candidatas) do espaço de busca foram explorados. Assim, um dos critérios de parada de um EA pode ser

estabelecido quando um determinado número de avaliações for atingido. Outro critério pode ser a convergência da população, o que pode ser medido pelo desvio padrão do *fitness* da população.

Algoritmo 1: Pseudocódigo de um Algoritmo Evolutivo

```

1:  $P' \leftarrow$  Inicializar aleatoriamente uma população
2:  $P \leftarrow$  Avaliar a população inicial  $P'$ 
3:  $g \leftarrow 1$ 
4: while Critério de convergência não é atingido do
5:    $S \leftarrow$  Aplicar método de seleção a partir de  $P$ 
6:    $O' \leftarrow$  Utilizar os indivíduos de  $S$  para gerar os filhos
7:    $O \leftarrow$  Avaliar os filhos  $O'$ 
8:    $P \leftarrow$  Aplicar método de substituição da população em  $P \cup O$ 
9:    $g \leftarrow g + 1$ 
10: end while
```

A seguir são apresentados alguns métodos de busca. Esses métodos foram utilizados neste trabalho como referência para avaliação do desempenho do Algoritmo de Estimação de Distribuição para PSP proposto nesta tese.

3.2 Métodos de busca de referência

Esta seção apresenta as técnicas Busca Aleatória (Seção 3.2.1), Monte Carlo (Seção 3.2.2), Algoritmo Genético (Seção 3.2.3) e Evolução Diferencial (Seção 3.2.4). Esses métodos são utilizados como referência para comparação com o Algoritmo de Estimação de Distribuição proposto apresentado na Seção 3.3.

3.2.1 Busca Aleatória

A Busca Aleatória (RW, do inglês, *Random Walk*) foi inicialmente proposta por Pearson (1905) e tem sido utilizada em várias áreas. É também conhecida como o “andar do bêbado”, pois a partir de um ponto inicial, sucessivos passos são dados em direções aleatórias.

A ideia original da RW foi elaborada a partir de um exemplo que considera uma pessoa parada no ponto inicial B que caminha g metros em linha reta. Em seguida, essa pessoa vira para qualquer direção e caminha mais g metros. Após repetir esse procedimento de alterar a direção e caminhar um certo número de vezes, a pessoa estará a uma distância r_{rw} do ponto inicial B , que é aleatória. Isso quer dizer que a caminhada aleatória pode produzir soluções bem diversificadas.

Essa ideia pode ser estendida para um algoritmo de otimização em que novas soluções podem ser geradas completamente aleatórias, dentro do intervalo definido pelas variáveis do problema. De certa forma, o procedimento de gerar soluções aleatórias possui relação com a geração da população inicial de um EA. Por exemplo, um milhão de avaliações do algoritmo RW tem praticamente o

mesmo efeito da geração da população inicial de um EA para a primeira geração, com um milhão de indivíduos.

3.2.2 Monte Carlo

O método de Monte Carlo (Metropolis & Ulam, 1949) foi inicialmente proposto para lidar com problemas matemáticos, utilizando uma abordagem estatística para estudar equações diferenciais. A ideia do MC é semelhante ao fato de fazer apostas e armazenar os resultados durante um jogo de cassino na vida real. Inicialmente, esse método foi estendido para a área de física nuclear buscando o desenvolvimento de armas. O MC consiste em obter amostras aleatórias repetidamente e depois analisar o resultado. Com o surgimento dos primeiros computadores esse princípio tornou-se interessante, pois simulações computacionais baseadas no MC podiam ser realizadas para obter aproximações numéricas de funções.

Um dos exemplos mais comuns que utilizam o método de MC é o cálculo do valor de π , que, por ser irracional, não pode ser escrito na forma de uma razão. Considere um círculo inscrito em um quadrado de lado l . Dentro desse quadrado n_p pontos são gerados aleatoriamente. Para cada ponto gerado é verificado se o ponto está dentro ou fora do círculo. Se a distância de determinado ponto até o ponto de origem (centro do círculo) for menor ou igual ao raio do círculo é considerado como ponto dentro do círculo, incrementando o contador h_{mc} . Assim, a probabilidade de um ponto estar dentro do círculo é dada por $\frac{h_{mc}}{n_p}$. Considerando a área do círculo como πl^2 e a área do quadrado $4l^2$ tem-se a razão definida pela Equação 3.1.

$$\frac{\pi l^2}{4l^2} = \frac{\pi}{4}. \quad (3.1)$$

Considerando que a probabilidade obtida pela Equação 3.1 é igual a $\frac{h_{mc}}{n_p}$ pode-se reescrever π pela Equação 3.2.

$$\frac{h_{mc}}{n_p} = \frac{\pi}{4} \implies \pi = \frac{4h_{mc}}{n_p}. \quad (3.2)$$

Assim, quanto mais pontos n_p forem amostrados aleatoriamente, melhor será a estimativa do valor de π utilizando o método de MC. A Figura 3.3 mostra um exemplo para o cálculo do valor de π utilizando MC em que $n_p = 5.000$.

Existem de certa forma contradições com relação ao uso do nome Monte Carlo na literatura. Para alguns autores, MC é a modelagem de simulações estocásticas (Ripley & Corporation, 1987) enquanto que outros autores consideram como uma simulação fictícia da realidade (Sawilowsky & Fahoome, 2002).

O algoritmo de Metropolis-Hastings é uma variação do MC que utiliza um critério de aceitação/rejeição para uma amostra. Em alguns casos, mesmo que uma nova solução amostrada possua qualidade inferior da qual tal solução se originou, a solução inferior pode ser aceita. Em otimização, isso pode ser um mecanismo útil para o algoritmo sair de ótimos locais. O Algoritmo 2 mostra

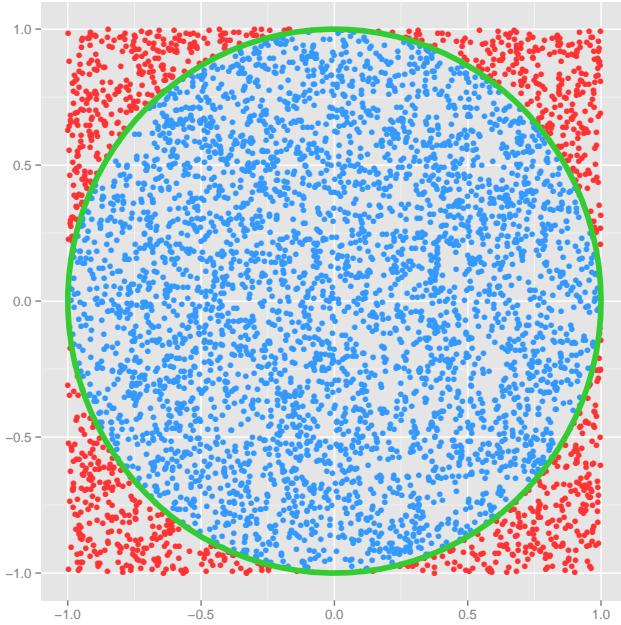


Figura 3.3: Método de MC aplicado para aproximar π . Neste exemplo, o círculo inscrito no quadrado tem raio $l = 1$ e origem em $[0, 0]$, $n_p = 5.000$ e $h_{mc} = 3.908$, então $\pi \approx 3,1264$ (Equação 3.2).

um pseudocódigo do método de MC com o critério de aceitação/rejeição (linha 9 do pseudocódigo) aplicado para geração de números aleatórios. Considere a Função Densidade Probabilidade para uma distribuição normal $pdf(x)$ com média 0 e desvio padrão 1 definida pela Equação 3.3.

Algoritmo 2: Monte Carlo para geração de números aleatórios.

```

1:  $v \leftarrow$  Inicializar vetor de zeros de tamanho  $n_p$ 
2:  $x \leftarrow 0$ 
3:  $v_1 \leftarrow x$ 
4: while  $i = 2$  to  $n_p$  do
5:    $y \leftarrow$  Gerar um número com distribuição uniforme  $U(-0,5; +0,5)$ 
6:    $s \leftarrow x + y$ 
7:    $p_m \leftarrow$  Mínimo entre 1 e  $\frac{pdf(s)}{pdf(x)}$ 
8:    $u \leftarrow$  Gerar um número com distribuição uniforme  $U(0; 1)$ 
9:   if  $u < p_m$  then
10:     $x \leftarrow y$ 
11:   end if
12:    $v_i \leftarrow x$ 
13: end while
```

$$pdf(x) = \frac{1}{\sqrt{2\pi}} e^{\frac{x^2}{2}}. \quad (3.3)$$

3.2.3 Algoritmo Genético

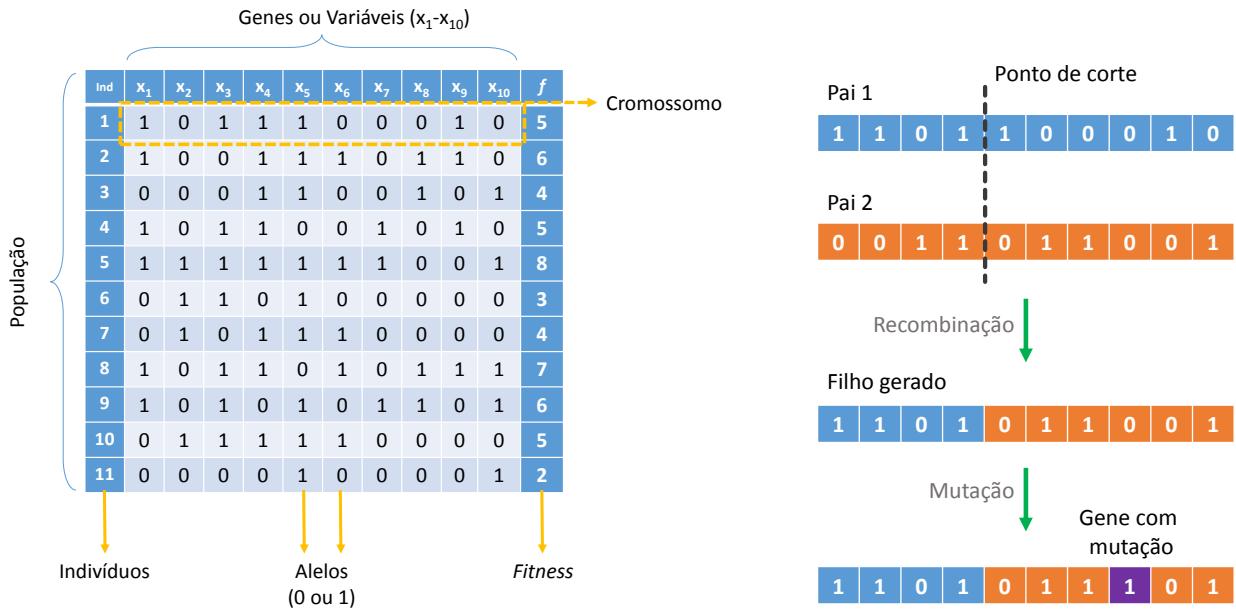
O Algoritmo Genético (GA, do inglês, *Genetic Algorithm*) pertence à classe dos EAs e foi inicialmente proposto por Holland (1975). Além de elementos comuns aos EAs como, por exemplo, a população, o GA utiliza os conceitos de cromossomos, genes e alelos. No GA, o conjunto de todas as variáveis de certo indivíduo é chamado de cromossomo. Cada parte do cromossomo é dividido em genes, em que cada um representa uma variável do problema. Alelo é nome dado aos possíveis valores que um gene pode receber. Por exemplo, em problemas envolvendo variáveis binárias os alelos podem ser 0 e 1. A Figura 3.4(a) mostra um exemplo de uma população e da nomenclatura utilizada em GAs.

As etapas básicas de um GA são as mesmas descritas na Seção 3.1 para EAs típicos: (1) criação da população inicial, (2) avaliação das soluções, (3) seleção (4) geração de novas soluções e (5) substituição. Os GAs caracterizam-se pelos operadores de recombinação e de mutação.

Os operadores de recombinação são estratégias que os GAs utilizam para imitar o cruzamento de dois cromossomos, como ocorre na natureza. Desse processo, chamado de recombinação, o filho recebe uma parte do material genético de cada pai. Em GAs, isso significa que uma certa solução gerada a partir do cruzamento de dois pais contém uma cópia de um segmento do cromossomo de um pai e o restante é cópia do cromossomo do outro pai. Se a combinação dos pais gerar um filho com melhor aptidão (*fitness*), pode se dizer que o cruzamento foi bem sucedido. A probabilidade de recombinação c_r é um parâmetro que define a taxa com que o operador de recombinação será aplicado na geração de novas soluções (Etapa 4, Seção 3.1). O ponto de corte, isto é, o ponto no qual o cruzamento ocorre, pode ser gerado utilizando uma distribuição uniforme aleatória discreta dentro do intervalo $1 \dots d - 1$, sendo d o número de dimensões do problema. GAs mais elaborados podem utilizar múltiplos pontos de corte e inclusive podem utilizar múltiplos pais para gerar um filho.

O segundo operador do GA é a mutação, que provoca pequenas mudanças nos genes. A probabilidade de um gene ser alterado pela mutação é dada por um parâmetro m_r , chamado taxa de mutação. A mutação em problemas binários pode ser simplesmente considerada como a troca do valor de um gene de 0 para 1, ou vice-versa. Para problemas do domínio contínuo, a mutação pode ser entendida como uma perturbação do valor de um gene. Essa perturbação pode ser definida somando-se um valor aleatório gerado dentro de um intervalo definido por um parâmetro m_f , chamado fator de mutação. A mutação pode introduzir diversidade à população, porém altas taxas de mutação podem também resultar em convergência lenta. A Figura 3.4(b) mostra um exemplo de recombinação e mutação de um indivíduo. O Algoritmo 3 mostra um pseudocódigo de um GA simples.

Após gerar os novos indivíduos utilizando os operadores de recombinação e mutação, os novos indivíduos são avaliados e a forma com que esses novos indivíduos irão substituir os indivíduos da população atual depende do método de substituição escolhido (Etapa 5, Seção 3.1).



(a) Representação de uma população em um GA com variáveis binárias. Neste exemplo, existem 11 indivíduos ($n = 11$) e um problema com 10 dimensões ($d = 10$).

(b) Geração de novo indivíduo. O ponto de corte foi escolhido aleatoriamente (ponto após o quarto gene). Seguida à recombinação, foi realizada uma mutação no oitavo gene do novo filho gerado.

Figura 3.4: Representação de uma população e geração de novo indivíduo para o GA.

Diversas análises teóricas têm sido realizadas com o objetivo de demonstrar em que casos um GA pode funcionar ou falhar. Essas análises buscam também fornecer os parâmetros mais adequados para o funcionamento do GA como, por exemplo, o tamanho da população (Schmitt, 2001). Em geral, tais análises têm sido realizadas para problemas com representação discreta, em geral binária, pois com base na cardinalidade de cada variável pode-se estimar probabilidades de ocorrer efeitos combinados de valores para mais de uma variável. Em geral, tais estimativas para variáveis contínuas são mais difíceis de serem calculadas.

3.2.4 Evolução Diferencial

A Evolução Diferencial (DE, do inglês, *Differential Evolution*) foi inicialmente proposta por Storn & Price (1997). A DE também pertence à classe dos EAs, pois herda várias características, como, a utilização de uma população, função de avaliação, geração dos filhos e substituição da população atual pelos filhos. Basicamente, a principal diferença entre um GA e uma DE está relacionada com a maneira com que geram novos filhos, isto é, como são feitas a recombinação e mutação (ver Algoritmo 4). Além disso, a DE foi projetada inicialmente para problemas com variáveis contínuas.

Assim como nos GAs, a DE utiliza um parâmetro c_r que controla a taxa de recombinação. No entanto, ao invés de utilizar o parâmetro para determinar se ocorre ou não a recombinação

Algoritmo 3: Pseudocódigo de um Algoritmo Genético típico

```

1:  $P' \leftarrow$  Criar população inicial de tamanho  $n$ 
2:  $P \leftarrow$  Avaliar a população  $P'$ 
3:  $g \leftarrow 1$ 
4: while  $g < g_{max}$  do
5:    $S \leftarrow$  Selecionar  $s$  indivíduos de  $P$ 
6:    $O' \leftarrow$  Recombinar os indivíduos de  $S$  em pares
7:    $O \leftarrow$  Mutar alguns genes dos indivíduos de  $O'$ 
8:    $O \leftarrow$  Calcular o fitness dos indivíduos de  $O$ 
9:    $P \leftarrow$  Substituir alguns indivíduos de  $P$  por indivíduos de  $O$ 
10:   $g \leftarrow g + 1$ 
11: end while

```

de um determinado indivíduo, a DE utiliza o parâmetro de recombinação para determinar se a recombinação ocorre ou não para cada variável do problema (em analogia aos genes de um GA). Assim, um valor aleatório uniforme no intervalo $[0, 1]$ é gerado e comparado com c_r para cada variável e, caso o valor gerado seja menor, a recombinação de tal variável é realizada (linha 5 do Algoritmo 4). Caso contrário, o valor da variável do filho recebe o mesmo valor da variável do pai.

Os operadores de mutação e a recombinação estão fortemente relacionadas na DE, pois a mutação somente ocorre se a recombinação ocorrer. Na verdade, o parâmetro que define a taxa de recombinação tem dois propósitos: a recombinação e a mutação. Uma das características interessantes da DE é o fato da mutação utilizar três indivíduos pais para compor o filho. Os três indivíduos selecionados, chamados de p^1, p^2, p^3 devem ser diferentes entre si. O valor da variável j do filho i é calculada conforme mostra a Equação 3.4.

$$o_j^i = p_j^1 + F(p_j^2 - p_j^3), \quad (3.4)$$

em que F é definido como uma constante no intervalo $[0, 2]$, que controla a amplitude da variação da diferença $(p_j^2 - p_j^3)$ sobre o_j^i .

A escolha dos parâmetros c_r e F são essenciais para o sucesso da DE. Storn & Price (1997) definem que o tamanho de população adequado n varia entre $5 \cdot d$ e $10 \cdot d$, em que d é o número de dimensões do problema. O parâmetro F pode inicialmente ser definido em 0,5, porém, caso haja convergência prematura, os parâmetros F e n devem ser aumentados. Para valores de F menores que 0,4 e maiores que 1,0 pode não ser possível explorar adequadamente o espaço de busca. A taxa de recombinação c_r é geralmente definida como 0,1, porém valores entre 0,9 e 1,0 podem ser utilizados para analisar se alguma solução é encontrada rapidamente. O Algoritmo 4 mostra como é realizada a recombinação e mutação de um indivíduo na DE.

Storn & Price (1997) sugerem também algumas variações da DE em que mais indivíduos (mais do que três) sejam utilizados para compor as diferenças na mutação. Sugere também que, ao invés de utilizar o indivíduo p^1 da Equação 3.4, seja utilizado o melhor indivíduo da população.

Algoritmo 4: Pseudocódigo da recombinação e mutação utilizada na DE para um indivíduo i

```

1:  $i \leftarrow$  índice de um indivíduo de  $P$ 
2:  $p^{1,2,3} \leftarrow$  Selecionar três indivíduos diferentes da população  $P$ 
3: for  $j = 1$  to  $d$  do
4:    $u \leftarrow$  Gerar número aleatório uniforme  $U[0; 1]$ 
5:   if  $u < c_r$  then
6:     Novo indivíduo  $o_j^i = p_j^1 + F(p_j^2 - p_j^3)$ 
7:   else
8:     Novo indivíduo  $o_j^i = P_j^i$ 
9:   end if
10: end for

```

A DE tem mostrado ser promissora em aplicações do mundo real. Alguns trabalhos relativamente recentes têm buscado evidenciar esse potencial da DE: Van Sickel et al. (2007); Rocca et al. (2011); Li & Yin (2012).

3.3 Algoritmos de Estimação de Distribuição

A dificuldade que existe em explorar adequadamente o espaço de busca por soluções promissoras sem que o algoritmo fique preso a ótimos locais fomentou pesquisas que resultam em um novo tipo de EA. Esses, foram chamados de Algoritmos de Estimação de Distribuição (EDAs, do inglês, *Estimation of Distribution Algorithms*). Entre seus precursores está o trabalho proposto por Muehlenbein & Paab (1996). Os EDAs utilizam um modelo probabilístico e amostragens a partir desse modelo ao invés de aplicar operadores de reprodução como a mutação e recombinação. Antes do surgimento dos EDAs, Holland (1975) já tinha verificado que um GA que fosse capaz de modelar relacionamentos entre variáveis poderia explorar melhor o espaço de busca do que um GA típico. Isso motivou o desenvolvimento de GAs mais sofisticados, capazes de modelar relacionamentos entre variáveis (Larranaga & Lozano, 2002).

Os EDAs têm se mostrado capazes de resolver vários problemas complexos e de larga-escala e têm sido aplicados em diversos campos como, por exemplo, área militar (Yu et al., 2006), bioinformática (Bacardit et al., 2007), econômica (Chen & Chen, 2007), entre outras. É importante destacar que para um número significativo de aplicações em que os EDAs tiveram sucessos nenhuma outra técnica foi capaz de encontrar soluções superiores aos EDAs, ou mesmo resolver os problemas de tamanho e complexidade semelhantes (Hauschild & Pelikan, 2011).

Os EDAs também utilizam um conjunto de indivíduos (soluções candidatas) chamado de população que, inicialmente, deve ser gerada utilizando uma distribuição aleatória dos possíveis valores que cada variável pode assumir. A partir da população, ocorre a seleção de soluções promissoras por um dos métodos de seleção descritos na Seção 3.1. Com base nessa amostragem de indiví-

duos selecionados, um modelo probabilístico é construído (Figura 3.5). A construção do modelo probabilístico é uma das etapas mais importantes de um EDA.

O modelo probabilístico de um EDA busca estimar a distribuição dos valores das variáveis dos indivíduos selecionados. Depois, o modelo probabilístico é utilizado para amostrar novas soluções. Após a geração dos novos indivíduos, a população antiga pode ser parcialmente ou completamente substituída pela nova população. Todo esse processo é repetido até que um critério de convergência estabelecido seja atingido como, por exemplo, a convergência das variáveis (quando o desvio padrão das variáveis for menor que um valor determinado) ou um quando um número máximo de avaliações for atingido (Hauschild & Pelikan, 2011). A Figura 3.5 mostra o esquema de funcionamento de um EDA.

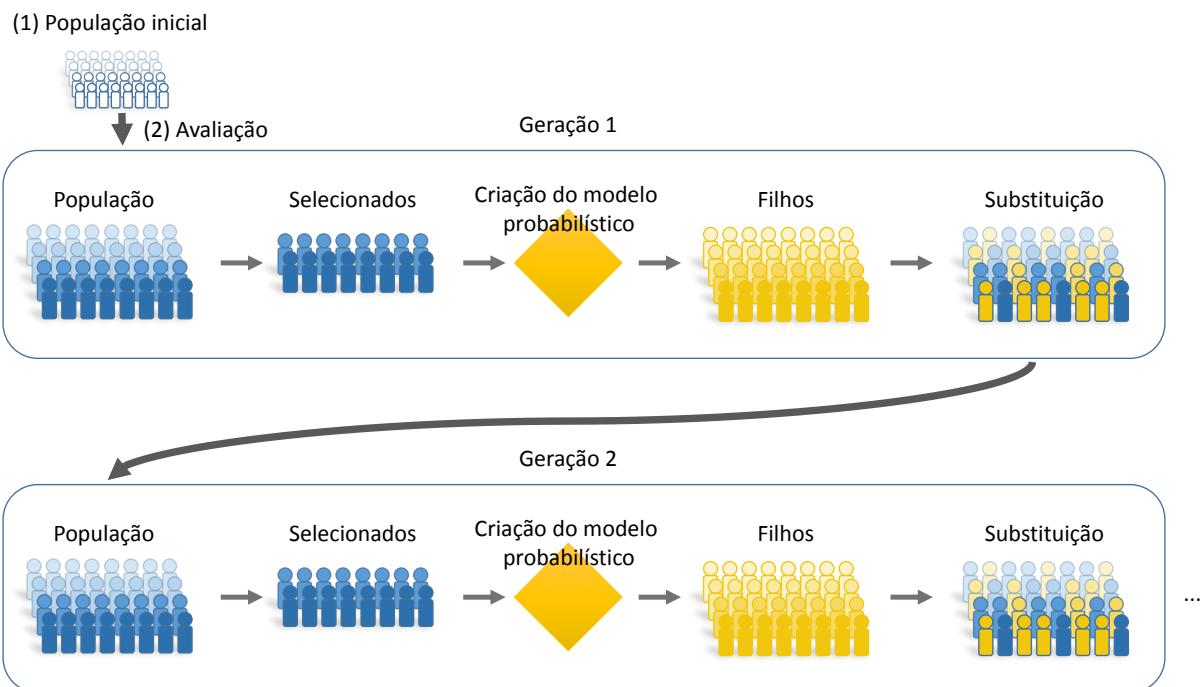


Figura 3.5: Esquema do funcionamento de um EDA.

A principal diferença entre os algoritmos de busca como o MC, GA e DE com os EDAs, está no modelo que os EDAs constroem a partir da distribuição de probabilidade das soluções promissoras, responsável pela geração de novos indivíduos. O modelo probabilístico gerado precisa ser elaborado cuidadosamente, pois a distribuição de probabilidades não pode privilegiar totalmente as melhores soluções, mas sim representar uma distribuição mais geral da distribuição dos valores das variáveis. Além disso, os EDAs exigem métodos eficientes de construção dos modelos probabilísticos, pois, em geral, a construção desses modelos é a etapa do algoritmo que requer mais tempo de computação (Hauschild & Pelikan, 2011). O Algoritmo 5 mostra o pseudocódigo do funcionamento de um EDA.

Embora existam vários EDAs com desempenho relevante para os problemas em que foram aplicados como, o *Bayesian Optimization Algorithm* (BOA) e o *Extended compact Genetic Algorithm* (EcGA) (Seção 3.3.2) a qualidade do modelo depende significativamente da capacidade do

Algoritmo 5: Pseudocódigo de um Algoritmo de Estimação de Distribuição.

```

1:  $P' \leftarrow$  Inicializar a população aleatória
2:  $P \leftarrow$  Avaliar a população inicial  $P'$ 
3:  $g \leftarrow 1$ 
4: while  $g < g_{max}$  do
5:    $S \leftarrow$  Aplicar método de seleção a partir de  $P$ 
6:    $\hat{\theta} \leftarrow$  Criar modelo probabilístico utilizando  $S$ 
7:    $O' \leftarrow$  Gerar novos indivíduos utilizando o modelo probabilístico  $\hat{\theta}$ 
8:    $O \leftarrow$  Avaliar os filhos  $O'$ 
9:    $P \leftarrow$  Aplicar método de substituição da população em  $O$ 
10:   $g \leftarrow g + 1$ 
11: end while

```

método utilizado para se descobrir correlações entre variáveis relacionadas e do custo computacional requerido pelo método. Assim, as características de cada problema, o tamanho das instâncias a serem tratadas e o tempo de computação que é aceitável podem afetar significativamente o sucesso em se construir um modelo de alta qualidade. De forma a mostrar como esses aspectos têm sido considerados no desenvolvimento de melhores EDAs as seções seguintes introduzem os principais modelos probabilísticos desenvolvidos para EDAs

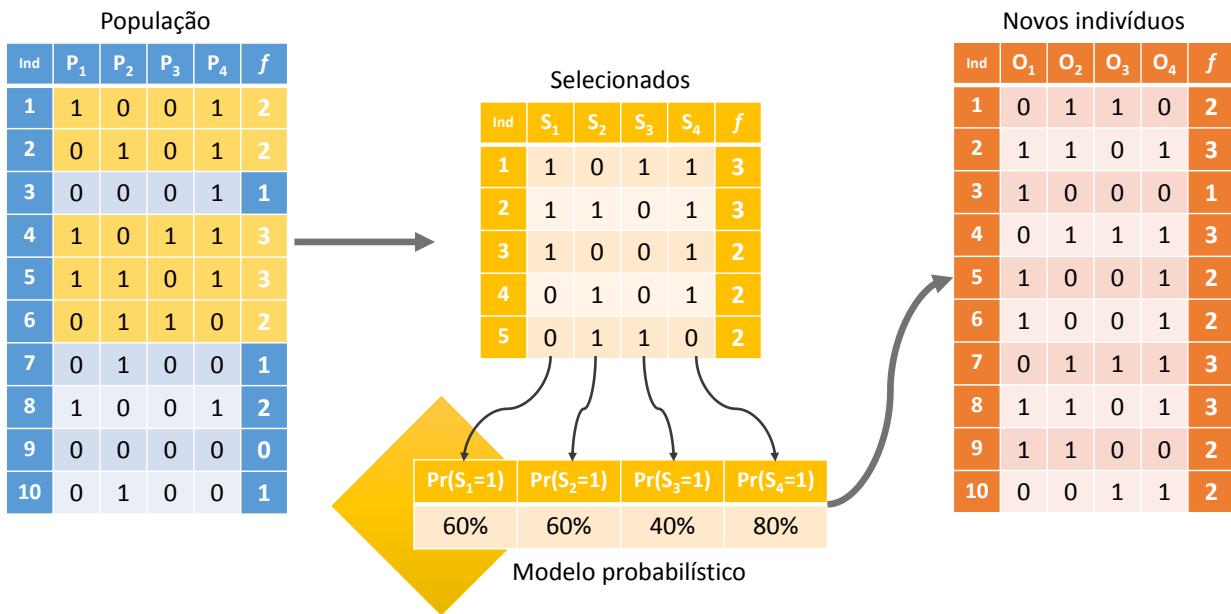
Primeiramente, a Seção 3.3.1 apresenta um exemplo básico de um EDA univariado e bivariado para variáveis binárias. As Seções 3.3.2 e 3.3.3 discutem os EDA para variáveis discretas e contínuas, respectivamente.

3.3.1 Exemplo de um EDA

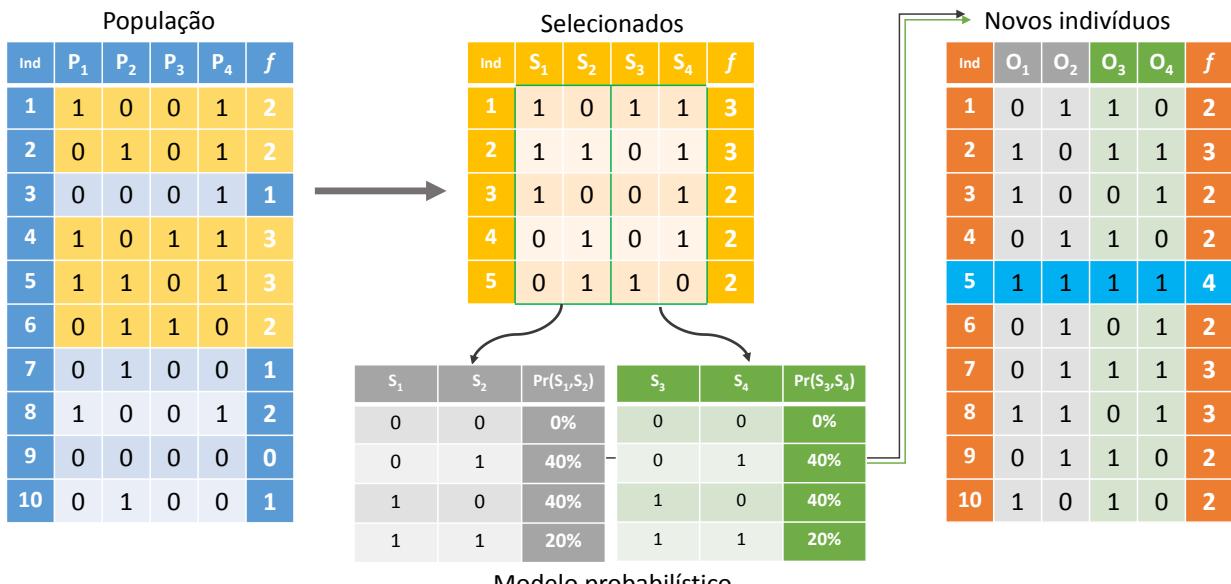
O funcionamento de um EDA pode ser melhor entendido utilizando um exemplo de um simples problema de otimização binário. Considere que seja necessário maximizar a função *OneMax* definida no espaço 4-dimensional ($d = 4$), como $f(x) = \sum_{i=1}^4 x_i$, sendo $x_i \in [0, 1]$. O ótimo global dessa função é $f(x) = 4$, ou seja, quando $x = 1111$. No caso univariado, são criados quatro modelos probabilísticos, um para cada variável do problema.

A população inicial pode ser gerada aleatoriamente seguindo uma distribuição uniforme para cada variável com 50% de chances de ser 0 ou 1. Essa probabilidade é então ajustada de acordo com o processo evolutivo e pode ser utilizada para amostrar novos indivíduos. Considere a população representada pela Figura 3.6. Utilizando algum método de seleção como, por exemplo, o método por truncamento é montado o conjunto dos indivíduos selecionados, que correspondam a indivíduos que possivelmente estão em regiões promissoras do espaço de busca.

No exemplo da Figura 3.6 foi criado uma população de tamanho 10 ($n = 10$) e utilizado $\tau = 0,5$. Assim, o tamanho do conjunto dos selecionados S é igual a 5 ($s = 5$) e S é uma matriz tal que cada elemento S_j^i , $i = 1 \dots s$, é o valor da variável j do indivíduo selecionado i . Nesse exemplo, os cinco indivíduos do conjunto dos selecionados S representam os cinco melhores indivíduos da população P . Utilizando S é possível calcular a distribuição de frequência de cada



(a) Esquema de funcionamento de um EDA univariado.

(b) Esquema de funcionamento de um EDA bivariado. No conjunto dos novos indivíduos foi destacado o indivíduo 5 que possui o ótimo global da função ($f = 4$).**Figura 3.6:** Exemplos de EDAs para o problema *OneMax*.

variável, ou seja, o modelo probabilístico utilizado neste exemplo. Considere Pr_j como sendo a probabilidade dos selecionados S_j assumir o valor 1 com base em todos os s indivíduos selecionados. Assim, no exemplo da Figura 3.6(a), caso $j = 1$, $Pr_j = 0, 6$. Essa probabilidade é calculada para todas as variáveis e o conjunto dessas probabilidades é representada por Pr , chamado de vetor de probabilidades. A construção do vetor de probabilidades Pr pode ser considerada o “modelo probabilístico” desse EDA.

Usando o vetor de probabilidades Pr , novos indivíduos podem ser gerados. Por exemplo, considere que $Pr_1 = 0,6$, é esperado que os valores de O_1 sejam 1 em 60% dos indivíduos e 0 no restante. Esse procedimento é repetido para todas as demais variáveis ($j = 2 \dots d$) até que todas as posições de O sejam preenchidas. Isso conclui o processo de amostragem dos novos indivíduos.

Em seguida, assim como em um EA comum, é calculado o *fitness* dos filhos que podem também ser agrupados com a população P , tal que $M = P \cup O$. Então, apenas os n melhores indivíduos de M são escolhidos para substituir toda a população P . Todo esse procedimento caracteriza uma geração (ou iteração) do EDA. Com base no exemplo da Figura 3.6(a) é possível verificar que a nova população possui maior quantidade de números 1 em relação à população inicial. Dessa maneira, espera-se que a quantidade de números 1 da população continue aumentando de acordo com as gerações até atingir a solução correta, que conterá o valor 1111.

A segunda característica fundamental de um EDA é o tratamento de relações entre variáveis. Para facilitar o entendimento, suponha que a mesma função *OneMax* seja utilizada para exemplificar um EDA que trate relação entre duas variáveis, isto é, um EDA bivariado. Assim, o problema no espaço 4-dimensional pode ser dividido em problemas bivariados (ao invés de quatro univariados) em que a relação das variáveis vizinhas duas a duas são estabelecidas. Neste exemplo, consideram-se os pares (x_1, x_2) e (x_3, x_4) como variáveis relacionadas. Assim, os pares de vetores $[S_1, S_2]$ e $[S_3, S_4]$ devem ser modelados conjuntamente. Dessa maneira, o modelo probabilístico e a amostragem devem ser capazes de lidar com tal relacionamento.

Assim como o exemplo univariado, a partir de uma população aleatória, uma certa quantidade de indivíduos promissores é selecionada. No entanto, ao invés de calcular a probabilidade de cada variável individual é criada uma tabela de probabilidade conjunta para cada grupo de par de variáveis relacionadas e atribuído uma probabilidade para cada combinação possível dos valores das variáveis. Cada tabela de probabilidade deverá ter dimensão 2^2 linhas por 2 colunas, isto é, $c^d \times d$ em que c é a cardinalidade das variáveis. No caso binário $c = 2$, pois os únicos valores que as variáveis podem assumir é 0 ou 1.

A Figura 3.6(b) mostra como ficaria essas tabelas para o conjunto de selecionados. Nesse caso, duas tabelas foram construídas, uma para cada par de variáveis. A tabela que mostra o relacionamento entre $[S_1, S_2]$ tem as seguintes probabilidades: $Pr(S_1 = 0; S_2 = 0) = 0,0$; $Pr(S_1 = 0; S_2 = 1) = 0,4$; $Pr(S_1 = 1; S_2 = 0) = 0,4$; e $Pr(S_1 = 1; S_2 = 1) = 0,2$. É possível notar também que na tabela das variáveis $[S_3, S_4]$ a probabilidade de ambas as variáveis assumirem o valor 0 é de 0,0.

Após calcular as probabilidades de cada tabela, os valores das variáveis dos novos indivíduos são amostrados de dois em dois, devido a utilização do modelo bivariado, para cada indivíduo. Assim, ao invés de amostrar O_1 isoladamente, são amostrados os valores de $O_{1,2}$, isto é, valores de O_1 e O_2 ao mesmo tempo com base na distribuição da tabela Pr_{S_1, S_2} . Esse processo é repetido também para as variáveis S_3 e S_4 utilizando a distribuição da tabela Pr_{S_3, S_4} . É interessante observar que utilizando a abordagem bivariada, nenhum indivíduo com o valor 0000 foi gerado, devido a probabilidade de $Pr(S_1 = 0; S_2 = 0) = 0,0$ e $Pr(S_3 = 0; S_4 = 0) = 0,0$. Dessa

forma, foi evitado a geração de indivíduos relativamente ruins. Além disso, com a combinação da amostragem dos dois blocos de variáveis foi gerado o indivíduo ótimo já na primeira geração do EDA, representado pelo quinto indivíduo (em destaque) do conjunto dos filhos da Figura 3.6(b). Assim, o processo de amostragem dos filhos O da primeira geração é concluído.

3.3.2 EDAs no domínio discreto

Os EDAs projetados para trabalhar com variáveis discretas utilizam cardinalidade finita como, por exemplo, a representação binária. Os EDAs podem ser divididos em univariados ou multivariados, assim como mostra o exemplo da Seção 3.3.1, em que a Figura 3.6(a) mostra um exemplo de um EDA univariado e a Figura 3.6(b), um exemplo de um EDA bivariado (ou multivariado com dimensão dois). No entanto, diferentemente desse exemplo, em que foi pré-fixada as variáveis vizinhas que compunham um par, alguns EDAs também possuem mecanismos para aprender quais variáveis possuem correlação a partir do conjunto dos indivíduos selecionados. A capacidade de encontrar tais relacionamentos depende da complexidade do modelo probabilístico utilizado.

O *Univariate Marginal Distribution Algorithm* (UMDA) (Mühlenbein, 1997) é um exemplo de um EDA univariado. Outros EDAs semelhantes ao UMDA também têm sido pesquisadas na literatura como, por exemplo, o PBIL e o cGA. Estes, são considerados EDAs incrementais, pois utilizam um vetor de probabilidades ao invés de uma população, semelhante ao exemplo da Figura 3.6(b). Dessa maneira, em geral requerem menos recursos computacionais.

O *Population-Based Incremental Learning* (PBIL) foi desenvolvido por Baluja (1994) e também lida com variáveis binárias. Assim como o UMDA, o PBIL utiliza um vetor de probabilidades. A cada geração, um pequeno conjunto de soluções é gerado com base no vetor de probabilidades. A melhor solução desse pequeno conjunto é escolhida e os valores do vetor de probabilidades são atualizados na direção dos valores do indivíduo escolhido. São também utilizados dois parâmetros, que define a taxa de alteração do vetor de probabilidades pelas variáveis da solução escolhida e outro parâmetro que provoca uma perturbação no mesmo valor do vetor de probabilidades.

O cGA (*compact Genetic Algorithm*) (Harik et al., 1998) é semelhante ao PBIL. A principal diferença entre eles está na maneira com que o vetor de probabilidades é atualizado a cada geração. No cGA, dois indivíduos são gerados a partir do vetor de probabilidades e avaliados. As variáveis dos dois indivíduos gerados são comparadas uma a uma. Caso sejam diferentes, o vetor de probabilidades é atualizado na direção do valor da variável melhor avaliada, ou seja, os índices correspondentes do vetor de probabilidades que necessitam ser alterados são deslocados $1/n$ na direção do valor da variável do melhor indivíduo, em que n representa um tamanho de população teórico, isto é, o mesmo tamanho que a princípio seria necessário para um GA simples.

O UMDA, PBIL e cGA são exemplos de EDAs univariados para problemas binários. É possível basear em algumas dessas ideias para projetar EDAs capazes de lidar com relacionamento de variáveis. Um exemplo de EDA bivariado é o BMDA, que é uma extensão do UMDA. Exemplos

de EDAs multivariados, capazes de tratar múltiplas variáveis correlacionadas é o EcGA (extensão do cGA) e o BOA.

O *Bivariate Marginal Distribution Algorithm* (BMDA) foi proposto por Pelikan & Muehlenbein (1999). O BMDA utiliza um grafo orientado (Trudeau, 2013) para modelar o relacionamento de pares de variáveis. O grafo é reconstruído a cada geração utilizando a estatística χ^2 de Pearson (Pearson, 1895). Novos indivíduos são amostrados a partir do relacionamento das variáveis que foram mapeadas pelo grafo.

Os EDAs que tratam correlações múltiplas são também chamados de EDAs multivariados. Alguns deles utilizam o conceito de agrupamento de variáveis em blocos. Assim, o agrupamento de variáveis, chamado de blocos construtivos (BBs, do inglês, *Building Blocks*) são tratados como subproblemas (problemas menores) e a princípio podem, em certos casos, ser resolvidos de forma independente. O BOA, *Bayesian Optimization Algorithm* (Pelikan et al., 1999), utiliza uma BN (do inglês, *Bayesian Network*), um grafo orientado ao qual são associados probabilidades condicionais (Jordan & Division, 1998) para representar a interação que ocorre entre as variáveis do problema. A partir do conjunto de soluções promissoras, o BOA busca aprender uma BN adequada que represente corretamente a interação entre as variáveis e calculando a probabilidade condicional entre tais variáveis. Utilizando a melhor BN encontrada, novas soluções são geradas podendo substituir parte ou mesmo toda a população antiga com os novos indivíduos. Esse processo é repetido até que um critério previamente estabelecido seja atingido.

Um dos grandes desafios do BOA é a construção da BN que melhor represente relacionamentos entre variáveis. Nessa etapa é utilizado um procedimento de busca usando uma métrica para avaliação do modelo. O procedimento procura por estruturas de BNs que tenham a melhor pontuação obtida na avaliação pela métrica. Tal procedimento pode ser entendido como um problema combinatório em si. Portanto, em geral, o procedimento de busca utilizado pelo BOA corresponde a um algoritmo guloso (Heckerman et al., 1995), de forma a ser computacionalmente viável. A cada aresta acrescentada ou removida, o novo modelo é comparado com o modelo anterior. Caso a adição ou remoção da aresta melhore o modelo segundo a métrica, a alteração da aresta permanece. Esse processo é repetido para todas as variáveis até que um número máximo estabelecido de grau para nós da BN seja atingido (passando para a próxima variável) ou até que o processo não seja mais capaz de melhorar a qualidade do modelo. O grau máximo de uma variável é definido pela quantidade de saltos necessários, a partir de certa variável, até a variável que tem a relação mais distante. Por exemplo, em $x_1 \rightarrow x_2 \rightarrow x_3$, x_1 tem grau 2 (pois tem influência sobre os valores de x_2 e x_3), e x_2 tem grau 1 (tem influência somente sobre x_1). A métrica de avaliação utilizada no aprendizado de BNs é, em geral, a métrica chamada K2 (Cooper & Herskovits, 1992).

A geração de novos indivíduos no BOA ocorre a partir do melhor modelo obtido. Considerando que há relacionamento entre variáveis, é necessário, em primeiro lugar, que as variáveis pais sejam geradas na sequência dada pelas arestas orientadas da BN. Para isso, deve-se calcular a ordem de ancestrais das variáveis (Pelikan, 2005).

O *Extended compact Genetic Algorithm* (EcGA) (Harik, 1999) é semelhante ao cGA, a diferença é que o EcGA detecta relacionamentos de variáveis assim como o BOA, portanto, o EcGA é um EDA multivariado. Diferentemente do cGA, o EcGA é capaz de aprender e utilizar a informação dos BBs para melhorar a exploração do espaço de busca

O funcionamento do EcGA parte do mesmo princípio do BOA. Primeiramente, a população inicial pode ser inicializada aleatoriamente. Uma amostra de tamanho previamente definido é selecionada da população, utilizando um dos métodos de seleção (Seção 3.1). A partir da população selecionada é construído um modelo para que represente relacionamentos de variáveis. No caso do EcGA, é utilizado o Modelo Produto Marginal (MPM) (Harik et al., 1998) para agrupar as variáveis que estão relacionadas. Inicialmente são formados d grupos unitários, em que d é a quantidade de variáveis do problema. As variáveis são combinadas de forma a minimizar a Complexidade Combinada (CC), pois distribuições mais simples são consideradas melhores que as mais complexas para se obter uma melhor CC. Assim, a nova população pode ser gerada utilizando o MPM com a melhor CC. Todo esse processo é repetido até que um critério de parada seja estabelecido como, por exemplo, convergência da população ou quando o algoritmo encontrar uma solução suficientemente boa.

A geração de indivíduos no EcGA é realizada utilizando a distribuição da frequência da variável na população da mesma forma que é ocorre no cGA. No entanto, considerando que o EcGA pode ter BBs maiores que 1, uma tabela de probabilidades é calculada para cada BB (assim como o segundo exemplo da Seção 3.3.1). Por exemplo, considerando um BB de tamanho 3 para um problema com variáveis binárias, a tabela de probabilidades deve ter tamanho $2^3 = 8$. Dado a distribuição de probabilidades para determinado BB de tamanho 3, a sequência de bits 111 tem mais chances de ser escolhida para compor parte do indivíduo, do que teria a sequência de bits 000, que poderia ser um ótimo local. Isso revela grande vantagem que o EcGA tem em relação aos métodos de busca que não detectam relacionamentos entre variáveis. O tratamento de variáveis relacionadas como BB, possibilita explorar melhor o subespaço das variáveis de cada BB e promover a recombinação entre BBs (Goldberg, 2002).

3.3.3 EDAs no domínio contínuo

Os EDAs foram inicialmente projetados para trabalhar com problemas discretos, mais precisamente binários. A complexidade da geração dos modelos probabilísticos dos EDAs pode estar relacionada com os fatores tamanho do BB e com a cardinalidade das variáveis. Assim, dependendo desses fatores, a tabela de probabilidades pode tornar-se relativamente grande. Por exemplo, para um problema em que a cardinalidade das variáveis é 10, a tabela terá 10^3 registros para um BB de tamanho 3. Nesse caso, a CC de um EcGA, por exemplo, (Seção 3.3.2) irá crescer de acordo com o tamanho de certo BB b elevado a sua cardinalidade c , isto é, b^c . Além desse aspecto desvantajoso, tais EDAs não podem, em geral, ser aplicados diretamente para problemas contínuos.

Para tornar possível o uso dos EDAs no domínio contínuo é necessário realizar certos ajustes na geração dos modelos probabilísticos. Devido a cardinalidade das variáveis de problemas contínuos ser infinita, não há como mapear a probabilidade de ocorrência das variáveis. Basicamente, existem duas abordagens principais que tornam possível o uso de EDAs para variáveis do domínio contínuo: 1) utilizar um EDA para variáveis discretas mapeando as variáveis de um problema contínuo no domínio discreto (Seção 3.3.2); 2) desenvolver EDAs que são baseados em distribuição de probabilidade de variáveis contínuas (Hauschild & Pelikan, 2011).

Uma das formas mais direta de utilizar um EDA para problemas que tenham variáveis contínuas pode ser, simplesmente, discretizar (truncar) as variáveis e tratar os passos seguintes (a partir da construção do modelo probabilístico em diante) da mesma maneira que um EDA discreto. No entanto, isso pode ser um problema quando os valores das variáveis contínuas estiverem próximos uns dos outros, pois a discretização pode esconder valores interessantes para o algoritmo ou mesmo esconder as regiões mais densas em que os valores frequentes estão mais concentrados. Para contornar esse problema, métodos de discretização que sejam capazes de amostrar bem as regiões mais concentradas e menos concentradas na faixa de valores admissíveis podem ser utilizados (Hauschild & Pelikan, 2011). Vários métodos têm sido desenvolvidos para representar mais adequadamente a distribuição dos valores das variáveis contínuas no domínio discreto (Tsutsui et al., 2001; Pelikan et al., 2003; Chen et al., 2006; Suganthan et al., 2005).

Duas maneiras de discretização, por histogramas e o *k-means* (Kaufman & Rousseeuw, 2008), foram utilizados no BOA buscando resolver problemas do mundo real, como mostrado em Pelikan et al. (2003). Após mapear as variáveis discretas para o domínio contínuo, utilizou-se também mutação adaptativa (Bäck, 1996). Os resultados foram satisfatórios, indicando que a técnica tem escalabilidade para vários problemas do domínio contínuo.

Em Chen et al. (2006) foi proposto o *Split-on-Demand* (SoD) para tornar possível o uso do EcGA com variáveis contínuas, chamado de *real-coded Extended compact Genetic Algorithm* (rEcGA). O SoD ajusta variáveis contínuas para o domínio discreto em tempo real. Este método utiliza procura por regiões em que há mais informações para fazer a divisão, aumentando a capacidade de exploração, em que cada divisão é identificada por um único valor. A principal diferença entre o EcGA e o rEcGA é que, entre as etapas de seleção dos indivíduos e a geração do modelo probabilístico, deve ser aplicado o SoD para codificar as variáveis. O uso do SoD no rEcGA obteve sucesso em um conjunto de funções de *benchmark*.

Os EDAs que utilizam mecanismos para representar variáveis contínuas no domínio das variáveis discretas utilizam EDAs projetados para variáveis discretas. No entanto, os EDAs podem trabalhar diretamente com números reais, sem necessidade de discretização. Nesse caso, esses algoritmos precisam ser projetados para tratar diretamente variáveis contínuas. Em geral, isso pode ser feito mantendo a estrutura do EDA e adaptando somente os trechos em que as variáveis são manipuladas para construção dos modelos probabilísticos.

Assim como os EDAs discretos, os EDAs contínuos são classificados de acordo com a maneira com que mapeiam os relacionamentos de variáveis. Os EDAs mais simples consideram que todas

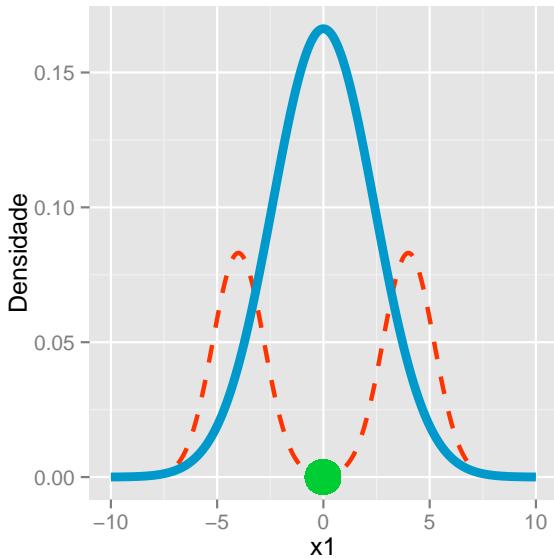
as variáveis são independentes como, por exemplo, o UMDA_c (Muehlenbein & Mahnig, 1999) e o SHCLVND (Rudlof & Köppen, 1996).

Um dos primeiros EDAs propostos capaz de lidar com variáveis reais foi o SHCLVND (*Stochastic Hill Climbing with Learning by Vectors of Normal Distributions*) (Rudlof & Köppen, 1996). Este algoritmo representa as variáveis por meio de um vetor, em que cada variável é formada por uma distribuição normal independente. Os valores das variáveis são representados por um valor médio de cada variável e por um único desvio padrão. O valor da média das variáveis tende a deslocar-se em direção das soluções ótimas e o desvio padrão é reduzido por um fator multiplicativo. O fato de usar o mesmo desvio padrão para todas as variáveis é uma desvantagem para o SHCLVND, além de utilizar a mesma distribuição para todas as variáveis. Uma melhoria nesse algoritmo foi feita por Sebag & Ducoulombier (1998), adicionando o parâmetro para armazenar o desvio padrão de cada variável, obtendo resultados mais significativos.

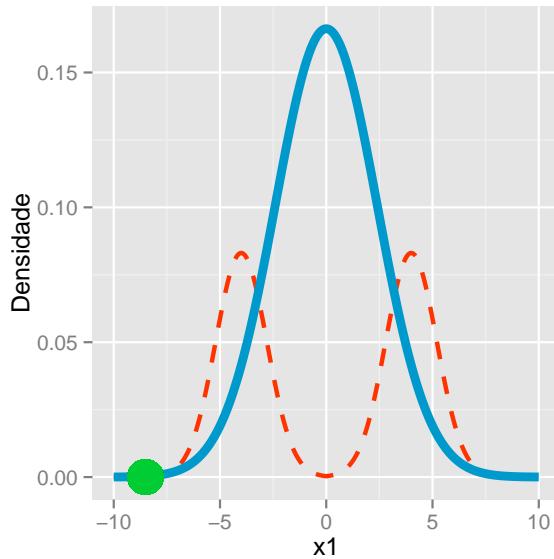
O *Univariate Marginal Distribution Algorithm* contínuo (UMDA_c) proposto por Muehlenbein & Mahnig (1999) é outro exemplo de EDA contínuo que assume que as variáveis são independentes. Para cada variável são executados testes estatísticos para determinar a função de densidade mais apropriada para cada variável. Em geral, utiliza-se a distribuição normal. Assim, a partir do conjunto de indivíduos selecionados são estimados os valores da média e do desvio padrão de cada variável. Em seguida, novos indivíduos podem ser amostrados utilizando a Função Densidade de Probabilidade (FDP) da distribuição normal com a média e desvio padrão de cada variável. A abordagem do UMDA_c com distribuição normal é especialmente interessante para problemas com relativamente poucos ótimos locais, pois não é capaz de representar corretamente a distribuição de frequência de uma variável com mais de uma moda.

Por exemplo, considere a Figura 3.7(a) em que o valor ótimo da variável x_1 é o ponto 0, 0. A distribuição de frequência real dos dados (do conjunto dos indivíduos selecionados) é representada pela linha tracejada vermelha, isto é, os dados possuem duas modas. Apesar de os dados reais serem representados por duas modas, o modelo probabilístico do UMDA_c (utilizando apenas média e desvio padrão) não foi capaz de representar adequadamente a distribuição real dos dados. Mesmo assim, considerando que o ótimo global da variável x_1 está no ponto 0, 0 (representado pelo ponto verde da Figura 3.7(a)), o UMDA_c, ao amostrar novos dados poderá conseguir amostrar relativamente boas soluções, pois a média obtida pelo modelo probabilístico coincidiu com o ótimo global, que apareceu justamente no meio das duas modas da distribuição real das variáveis. No entanto, a Figura 3.7(b) mostra um caso em que o ótimo global da variável x_1 está definido em -8, 5. Nesse caso, utilizando a mesma média e desvio padrão da Figura 3.7(a), em que distribuição de frequência real dos dados são iguais, o modelo probabilístico gerado pelo UMDA_c dificilmente irá gerar valores próximos do ótimo global (definido em -8, 5 pelo ponto verde da Figura 3.7(b)).

Ambos os EDAs univariados SHCLVND e UMDA_c são promissores para problemas que não exista correlação de variáveis ou mesmo que não tenham múltiplos ótimos locais. Por outro lado, EDAs que tratam relacionamentos de variáveis são computacionalmente mais complexos, especialmente para o caso contínuo. Exemplos de EDAs multivariados e contínuos são o EGNA (Lar-



(a) Ótimo global definido em 0, 0. Nesse caso, o modelo probabilístico utilizado pode ser beneficiado pelo fato do ótimo estar entre duas modas.



(b) Ótimo global definido em -8, 5. Nesse caso, o ótimo global poderá não ser atingido pelo modelo probabilístico utilizado.

Figura 3.7: Distribuição dos valores da variável x_1 . A linha tracejada vermelha representa a distribuição real dos dados. A linha azul representa o modelo probabilístico construído pelo UMDA_c e o ponto verde representa o valor ótimo para a variável x_1 .

ranaga et al., 1999), IDEA (Bosman & Thierens, 2000), mIDEA (Bosman & Thierens, 2001) e rBOA (Ahn et al., 2004).

O Algoritmo de Estimação de Redes Gaussiana (EGNA, do inglês, *Estimation of Gaussian Networks Algorithm*) (Larranaga et al., 1999) utiliza redes Gaussiana para modelar o relacionamento de variáveis, a partir da população dos indivíduos selecionados. O conceito é o mesmo utilizado em BNs, a diferença é que a rede Gaussiana utiliza variáveis reais. Para isso, cada variável é representada pela média e desvio padrão a partir de uma função de ajuste dos valores dos pais. O EGNA também possui uma métrica que penaliza os modelos complexos (assim como o BOA e o EcGA possuem) e a melhor estrutura do modelo utilizada para gerar a nova população pode ser aprendida utilizando uma busca gulosa.

O framework IDEA (Bosman & Thierens, 2000) também trata relacionamentos de variáveis utilizando FDP Gaussiana e distribuição kernel (Rosenblatt, 1956). Experimentos mostraram que o IDEA é, em geral, capaz de produzir resultados mais significantes que outros EDAs que fazem somente uso do vetor de distribuição Gaussiana (o mesmo utilizado pelo SHCLVND). Bosman & Thierens (2001) utilizaram mistura de distribuições normal para compor um novo e mais sofisticado algoritmo, chamado mIDEA. Os modelos do mIDEA são construídos agrupando variáveis e, em seguida, é feito um ajuste da distribuição de probabilidade para cada agrupamento. A soma ponderada das distribuições individuais é assumida como a melhor distribuição. A avaliação dos

modelos gerados pode ser calculada utilizando o *Bayesian Information Criterion* (BIC) (Schwarz, 1978).

O critério de avaliação do modelo BIC é também utilizado no rBOA, *real-value Bayesian Optimization Algorithm* (Ahn et al., 2004). O rBOA mostrou desempenho superior em relação ao mIDEA em vários problemas com subótimos. O rBOA busca estender os princípios do BOA para um EDA capaz de resolver problemas que utilize números reais e com relacionamento entre variáveis. A probabilidade conjunta no rBOA é determinada pelo produto da densidade da FDP de cada variável relacionada. Análises teóricas do rBOA como, por exemplo, escalabilidade, número de avaliações, tamanho da população e número de gerações, são apresentadas em Ahn & Ramakrishna (2008)

3.4 EDAs para problemas hierárquico

Embora o BOA e o EcGA sejam capazes de decompor o problema em subproblemas e tratá-los de forma independente, há problemas em que subproblemas tem relações com outros subproblemas. Para esses casos, tem sido proposto a criação de um mecanismo de hierarquia de subproblemas, permitindo criar relacionamento de subproblemas em vários níveis, em que o nível mais baixo é representado pelas próprias variáveis do problema e os níveis mais altos pelos agrupamentos hierárquicos de subproblemas. Portanto, o relacionamento de variáveis que estão no mesmo subproblema ou que descendem de um mesmo subproblema é mais forte e devem ser tratados como um conjunto.

É possível visualizar hierarquia em muitos objetos da vida real como, por exemplo, no corpo humano, que é formado por órgãos em um nível, células em outro, átomos e assim por diante. Muitos problemas difíceis podem ser resolvidos utilizando decomposição hierárquica, pois a partir da visualização das variáveis agrupadas em níveis, pode se ter ideia de como resolver o problema original. O BOA e o EcGA decompõem o problema em pequenos subproblemas e tenta resolvê-los. No entanto, nem todo subproblema pode ser decomposto em subproblemas em somente um nível devido ao relacionamento que subproblemas podem ter em vários níveis. A cada decomposição hierárquica do problema diminui também a complexidade, isto é, conforme os agrupamentos de subproblemas aumentam de nível, pode ficar relativamente mais simples tratar o problema. A composição da hierarquia é, então, feita até o nível que torne possível tratar o subproblema (Pelikan, 2005).

Para problemas desse tipo, Pelikan et al. (2001) desenvolveu o *hierarchical Bayesian Optimization Algorithm* (hBOA), capaz de resolver problemas complexos para variáveis discretas. O hBOA (Pelikan et al., 2001; Pelikan & Goldberg, 2003) é um EDA que herda as características do BOA para representar o modelo probabilístico que separa os subproblemas em níveis hierárquicos. Os conceitos principais utilizados na hierarquia são a decomposição e agrupamento. A decomposição é a mesma utilizada no BOA (que gera o modelo probabilístico), formando os subproblemas. O

procedimento de agrupamento pode juntar subproblemas que podem ser tratados como uma única variável em um primeiro nível (chamados de BBs também). O mesmo processo é refeito para os BBs que foram agrupados, procurando por agrupamentos de BBs em um segundo nível, até não ser mais possível realizar agrupamentos. Além disso, o hBOA busca garantir alta diversidade de indivíduos na população para ser capaz de sair de ótimos locais. Isso é feito realizando uma competição de certo indivíduo com apenas indivíduos similares, ou seja, com valor de *fitness* parecido. Indivíduos que são bem diferentes raramente competem com outros indivíduos. Esse processo é chamado de *niching*, pois indivíduos semelhantes em geral compartilham o mesmo nicho (Pelikan, 2005).

O hBOA tem várias características do BOA. A diferença é que o hBOA utiliza grafos de decisão (Oliver, 1993) para garantir a decomposição correta e o agrupamento. Assim, a construção do modelo e a geração de novos indivíduos precisa adequar-se à representação de grafos de decisão. Além disso, a substituição dos indivíduos é feita pelo processo chamado de substituição de torneio restrita (RTR, do inglês, *Restricted Tournament Replacement*). Para cada indivíduo gerado por meio do modelo aprendido é selecionado um conjunto de indivíduos aleatórios da população original (v). Utilizando uma medida de similaridade, o indivíduo mais similar entre o conjunto de indivíduos escolhido aleatoriamente e o novo indivíduo gerado competem entre si. Se o novo indivíduo gerado tiver *fitness* melhor, o indivíduo corrente da população é substituído. Caso contrário, o novo indivíduo não entra na população, permanecendo o antigo (Pelikan et al., 2001). Baseado na experiência de (Pelikan, 2005), o valor adequado para v deve ser igual ao número de variáveis do problema.

Para avaliar o hBOA, várias funções de *benchmark* hierárquicos foram desenvolvidas e avaliadas conseguindo atingir o ótimo rapidamente (Pelikan et al., 2001; Pelikan, 2005). Além disso, o hBOA também foi capaz de encontrar soluções para problemas do mundo real como, por exemplo, o *Ising spin glass* de duas e três dimensões (Naudts & Naudts, 1998; Pelikan & Goldberg, 2003) e instâncias do *Maximum Satisfiability* (MAXSAT) (Yannakakis, 1992). Utilizando o NSGA-II (Deb et al., 2002), o hBOA foi também estendido para problemas multi-objetivo (Pelikan et al., 2005).

Além do hBOA, existem os GAs que utilizam árvores filogenéticas para representar a hierarquia entre as variáveis (Lewis, 1998). O Algoritmo Filogenético, que foi desenvolvido pelo próprio grupo de pesquisa do LCR, adapta o *Neighbor Joining* (Saitou & Nei, 1987) para detectar os níveis hierárquicos (Vargas et al., 2010). Esses algoritmos podem chegar no ótimo com número pequeno de gerações, pois podem criar um modelo adequado na primeira geração.

3.5 Considerações finais

Este capítulo mostrou os principais componentes que caracterizam um EDA, como a obtenção da distribuição de probabilidade dos indivíduos selecionados e a construção de um modelo pro-

babilístico para identificação da interação de variáveis. Embora alguns EDAs não identifiquem o relacionamento entre variáveis, também são considerados EDAs, pois, a partir de uma população de indivíduos promissores, utilizam estatísticas dos valores mais prováveis das variáveis para guiar o processo de busca, sem descartar valores que não parecem ser importantes.

Vários EDAs têm recebido destaque significativo entre os EAs, pois têm sido capazes de resolver problemas do mundo real, que não tinham sido resolvidos por outras técnicas (Hauschild & Pelikan, 2011). Embora a representação discreta (mais precisamente a binária) seja comum nos EDAs, foi mostrado também o funcionamento de EDAs que representam diretamente variáveis contínuas.

Esse tipo de EDA é fundamental para representar os ângulos diedrais para o problema de PSP, foco desta tese. Considerando também que as estruturas de proteínas são organizadas em hierarquias, este capítulo também mostrou o princípio de um EDA que seja capaz de decompor o problema original em subproblemas separados em uma hierarquia. De certa forma, essas técnicas são utilizadas como inspiração no desenvolvimento de uma técnica específica para o problema de PSP.

De uma forma geral, vários aspectos das técnicas apresentadas neste capítulo são utilizados no desenvolvimento de um novo EDA específico para o problema de PSP, conforme explicado a seguir. Considerando que as variáveis do problema de PSP *full-atom* são representadas pelo par de ângulos (ϕ, ψ) de cada aminoácido, pode-se assumir que cada par de ângulos é um BB de tamanho dois. Utilizando o EcGA, BOA ou mesmo o rBOA, a detecção de BBs (ϕ, ψ) de um mesmo aminoácido pode ser evitada se considerar que todo par (ϕ, ψ) de cada aminoácido é um BB por si só, pois estão correlacionados pelo ângulo diedral do próprio aminoácido. O fator escala também pode ser um problema se utilizar EDAs da literatura, pois, em geral, as proteínas precisariam ser representadas utilizando várias dezenas de variáveis. A estrutura de hierarquia presente nas proteínas na natureza também deve ser levada em consideração na elaboração de um EDA específico para PSP, pois a hierarquia da proteína é formada por níveis já estabelecidos na natureza que podem ser incorporados em um EDA.

Em resumo, um EDA para PSP *ab initio* e *full-atom* deve ter: (1) a capacidade de tratar relacionamento de variáveis (como por exemplo o EcGA), (2) capacidade de lidar com números reais (rBOA) e (3) de decomposição do problema em níveis hierárquicos (hBOA). No entanto, ainda não existe um *real-valued hierarchical* BOA ou semelhante para o problema de PSP. Assim, o Capítulo 4 propõe um EDA desenvolvido neste trabalho, de acordo com a demanda do problema de PSP *ab initio* e *full-atom*.

Algoritmos de Estimação de Distribuição para Predição de Estruturas de Proteínas

Este capítulo mostra como foi desenvolvido o novo Algoritmo de Estimação de Distribuição (EDA) específico para o Problema de Predição de Estruturas de Proteínas (PSP) com modelagens puramente *ab initio* e *full-atom*. O ProtPred é um algoritmo desenvolvido inicialmente por Lima & Delbem (2007) para predição de estruturas de proteínas *ab initio* e *full-atom*. O ProtPred utiliza um Algoritmo Genético (GA) para minimizar a energia da proteína. Foi observado no Capítulo 3 que métodos de otimização como os EDAs podem ser capazes de explorar melhor o espaço de busca do que os GAs.

A Seção 4.1 mostra os métodos de referência, ou seja, como foi adaptado alguns algoritmos de busca (Busca Aleatória, Monte Carlo, Algoritmo Genético e Evolução Diferencial) para o problema de PSP para serem comparados com o EDA proposto. Em seguida, três modelos probabilísticos foram desenvolvidos para PSP puramente *ab initio*, produzindo três novos EDAs: *Univariate model-based Optimization* (UNIO), *Kernel Density Estimation model-based Optimization* (KDEO) e *Finite Gaussian Mixture model-based Optimization* (FGMO), descritas na Seção 4.2. A extensão hierárquica desses EDAs é apresentada na Seção 4.3. Por fim, a Seção 4.5 apresenta as considerações finais deste capítulo.

4.1 Algoritmos de referência

Esta seção descreve os algoritmos de referência utilizados para comparar o EDA proposto. Foi utilizado quatro algoritmos de otimização de referência: Busca Aleatória (Seção 4.1.1), Monte Carlo (Seção 4.1.2), Algoritmo Genético (Seção 4.1.3) e Evolução Diferencial (Seção 4.1.4), produzindo quatro diferentes algoritmos de otimização para PSP. Tais algoritmos foram inspirados nas informações apresentadas na Seção 3.2.

4.1.1 Busca Aleatória

A Busca Aleatória (RW) desenvolvida neste trabalho foi inspirada na RW mostrada na Seção 3.2.1. Foi realizado uma adaptação de conceitos comuns de EAs para a RW proposta como, por exemplo, a utilização de indivíduos e população para representar as soluções, o termo *fitness* para representar a aptidão dos indivíduos, o conjunto dos indivíduos filhos e a substituição da população atual pelos filhos utilizando um método de seleção. Assim, a RW desenvolvida pode ser vista como uma metaheurística.

A cada geração um número definido de conformações são construídas de forma completamente aleatória para todos os ângulos da cadeia principal e lateral das conformações de proteínas. Novos indivíduos são gerados e mesclados à população e o melhor indivíduo (com melhor *fitness*) é escolhido como a solução da geração correspondente. Conforme novas gerações são realizadas, o melhor indivíduo tende a ser substituído por outros melhores.

Dessa forma, embora tenha sido implementado o parâmetro que define o tamanho da população na RW, o número máximo de avaliações é o parâmetro mais importante. Pois, utilizando a mesma semente em execuções da RW com diferentes tamanhos de população e mesmo número de avaliações irá sempre produzir o mesmo resultado. A Figura 4.1 mostra um exemplo com três tamanhos de populações diferentes. É possível notar que no final das 200.000 avaliações, todas as três execuções possuem os mesmos valores de energia.

4.1.2 Monte Carlo

O método de MC desenvolvido neste trabalho foi inspirado no método de MC apresentado na Seção 3.2.2. De forma semelhante à RW desenvolvida, o MC desenvolvido também utilizou elementos comuns de um EA como, por exemplo, o encapsulamento de soluções (indivíduos) em uma população, a geração de novas soluções (os filhos), o cálculo da qualidade de cada solução (o *fitness*) e um método de substituição da população atual pelos filhos. Assim, o método de MC proposto, possuindo vários elementos em comum com um EA pode ser considerado uma metaheurística.

O método de Monte Carlo (MC) foi também adaptado para o problema de PSP. Na verdade, foi utilizado também o conceito de Metropolis-Hastings, para controlar a aceitação/rejeição de con-

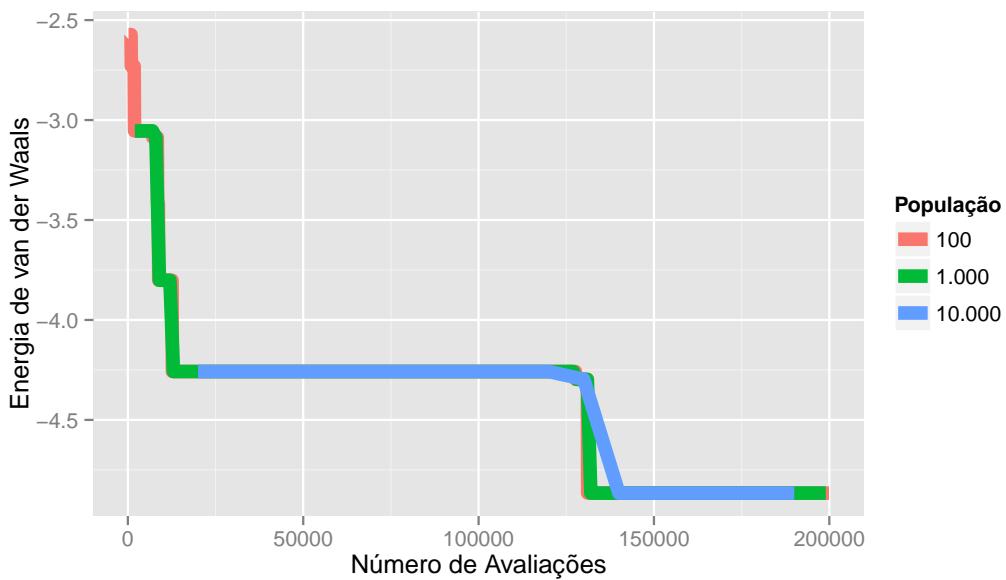


Figura 4.1: Exemplo do processo evolutivo para a RW para com a proteína 1A11.

formações de proteínas. O MC para PSP também foi implementado na forma de um EA, em que a partir de uma população inicial gerada aleatoriamente, os indivíduos recebem uma pequena mutação definida por um parâmetro chamado s_{mc} . Isto é, para cada variável do problema é somado ou subtraído o valor s_{mc} , produzindo uma nova solução. Para cada nova solução gerada é calculado o *fitness* e utilizado o conceito de Metropolis-Hasting, para controlar a aceitação/rejeição, ou seja, a substituição do indivíduo pai na população atual pelo filho ocorre se a Equação 4.1 for satisfeita:

$$\frac{e^{(-(f'_{mc} - f_{mc}))}}{C_1} > u, \quad (4.1)$$

em que f'_{mc} é o *fitness* do indivíduo filho gerado, f_{mc} é o *fitness* do indivíduo pai, C_1 é uma constante como, por exemplo, a constante de Boltzmann, isto é, $C_1 = 0,259$ e u é um número aleatório uniforme gerado no intervalo $[0; 1]$. Isso permite que, mesmo que um filho gerado tenha seu *fitness* inferior ao pai que o originou, o pai é substituído pelo filho, podendo permitir que o processo de busca consiga vencer ótimos locais. Ao término de uma geração, a população terá uma mistura de indivíduos pais e filhos e tal processo é repetido até atingir o critério de convergência, definido por um número máximo de avaliações ou pelo desvio padrão do *fitness* da população ser inferior a um determinado valor.

A Figura 4.2 mostra um exemplo de como o método de MC gera novas soluções quando aplicado ao problema de PSP. A partir de uma conformação de proteína (a conformação pai, cinza), os ângulos diedrais são modificados no sentido $\pm s_{mc}$ produzindo uma nova conformação (o filho, sobreposta à cinza). É possível perceber que mesmo o valor s_{mc} sendo baixo, conformações relativamente diferentes podem ser geradas. Isso pode ocorrer porque as modificações nos ângulos diedrais dos primeiros resíduos podem ser propagadas para os resíduos seguintes. Por exemplo,

primeiro resíduo da nova conformação mostrada na Figura 4.2 é praticamente sobreposto com o da conformação anterior, enquanto que o último resíduo das duas conformações está mais distante.

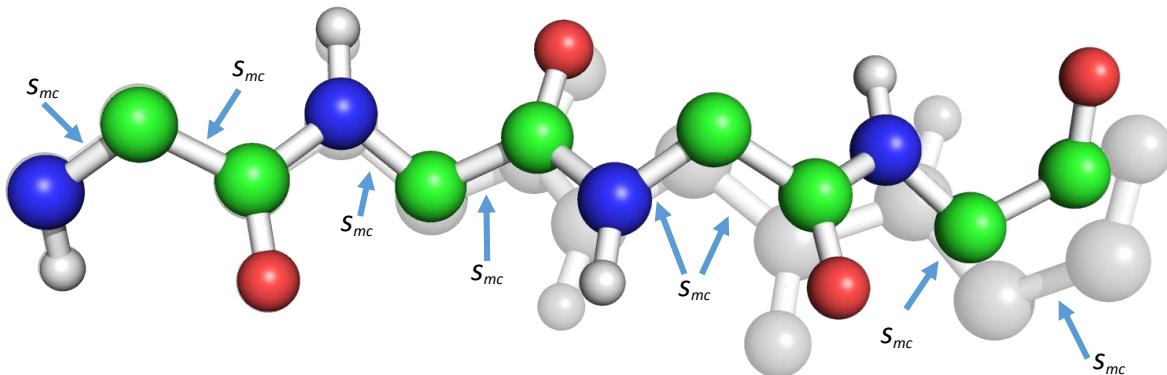


Figura 4.2: Geração de nova conformação utilizando o método de MC para PSP.

A princípio, o MC é melhor do que a RW, pois a partir dos indivíduos gerados aleatoriamente na primeira geração, cada indivíduo é responsável por explorar localmente uma região do espaço de busca até atingir o critério de convergência. Em outras palavras, ao introduzir pequenas modificações nos indivíduos, pode-se dizer que, após o término da execução do MC, cada indivíduo fez uma exploração local do espaço de busca. A RW, entretanto, mesmo podendo amostrar soluções ruins e ao tempo relativamente próximas de regiões promissoras, uma nova solução amostrada pela RW poderá ser amostrada em uma região ruim, descartando-se o fato de já possuir uma solução relativamente próxima de uma região promissora. No entanto, pelo fato da RW não possuir o “conceito de memória”, isto é, a capacidade de evoluir as soluções, será provável que a RW não consiga amostrar um novo indivíduo na região promissora do espaço de busca, mesmo que o indivíduo esteja relativamente próximo de uma região promissora.

4.1.3 Algoritmo Genético

O Algoritmo Genético (GA) baseou-se na abordagem de um GA simples. Diferentemente da versão do ProtPred originalmente proposta por Lima & Delbem (2007), que possui dois operadores de recombinação e três operadores de mutação, o GA simples possui apenas um operador de recombinação e um de mutação.

Existe uma certa relação entre a RW e o MC (no formato metaheurística) com o GA. Em um GA simples, a população é gerada uniformemente aleatória, da mesma maneira que é gerada na primeira geração da RW. A segunda geração da RW também produz indivíduos aleatórios, que são combinados com a população atual para que as piores soluções sejam substituídas pelas melhores soluções entre a população atual e os filhos. Assim, após um certo número de gerações a RW produzirá a mesma solução que um GA seria capaz de produzir se fosse utilizado uma população inicial com o mesmo número de avaliações da RW. Por exemplo, considere uma RW com 10 mil avaliações e com população de 100 indivíduos. Ao final das 10 mil avaliações haverá um certo

conjunto de soluções. No entanto, um GA com população inicial de 10 mil indivíduos pode ser capaz de misturar as soluções e, com os operados genéticos de mutação e recombinação, poderá, em geral, obter soluções mais significativas. O método de MC também pode ser comparado a um GA com apenas mutação, que utiliza taxa de mutação $m_r = 1,0$ (mutação para todos os genes) e fator de mutação $m_f = \pm s_{mc}$, em que s_{mc} poderia ser o passo dado de cada variável no MC. Isso pode ser eficaz na exploração do espaço de busca apenas de ótimos locais. No entanto, os GA, por utilizarem o operador de recombinação, que provoca relativamente grandes mudanças nos valores das variáveis, poderá ter mais chances de sair de ótimos locais.

No GA simples desenvolvido para PSP neste trabalho, a partir de dois indivíduos escolhidos aleatoriamente da população, isto é, dois pais, um novo filho é criado utilizando o operador de recombinação. O ponto de corte é escolhido aleatoriamente dentro do intervalo $[1; d - 1]$, onde d é a quantidade total de ângulos diedrais, ou seja, a quantidade de ângulos diedrais da cadeia principal somada a quantidade de ângulos diedrais das cadeias laterais.

A mutação é controlada pelo parâmetro taxa de mutação (m_r). Para cada gene do novo indivíduo criado com a recombinação dos pais, é gerado um número aleatório uniforme com intervalo $[0; 1]$ e comparado com o valor de m_r . Caso o valor aleatório gerado seja menor do que m_r significa que a mutação de determinado gene irá ocorrer. Para isso, é somado ao valor atual do gene um valor aleatório uniforme dentro do intervalo $[-m_f; +m_f]$, definido pelo fator de mutação m_f .

A Figura 4.3 mostra um exemplo de recombinação e mutação utilizado no GA simples. A partir de dois pais (Pai 1 e Pai 2), um ponto de corte é escolhido aleatoriamente como, por exemplo, a posição entre os ângulos diedrais χ_2^2 e ϕ_3 . Neste exemplo, foi destacado (em verde) os valores dos ângulos diedrais do Pai 1 que foram propagados para o Filho ($\phi_1, \psi_1, \chi_1^1, \phi_2, \psi_2, \chi_2^1$ e χ_2^2) até o ponto de corte, e em azul os valores dos ângulos diedrais do Pai 2 ($\phi_3, \psi_3, \chi_3^1, \chi_3^2, \phi_4, \psi_4, \phi_5, \psi_5$ e χ_2^1) que foram propagados para o Filho, a partir do ponto de corte. É também mostrado um exemplo de mutação no ângulo diedral identificado por ϕ_4 do Filho, alterando o valor de $-155,2$ para $106,0$ graus, destacado em laranja na Figura 4.3(a) correspondente à Figura 4.3(c).

O *fitness* dos novos indivíduos são calculados e em seguida, os indivíduos da população e filhos são mesclados e ordenados pelo valor do *fitness* para que os n melhores (mesmo tamanho da população) substitua todos os indivíduos da população, formando a população da próxima geração (substituição por truncamento). Todo esse processo de recombinação, avaliação e substituição é repetido até atingir o critério de convergência, estabelecido ou pelo número máximo de avaliações ou quando o desvio padrão do *fitness* da população for menor que um valor pré-determinado.

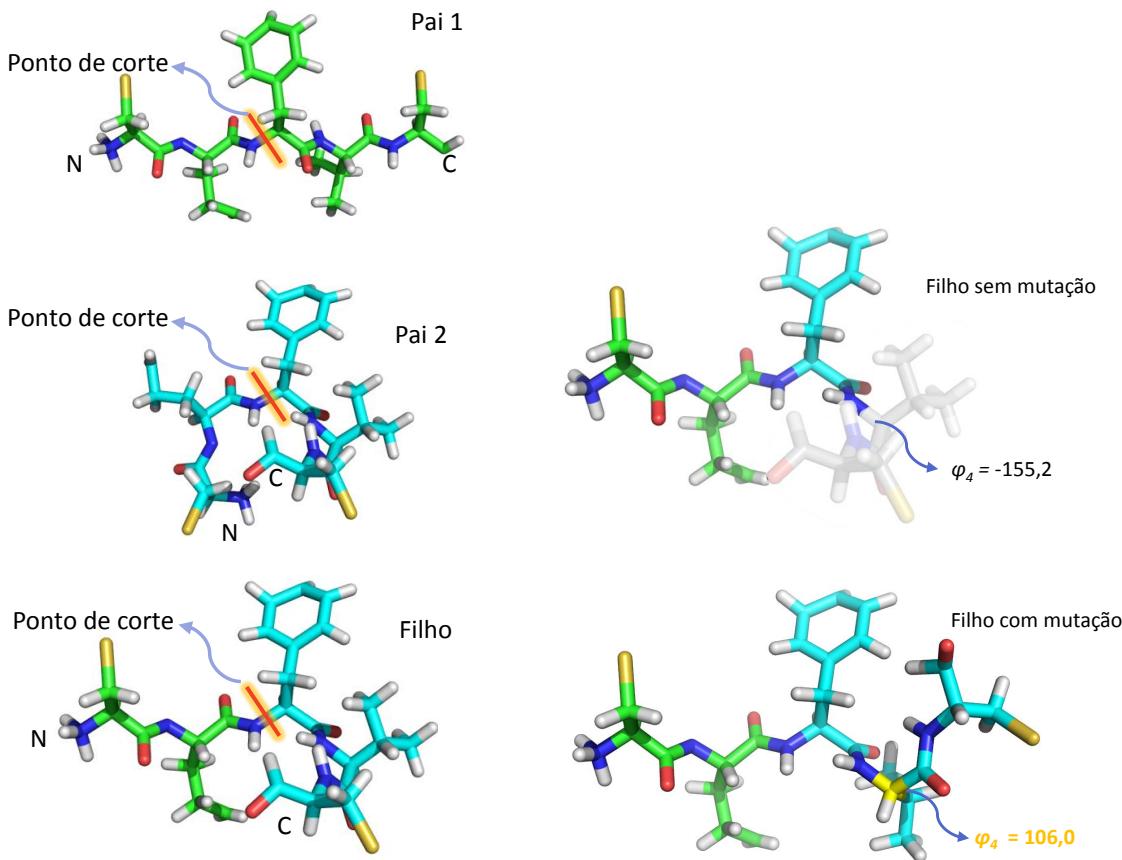
4.1.4 Evolução Diferencial

A Evolução Diferencial (DE) utilizada no algoritmo de PSP neste trabalho é semelhante a um DE simples, dentro das outras variações da DE. Para cada indivíduo da população, um novo indivíduo é criado utilizando o procedimento da DE. As variáveis neste caso, são também representadas pelo conjunto de todos os ângulos diedrais da conformação de uma proteína.

	Ponto de corte																	
	φ_1	ψ_1	X_1^1	φ_2	ψ_2	X_2^1	X_2^2	φ_3	ψ_3	X_3^1	X_3^2	φ_4	ψ_4	X_4^1	φ_5	ψ_5	X_5^1	
Pai 1	-133,8	127,6	-112,8	-144,6	131,3	-123,6	-118,8	-130,7	127,9	-109,8	-124,5	-134,4	136,7	-100,1	-151,9	151,2	-116,9	
Pai 2	-153,3	-56,0	-107,2	-137,8	-24,2	-116,1	-114,7	-144,9	-58,2	-128,8	-120,0	-155,2	-41,0	-114,4	-150,8	-51,6	-109,6	
Filho	-133,8	127,6	-112,8	-144,6	131,3	-123,6	-118,8	-144,9	-58,2	-128,8	-120,0	-155,2	-41,0	-114,4	-150,8	-51,6	-109,6	

↓
106,0 Valor alterado
pela mutação

(a) Vetores de ângulos diedrais (representação computacional das soluções) do Pai 1, Pai 2 e Filho gerado.

(b) Geração do filho a partir da recombinação de dois pais. Neste caso, foi estabelecido o ponto de corte entre as variáveis χ_2^2 e ϕ_3 .(c) Mutação do valor de ϕ_4 de $-155,2$ para $106,0$ graus do filh0.**Figura 4.3:** Exemplo de recombinação e mutação do GA para PSP para a proteína 2P7R (com cinco resíduos). Os valores dos ângulos diedrais mostrados em (a) correspondem às respectivas estruturas mostradas em (b), destacadas pelas cores (verde e azul) bem com o ponto de corte.

Basicamente, a diferença entre a DE com o GA, MC e RW está na maneira com que geram novas soluções. Para compor o novo indivíduo na DE, três indivíduos pais são selecionados diretamente da população. É verificado se a probabilidade da taxa de recombinação c_r é menor do que um valor aleatoriamente gerado no intervalo $[0; 1]$ para cada variável do problema. Se for menor,

é utilizado a Equação 3.4 para gerar o valor da nova variável no indivíduo filho. Em alguns casos, tanto no GA quanto na DE, podem ser gerados valores fora do intervalo definido pelos ângulos diédrais em PSP. Para contornar esse problema, caso a o valor da variável ultrapasse um dos limites inferior ou superior é somado ou subtraído 360 graus ao valor dessa variável para que permaneça dentro do intervalo factível $[-180; +180]$. Caso a condição da taxa de recombinação não seja satisfeita, o valor da variável do indivíduo filho recebe o mesmo valor da variável do indivíduo pai. Esse procedimento é repetido para todas as variáveis e também para todos os indivíduos filhos. De forma semelhante ao GA, o *fitness* dos filhos são calculados e os indivíduos são mesclados juntos com a população. Todo processo de recombinação, avaliação, substituição é repetido até atingir o critério de convergência, o mesmo estabelecido pela RW, MC e GA.

4.2 Algoritmos de Estimação de Distribuição

Os EAs que utilizam um modelo probabilístico sobre um conjunto de soluções promissoras para estimação dos valores das variáveis para que novas soluções sejam amostradas podem ser considerados EDAs. Os modelos probabilísticos dos EDAs são mecanismos que tentam extrair estatísticas relevantes do conjunto das soluções promissoras. Assim, utilizando o modelo probabilístico gerado a partir dos indivíduos promissores, novas soluções poderão ser amostras em regiões promissoras do espaço de busca, pois são regiões em que as soluções promissoras são mais frequentes. No entanto, os modelos probabilísticos de um EDA pode tornar-se o gargalo de um EDA em termos de tempo computacional. Há vários fatores envolvidos na complexidade de um modelo probabilístico de um EDA como, por exemplo, o número de variáveis (dimensão do problema), variáveis que se relacionam e o aspecto da multimodalidade. Pode-se melhorar as chances de sucesso de um EDA se modelos probabilísticos mais adequados forem utilizados. Entretanto, modelos probabilísticos inadequados ou pouco representativos (incapazes de extrair informações relevantes do conjunto de indivíduos selecionados) ao problema podem não ser capazes de explorar o espaço de busca da mesma forma que um GA simples ou DE (ver experimento da Figura 4.5).

Em uma análise preliminar dos dados para o problema de PSP com representação *full-atom*, foi identificado que os dados dos ângulos diédrais possuem distribuição multimodal. Dessa forma, é necessário utilizar um modelo probabilístico que seja capaz de tratar esse tipo de distribuição.

Primeiramente, foi avaliado a possibilidade de se utilizar o UMDA_c como modelo probabilístico para o ProtPred-EDA. No entanto, os resultados não se mostraram satisfatórios, pois o modelo probabilístico criado pelo UMDA_c não foi capaz de estimar corretamente os valores das variáveis para o problema de PSP. Isso ocorreu porque o modelo probabilístico criado pelo UMDA_c possui uma variância relativamente grande e com média próxima de zero. Assim, ao gerar novos indivíduos com média zero e desvio padrão grande produz um efeito semelhante a gerar um valor aleatório uniforme, dificultando encontrar soluções boas, principalmente durante as primeiras gerações, em que a distribuição dos dados é mais uniforme ao longo da faixa de valores admissíveis

$[-180, 0; +180, 0]$. A Figura 4.5 mostra uma comparação do processo evolutivo entre o GA, DE e um EDA utilizando o modelo probabilístico UMDA_c. É possível verificar que neste caso o EDA não foi superior que outras metaheurísticas consideradas mais simples como a DE e GA, pois o UMDA_c não foi capaz de estimar e amostrar corretamente novas soluções.

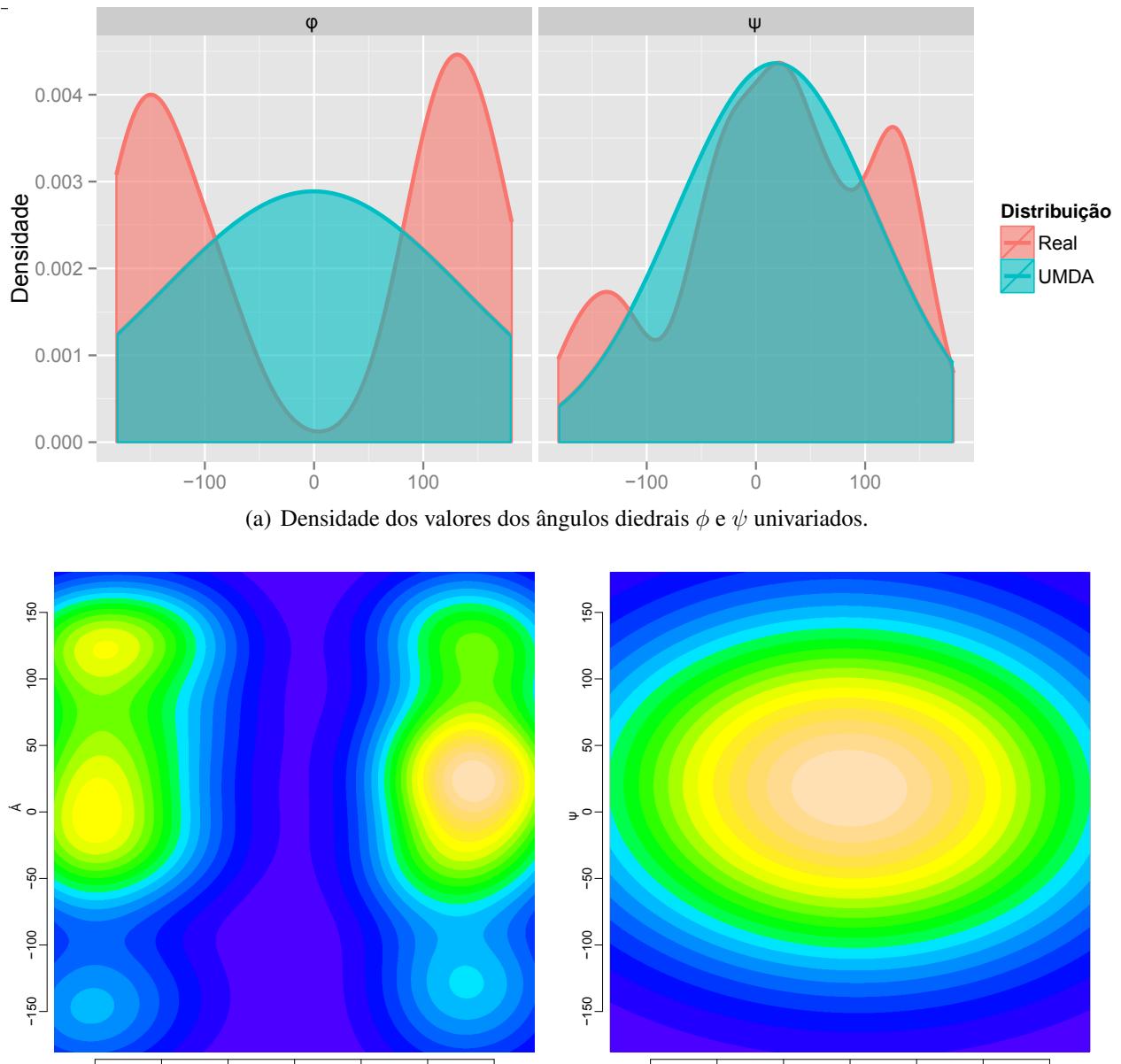
Foi verificado que, de fato, EDAs unimodais podem não ser adequados para o problema de PSP. A Figura 4.4 mostra um exemplo de um modelo probabilístico que trate apenas o aspecto unimodal como, por exemplo, o UMDA_c e BMDA_c. A partir de certa geração do processo evolutivo os valores dos ângulos diedrais ϕ e ψ foram obtidos da população. A Figura 4.4(a) mostra uma comparação entre a densidade real dos dados (obtidas pelo método do kernel) e a densidade calculada utilizando a média e desvio padrão, isto é, os mesmos valores utilizados no modelo probabilístico do UMDA_c. É possível notar que tais valores são bem diferentes da densidade real para o ângulo ϕ . Para o ângulo ψ a diferença é menor, porém ainda não é suficiente para amostrar as três modas presentes na distribuição real. Calculado a probabilidade conjunta (ϕ, ψ) , a estimação utilizando apenas uma moda é ainda menos significante. É possível comparar a distribuição real dos dados (ϕ, ψ) na Figura 4.4(b) com a distribuição estimada utilizando um modelo probabilístico unimodal, porém bivariado (como, por exemplo, o BMDA_c) mostrado na Figura 4.4(c). Nessa comparação é possível verificar que muitos valores impróprios podem ser amostrados utilizando o modelo obtido pela Figura 4.4(c), pois não há nenhuma amostra na população com valores em que $\phi = 0$.

Considerando que as variáveis do problema de PSP são representadas por ângulos que podem assumir valores circulares como, por exemplo, o valor do ângulo $-185,0$ graus, que também pode ser representado por $+175,0$ graus, foi implementado uma distribuição circular normal, também conhecida como distribuição von Mises (Mardia & Zemroch, 1975; Mardia & Jupp, 2008). Esperava-se que a distribuição von Mises fosse capaz de manipular melhor os valores das variáveis que eram gerados fora do intervalo $[-180; +180]$. No entanto, o mesmo efeito do UMDA_c com distribuição normal ocorreu com a distribuição von Mises. Ou seja, ambas as distribuições estimaram a média próxima ao ponto zero e com desvio padrão relativamente grande.

A partir dos dois exemplos do UMDA_c, com distribuição normal e distribuição circular normal, concluiu-se que tais modelos probabilísticos não eram adequados para o problema de PSP *full-atom*. Para que fossem adequados seria necessário haver alguma rotina para identificação das modas e tratá-las como componentes de misturas. Assim, cada mistura teria seus próprios parâmetros (Seção 4.2.3).

Outra opção seria utilizar um EDA mais sofisticado para variáveis discretas como, por exemplo, o EcGA no problema de PSP. No entanto, sabe-se que a precisão dos ângulos diedrais é importante na representação *full-atom*, pois mesmo uma pequena variação no ângulo diedral de uma conformação pode produzir grandes diferenças no valor da energia. Por esta razão, a hipótese de utilizar um EDA discreto para PSP foi descartada.

Assim, considerando somente os EDAs para variáveis reais da literatura foi escolhido o rBOA para ser utilizado no problema de PSP, pois o rBOA tem mostrado resultados relevantes em vá-



(b) Densidade real dos valores do par de ângulos dierdrais ϕ, ψ bivariados. (c) Densidade estimada pela média e desvio padrão dos valores do par de ângulos dierdrais ϕ, ψ bivariados.

Figura 4.4: Comparação entre a densidade real e estimada utilizando modelo probabilístico unimodal.

rias funções de *benchmark* bem como é capaz de tratar relacionamentos de variáveis e o aspecto multimodal. No entanto, a grande quantidade de memória exigida pelo rBOA, mesmo para uma proteína muito pequena, impossibilitou seu uso nos experimentos deste trabalho. Estima-se que mesmo para uma proteína pequena, com 28 resíduos, o rBOA precisa de mais de 2 GB de memória RAM. Além disso, os grafos dos modelos de relacionamento de variáveis no rBOA revelam que muitos relacionamentos (ϕ, ψ) podem ocorrer. No entanto, sabe-se que tal relacionamento é fortemente presente nas estruturas de proteínas, pois alterações em ϕ produzem, em geral, restrições no ângulo ψ , do mesmo resíduo (Apêndice A).

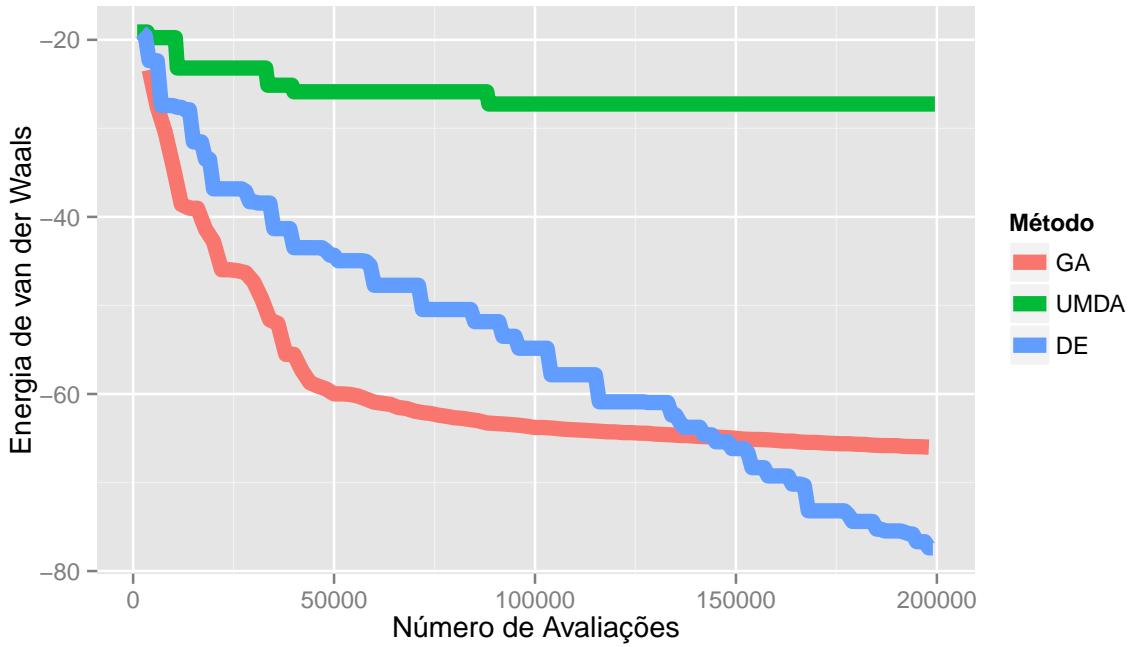


Figura 4.5: Comparação entre o GA, DE e UMDA_c para 200 mil avaliações no problema de PSP com a proteína 1A11.

Por fim, considerando as vantagens e desvantagens dos modelos probabilísticos da literatura e os requisitos necessários para explorar adequadamente o espaço de busca em PSP, foi criado três modelos probabilísticos próprios, específicos para o problema de PSP. Dessa forma, é possível executar o ProtPred-EDA de três maneiras diferentes, utilizando o modelo univariado (UNIO, Seção 4.2.1) ou um dos dois modelos bivariados (KDEO e FGMO, Seções 4.2.2 e 4.2.3).

4.2.1 Univariado

O primeiro modelo probabilístico implementado foi um simples modelo Univariado (UNI), pois não trata relacionamento de variáveis. A vantagem do UNI em relação ao UMDA_c é que o UNI é capaz de tratar o aspecto multimodal, presente nos problemas com vários ótimos locais de uma maneira eficiente, sem a necessidade de estimar parâmetros como, por exemplo, média e desvio padrão. A ideia do UNI é relativamente simples, pois simula o processo do *Kernel Density Estimation* em uma dimensão para cada variável do problema. Sabe-se que construir uma distribuição kernel para todas as variáveis do problema pode ter alto impacto computacional, pois seria necessário iterar sobre todos os valores das variáveis do conjunto dos selecionados e para todas as variáveis do problema.

O método de estimação baseado no método do kernel (KDE, do inglês, *Kernel Density Estimation*) utiliza a soma da diferença entre o ponto S^i em que se quer saber a densidade e todas as observações do mesmo conjunto (S^1, S^2, \dots, S^s), onde s é o tamanho do conjunto de dados. É utilizado um valor de janela (*bandwidth*) definido por h que controla a dispersão de cada ponto. A

ideia dos modelos probabilísticos é estimar a distribuição para depois amostrar novos indivíduos. Assim, para amostrar novos indivíduos utilizando o KDE seria necessário construir a FDP para o conjunto de dados S . Para isso, é necessário dividir a faixa de valores do conjunto de dados S em pedaços e, para cada um desses pedaços deve-se calcular o KDE. Basicamente, esse procedimento não é computacionalmente custoso quando utilizado para estimar apenas uma variável. No entanto, considerando que o KDE esteja sendo utilizado como modelo probabilístico em um EDA seria necessário fazer várias chamadas à função do KDE. Por exemplo, considere certo EDA em que o tamanho do conjunto de selecionados é 1.000 ($s = 1.000$), a quantidade de pontos necessários para construir a FDP sendo 500 e para um problema com 100 dimensões, seria necessário chamar a função KDE 1.000 multiplicado por 500 vezes 100, resultando em 50.000.000 de vezes por geração.

Para manter a eficiência computacional e ainda manter a capacidade de estimação semelhante ao KDE, foi proposto uma extensão da ideia do KDE, mais eficiente. Ao invés de construir uma FDP a partir dos kernels para que em seguida novos valores possam ser amostrados utilizando a FDP, os valores dos novos indivíduos são gerados diretamente a partir de valores extraídos aleatoriamente do conjunto dos selecionados. Em seguida, é somado uma pequena perturbação ao valor da nova variável extraída como, por exemplo, uma distribuição $N(0, 1)$.

Considere o conjunto de selecionados S_j^i onde $i = 1 \dots n$ e $j = 1 \dots d$, onde n é o tamanho do conjunto dos selecionados e d é o número de variáveis do problema. A ideia é preencher todos os valores dos filhos O_j^k onde $k = 1 \dots f$ e f é a quantidade de filhos que serão gerados. Considere um número z inteiro, como sendo amostrado aleatoriamente dentro do intervalo $[1; n]$ para certa variável j . O valor da primeira variável ($j = 1$) do primeiro filho ($k = 1$) será $O_j^k = S_j^z + N(0, 1)$. Esse processo é repetido para todas as variáveis do problema ($j = 1 \dots d$) e também para todos os filhos ($k = 1 \dots f$). Para esse procedimento deu-se o nome de UNI. Em outras palavras, UNI é o modelo probabilístico que pode ser utilizado em um EDA para geração de novas soluções. O Algoritmo 6 mostra um pseudocódigo de como essa técnica funciona.

Algoritmo 6: Pseudocódigo do modelo probabilístico UNI - Gera o conjunto de filhos O a partir do conjunto de selecionados S .

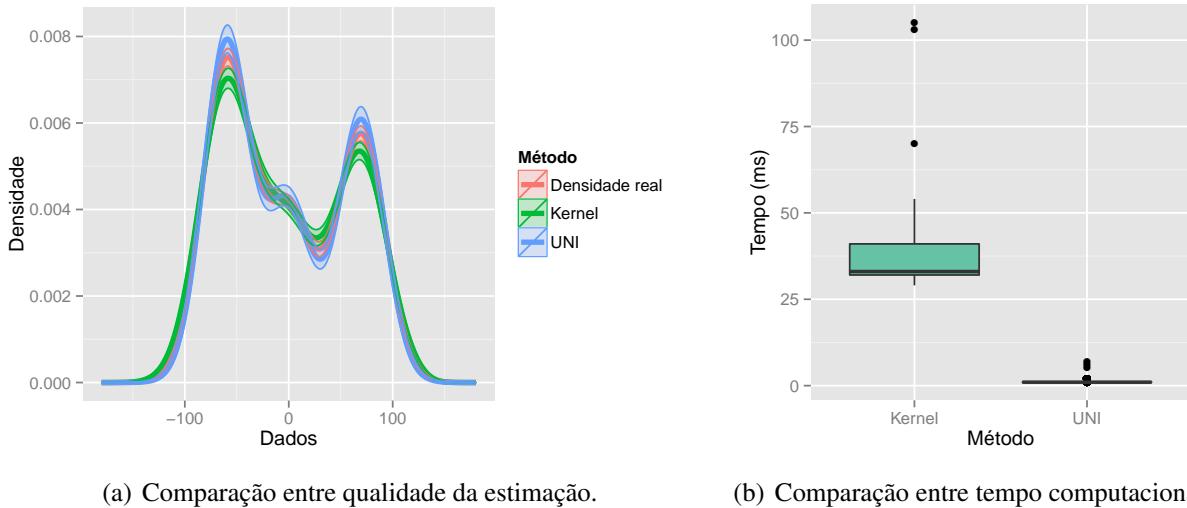
```

1: for  $j = 1$  to  $d$  do
2:    $z \leftarrow$  Amostrar  $f$  valores a partir da distribuição uniforme discreta no intervalo  $[1, n]$ 
3:    $O^j \leftarrow S_j^z + N(0, 1)$ 
4: end for

```

Um experimento foi executado para determinar se o método proposto UNI é equivalente ao KDE, avaliando a qualidade de estimação e a eficiência computacional dos métodos. Neste experimento, notou-se que, de fato, o UNI não tem a mesma precisão de estimação do que o KDE, mas é computacionalmente mais eficiente. Assim, utilizando o UNI é possível obter uma estimativa bem aproximada ao kernel, porém com tempo computacional significativamente menor.

A Figura 4.6 mostra um exemplo da comparação da amostragem utilizando o método proposto UNI e o KDE tradicional. A partir de um conjunto de dados com três modas, novos dados foram amostrados utilizando UNI e KDE. O processo de amostragem de novos valores foi repetido 30 vezes para cada uma das técnicas. Na Figura 4.6(a) a linha vermelha representa a densidade real dos dados, obtida pelo método do kernel. Os valores médios amostrados pelos métodos UNI e KDE estão representados pelas linhas e o desvio padrão está representado pela área suavizada de mesma cor referente ao método. Analisando o tempo requerido entre os métodos kernel e UNI pela Figura 4.6(b) é possível verificar que o UNI possui um tempo de computação relativamente mais baixo que o kernel para amostrar novos dados e possui uma capacidade de amostragem suficientemente boa.



(a) Comparação entre qualidade da estimação.

(b) Comparação entre tempo computacional.

Figura 4.6: Métodos de amostragem univariados.

Ponderando a qualidade de estimação e a eficiência computacional entre UNI e o KDE, decidiu-se implementar o modelo probabilístico UNI no ProtPred-EDA. Isso produziu um novo EDA para PSP, chamado *Univariate model-based Optimization*, abreviado por UNIO. Este pode ser considerado o mais rápido entre os métodos propostos neste trabalho e pode servir também de referência para outros EDAs para PSP, em termos de tempo computacional.

4.2.2 2-D Kernel

Dado que todos os ângulos diedrais (variáveis) em PSP pode ter certa interação, poderia ser criado um modelo probabilístico que tratasse todas as variáveis como correlacionadas, pois uma alteração de um ângulo diedral de um resíduo em certo ponto da proteína, poderia criar restrições em outros ângulos diedrais que estejam afastados na cadeia principal. Para isso, seria necessário criar um modelo probabilístico d -dimensional, tornando o problema bem mais complexo, em que d é o número de variáveis do problema. No entanto, o relacionamento entre ângulos diedrais distantes é, em geral, fraco e pode não ocorrer.

Sabe-se que os ângulos diedrais ϕ e ψ de um mesmo resíduo possuem forte relacionamento, pois alterações em ϕ produz, em geral, restrições em ψ . Além disso, sempre que o diagrama de Ramachandran é mostrado está sendo relacionado o par de ângulos (ϕ, ψ) do mesmo resíduo. Assim, ao invés de utilizar um modelo probabilístico d dimensional que relate todas as variáveis do problema, optou-se por utilizar um modelo probabilístico bidimensional, criando um modelo probabilístico bivariado para cada resíduo em uma proteína.

O primeiro modelo probabilístico bivariado proposto é o KDE de duas dimensões, chamado KDE2D. Neste caso, os valores de ψ são gerados condicional ao valor de ϕ com base na FDP conjunta de (ϕ, ψ) . Primeiramente, um novo valor ϕ' é gerado a partir de uma distribuição univariada de ϕ , utilizando o modelo UNI. Em seguida, é criado um mapa bidimensional a partir da distribuição kernel bivariada para as variáveis (ϕ, ψ) , assim como mostra a Figura 4.7. O número de partições do mapa bidimensional está diretamente relacionado com o desempenho, isto é, a precisão do estimador aumenta conforme o número de divisões, porém diminui o desempenho. Foi estabelecido um total de 150 divisões para cada variável ϕ e ψ . Em seguida, a partir dos valores das divisões de ϕ é escolhido um outro valor de ϕ' na escala das divisões de ϕ que seja o mais próximo de ϕ' . Com o novo valor de ϕ' é possível utilizar a distribuição de ψ para amostrar o valor de ψ' condicional a ϕ' .

Foi utilizado as operações do método do kernel específico para o caso bivariado, para garantir melhor desempenho computacional da estimativa da FDP kernel bivariada. Dessa forma, a estimativa da FDP kernel para o caso bivariado pode ser escrita pela Equação 4.2:

$$\hat{f}_h(x) = \frac{1}{n} \sum_{i=1}^n \frac{1}{h_1 h_2} K\left(\frac{x_1 - X_{i1}}{h_1}\right) K\left(\frac{x_2 - X_{i2}}{h_2}\right), \quad (4.2)$$

onde $K(\cdot)$ é o kernel. Neste trabalho, foi utilizado o kernel normal, como mostra a Equação 4.3:

$$K(x) = (2\pi)^{-1} e^{-0.5x^2} \quad (4.3)$$

e a janela h_j pode ser calculada utilizando a Equação 4.4:

$$h_j = \begin{cases} 4 \cdot 1,06 \cdot \hat{\sigma}_j \cdot n^{-1/5} & \text{se } \hat{\sigma}_j < r_k, \\ 4 \cdot 1,06 \cdot r_k \cdot n^{-1/5} & \text{se } \hat{\sigma}_j \geq r_k, \end{cases} \quad (4.4)$$

em que 4 é o fator multiplicativo (Silverman, 1986; Ripley & Venables, 1994) e r_k é representado pela Equação 4.5:

$$r_k = \frac{Q3 - Q1}{1,34}, \quad (4.5)$$

onde $Q3$ e $Q1$ são o terceiro e primeiro quartis, respectivamente.

Todo esse processo requer mais recursos computacionais em relação a modelos probabilísticos mais simples. Assim, para acelerar o processo evolutivo é verificado se ϕ e ψ possuem distribuição

normal e, caso possuam, é utilizado uma distribuição bivariada normal ao invés do kernel bivariado. Para calcular a normalidade dos dados é utilizado o teste de Anderson-Darling (Kac et al., 1955) com 95% de intervalo de confiança. Se a estatística do teste for inferior a 0,05 o método do kernel não é utilizado. Ao invés disso, é chamado a rotina para amostrar novos dados a partir da distribuição normal. Essa estratégia é interessante porque no começo do processo evolutivo, as variáveis geralmente possuem distribuições com múltiplas modas. No entanto, conforme o decorrer do processo evolutivo, os valores das variáveis tendem-se a permanecer em uma moda. Nesse caso, o método do kernel pode ser substituído pela distribuição normal bivariada, pois poderá ser capaz de amostrar dados com qualidade semelhante, porém de forma mais eficiente. O Algoritmo 7 mostra o funcionamento do modelo probabilístico KDE2D proposto para o ProtPred-EDA.

Algoritmo 7: Amostragem com o KDE2D - Gera os valores (ϕ, ψ) para o filho i .

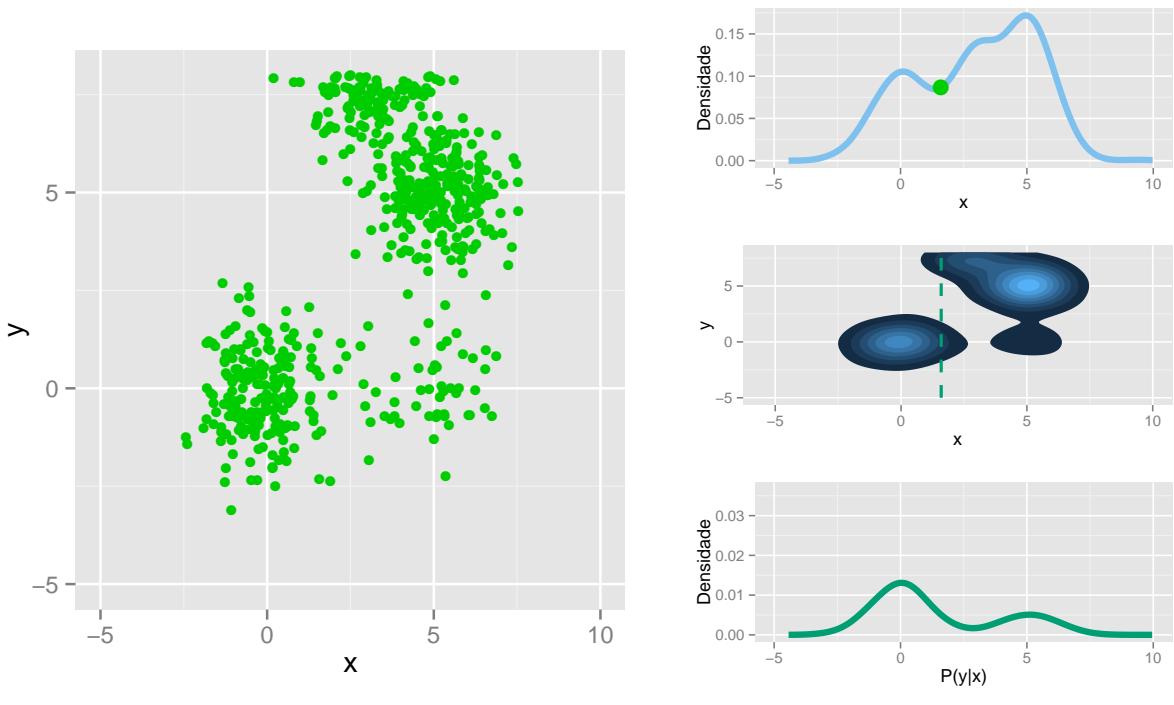
```

1:  $r \leftarrow$  número de resíduos
2: for  $i$  to  $r$  do
3:    $\phi \leftarrow$  Busca o vetor  $\phi$  do resíduo  $i$  de  $S$ 
4:    $\psi \leftarrow$  Busca o vetor  $\psi$  do resíduo  $i$  de  $S$ 
5:   if p-valor do teste de Anderson-Darling para  $[\phi, \psi] < 0,05$  then
6:      $\phi' \leftarrow \text{UNI}(\phi, o)$ 
7:      $P \leftarrow$  2D kernel PDF  $(\phi, \psi)$ 
8:      $\psi' \leftarrow$  Amostrar a partir de  $P$  condicional a  $\phi'$ 
9:   else
10:     $\phi', \psi' \leftarrow$  Amostrar  $o$  valores da distribuição bivariada normal  $N([\mu_\phi, \mu_\psi], \Sigma_{\phi\psi})$ 
11:   end if
12:    $O^i \leftarrow \phi', \psi'$ 
13: end for
  
```

Considere o seguinte exemplo para o KDE2D. De forma similar aos ângulos diedrais (ϕ, ψ) , considere x_i e y_i como sendo dois vetores distribuídos conforme mostra a Figura 4.7(a), em que $i = 1 \dots n$ e n é a quantidade de dados dos vetores. Primeiramente, um novo vetor x' é amostrado a partir da distribuição independente de x (linha 6 do Algoritmo 7). Em seguida, é criado o mapa bidimensional do KDE para o par $[x, y]$ (linha 7). Para cada ponto de x' um novo valor para y' é amostrado condicional ao valor de x' (linha 8). Considere $x'_1 = 1,594476$ e o valor mais próximo no mapa bidimensional como sendo 1,6. Utilizando a probabilidade condicional $x'_1 = 1,6$ é gerado um novo valor y'_1 . Tal exemplo é mostrado na Figura 4.7(b), em que a partir de um ponto verde (acima) em que $x'_1 = 1,594476$, é escolhido o ponto mais próximo no mapa de densidade bivariado (meio). Tal ponto é, então, utilizado como FDP para amostrar o valor de y'_1 condicional a x'_1 (abaixo)¹.

Assim, para o ProtPred-EDA com o modelo probabilístico KDE2D deu-se o nome de *Kernel Density Estimation model-based Optimization*, abreviado por KDEO.

¹Uma animação do funcionamento do KDE2D pode ser vista em:
<http://lcrserver.icmc.usp.br/~daniel/ani/anikde2d/>.



(a) Conjunto de dados de exemplo.

 (b) Criação da FDP condicional. Utilizando a FDP de x (acima) e da FDP kernel bivariada de x, y (meio) é possível encontrar a FDP de x condicional a y (abaixo).

Figura 4.7: Estimação e amostragem com o KDE2D.

4.2.3 Misturas Finitas Gaussianas

Outro modelo probabilístico bivariado também foi implementado no ProtPred-EDA como uma alternativa ao KDE2D. Esse modelo utiliza Misturas Finitas Gaussianas (FGM, do inglês, *Finite Gaussian Mixtures*) (McLachlan & Peel, 2004) para criar as estimativas que serão utilizadas para amostrar novas soluções. Embora tenha sido implementada a versão d -dimensional do FGM, apenas a bivariada foi utilizada. O modelo de FGM combina misturas de distribuições Gaussianas para estimar a densidade de dados com múltiplas modas. Cada componente de mistura k , também chamado de *cluster*, tem sua própria média μ_k , desvio padrão Σ_k e peso da mistura π_k . A partir de um número de misturas K , pretende-se estimar os parâmetros $\theta = [\mu, \Sigma, \pi]$ para todos os componentes de mistura.

O algoritmo *Expectation-Maximization* (EM) (Moon, 1996) é geralmente utilizado para estimar os parâmetros das misturas Gaussianas. Para estimar o parâmetro $\hat{\theta}$ o algoritmo EM utiliza um conjunto de parâmetros como estimativa inicial $\hat{\theta}_0$ e itera entre *E-Step* e *M-Step*. O algoritmo EM converge quando a diferença da máxima verossimilhança for menor que um valor determinado (1, 5 no caso dos experimentos deste trabalho). Considere X como uma matriz de dados d dimensional, o *E-Step* atualiza uma matriz de probabilidades $w_{i,k}$ definida pela Equação 4.6:

$$w_{j,k} = \frac{\hat{\pi}_k |\hat{\Sigma}_k|^{-1/2} \exp \left\{ -\frac{1}{2} (x_j - \hat{\mu}_k)^T \hat{\Sigma}_k^{-1} (x_j - \hat{\mu}_k) \right\}}{\sum_{l=1}^K \hat{\pi}_l |\hat{\Sigma}_l|^{-1/2} \exp \left\{ -\frac{1}{2} (x_j - \hat{\mu}_l)^T \hat{\Sigma}_l^{-1} (x_j - \hat{\mu}_l) \right\}} \quad (4.6)$$

e o *M-Step* atualiza os parâmetros $\hat{\pi}_k$, $\hat{\mu}_k$, $\hat{\Sigma}_k$, definidos pelas Equações 4.7, 4.8 e 4.9, respectivamente:

$$\hat{\pi}_k = \frac{1}{n} \sum_{i=1}^n w_{ik}, \quad (4.7)$$

$$\hat{\mu}_k = \frac{\sum_{i=1}^n w_{ik} y_i}{\sum_{i=1}^n w_{ik}}, \quad (4.8)$$

$$\hat{\Sigma}_k = \frac{1}{\sum_{i=1}^n w_{ik}} \sum_{i=1}^n w_{ik} (y_i - \hat{\mu}_k)(y_i - \hat{\mu}_k)^T. \quad (4.9)$$

Assim, a FDP do método FGM é definida por:

$$f(x|\hat{\theta}) = \left(\sum_{k=1}^K \hat{\pi}_k \Phi \hat{\mu}_k, \hat{\Sigma}_k(X_i) \right), \quad (4.10)$$

onde $\Phi \hat{\mu}_k$, $\hat{\Sigma}_k(x_i)$ é definido pela Equação 4.11 para o caso multivariado:

$$\frac{1}{(2\hat{\pi})^{d/2} |\hat{\Sigma}_k|^{1/2}} \exp \left\{ -\frac{1}{2} (X_i - \hat{\mu}_k)^T \hat{\Sigma}_k^{-1} (X_i - \hat{\mu}_k) \right\}. \quad (4.11)$$

A máxima verossimilhança é calculada utilizando a Equação 4.12:

$$\sum_{i=1}^n \log f(X_i|\hat{\theta}) = \sum_{i=1}^n \log \left(\sum_{k=1}^K \hat{\pi}_k \Phi \hat{\mu}_k, \hat{\Sigma}_k(X_i) \right). \quad (4.12)$$

Para cada par de ângulos diedrais (ϕ, ψ) do conjunto dos indivíduos selecionados é executado um algoritmo completo do EM para um dado número K de componentes de mistura para estimar $\hat{\theta}$. A partir de $\hat{\theta}$, um componente de mistura é aleatoriamente escolhido utilizando uma distribuição uniforme com peso $\hat{\pi}_k$. O componente de mistura k escolhido é utilizado para gerar o par de valores (ϕ', ψ') de uma vez, utilizando um gerador de números aleatórios bivariado definido por $N(\hat{\mu}_k, \hat{\Sigma}_k)$, sendo $\hat{\mu}_k$ um vetor de tamanho 2 e $\hat{\Sigma}_k$ uma matriz 2×2 (caso bivariado). Dessa forma, $\hat{\theta}$ é utilizado para amostrar todos os filhos para determinado resíduo. Para o par de ângulos diedrais do resíduo seguinte na cadeia principal é necessário estimar novamente um novo $\hat{\theta}$ e utilizar esse novo conjunto de parâmetros para amostrar (ϕ', ψ') . Esse processo de estimação e amostragem é repetido para todos os resíduos da proteína.

Por exemplo, considere uma proteína com cinco resíduos. O conjunto dos indivíduos selecionados terá cinco pares (ϕ, ψ) ($\phi_1^i, \psi_1^i; \dots; \phi_5^i, \psi_5^i$), onde i representa o indivíduo. Para o par

(ϕ_1, ψ_1) é calculado o estimador $\hat{\theta}_1$ utilizando o algoritmo EM. Assim, o par (ϕ'_1, ψ'_1) é amostrado utilizando o estimador $\hat{\theta}_1$ a partir da distribuição normal bivariada na forma $N(\hat{\mu}_k, \hat{\Sigma}_k)$. Em seguida os parâmetros de $\hat{\theta}_2$ são estimados a partir de (ϕ_2, ψ_2) e depois são amostrados os novos dados (ϕ'_2, ψ'_2) . Esse processo continua até o quinto resíduo da proteína. É interessante observar que existem cinco algoritmos EM sendo executados, ou seja, cinco algoritmos EM são executados por geração para uma proteína com cinco resíduos.

Devido ao grande custo computacional exigido pelo EM em percorrer todas as variáveis do problema de todas as gerações do processo evolutivo, foi implementado uma versão do EM específica para o caso bivariado onde a Equação 4.11 é substituída pela Equação 4.13, evitando a operação de inversão da matriz (que requer alto custo computacional).

$$f(x_1, x_2) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} \exp\left\{-\frac{z}{2(1-\rho^2)}\right\}, \quad (4.13)$$

em que, em PSP, x_1 e x_2 podem ser representados por ϕ e ψ , e z é definido pela Equação 4.14:

$$z \equiv \frac{(x_1 - \mu_1)^2}{\sigma_1^2} + \frac{2\rho(x_1 - \mu_2)(x_2 - \mu_2)}{\sigma_1\sigma_2} + \frac{(x_2 - \mu_2)^2}{\sigma_2^2}. \quad (4.14)$$

Além disso, antes de executar o método FGM é calculado o desvio padrão do par (ϕ, ψ) do conjunto dos selecionados. Se o desvio padrão de ϕ e ψ for menor que um certo valor como, por exemplo, 0,01, isto é, o mesmo valor utilizado nos experimentos, o algoritmo de FGM é pulado e utilizado uma distribuição Gaussiana bivariada (linha 5 do Algoritmo 8).

Durante as iterações do EM é preciso tratar casos especiais para evitar divisão por zero na Equação 4.13, que pode ocorrer quando há uma grande concentração de valores em uma determinada região e há um valor bem distante dessa concentração. Em geral, isso ocorre quando a população do processo evolutivo está perto de convergir. Para evitar divisões por zero, foi adicionado uma rotina ao algoritmo EM que retorna o último $\hat{\theta}$ válido, caso a Equação 4.13 possua divisão por zero. Assim, esse último $\hat{\theta}$ válido será utilizado para amostrar novos indivíduos. Além disso, para evitar a geração de matrizes singulares foi utilizado a abordagem de Einbeck & Hinde (2006), que utiliza um parâmetro λ para definir a troca de informação entre os componentes de mistura.

O Algoritmo 8 mostra como funciona o método estimação de $\hat{\theta}$ utilizando o FGM. A primeira parte do algoritmo é encontrar o par de ângulos (ϕ, ψ) para determinado resíduo i (linhas 3 e 4) a partir dos indivíduos selecionados S . Em seguida, é verificado se o desvio padrão dos dados (ϕ, ψ) é maior que 0,01 (linha 5) e, caso sejam, é utilizado o EM para estimar $\hat{\theta}$ (linha 6). A amostragem dos novos valores de (ϕ', ψ') (linha 7) é realizada pelo Algoritmo 9. Caso não satisfaça a condição da linha 5 os novos valores de (ϕ', ψ') são gerados utilizando uma distribuição Gaussiana bivariada (linha 9). Por fim, os índices referentes as variáveis (ϕ, ψ) do resíduo i no filho O recebem os novos valores de (ϕ', ψ') .

O Algoritmo 9 mostra como novos dados são amostrados a partir de $\hat{\theta}$. Primeiramente, é gerado um valor aleatório uniforme no intervalo $[0, 1]$ (linha 1). Em seguida, um vetor c_s recebe a soma cumulativa dos valor de $\hat{\theta}_\pi$ (linha 2). A variável k procura pelo componente de mistura que será utilizado para gerar os novos valores (linhas 3, 4 e 5). Por fim, é utilizado a mistura k para gerar f pares de valores (linha 7) por meio da Equação 4.13.

Algoritmo 8: Pseudocódigo do modelo probabilístico FGM - Gera o conjunto de filhos O a partir do conjunto de selecionados S .

```

1:  $r \leftarrow$  número de resíduos
2: for  $i = 1$  to  $r$  do
3:    $\phi \leftarrow$  Busca o vetor  $\phi$  do resíduo  $i$  de  $S$ 
4:    $\psi \leftarrow$  Busca o vetor  $\psi$  do resíduo  $i$  de  $S$ 
5:   if Desvio padrão de  $[\phi, \psi] > 0,01$  then
6:      $\hat{\theta} \leftarrow$  Expectation-Maximization ( $[\phi, \psi], K$ )
7:      $[\phi'; \psi'] \leftarrow$  Amostrar  $o$  valores utilizando o modelo ajustado  $\hat{\theta}$ 
8:   else
9:      $\phi', \psi' \leftarrow$  Amostrar  $o$  valores utilizando a distribuição normal bivariada  $N([\mu_\phi, \mu_\psi], \Sigma_{\phi\psi})$ 
10:  end if
11:   $O^i \leftarrow [\phi'; \psi']$ 
12: end for
```

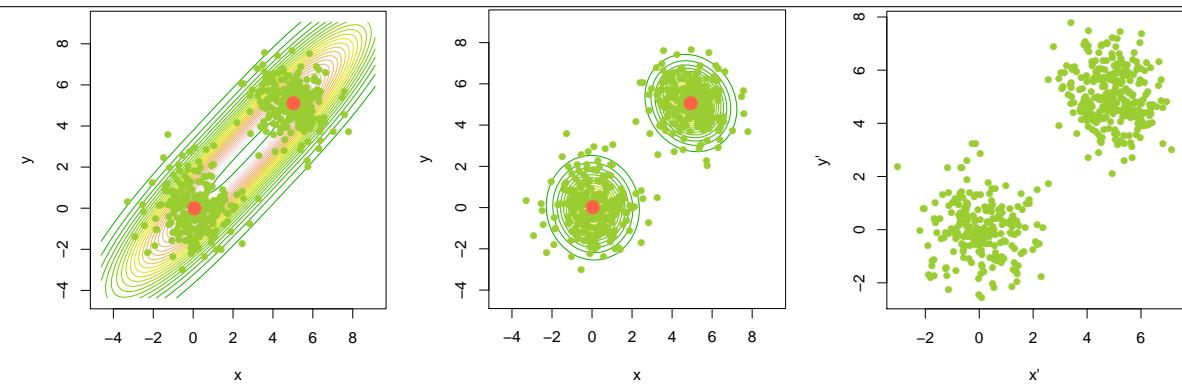
Algoritmo 9: Pseudocódigo do modelo probabilístico FGM - Amostragem.

```

1:  $x \leftarrow$  Amostrar um valor da distribuição  $U(0, 1)$ 
2:  $c_s \leftarrow$  Soma cumulativa de  $\hat{\theta}_\pi$ 
3:  $k \leftarrow 1$ 
4: while  $x < c_{s_k}$  do
5:    $k \leftarrow k + 1$ 
6: end while
7:  $s \leftarrow$  Amostrar de  $N(\hat{\theta}_{\mu_k}, \hat{\theta}_{\sigma_k})$ 
```

A Figura 4.8 ilustra um exemplo de como o FGM funciona. Na primeira iteração os parâmetros são definidos de acordo com a condição inicial. Com o decorrer das iterações do EM os parâmetros tendem a convergir (Figura 4.9(b)). Após a convergência dos parâmetros, os mesmos são utilizados para gerar novos indivíduos (Figura 4.8(c)).

Por fim, o ProtPred-EDA com o modelo probabilístico FGM recebeu o nome de *Finite Gaussian Mixtures model-based Optimization*, abreviado pela sigla FGMO. A princípio, o tempo de computação do método FGMO está diretamente relacionado com o número de iterações do EM e com o número de misturas K .



(a) Primeira iteração do algoritmo FGM.
 (b) Última iteração do algoritmo FGM.
 (c) Novos valores amostrados.

Figura 4.8: Estimação e amostragem com o FGM.

4.3 Hierárquico

A versão original do ProtPred obteve sucesso em previsões de proteínas pequenas, com algumas dezenas de aminoácidos. Sabe-se que a quantidade de aminoácidos utilizados em PSP está diretamente relacionada com a dificuldade do problema. Proteínas mais pequenas possuem menos variáveis para minimizar (menos graus de liberdade) enquanto que proteínas maiores possuem mais variáveis (mais graus de liberdade). Mesmo utilizando modelos probabilísticos mais elaborados, específico para o problema de PSP (conforme mostrado na seção anterior) a dificuldade em minimizar a energia de proteína aumenta de acordo com a quantidade de aminoácidos. Por exemplo, as proteínas a partir de certo tamanho tendem a formar dobramentos isolados. Esses dobramentos isolados podem interagir entre si, porém de uma forma mais fraca. No entanto, tratando toda a cadeia da proteína como um único problema, pedaços da proteína que deveriam ser tratados isoladamente começam a sobrepor-se. Isso prejudica o processo evolutivo, pois podem impedir que subestruturas sejam formadas.

Para lidar com esse problema, foi desenvolvido uma extensão hierárquica do EDA proposto específica para o problema de PSP. O EDA hierárquico em si não possui modelo probabilístico específico, isto é, não estima e amostra novas soluções. O EDA hierárquico precisa de modelos probabilísticos como, por exemplo, os apresentados na Seção 4.2 para ser capaz de estimar e amostrar novas soluções. Diferentemente de um EDA comum, o EDA hierárquico divide o problema em subproblemas menores utilizando um método de agrupamento hierárquico e tenta tratar cada subproblema de forma independente. Em seguida, utiliza a solução de cada subproblema para tentar resolver o problema inicial. Assim, a interação entre os subproblemas é realizada somente entre as junções entre eles, impedindo que subproblemas já resolvidos modifiquem outros subproblemas.

Basicamente, o EDA hierárquico consiste em quatro passos. O primeiro passo (1) consiste em criar uma população inicial P gerada utilizando uma distribuição aleatória uniforme. A partir da população P é aplicado um método de seleção produzindo o conjunto de selecionados S .

No segundo passo (2), é executado uma rotina de agrupamento hierárquico utilizando os dados de S . Nesta etapa, as variáveis do problema representadas em ângulos diedrais são convertidas em uma matriz de coordenadas Cartesiana A com três colunas e a número de linhas, onde a é o número total de átomos da conformação da proteína. A matriz A é filtrada pelos $C\alpha$'s da cadeia principal e, assim, ao invés de a linhas a matriz A filtrada terá r linhas, onde r é o número de resíduos da proteína. A estrutura de dados D conterá o dendrograma com as informações sobre os agrupamentos. Essa estrutura de dados é construída a partir da matriz A utilizando um método de agrupamento hierárquico (Hartigan, 1975). Tal método de agrupamento hierárquico é o mesmo utilizado pela função *hclust* do pacote *stats* da Linguagem R (R Core Team, 2014). Essa função foi implementada em C para manter a eficiência de todo o algoritmo. O parâmetro m define a quantidade de subproblemas que será criado a partir de D . No caso $m = 2$, o dendrograma é dividido em dois grupos utilizando um método de corte, produzindo dois subproblemas. O método de corte utilizado é o mesmo utilizado pela Linguagem R do mesmo pacote *stats*, chamado *cutree* (Becker et al., 1988).

A Figura 4.9 mostra um exemplo da criação de dois subproblemas a partir do Dendrograma da proteína nativa 1A11. Foi também adicionado uma rotina para impedir que sejam criados subproblemas de tamanho um, pois não é possível tratar um subproblema com apenas um resíduo. Por fim, V_k^p é responsável por armazenar todos os agrupamentos, onde p é a identificação de cada subproblema, k é um conjunto de vetores que contém os índices dos resíduos de cada subproblema e $k = 1 \dots v$, onde v é o tamanho do subproblema p .

Em seguida, cada subproblema é independentemente otimizado no passo (3). Para cada subproblema k é executado duas otimizações, utilizando os processos comuns de um EDA (criação da população inicial, avaliação, seleção, estimação, amostragem, substituição e assim por diante). Nesta etapa, é informado qual o modelo probabilístico será utilizado para a execução do EDA com os subproblemas. Na primeira execução da otimização é utilizado uma população gerada aleatoriamente e a segunda utiliza a mesma população P , porém utiliza somente as variáveis referentes ao subproblema que está sendo otimizado. Assim, cada EDA que está otimizando um subproblema terá um número de variáveis menor do que o problema original. A Figura 4.10 mostra um exemplo da divisão de certa proteína em dois subproblemas e a ligação entre eles. Foi adicionado um parâmetro para definir o número máximo de avaliações e_{max} para o subproblema. Considerando um problema com tamanho $d = 25$ dividido em dois subproblemas $V_{1,2,3,\dots,12}^1$ e $V_{13,14,15,\dots,25}^2$ o EDA para PSP é executado quatro vezes. Os melhores indivíduos de cada subproblema (soluções parciais) que utilizam a população P como inicial são armazenados em U_p e os melhores indivíduos dos subproblemas que utilizam população aleatória são armazenados em W_p , para o subproblema p .

O último passo (4) utiliza os subproblemas resolvidos U e W para otimizar o problema original. Para isso, todos os subproblemas de U são combinados com os subproblemas de W . No caso em que $m = 2$ resultará em quatro combinações possíveis de população inicial nova: $L_1 \leftarrow U_1 \cup U_2$; $L_2 \leftarrow U_1 \cup W_2$; $L_3 \leftarrow W_1 \cup W_2$ e $L_4 \leftarrow W_1 \cup U_2$. Para montar todas as combinações possíveis da

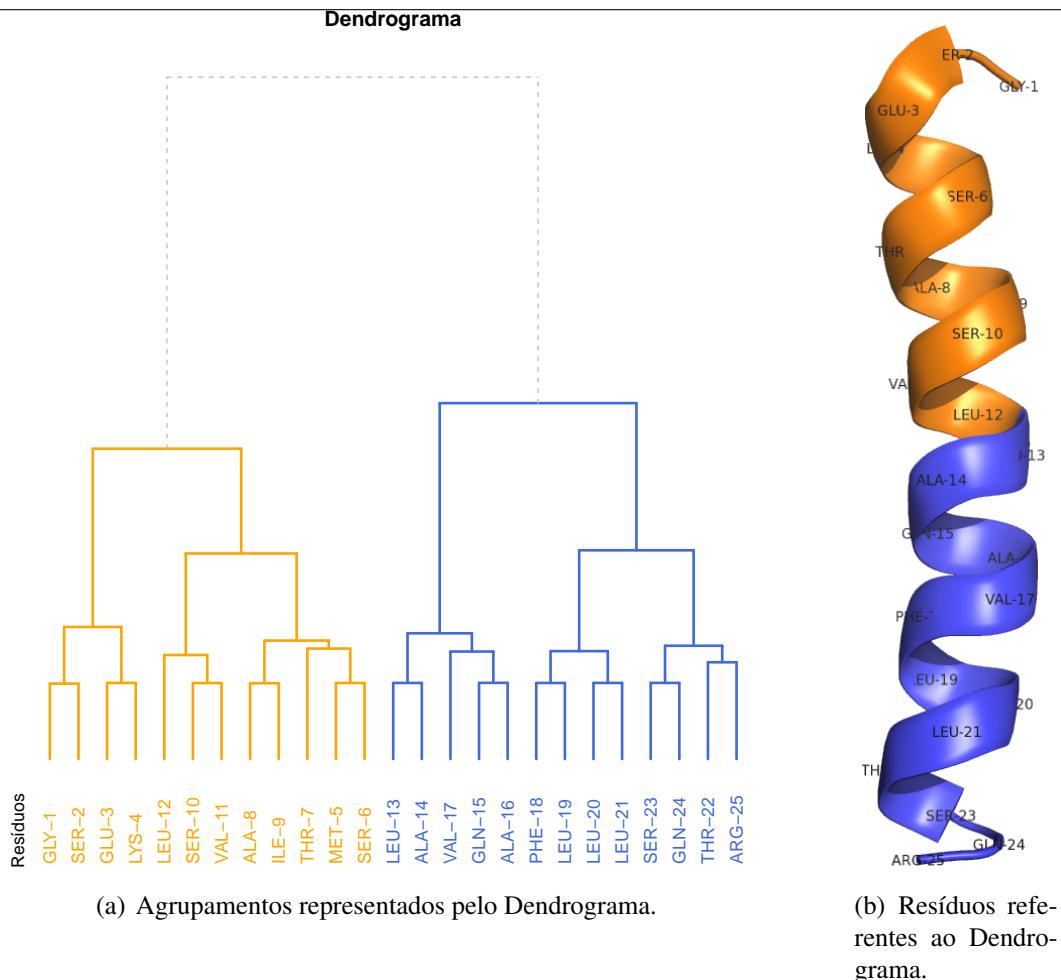


Figura 4.9: Agrupamento hierárquico dividido em dois grupos ($m = 2$) para a proteína nativa 1A11. É possível notar que os resíduos mais próximos permaneceram em um subproblema (1 – 12) enquanto que os outros resíduos (13 – 25) estão em outro subproblema.

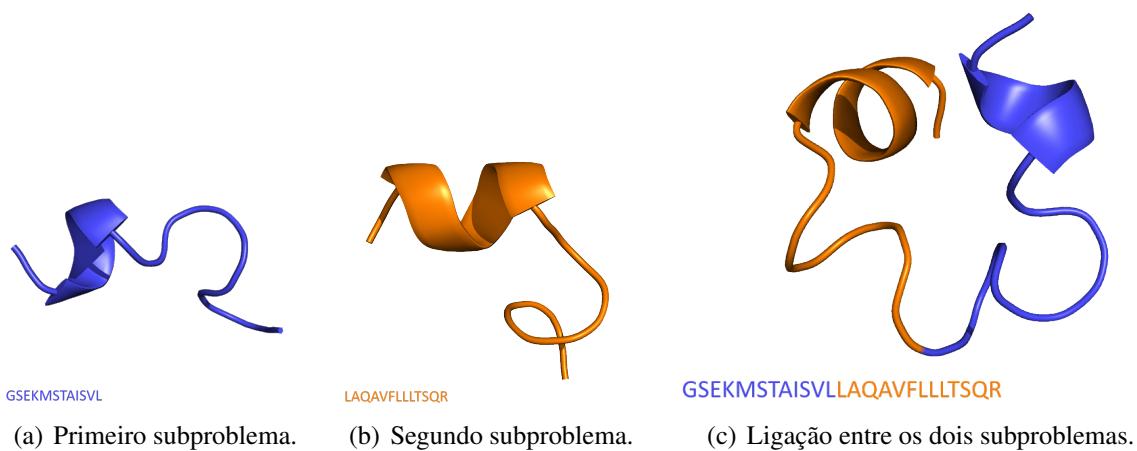
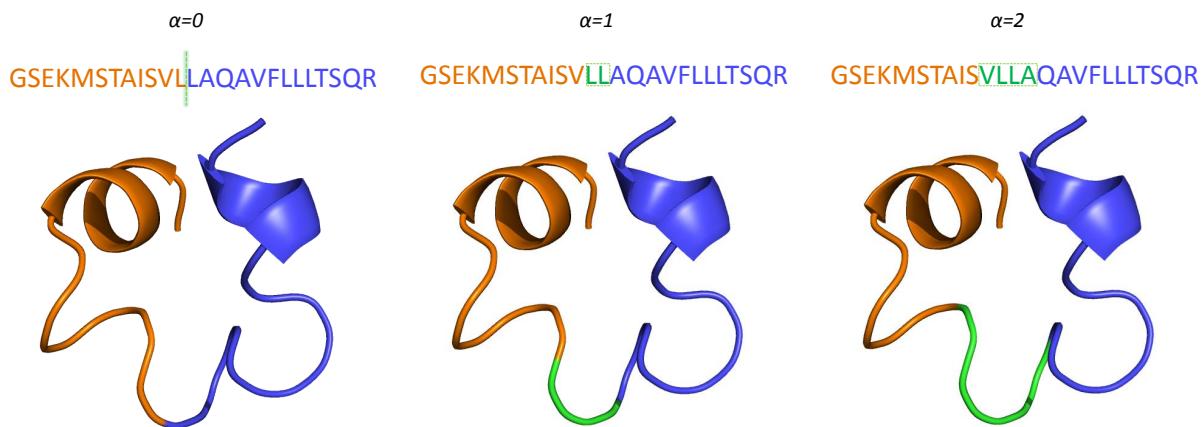


Figura 4.10: Exemplo da representação de união de dois subproblemas. Cada trecho da proteína é representado por um subproblema. Indicado também a sequência de aminoácidos de cada subproblema para uma conformação da proteína 1A11.

população inicial de L foi utilizado uma tabela binária de combinações de acordo com o número de subproblemas, sendo U representado pelo valor 0 e W representado pelo valor 1 (ver Tabela 4.1). As variáveis utilizadas nas conexões entre os subproblemas são determinadas utilizando um parâmetro chamado α . Caso α seja zero significa que os subproblemas são concatenados apenas. Caso contrário, as conexões entre os subproblemas são substituídas pelas respectivas variáveis de P a quantidade de vezes definido em α para esquerda e para a direita, a partir da junção dos subproblemas. Por exemplo, assumindo que α seja igual a dois e a conexão entre os dois subproblemas ocorra entre as variáveis 12 – 13, significa que as variáveis 11, 12, 13, 14 de L serão substituídas pelas mesmas variáveis da população inicial P . A Figura 4.11 mostra um exemplo com três tipos de sobreposição. É interessante observar que todas as outras variáveis em L que não sejam as variáveis de conexão terão os mesmos valores dos indivíduos das respectivas populações finais de cada subproblema. Por fim, cada população L é utilizada para executar uma instância do EDA completo. Neste EDA, pode ser utilizado o mesmo modelo probabilístico utilizado no EDA utilizado para resolver os subproblemas. Ao final da execução dos EDAs referentes aos problemas completos, haverão 2^m melhores indivíduos, um de cada combinação dos subproblemas, que serão utilizados como solução do problema original. A Figura 4.12 mostra um esquema de como o EDA hierárquico funciona, incluindo todos os passos (1-4).



- (a) Sem sobreposição. Os subproblemas são apenas unidos para compor a solução ao problema original.
- (b) Com sobreposição em que $\alpha = 1$. Neste caso, os valores dos ângulos diédrais referentes aos resíduos *LL* são edrais referentes aos resíduos *VLLA* utilizados da população do problema são utilizados da população do problema inicial.
- (c) Com sobreposição em que $\alpha = 2$. Neste caso, os valores dos ângulos diédrais referentes aos resíduos *LL* são edrais referentes aos resíduos *VLLA* utilizados da população do problema são utilizados da população do problema inicial.

Figura 4.11: Exemplo de sobreposição de subproblemas.

O EDA hierárquico consome mais tempo computacional do que o EDA comum (com os respectivos modelos probabilísticos), pois para cada uma das melhores soluções obtidas podem ser feitas 2^m chamadas à função objetivo. No entanto, acredita-se que utilizando o mesmo número de avaliações do EDA hierárquico, os resultados do EDA comum ainda seriam inferiores. Por exemplo, considere um EDA para PSP com número máximo de avaliações definido em um milhão. Considere $2^m = 2^2 = 4$ e, neste caso, o EDA hierárquico que utiliza o EDA com um milhão de

Tabela 4.1: Combinação de subproblemas criada a partir da combinação da tabela binária em que 0 representa U e 1 representa W .

Subproblema 1	Subproblema 2	Subproblema 1	Subproblema 2
0	0	U_1	U_2
0	1	U_1	W_2
1	0	W_1	U_2
1	1	W_1	W_2

avaliações, será necessário quatro milhões de avaliações no total para que o EDA hierárquico seja executado completamente. No entanto, utilizando somente um EDA comum com quatro milhões de avaliações acredita-se que não seja capaz de minimizar a energia melhor do que o EDA hierárquico. Isso pode ser especialmente favorável para proteínas acima de 50 aminoácidos, tamanho que deverá favorecer a formação de subestruturas.

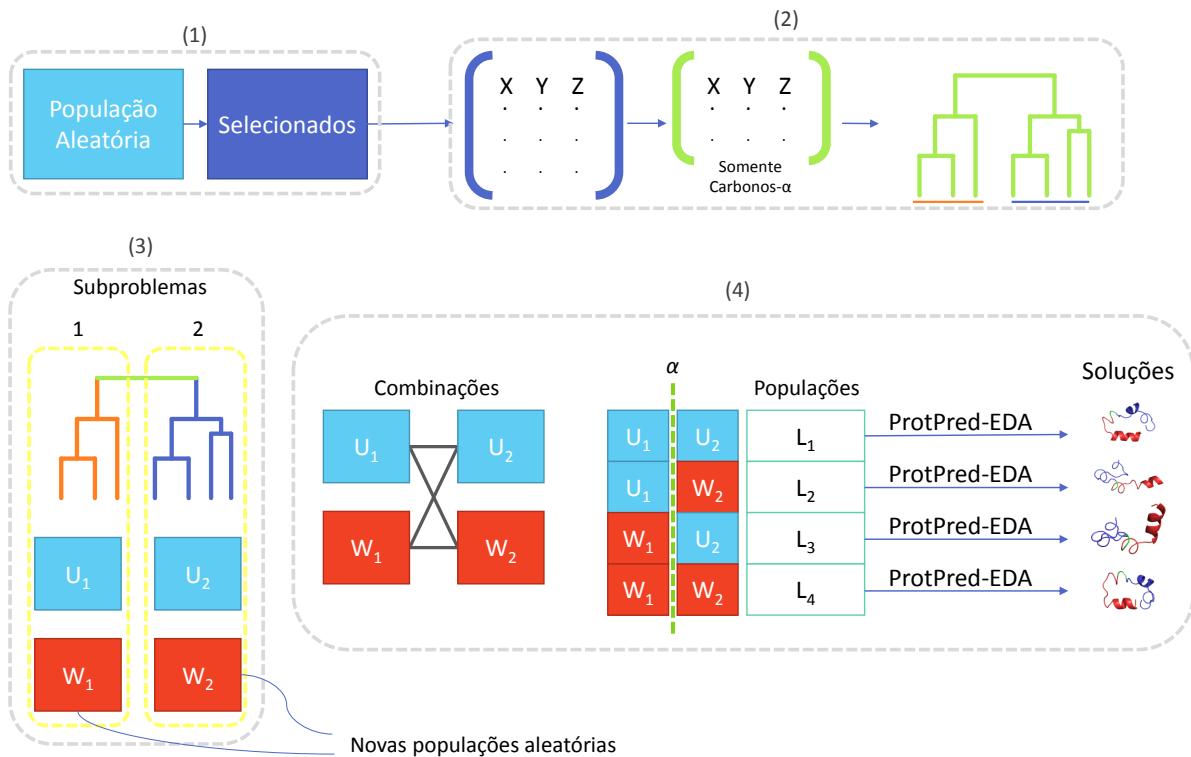


Figura 4.12: Esquema de funcionamento do EDA hierárquico: (1) gerado uma população aleatória e criado o conjunto dos selecionados; (2) convertidos os ângulos diedrais em coordenadas Cartesianas, filtrado pelos Carbono- α 's e agrupado os dados hierarquicamente; (3) resolvido os subproblemas de forma independente e; (4) combinado os subproblemas e executado uma instância do EDA comum para cada combinação.

Para cada combinação do EDA hierárquico com um dos três modelos probabilísticos propostos (UNI, KDE2D e FGM) deu-se o nome de: (1) hUNIO, que significa *hierarchical Univariate model-based Optimization*, quando combinado com o modelo probabilístico UNI; (2) hKDEO,

que significa *hierarchical Kernel Density Estimation model-based Optimization*, quando combinado com o modelo probabilístico KDE2D; e (3) hFGMO, que significa *hierarchical Finite Gaussian Mixtures model-based Optimization*, quando combinado com o modelo probabilístico FGM. Dessa forma, o EDA hierárquico é representado de acordo com o modelo probabilístico que utiliza (hUNIO, hKDEO e hFGMO).

4.4 Energia de Solvatação em GPU

O cálculo da energia de solvatação exige mais recursos computacionais do que a energia de van der Waals. Embora a energia utilizada na função de avaliação deste trabalho seja somente a energia de van der Waals, surgiu a oportunidade para integrar uma versão do cálculo da energia de solvatação eficiente, que utiliza lista de células e GPU (*Graphic Processing Units*) proposta por Zhang et al. (2013). Assim, a energia de solvatação do ProtPred-EDA foi substituída pela proposta por Zhang et al. (2013).

A complexidade do cálculo da energia de solvatação utilizada no ProtPred-EDA passou de $O(n^2)$ para $O(n)$ utilizando as mesmas rotinas utilizadas no pacote MURCIA (Quiñonero et al., 2011). Foi utilizado ambas as versões em CPU e em GPU do pacote MURCIA, permitindo realizar comparações de tempo computacional entre as duas versões. Foi validado o valor do cálculo da energia de solvatação entre o ProtPred-EDA com as versões em CPU, GPU e também com o próprio MURCIA. Foi verificado que todas as versões produzem o mesmo valor de energia para uma mesma conformação, isto é, os valores das energias estão de acordo. Para calcular a energia de solvatação, foi utilizado o procedimento descrito na Seção 2.5.2. A partir da SASA calculada de cada átomo é associado o parâmetro ASP. Assim, multiplicando e depois somando o ASP de cada átomo com a área SASA resulta na energia de solvatação, que pode ser utilizada na função de *fitness*.

As proteínas são representadas utilizando ângulos diedrais no ProtPred-EDA, no entanto, o pacote MURCIA utiliza coordenadas Cartesianas. Assim, foi necessário ajustar as estruturas de dados entre o ProtPred-EDA e as funções extraídas do pacote MURCIA para que fosse possível realizar o cálculo da energia de solvatação. Além da conversão para coordenadas Cartesianas é também passado os valores dos parâmetros ASP utilizado no cálculo.

Existem duas versões do MURCIA que calculam a energia de solvatação, chamadas *kernel 1* e *kernel 2*. A diferença entre elas é que a *kernel 1* precisa iterar sobre todos os átomos para encontrar os vizinhos e a *kernel 2* utiliza um algoritmo mais elaborado, com interpolação de *hardware* com memória de textura da GPU. Assim, o ProtPred-EDA recebeu a versão *kernel 2* do pacote MURCIA (Figura 4.13).

Durante os primeiros experimentos com as energias van der Waals e de solvatação notou-se que as energias não possuem mesma escala. Por exemplo, enquanto a energia de van der Waals está em torno de -100 kcal/mol, a energia de solvatação pode estar na faixa de $+6.000$ kcal/mol. Isso

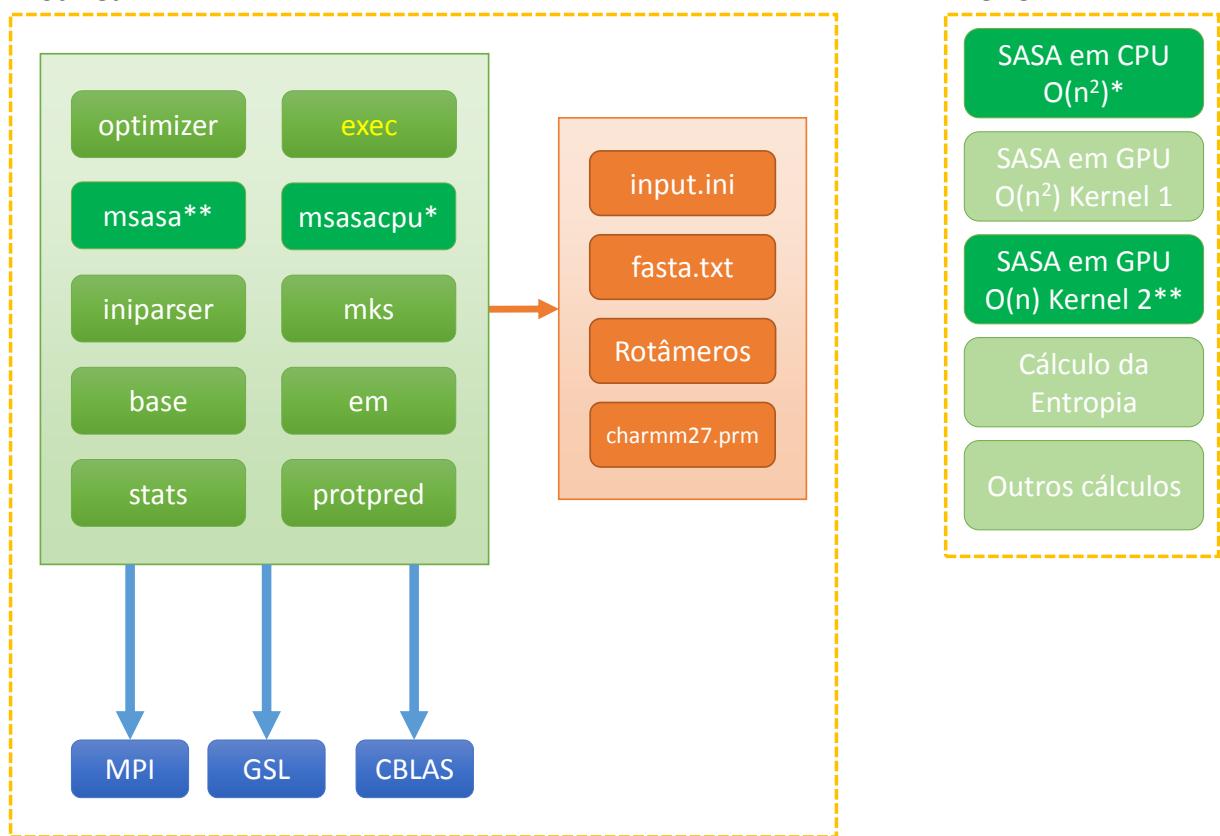


Figura 4.13: Ilustração entre o ProtPred-EDA e MURCIA. Foi destacado que o cálculo da energia de solvatação em CPU (SASA) do MURCIA foi convertido para uma biblioteca no ProtPred-EDA chamada *msasacpu* e o mesmo cálculo, porém para GPU e utilizando a versão *kernel 2*, foi convertido para a biblioteca chamada *msasa*.

torna o problema mais complexo de resolver pois, o processo de determinação de pesos adequados para cada energia já é um problema difícil em si. Além disso, para cada tamanho de proteína existe uma diferença de escala. De certo modo, isso confunde o processo evolutivo, pois, o *fitness* das conformações das proteínas é a soma ponderada de cada energia. Se o valor de uma energia é muito maior que o valor de outra energia, o processo evolutivo pode tender a energia de maior valor, neste caso, a energia de solvatação e desfavorecer a energia de van der Waals.

Embora alguns experimentos tenham sido executados utilizando a energia de solvatação, espera-se que uma versão multi-objetivo do ProtPred-EDA seja capaz de manipular adequadamente duas ou mais energias, mesmo em escalas diferentes. Por esse motivo, experimentos com energia de solvatação não foram totalmente desempenhados neste trabalho, deixando essa tarefa para uma instância futura.

4.5 Considerações Finais

Este capítulo mostrou as técnicas utilizadas e desenvolvidas neste trabalho de doutorado. Mostrou que, em primeiro lugar, antes de desenvolver o objetivo principal deste trabalho foi necessário realizar algumas etapas preliminares para certificar-se de que o objetivo principal poderia ser bem desenvolvido. Assim, diversos processos do algoritmo de PSP foram melhorados como, por exemplo, gerenciamento de memória, estrutura do código-fonte, elaboração de bibliotecas, melhorado a passagem de parâmetros, utilização de MPI e validação dos aminoácidos. Tudo isso garantiu uma maior flexibilidade do algoritmo, permitindo que seja estendido ou que seja rapidamente encontrado e corrigido problemas. Foi descrito também os algoritmos de referência para PSP implementados como, a Busca Aleatória, Monte Carlo, Algoritmo Genético e Evolução Diferencial.

Neste capítulo é também apresentado as técnicas que foram desenvolvidas e implementadas para atingir o objetivo principal deste trabalho: o desenvolvimento de EDAs aplicados ao problema de PSP *ab initio* e *full-atom*. Foram propostos três diferentes modelos probabilísticos que foram implementados na nova ferramenta para PSP, chamada ProtPred-EDA. Cada modelo probabilístico tem sua própria característica. O mais simples, UNI, não trata relacionamento de variáveis, mas é capaz de tratar múltiplas modas. Para o EDA com o modelo probabilístico UNI, deu-se o nome de UNIO. Os modelos probabilísticos KDE2D e FGM tratam relacionamento de variáveis entre os ângulos (ϕ, ψ) do mesmo aminoácido e também lidam com o aspecto multimodal. Para o EDA com os modelos probabilísticos KDE2D e FGM deu-se o nome de KDEO e FGMO, respectivamente. Todos os três EDAs desenvolvidos para PSP (UNIO, KDEO e FGMO) foram implementados levando em consideração o aspecto eficiência computacional. Por fim, pensando em lidar com proteínas com mais de 50 aminoácidos foi desenvolvido o EDA hierárquico, capaz de dividir o problema original em subproblemas e tratá-los de forma independente. O EDA hierárquico não tem um modelo probabilístico em si para estimação e amostragem de novos indivíduos. Ao invés disso, ele pode ser combinado com os modelos probabilísticos elaborados como o UNI, KDE2D ou FGM, produzindo o hUNIO, hKDEO e hFGMO. A Figura 4.14 mostra um esquema da combinação entre o EDA comum e o EDA hierárquico produzindo os métodos UNIO, KDEO, FGMO, hUNIO, hKDEO e hFGMO para PSP. Por fim, como parte do objetivo secundário deste trabalho foi descrito como foi melhorado a eficiência do cálculo da energia de solvatação do ProtPred-EDA a partir de um trabalho colaborativo.

No próximo capítulo, são mostrados os resultados obtidos por meio de experimentos realizados com as técnicas propostas (UNIO, KDEO, FGMO, hUNIO, hKDEO e hFGMO) e comparadas com os algoritmos de referência (RW, MC, GA e DE).

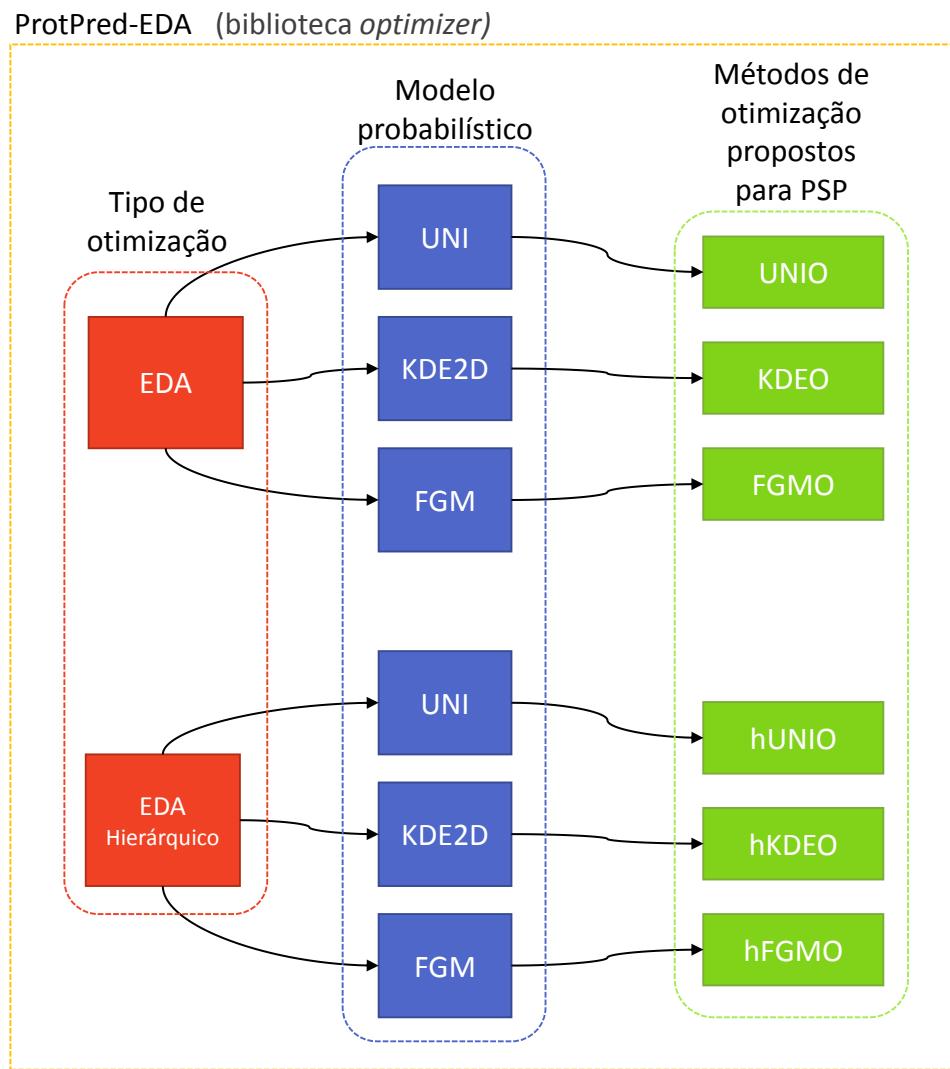


Figura 4.14: Métodos de otimização propostos para PSP. O EDA e o EDA hierárquico podem ser combinados com um dos três modelos probabilísticos propostos (UNI, KDE2D e FGM) e produzir a UNIO, KDEO e FGMO (quando utilizado o EDA) e hUNIO, hKDEO e hFGMO (quando utilizado o EDA hierárquico).

Resultados

Este capítulo apresenta experimentos e os resultados relativos aos EDAs (UNIO, KDEO e FGMO) baseados nos respectivos modelos probabilísticos propostos: UNI (Seção 4.2.1), KDE2D (Seção 4.2.2) e FGM (Seção 4.2.3); além dos resultados com suas extensões hierárquicas: hUNIO, hKDEO e hFGMO (Seção 3.4). Foram avaliados três aspectos diferentes para cada técnica proposta: (1) tempo computacional de execução; (2) a minimização da energia e (3) o RMSD, que estima a similaridade entre a proteína predita e a nativa. O Apêndice E mostra uma comparação estatística utilizando o teste de comparação não-paramétrico Wilcoxon (Gehan, 1965) para avaliar se houve diferença significativa dos resultados apresentados neste capítulo.

Como somente a energia de van der Waals foi utilizada como função de *fitness*, os resultados com relação a RMSD (Item (3)) podem não ser totalmente satisfatórios, pois outras energias potenciais que atuam na estabilização das proteínas não foram consideradas, com exceção dos resultados mostrados na Seção 5.1.2. No caso, utiliza-se também a energia de solvatação para ilustrar como o uso de outras energias pode melhorar os valores de RMSD. Porém, a consideração de outras energias para avaliação de conformações de proteínas por um EA é por si só uma linha de pesquisa. Conforme mostrado em Brasil et al. (2013) é importante utilizar algoritmos multi-objetivos (Srinivas & Deb, 1994) para se explorar não somente espaço de busca adequadamente, mas também o espaço dos objetivos (um objetivo para cada energia, por exemplo), gerando minimização de energia que corresponda a conformações mais plausíveis. Por outro lado, ao utilizar somente a energia de van der Waals (que tem uma contribuição relativamente alta para estabilização da proteína, ver Seção 2.5) foi possível investigar adequadamente a capacidade de exploração do espaço de busca (ao invés da exploração do espaço dos objetivos), permitindo compreender melhor o processo de minimização realizado pelos EDAs propostos em relação a outros métodos de otimização.

Dentre os outros métodos de otimização foram implementados e testados os algoritmos: Busca Aleatória (RW, Seção 4.1.1), Monte Carlo (MC, Seção 4.1.2), Algoritmo Genético (GA, Seção 4.1.3) e Evolução Diferencial (DE, Seção 4.1.4) como referência para comparação de desempenho com os EDAs propostos.

Antes de iniciar os experimentos, os parâmetros dos algoritmos de referência bem como das técnicas propostas foram calibrados utilizando a proteína 2LVG. A escolha dessa proteína para calibração deve-se ao fato de seu tamanho (em número de aminoácidos) corresponder a mediana do conjunto total de proteínas utilizadas nos experimentos.

Os experimentos foram executados em três *clusteres* de computadores diferentes: 1) *cluster* do LCR¹, do próprio laboratório de pesquisa, com 20 nós, cada um com um processador Intel Core i7 2,67 GHz com 4 núcleos físicos e 8 visíveis pelo sistema operacional, considerando a tecnologia *hyper-threading*. Cada nó tem 32 GB de RAM e sistema operacional Debian 4.6.3-14 64 bits. Além disso, cada nó tem dois adaptadores de rede, sendo um utilizado para operações do sistema de arquivos (NFS) e outro para comunicações de rede por MPI. O segundo *cluster* de computadores pertencem ao LNCC (Laboratório Nacional de Computação Científica - Rio de Janeiro, obtido acesso devido a necessidade de avaliar múltiplas configurações de parâmetros simultaneamente), chamado SunHPC (2), esse *cluster* possui 72 nós, modelo Sun Blade x6250 com 2 processadores Intel Xeon E5440 Quad Core, totalizando 8 núcleos físicos por nó e com 16 GB de RAM em cada nó. O sistema operacional utilizado é o Red Hat 4.1.2-28. O terceiro *cluster* de computadores (3) está localizado na Itália, chamado IBM PLX (obtido acesso para realização do trabalho Bonetti et al. (2014)), que faz parte de um consórcio entre várias universidades e centros de pesquisa chamado CINECA. Aparece na décima sétima posição da lista do Top500 (TOP500 Supercomputer Sites, 2014) lançada no primeiro semestre de 2014. O IBM PLX possui 2 processadores hexa-core Intel Westmere 2,4 GHz, 2 placas de vídeo NVidia Tesla M2070 e 48 GB de RAM por nó, e com um total de 274 nós, com sistema operacional Red Hat RHEL 5.6. Embora o IBM PLX seja mais rápido que os outros *clusteres* de computadores, não é totalmente adequado para o problema deste trabalho, pois é o único dos três *clusteres* que possui um parâmetro que limita o tempo de execução dos processos. Assim, o IBM PLX foi apenas utilizado para executar os experimentos com a energia de solvatação, pois é o único dos três *clusteres* que possui placas de vídeo de alto desempenho, tornando possível avaliar o desempenho do cálculo das energias em CPU e GPU. A eficiência desses cálculos é um aspecto não fundamental para a tese mas que também foi trabalhado de forma paralela pelo autor. O Apêndice C sumariza os resultados obtidos com relação a esse aspecto.

Considerando que o principal método experimental para determinação de estruturas de proteínas, a CRX, pode falhar na determinação de estruturas relativamente pequenas (até 100 aminoácidos, ver Capítulo 1) foi selecionado um conjunto de nove proteínas do PDB para representar proteínas dessa faixa de tamanho (menor que 100 aminoácidos). Além disso, optou-se por estrutu-

¹Laboratório de Computação Reconfigurável, Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo: <http://lcr.icmc.usp.br>.

ras de proteínas com uma ou mais α -hélices para que, de certa forma, a energia de van der Waals pudesse contribuir mais efetivamente (com exceção da proteína 2LLR, que possui uma folha- β). As proteínas utilizadas nos experimentos são apresentadas na Tabela 5.1, que mostra também a quantidade de resíduos de cada proteína, o número de variáveis que são utilizadas no processo de otimização e a quantidade de átomos de cada uma. Por exemplo, para a proteína 1R8T com 15 resíduos, o algoritmo de otimização terá 58 variáveis para otimizar, isto é, 15 ângulos ϕ mais 15 ângulos ψ mais 28 ângulos χ de cadeias laterais. A Figura 5.1 mostra as estruturas terciárias das proteínas nativas e a distribuição dos ângulos (ϕ, ψ).

Foram utilizados dois critérios de convergência para o processo evolutivo em todas as previsões. O algoritmo encerra a execução quando pelo menos um dos dois critérios de convergência é atingido. O primeiro critério de convergência é estabelecido pelo número máximo de avaliações, fixado em um milhão. O segundo é estabelecido quando o desvio padrão do *fitness* da população for menor do que o valor pré-determinado. Neste trabalho, o valor do desvio padrão do *fitness* utilizado para representar a convergência da população foi de 0,0001. A energia de van der Waals utilizada no cálculo do *fitness* é baseado no trabalho anterior (Bonetti et al., 2013), em que a complexidade do cálculo foi reduzida de $O(n^2)$ para $O(n)$, por meio de uma estrutura de vizinhança de células dispostas em uma grade tridimensional (Seção 2.5.1) que pode ser reconstruída de forma eficiente para cada nova conformação gerada. Isso é especialmente interessante para proteínas acima de 500 átomos, pois, assim como mostrado em (Bonetti, 2010; Bonetti et al., 2010a,b, 2013), o cálculo da energia de van der Waals utilizando lista de células tem ganho mais significativo para essa faixa de tamanho de proteínas.

Tabela 5.1: Proteínas utilizadas nos experimentos mostrando a quantidade de resíduos, quantidade de variáveis do problema e a quantidade de átomos de cada proteína.

Proteína	Resíduos	Variáveis	Átomos
1R8T	15	58	221
2LLR	22	86	358
1A11	25	95	390
2LX0	32	129	568
2LVG	40	169	627
2KK7	52	229	842
2X43	67	268	998
2A3D	73	300	1.141
2ZGG	92	357	1.364

A Seção 5.1 apresenta comparações de desempenho entre diferentes modelos propostos no ProtPred-EDA: UNIO, KDEO e FGMO. A Seção 5.2 mostra os resultados obtidos utilizando as extensões hierárquicas em conjunto com cada um dos modelos probabilísticos propostos. A FGMO e a hFGMO foram escolhidos para serem comparados com os algoritmos de referência (por apresen-

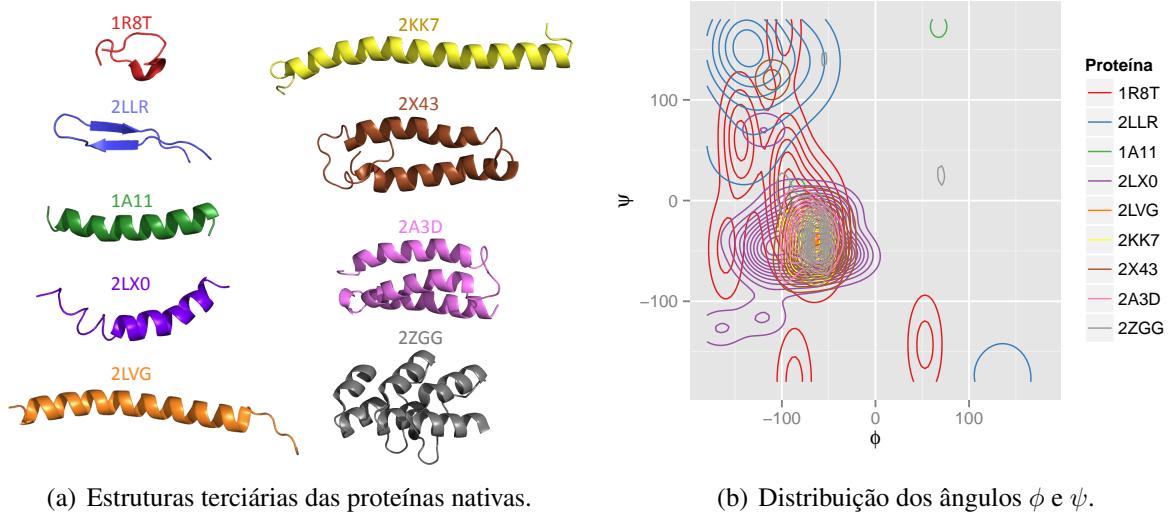


Figura 5.1: Proteínas nativas utilizadas nos experimentos.

tarem os melhores desempenhos entre os EDAs desenvolvidos de uma forma geral). A Seção 5.3 descreve todos os experimentos e resultados. A Seção 5.4 apresenta uma síntese desses resultados. Por fim, a Seção 5.5 apresenta algumas considerações finais deste capítulo.

5.1 Modelos probabilísticos propostos

Esta seção apresenta os experimentos e resultados obtidos para os EDAs propostos (Seção 4.2) utilizando um conjunto de nove proteínas. Antes da execução dos experimentos foi realizada uma calibração dos parâmetros: tamanho da população, pressão de seleção e a quantidade de componentes de misturas (parâmetro utilizado somente na FGMO). A calibração dos parâmetros é mostrada na Seção 5.1.1 e os experimentos com os parâmetros já calibrados são mostrados na Seção 5.1.2. Por último, a Seção 5.1.3 apresenta algumas análises complementares.

5.1.1 Calibração dos parâmetros dos EDAs

Utilizando a proteína de tamanho intermediário 2LVG foi realizada uma calibração dos parâmetros para a UNIO, KDEO e FGMO, variando os parâmetros: tamanho da população, pressão de seleção e quantidade de componentes de mistura para a FGMO. O tamanho da população foi variado em quatro níveis, enquanto que a pressão de seleção (conforme o parâmetro τ definido na Seção 3.1) e a quantidade de componentes de mistura foram variados em três níveis, conforme mostra a Tabela 5.2. Para equilibrar o número de execuções da FGMO com a UNIO e KDEO, as combinações de parâmetros da UNIO e KDEO foram variadas três vezes, isto é, utilizado três sementes diferentes para o gerador de números aleatórios para cada combinação. No total, as combinações de parâmetros produziram 36 configurações de parâmetros para cada método, isto é, quatro tamanhos de população vezes três diferentes tamanhos de pressão de seleção multiplicado

por três componentes de mistura para a FGMO e três sementes diferentes para o caso da UNIO e KDEO.

Tabela 5.2: Parâmetros utilizados para calibração.

Tamanho da População	Pressão de Seleção τ (torneio)	Misturas (FGMO)
200	0,5	5
500	1,0	10
1.000	1,5	15
2.000		

A Figura 5.2 mostra histogramas obtidos a partir das oito melhores execuções segunda cada modelo probabilístico (baseado somente na energia de van der Waals). A Tabela 5.3 lista todos os parâmetros, que são ordenados para cada modelo probabilístico de acordo com a pela energia de van der Waals para as oito melhores execuções. Com base nesta figura e tabela, foi determinado um conjunto de parâmetros que foi considerado mais adequado para cada modelo. O tamanho da população foi escolhido com base nesses histogramas, assim, o tamanho de população escolhido para a UNIO, KDEO e FGMO foi 1.000, 500 e 200 respectivamente. A pressão de seleção da UNIO e FGMO foi definida em 1,0, pois é o valor que aparece em primeiro lugar para ambos os modelos. Para a KDEO, a pressão de seleção foi definida em 0,5, devido a sua frequência ter sido superior às outras. A quantidade de componentes de mistura da FGMO foi definida em 10 (Figura 5.3), visto que os melhores valores foram 5 e 10, e o valor 10 possibilita modelos mais refinados, o que pode beneficiar modelos para outras proteínas (Tabela 5.3).

5.1.2 Experimentos e análises iniciais

Com os parâmetros calibrados, novos experimentos foram realizados com todas as proteínas mostradas na Tabela 5.1. Cada método (UNIO, KDEO e FGMO) foi executado 30 vezes para cada proteína. A cada execução foi adicionado um parâmetro relativo ao uso ou não na população inicial do banco de dados de ângulos diedrais (ADB, Seção 2.4). Isso pode ser importante para se verificar a relevância do ADB para os EDAs propostos. Assim, no total, foram necessárias $9 \times 3 \times 30 \times 2 = 1.620$ execuções, isto é, o número de proteínas vezes o número de modelos probabilísticos multiplicado pelo número de execuções e por dois (pelo fato de ter ou não ADB para gerar a população inicial). As 1.620 execuções foram divididas em 128 processos utilizando MPI (Gramma et al., 2003) e foram executadas no *cluster* SunHPC.

As Figuras 5.4-5.12 mostram os resultados obtidos pelos modelos probabilísticos propostos para as nove proteínas utilizando ADB. Em (a) é comparado os valores da energia de van der Waals obtidos do melhor indivíduo do processo evolutivo, em (b) o RMSD da conformação correspondente à proteína com a menor energia de van der Waals, em (c) é comparado o tempo de execução, em (d) é realizado um relacionamento entre a energia de van der Waals com o RMSD

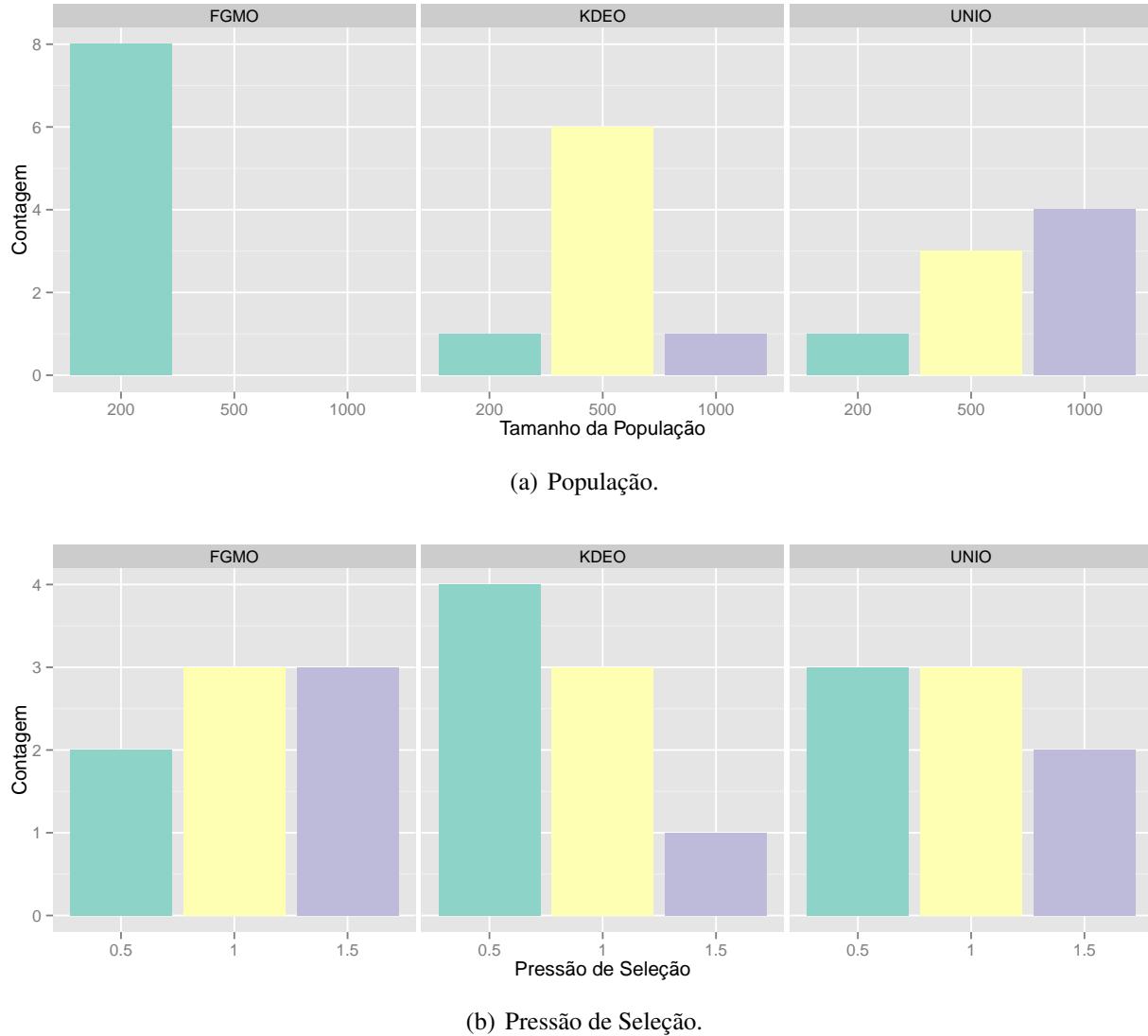


Figura 5.2: Calibração dos parâmetros mostrando a contagem das 8 melhores execuções para cada método.

bem como a Fronteira de Pareto (Tabela 5.4) e, por fim, (e) mostra o valor da energia de van der Waals do melhor indivíduo de acordo com o processo evolutivo. A Fronteira de Pareto é um conceito geralmente utilizado em algoritmos multi-objetivo (Zhou et al., 2011a) definido por um conjunto de soluções que dominam as demais. Nesse caso, a Fronteira de Pareto com base na energia de van der Waals e no RMSD contém as soluções que não são dominadas por outras, isto é, que não há soluções melhores que elas em pelo menos um dos critérios considerados (energia de van der Waals e RMSD).

Para a menor proteína 1R8T os resultados são mostrados na Figura 5.4. A KDEO e a FGMO tiveram resultados semelhantes considerando o fator energia de van der Waals. Embora a FGMO teve a menor mediana, a KDEO conseguiu o menor valor. Para o RMSD, a mediana da FGMO também foi menor que a da KDEO. Observando a Figura 5.4(c) é possível notar que, embora os pontos resultados entre FGMO e KDEO foram relativamente semelhantes em relação a energia de

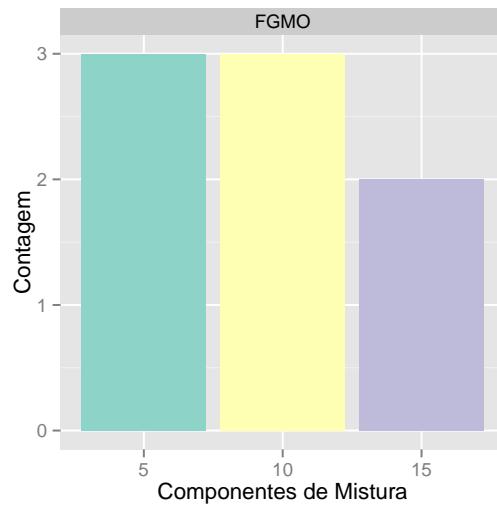


Figura 5.3: Calibração do número de misturas para a FGMO.

Tabela 5.3: Tabela dos parâmetros para as oito melhores execuções para os três modelos probabilísticos com a proteína 2LVG.

Método	População	Pressão seleção	Componentes de mistura	Energia de van der Waals	RMSD	Tempo (minutos)
UNIO	1000	0,5		-232,0	10,752	61,22
UNIO	200	1,5		-231,5	14,292	72,49
UNIO	1000	1,5		-228,5	16,205	62,04
UNIO	500	1		-227,7	8,680	57,82
UNIO	1000	1		-225,7	14,842	62,68
UNIO	1000	0,5		-224,3	13,616	64,07
UNIO	500	1		-223,2	16,617	69,59
UNIO	500	0,5		-222,9	11,338	66,74
KDEO	200	1		-251,5	11,821	128,66
KDEO	500	0,5		-248,3	14,373	111,86
KDEO	500	0,5		-247,8	10,267	97,76
KDEO	500	1		-247,2	9,493	121,73
KDEO	500	0,5		-247,0	7,505	94,01
KDEO	500	1		-245,2	7,158	116,72
KDEO	1000	0,5		-244,2	8,369	95,77
KDEO	500	1,5		-244,0	11,381	139,41
FGMO	200	1	5	-272,5	12,904	62,69
FGMO	200	1,5	15	-259,8	14,726	70,40
FGMO	200	1,5	5	-255,3	14,870	62,92
FGMO	200	0,5	5	-253,3	11,060	59,96
FGMO	200	1	10	-246,3	9,735	60,91
FGMO	200	1	15	-246,1	12,648	68,48
FGMO	200	1,5	10	-132,0	8,832	62,57
FGMO	200	0,5	10	-109,4	12,766	62,37

van der Waals e RMSD, o tempo de execução da KDEO foi praticamente o dobro dos tempos de execução da UNIO e FGMO.

O gráfico de dispersão entre energia de van der Waals e RMSD (Figura 5.4(d)) mostra a distribuição de cada execução com cada um dos modelos probabilísticos. Nesta figura, a Fronteira de Pareto mostra os três pontos que dominam os restantes, sendo dois pontos da KDEO e um da FGMO (Tabela 5.4). É interessante observar neste caso que as três execuções da Fronteira de Pareto possuem RMSD semelhante, porém energia de van der Waals diferentes. As estruturas referentes a menor energia e RMSD em comparação com a estrutura nativa é mostrada na Figura 5.14(a). Por fim, na Figura 5.4(e) é possível verificar que, no começo do processo evolutivo, todos os três modelos probabilísticos possuem energia de van de Waals relativamente semelhante. Então, a FGMO converge rapidamente com aproximadamente 100.000 avaliação, enquanto que a UNIO e KDEO convergem mais lentamente, especialmente a partir de 500.000 avaliações. Considerando os aspectos energia, RMSD e tempo de execução, a FGMO poderia obter valores de energia relativamente melhores (em comparação com a KDEO) utilizando apenas 10% da quantidade total de avaliações. Isso poderia reduzir o tempo de predição desta proteína em cerca de 18 minutos (Figura 5.4(c)-FGMO) para 1,8 minuto, mantendo a predição com qualidade relativamente alta.

Para a segunda proteína avaliada 2LLR (mostrada na Figura 5.5) os valores da energia de van der Waals entre KDEO e FGMO também foram semelhantes. No entanto, para esta proteína, a mediana da KDEO foi melhor do que a mediana da FGMO. Além disso, o valor do RMSD da KDEO foi relativamente melhor do que a FGMO e UNIO. Mesmo assim, o melhor RMSD da KDEO para esta proteína pode ainda ser considerada alto. Isso ocorreu porque a estrutura nativa da proteína 2LLR é uma folha- β . Assim, deixando de utilizar outros potenciais de energia fundamentais para a estabilização da folha- β como, por exemplo, a energia de ligações de hidrogênio, é pouco provável que a estrutura correta possa ser determinada somente utilizando a energia de van der Waals em PSP puramente *ab initio*. O tempo de execução para a proteína 2LLR seguiu o mesmo comportamento da proteína 1R8T, em que a KDEO precisou de quase o dobro do tempo dos outros modelos probabilísticos. Na Fronteira de Pareto (Figura 5.5(d)) apareceram três pontos pertencentes à KDEO, contendo as previsões com menor energia de van der Waals e RMSD, além de dois pontos referentes à FGMO. O processo evolutivo (Figura 5.5(e)) também mostrou que a FGMO começou a convergir por volta de 200,000 avaliações, enquanto que a KDEO e a UNIO começaram a convergir após 500,000 avaliações. A Figura 5.14(a) mostra que as duas estruturas preditas, com menor energia de van der Waals e menor RMSD são, de fato, diferentes da proteína nativa.

A Figura 5.6, referente a execução da proteína 1A11, mostra resultados diferentes dos apresentados para as proteínas 1R8T e 2LLR. Considerando apenas o aspecto energia de van der Waals, a KDEO foi relativamente superior aos outros dois modelos probabilísticos, pois a maioria dos valores estão entre -145 kcal/mol e -150 kcal/mol. No aspecto RMSD, a FGMO obteve uma maior concentração dos dados, inferior a 4,0 Å. Além disso, o tempo computacional da FGMO para a proteína 1A11 foi, em média, inferior ao da UNIO. Isso é um dos fatores que tornam a FGMO especialmente interessante, pois, observando o gráfico de convergência do processo evolutivo na Figura 5.6(e) é possível verificar que a FGMO foi capaz de reduzir significativamente a

energia de van der Waals com um número pequeno de avaliações. Assim, embora a FGMO seja mais complexa que a UNIO, pois o processo de estimação dos parâmetros das variáveis é iterativo (algoritmo EM, Seção 4.2.3), a FGMO foi mais rápida que a UNIO. Um dos motivos prováveis para isso é que o desvio padrão do par de variáveis (ϕ, ψ) a ser estimado seja pequeno a partir de certo número de avaliações, evitando o algoritmo EM e amostrando novos indivíduos utilizando uma FDP (Seção 3.3.3) bivariada normal. A Figura 5.6(d) mostra que mais pontos da KDEO apareceram na Fronteira de Pareto em relação à FGMO. No entanto, é possível observar que a maioria dos pontos referentes as execuções da UNIO estão afastados da Fronteira de Pareto, indicando que a UNIO foi menos adequada para a proteína 1A11.

Os resultados da proteína 2LX0, apresentados na Figura 5.7, mostram que os valores médios da energia de van der Waals entre KDEO e FGMO são parecidos. No entanto, a maior concentração dos valores da energia de van der Waals obtidos pela FGMO estão abaixo da mediana da KDEO. Com relação ao RMSD, a UNIO obteve apenas um ponto abaixo de 6,0 Å. Embora os melhores pontos referentes ao RMSD pertençam à KDEO, o melhor RMSD encontrado é de uma execução da FGMO, presente na Fronteira de Pareto (sendo cinco e quatro pontos referentes à FGMO e KDEO respectivamente). O destaque da Fronteira de Pareto (Figura 5.7(d)) é que a FGMO obteve pontos com menor energia de van der Waals e a KDEO obteve os pontos com menor RMSD, com exceção de um ponto de melhor RMSD da FGMO. A UNIO, semelhante a proteína 1A11 não aparece na Fronteira de Pareto, ficando distante dos outros dois modelos probabilísticos. A Figura 5.7(c) mostra que o tempo computacional para a proteína 2LX0 seguiu o mesmo comportamento das proteínas 1R8T e 2LLR, em que a KDEO foi o mais lento dos três EDAs propostos. O processo evolutivo, apresentado na Figura 5.7(e), mostra que a FGMO também começa a convergir mais rapidamente, isto é, com cerca de 200.000 avaliações; enquanto que a UNIO e a KDEO começam a convergir com mais de 500.000 avaliações.

O mesmo comportamento da proteína 2LX0 em relação a energia, RMSD e tempo de execução ocorreu com a proteína 2LVG (Figura 5.8). Nesse caso, a maioria dos valores de energia de van der Waals obtidos pela FGMO ficou abaixo de -250 kcal/mol, isto é, abaixo da mediana da KDEO. Embora a mediana do RMSD para a KDEO tenha ficado mais baixa que a mediana da FGMO, a maior parte dos valores do RMSD da KDEO ficaram abaixo de 10,0 Å. A Figura 5.8(d) mostra que todos os pontos da Fronteira de Pareto pertencem à FGMO, isto é, a FGMO dominou a KDEO e a UNIO tanto no aspecto de melhor energia de van der Waals quanto no aspecto do melhor RMSD. É possível verificar por meio da Figura 5.7(e) que a convergência da FGMO começa a ocorrer com cerca de 200,000 avaliações, com valores mais significativos do que a UNIO e KDEO. Comparando o processo evolutivo da UNIO e KDEO é possível notar que, por volta de 125.000 avaliações ambos os modelos probabilísticos possuem valores de energia de van der Waals semelhantes. Neste caso, é provável que pelo fato da KDEO ser bivariado seja capaz de estimar e amostrar valores mais adequados que a UNIO, evitando armadilhas (ótimos locais).

A Figura 5.9 mostra os resultados para a proteína 2KK7. Embora os valores da energia de van der Waals entre KDEO e FGMO sejam semelhantes, a FGMO possui três *outliers*, considerados

importantes, que ficaram abaixo de $-340,0$ kcal/mol. Em média, o RMSD da KDEO foi melhor que o da FGMO, ficando abaixo de $10,0$ Å. No entanto, o menor RMSD encontrado pertence à FGMO, presente na Fronteira de Pareto da Figura 5.9(d). A UNIO não obteve resultados relevantes tanto para energia de van der Waals quanto para o RMSD. O tempo computacional, mostrado pela Figura 5.9(c), foi semelhante ao obtido para a proteína 1A11, em que a mediana da FGMO é menor que a da UNIO, pois ao observar também a Figura 5.9(e) é possível perceber que a FGMO começa a convergir mais rapidamente que a UNIO e KDEO. O valor médio da energia de van der Waals da UNIO e KDEO permaneceram semelhantes até cerca de 125.000 avaliações. Após esse valor, a KDEO mostrou um ganho, apresentando valores próximos dos obtidos pela FGMO.

Os valores de energia de van der Waals obtidos para a proteína 2X43, mostrados na Figura 5.10, ficaram mais afastados entre os modelos probabilísticos, isto é, cada método ocupou uma faixa de valores de energia. A FGMO o melhor dos três modelos e o único que obteve valores abaixo de $-400,0$ kcal/mol. Embora o RMSD da UNIO tenha obtido a melhor mediana, a FGMO encontrou a estrutura com melhor RMSD. A Figura 5.10(d) mostra que a Fronteira de Pareto é formado por apenas dois pontos que pertencem à FGMO e, assim, a FGMO conseguiu encontrar o melhor RMSD e também a melhor energia. Com relação ao tempo de execução a UNIO foi mais rápida na média. A segunda mais rápida foi a FGMO. Por outro lado, a KDEO obteve uma mediana por volta de 160 minutos, isto é, aproximadamente 50% mais lenta que a UNIO. O processo evolutivo mostrado na Figura 5.10(e) está de acordo com o que foi obtido para a proteína 2LVG, em que a FGMO começa a convergir mais rapidamente para regiões promissoras, enquanto que a KDEO e UNIO são parecidas até certo número de avaliações, porém com valores da energia de van der Waals mais altos que a FGMO.

Para a proteína 2A3D (Figura 5.11), a FGMO mostrou ser melhor, em relação à KDEO e à UNIO, pois a FGMO foi a única a obter valores abaixo $-450,0$ kcal/mol. A mediana do RMSD entre os três EDAs avaliados (Figura 5.11(b)) são parecidos. No entanto, o melhor valor de RMSD pertence à FGMO. Na verdade, todos os pontos da Fronteira de Pareto (Figura 5.11(d)) pertencem à FGMO, mostrando que foi superior tanto no aspecto de energia de van der Waals quanto no aspecto qualidade da estrutura predita. Com relação ao tempo de execução (Figura 5.11(c)), a UNIO manteve-se a mais rápida e a KDEO, a mais lenta. É interessante observar que a FGMO foi relativamente mais lenta para a proteína 2A3D do que para as proteínas menores descritas anteriormente. Isso pode ter sido causado pela quantidade de vezes em que o algoritmo EM é chamado, pois para proteínas maiores que 73 resíduos, pode haver um impacto maior no tempo de execução. Mesmo assim, se ao invés de executar a FGMO com 1 milhão de avaliações, fosse executada com 500.000 avaliações, o tempo computacional da FGMO cairia pela metade e ainda seria capaz de encontrar valores de energia de van der Waals mais baixos que a UNIO com um milhão de avaliações ou mesmo a KDEO (Figura 5.11(e)).

Por fim, a Figura 5.12 mostra os resultados para a maior proteína utilizada neste experimento, a proteína 2ZGG. De forma semelhante a proteína 2A3D, a FGMO obteve os melhores resultados referentes a energia de van der Waals e RMSD, sendo a única a obter valores de energia de van der

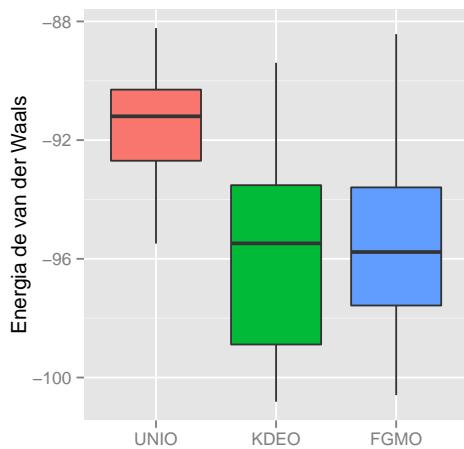
Waals abaixo de $-525,0$ kcal/mol. O RMSD da KDEO pode ser considerado o menos significativo entre os três EDAs propostos. No entanto, para esse tipo de proteína, que possui uma estrutura mais globular, formada por seis α -hélices, a medida de similaridade de estruturas por RMSD pode não ser totalmente adequada, pois conformações ruins e boas podem se misturar. Dessa forma, todos os pontos da Fronteira de Pareto mostrados na Figura 5.12(d) pertencem à FGMO. É possível verificar pela Figura 5.12(c) que o tempo de execução da FGMO aproxima-se do tempo da KDEO, pois, com o aumento da dimensionalidade do problema, mais chamadas ao algoritmo EM são necessárias para estimar todos os pares (ϕ, ψ) . No entanto, com a convergência rápida da FGMO para pontos ótimos melhores que a UNIO e a KDEO, o tempo de execução da FGMO pode ser diminuído utilizando um número de avaliações menor (Figura 5.12(e)).

A Figura 5.13 mostra uma síntese dos resultados obtidos pelos três EDAs propostos, considerando somente os 10% melhores resultados em relação aos aspectos energia de van der Waals, qualidade da estrutura e tempo de execução para todas as proteínas. Para proteínas menores como a 1R8T, 2LLR e 1A11 a KDEO encontrou o menor valor de energia de van der Waals. Para proteínas maiores que a 1A11, a FGMO foi melhor. Além disso, conforme aumenta-se o tamanho das proteínas, a FGMO mostra-se ser cada vez melhor. A UNIO não superou nem a KDEO nem a FGMO, mesmo considerando todas as proteínas.

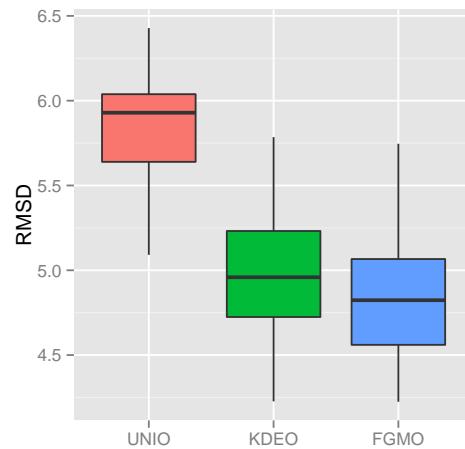
Os modelos probabilísticos bivariados (que tratam as relações (ϕ, ψ)) foram capazes de minimizar a energia de van der Waals melhor que o modelo probabilístico univariado. Esse é o primeiro indicador a revelar que modelos probabilísticos bivariados, que tratam as relações (ϕ, ψ) , contribuem, de fato, para obter soluções com menor energia de van der Waals (Figura 5.13(a)). Além disso, a Figura 5.13(b) mostra que os EDAs com modelos probabilísticos bivariados foram superiores à UNIO. Observe que na Figura 5.13(b) existe um pico no segundo ponto da escala das proteínas, em que o RMSD está entre 8 Å e 9 Å para a proteína com 22 resíduos (2LLR). Isso ocorreu devido a estrutura da proteína 2LLR ser uma folha- β e, por ter sido utilizado somente a energia de van der Waals nesses experimentos, não foi possível obter uma estrutura com qualidade. Os tempos de execução (Figura 5.13(c)) da UNIO e FGMO permanecem semelhantes para as seis proteínas menores. Para as proteínas 2X43, 2A3D e 2ZGG os tempos de execução entre a UNIO e a FGMO começam a divergir. No entanto, como foi visto nos gráficos do processo evolutivo da energia de van der Waals, sabe-se que a FGMO pode encontrar soluções superiores à UNIO utilizando um número menor de avaliações. O tempo de execução da KDEO foi mais alto do que a da UNIO e FGMO para todos os tamanhos de proteínas. Isso provavelmente ocorre devido à criação dos mapas bi-dimensionais utilizando o método do kernel.

As configurações de proteínas com melhor RMSD (em verde) e melhor energia de van der Waals (em vermelho) obtidos nos experimentos desta seção foram alinhadas com a estrutura da proteína nativa (em azul) e são apresentadas na Figura 5.14. A Tabela 5.4 mostra qual método obteve a melhor energia de van der Waals e RMSD para as nove proteínas avaliadas. Na Figura 5.14 é possível notar que, de fato, as melhores configurações obtidas foram estruturas que contém apenas uma α -hélice como, por exemplo, as proteínas 1A11, 2LX0, 2LVG e 2KK7. Isso pode ser

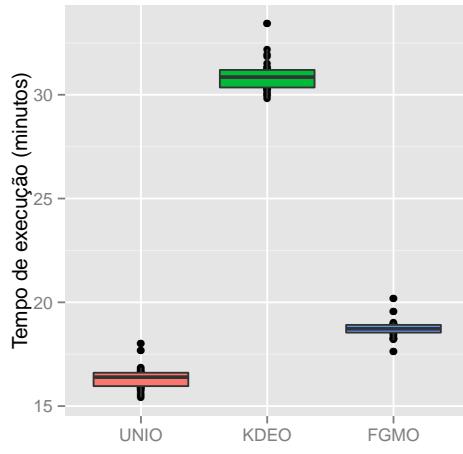
justificado pelo fato das predições terem sido realizadas utilizando somente a energia de van der Waals, que possui uma contribuição relativamente grande para estabilizar hélices. Para proteínas muito pequenas, como o caso da proteína 1R8T, o RMSD não foi tão baixo, pois houve a formação de uma α -hélice, mesmo a estrutura nativa não possuindo α -hélice. Houve formação de α -hélices também, para a proteína 2LLR, mesmo a estrutura nativa sendo uma folha- β . Para a proteína predita com menor RMSD para a estrutura 2X43, houve formação de três α -hélices, assim como na proteína nativa. Para a proteína 2A3D, com três α -hélices definidas, a melhor estrutura predita foi capaz de compactar a estrutura de forma que as voltas pudessem ocorrer em locais adequados. No entanto, a interação de uma α -hélice com outra provavelmente impediu a formação adequada das α -hélices. A maior proteína utilizada nesses experimentos, a 2ZGG, não obteve uma estrutura razoável, pois as α -hélices não puderam ser estabilizadas adequadamente. Com exceção das proteínas 1R8T e 2LLR, as proteínas preditas com menor energia de van der Waals (em vermelho) foram relativamente diferentes das proteínas com menor RMSD.



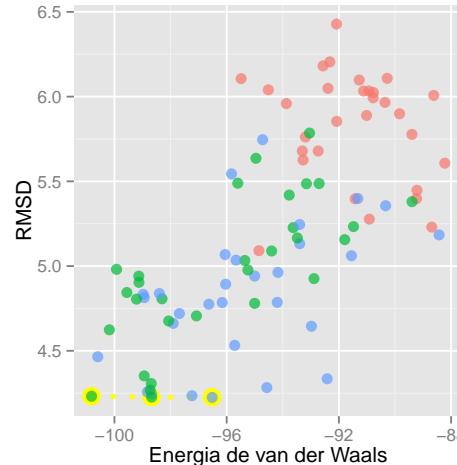
(a) Energia de van der Waals.



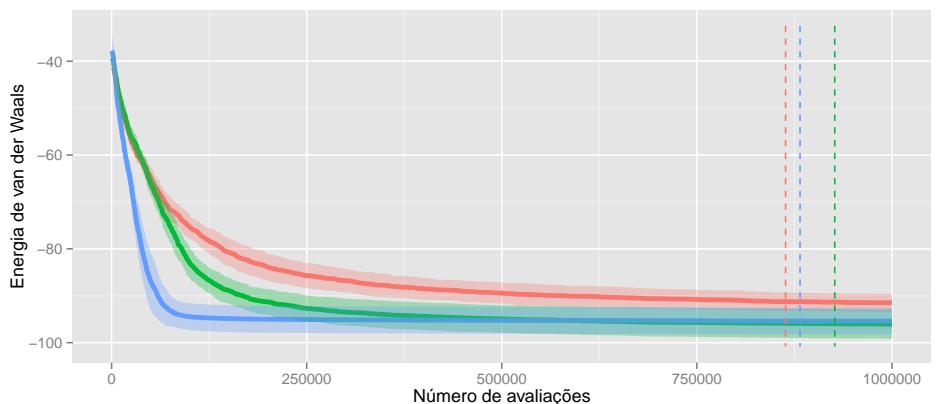
(b) RMSD.



(c) Tempo de execução.

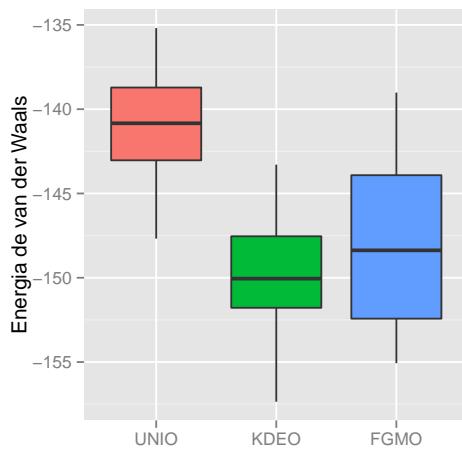


(d) Dispersão entre energia de van der Waals e RMSD, destacando a fronteira de Pareto em amarelo.

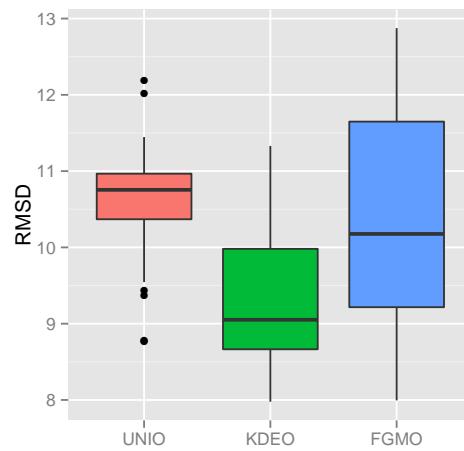


(e) Evolução da energia de van der Waals. A média das 30 execuções é representada pelas linhas e o desvio padrão pelas áreas suavizadas em torno das linhas de mesma cor. A linha vertical tracejada indica em que avaliação, em média, o EDA convergiu.

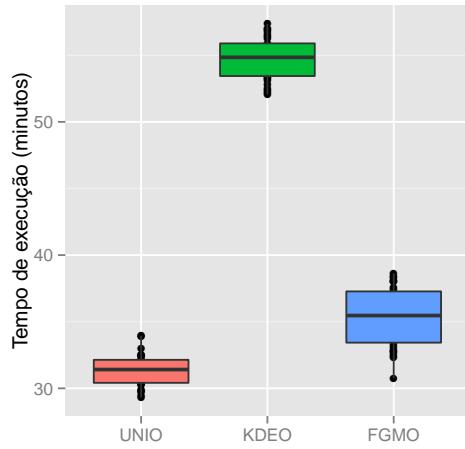
Figura 5.4: EDAs propostos para a proteína 1R8T para a melhor solução da última geração, segundo a energia de van der Waals.



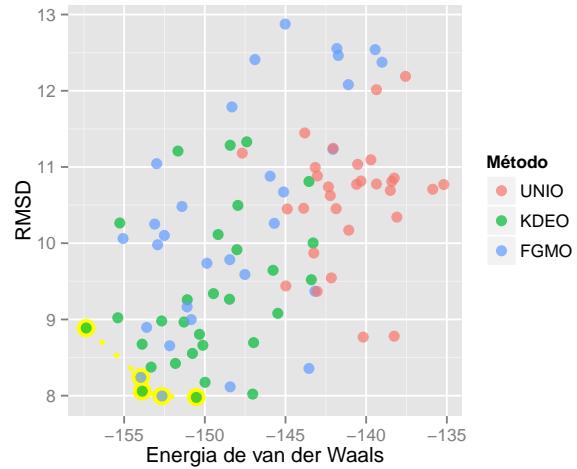
(a) Energia de van der Waals.



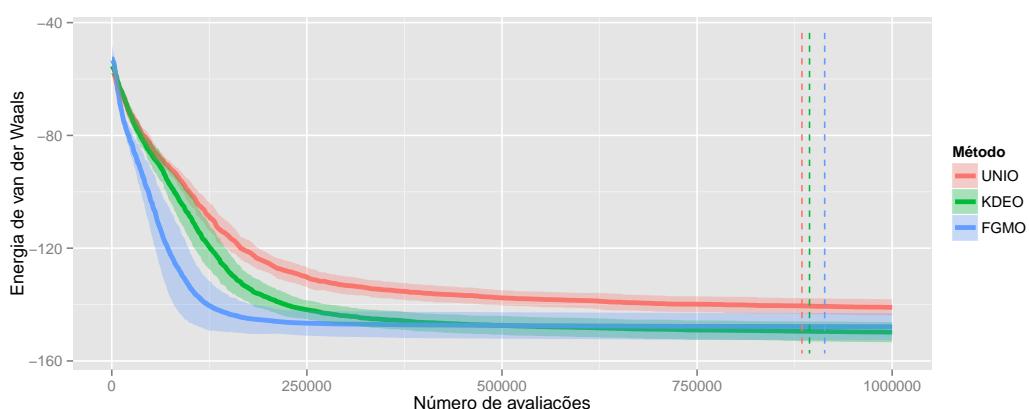
(b) RMSD.



(c) Tempo de execução.



(d) Dispersão entre energia de van der Waals e RMSD, destacando a fronteira de Pareto em amarelo.



(e) Evolução da energia de van der Waals. A média das 30 execuções é representada pelas linhas e o desvio padrão pelas áreas suavizadas em torno das linhas de mesma cor. A linha vertical tracejada indica em que avaliação, em média, o EDA convergiu.

Figura 5.5: EDAs propostos para a proteína 2LLR para a melhor solução da última geração, segundo a energia de van der Waals.

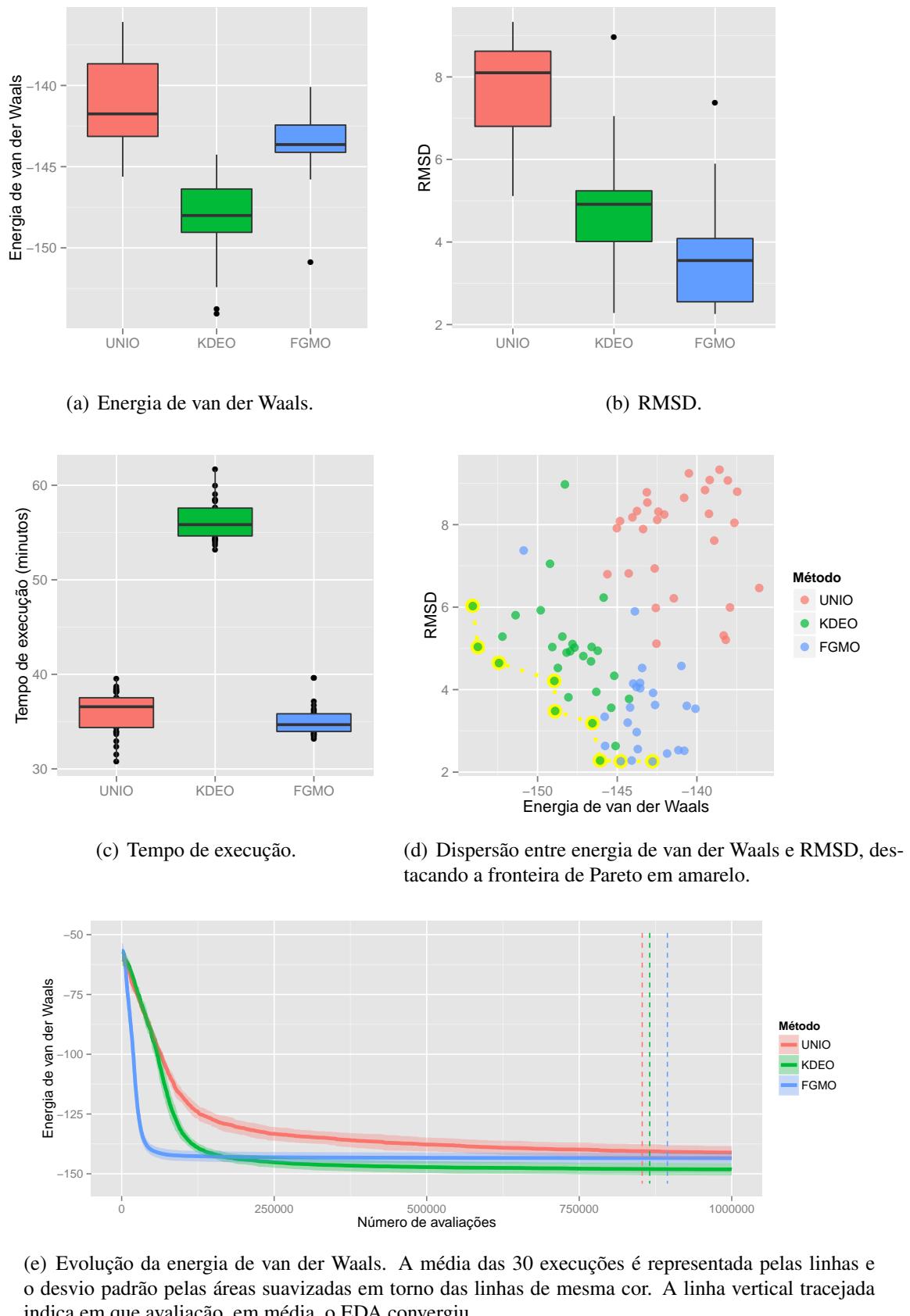
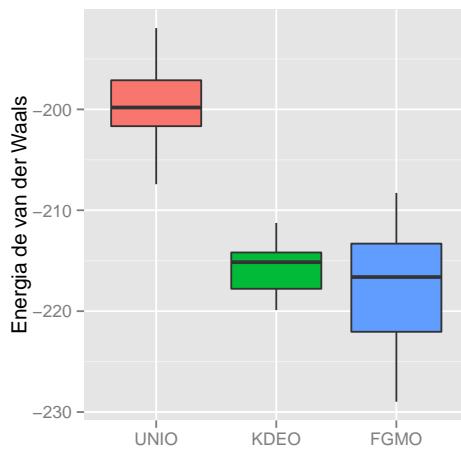
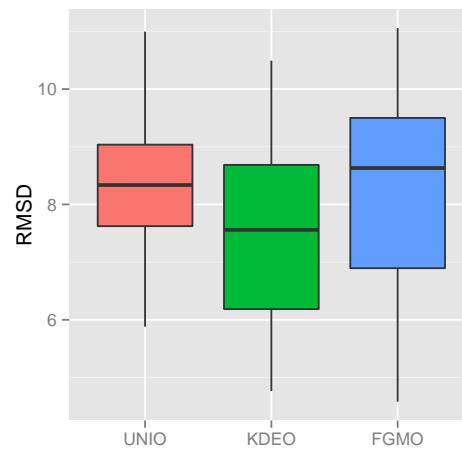


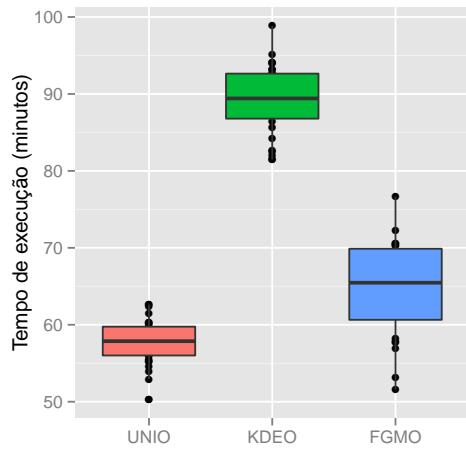
Figura 5.6: EDAs propostos para a proteína 1A11 para a melhor solução da última geração, segundo a energia de van der Waals.



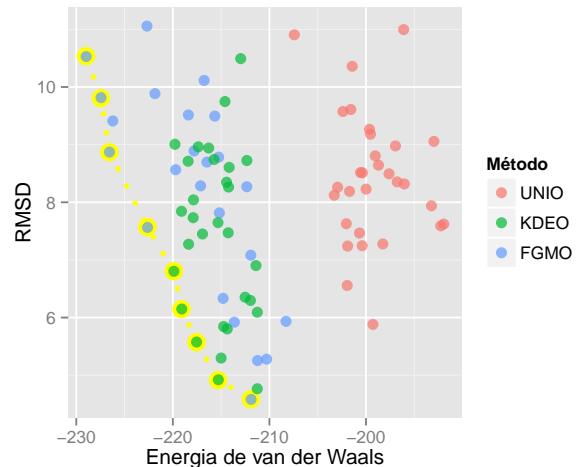
(a) Energia de van der Waals.



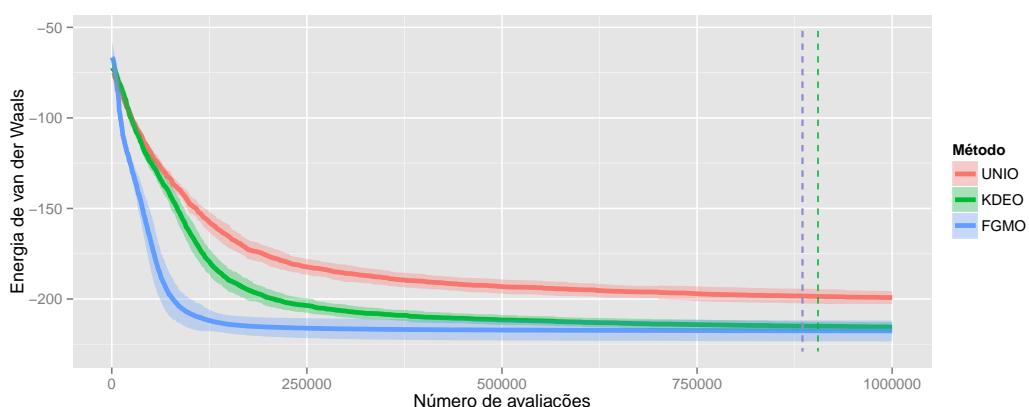
(b) RMSD.



(c) Tempo de execução.

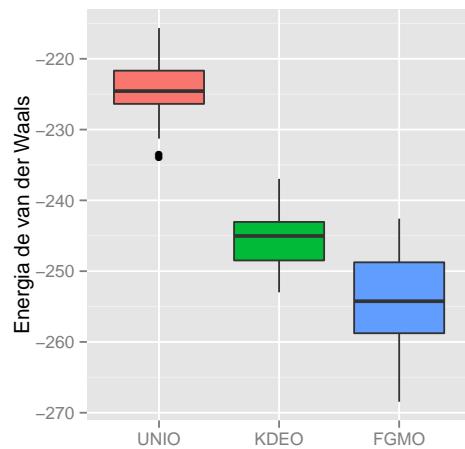


(d) Dispersão entre energia de van der Waals e RMSD, des- tacando a fronteira de Pareto em amarelo.

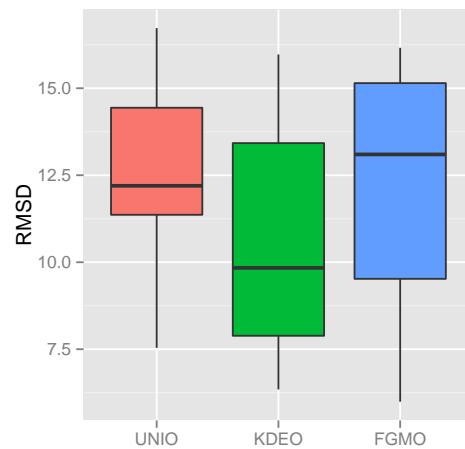


(e) Evolução da energia de van der Waals. A média das 30 execuções é representada pelas linhas e o desvio padrão pelas áreas suavizadas em torno das linhas de mesma cor. A linha vertical tracejada indica em que avaliação, em média, o EDA convergiu.

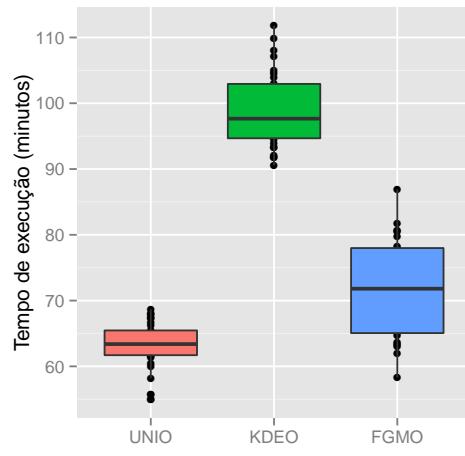
Figura 5.7: EDAs propostos para a proteína 2LX0 para a melhor solução da última geração, segundo a energia de van der Waals.



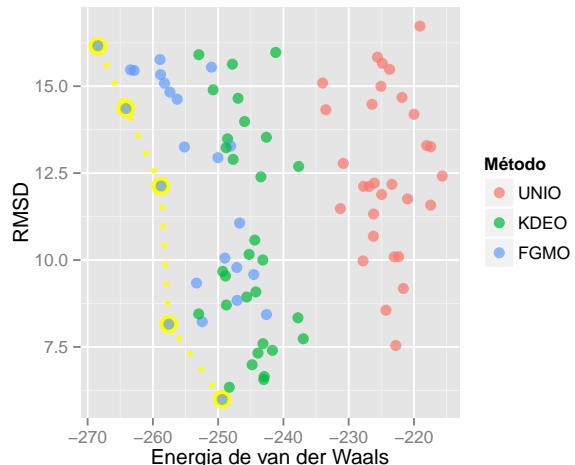
(a) Energia de van der Waals.



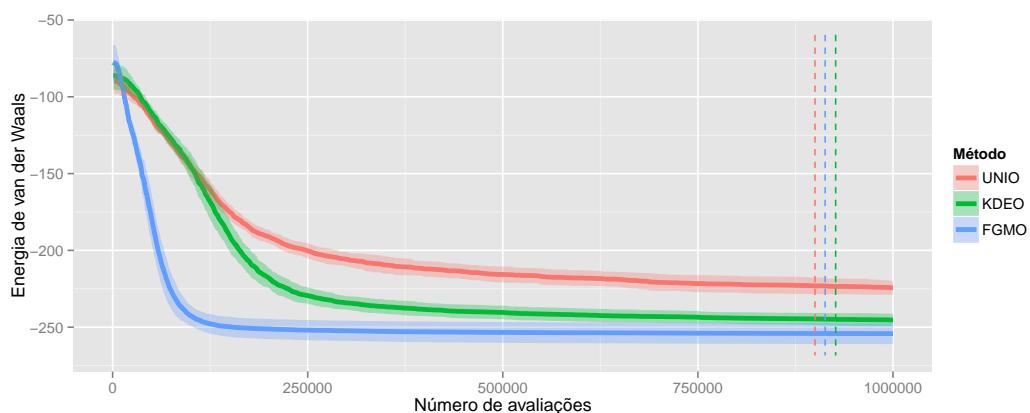
(b) RMSD.



(c) Tempo de execução.

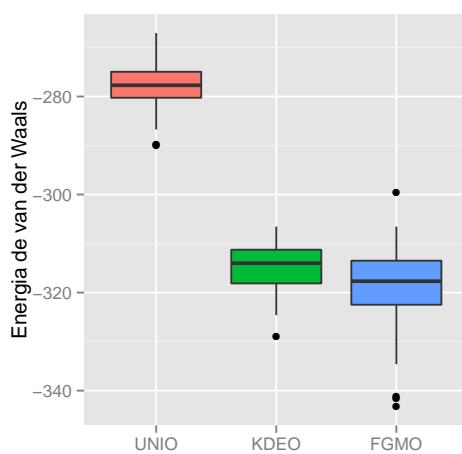


(d) Dispersão entre energia de van der Waals e RMSD, destacando a fronteira de Pareto em amarelo.

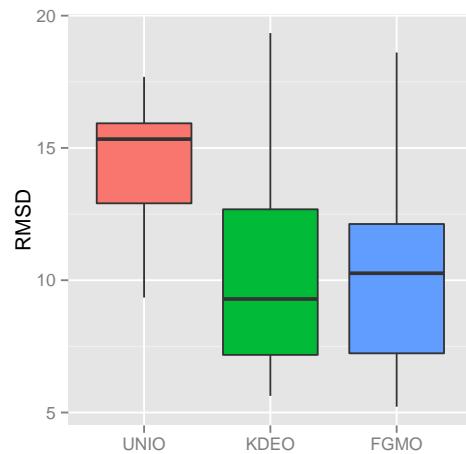


(e) Evolução da energia de van der Waals. A média das 30 execuções é representada pelas linhas e o desvio padrão pelas áreas suavizadas em torno das linhas de mesma cor. A linha vertical tracejada indica em que avaliação, em média, o EDA convergiu.

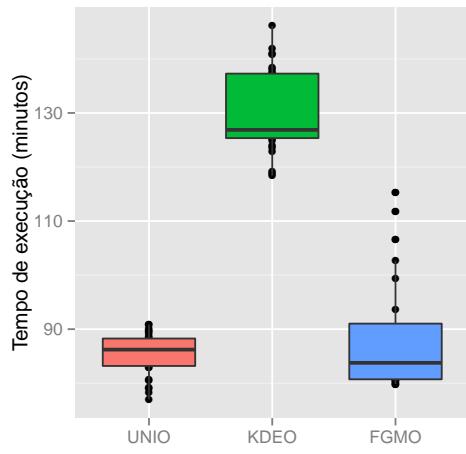
Figura 5.8: EDAs propostos para a proteína 2LVG para a melhor solução da última geração, segundo a energia de van der Waals.



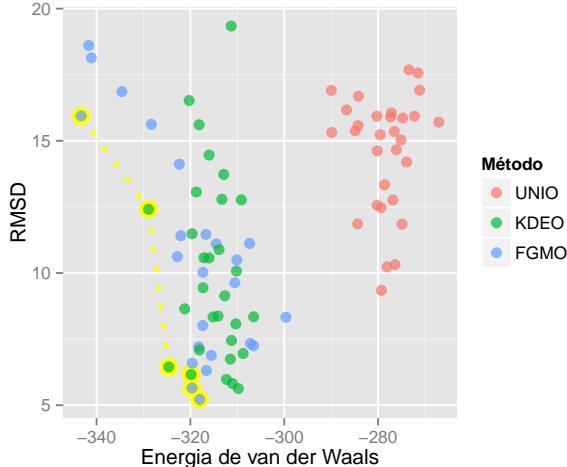
(a) Energia de van der Waals.



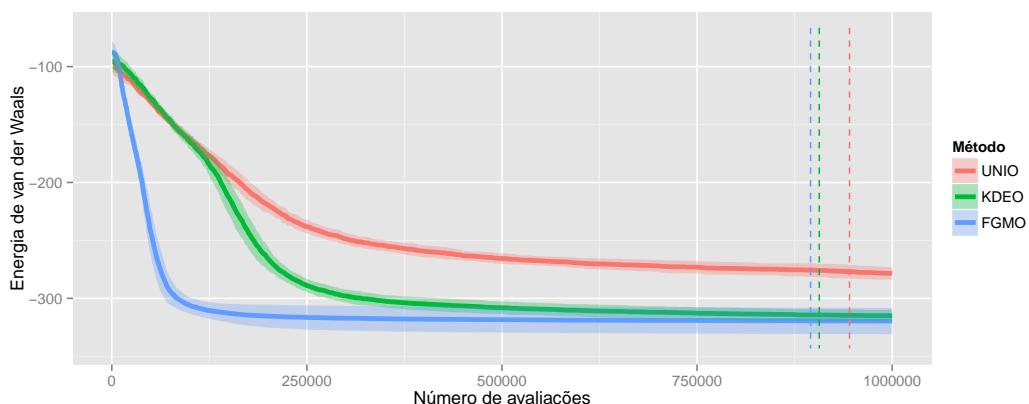
(b) RMSD.



(c) Tempo de execução.

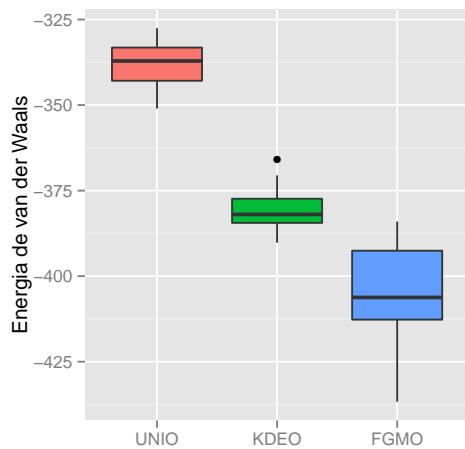


(d) Dispersão entre energia de van der Waals e RMSD, destacando a fronteira de Pareto em amarelo.

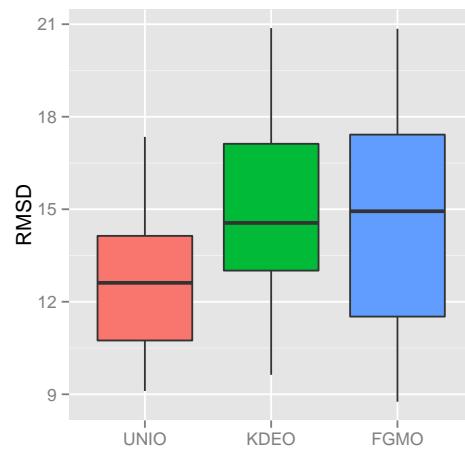


(e) Evolução da energia de van der Waals. A média das 30 execuções é representada pelas linhas e o desvio padrão pelas áreas suavizadas em torno das linhas de mesma cor. A linha vertical tracejada indica em que avaliação, em média, o EDA convergiu.

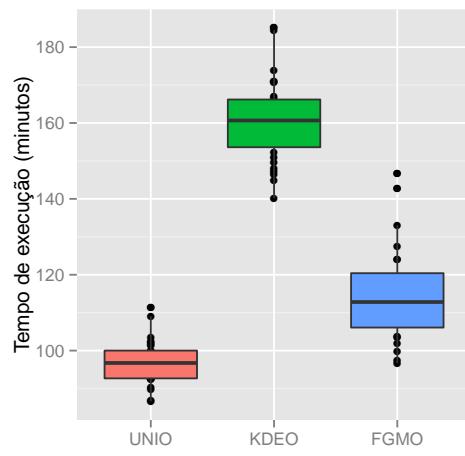
Figura 5.9: EDAs propostos para a proteína 2KK7 para a melhor solução da última geração, segundo a energia de van der Waals.



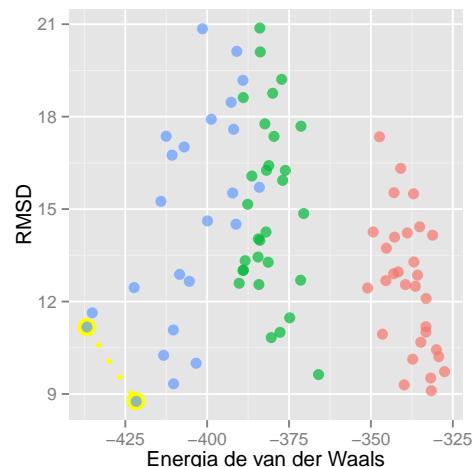
(a) Energia de van der Waals.



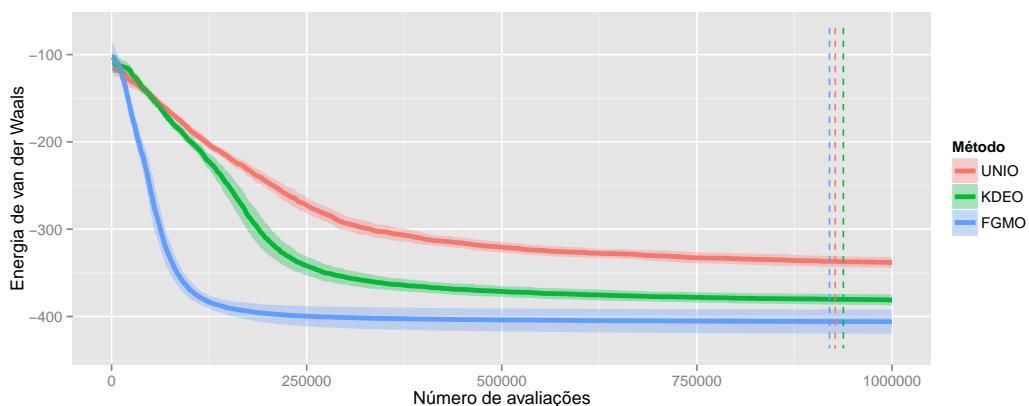
(b) RMSD.



(c) Tempo de execução.



(d) Dispersão entre energia de van der Waals e RMSD, destacando a fronteira de Pareto em amarelo.



(e) Evolução da energia de van der Waals. A média das 30 execuções é representada pelas linhas e o desvio padrão pelas áreas suavizadas em torno das linhas de mesma cor. A linha vertical tracejada indica em que avaliação, em média, o EDA convergiu.

Figura 5.10: EDAs propostos para a proteína 2X43 para a melhor solução da última geração, segundo a energia de van der Waals.

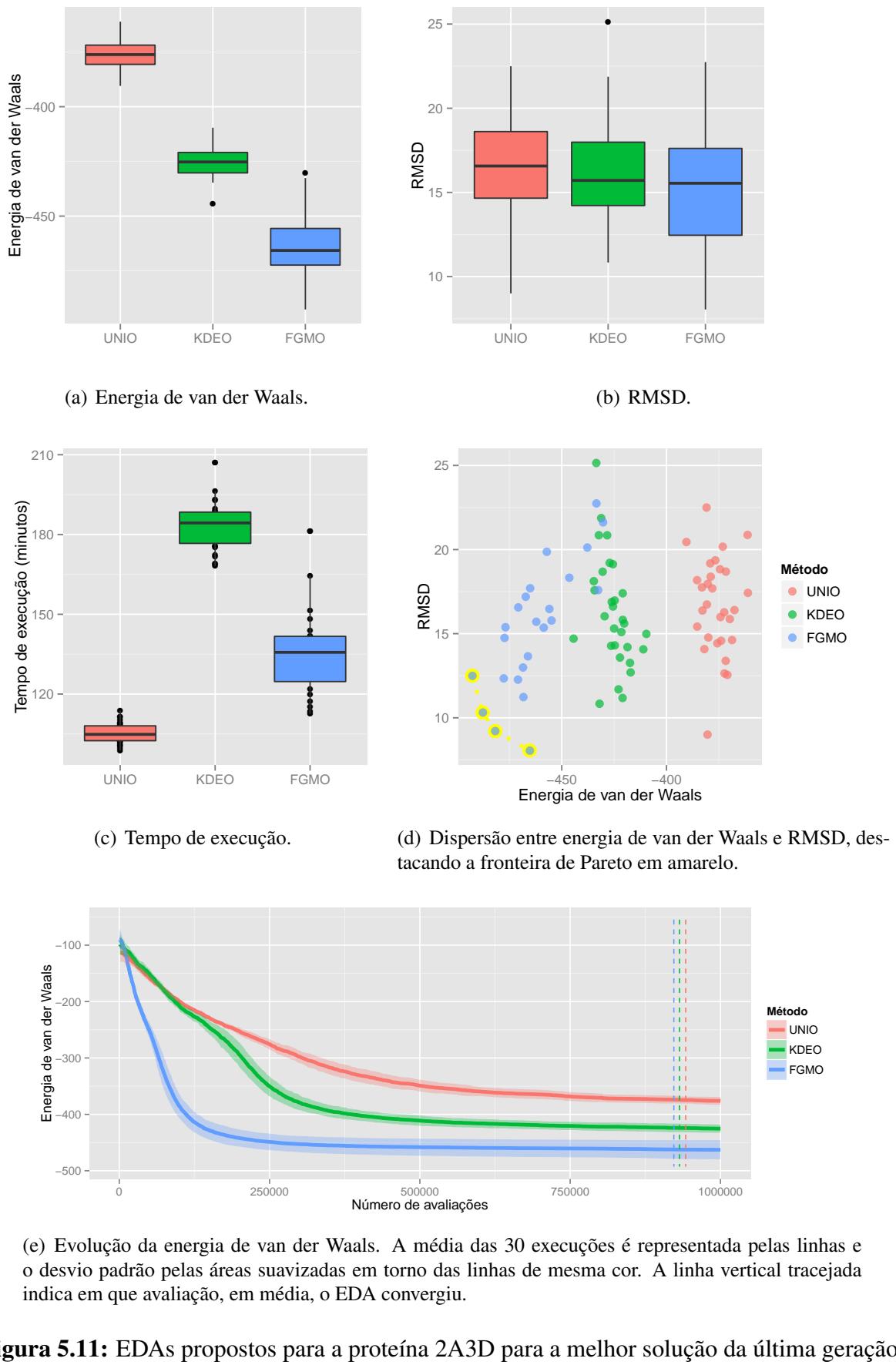
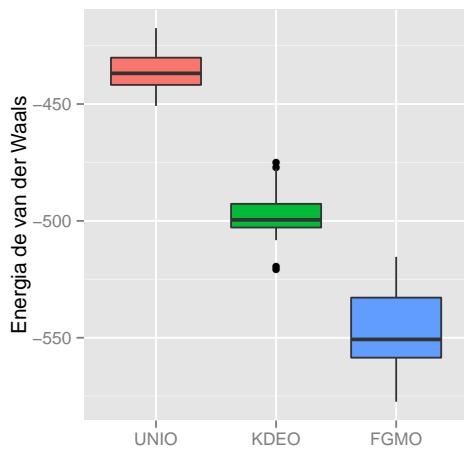
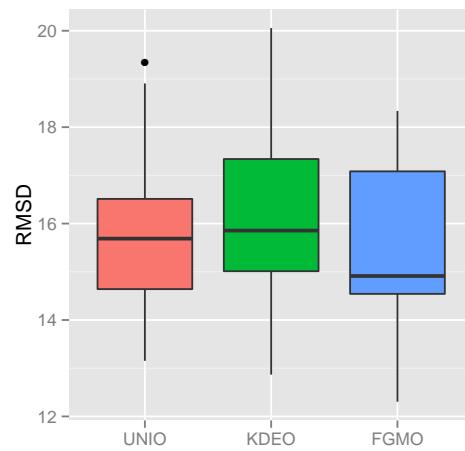


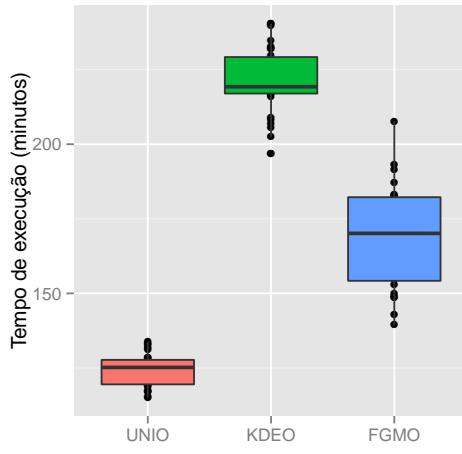
Figura 5.11: EDAs propostos para a proteína 2A3D para a melhor solução da última geração, segundo a energia de van der Waals.



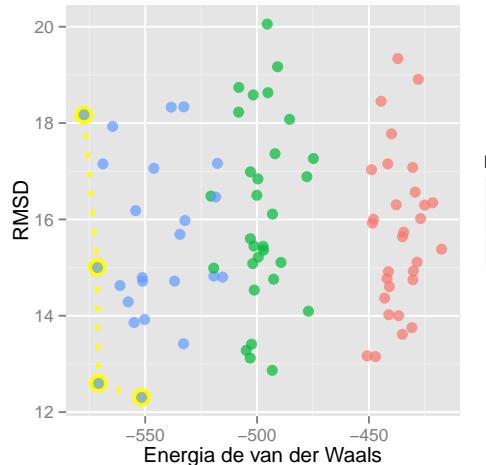
(a) Energia de van der Waals.



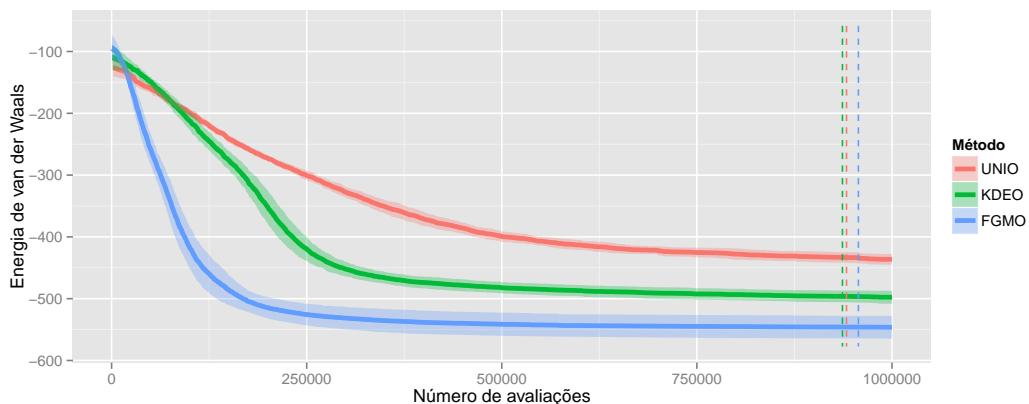
(b) RMSD.



(c) Tempo de execução.



(d) Dispersão entre energia de van der Waals e RMSD, destacando a fronteira de Pareto em amarelo.



(e) Evolução da energia de van der Waals. A média das 30 execuções é representada pelas linhas e o desvio padrão pelas áreas suavizadas em torno das linhas de mesma cor. A linha vertical tracejada indica em que avaliação, em média, o EDA convergiu.

Figura 5.12: EDAs propostos para a proteína 2ZGG para a melhor solução da última geração, segundo a energia de van der Waals.

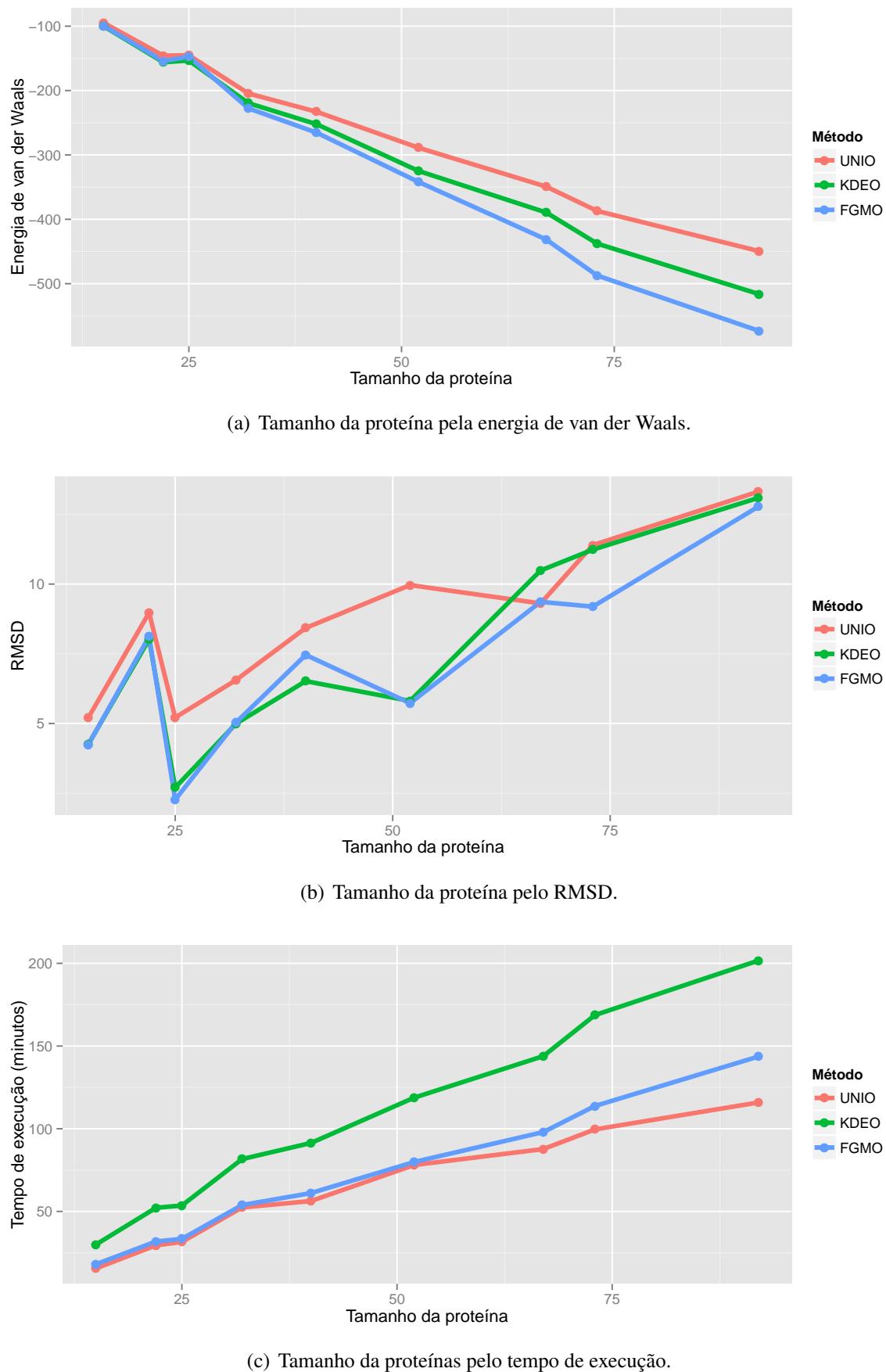


Figura 5.13: Síntese dos resultados com base nas 10% das melhores soluções, minimizando a energia de van der Waals.

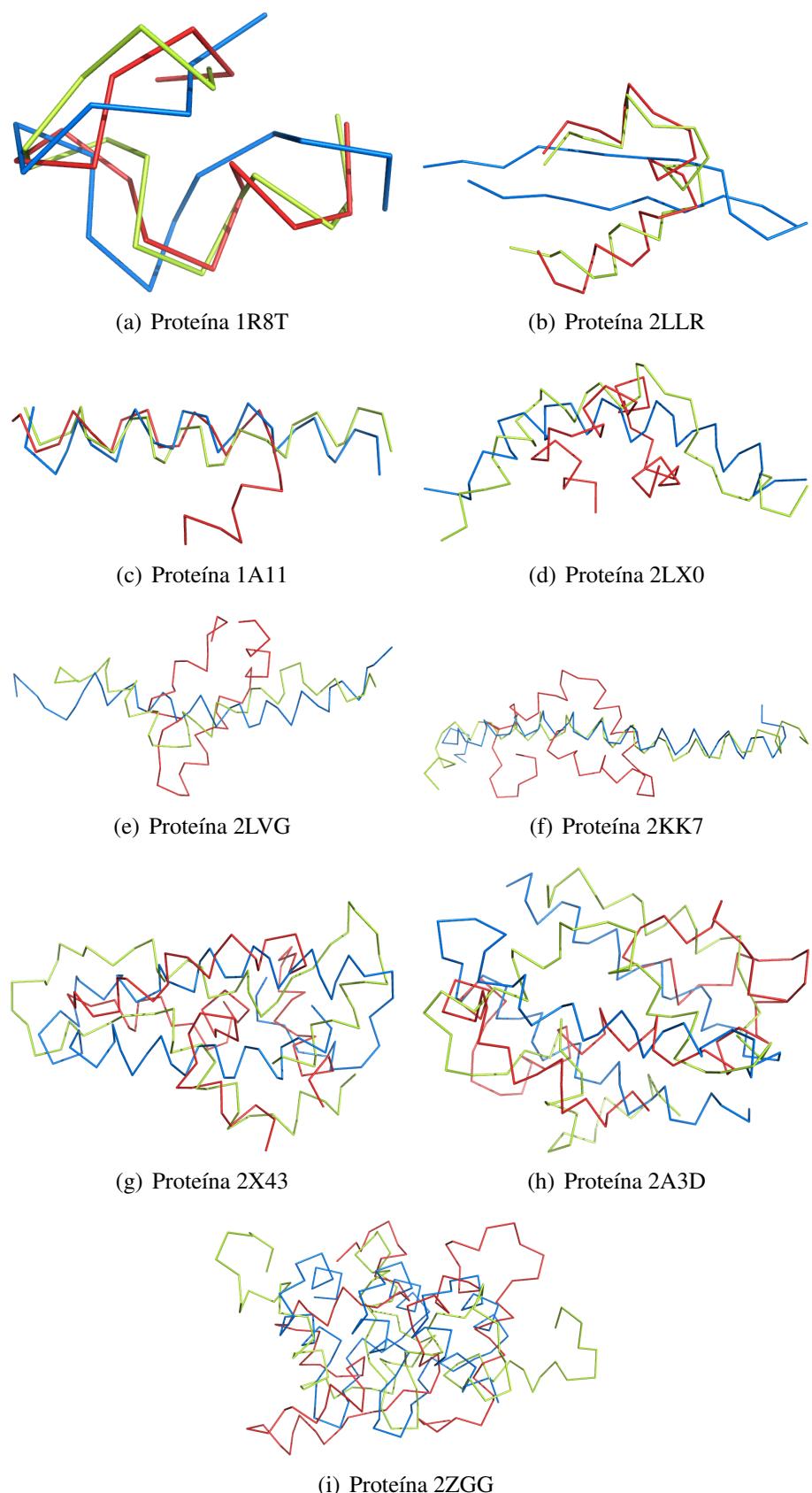


Figura 5.14: Estruturas de proteína preditas utilizando o ProtPred-EDA com energia de van der Waals. As proteínas nativas são representadas pela cor azul e as proteínas preditas com menor energia (em vermelho) e menor RMSD (em verde). As duas proteínas preditas estão alinhadas à nativa.

Tabela 5.4: Soluções da Fronteira de Pareto obtidas pelos critérios energia de van der Waals e RMSD.

Proteína	Resíduos	Método	Energia de van der Waals	RMSD
1R8T	15	KDEO	-100,82	4,23
	15	KDEO	-98,68	4,23
	15	FGMO	-96,52	4,22
2LLR	22	KDEO	-157,35	8,89
	22	FGMO	-153,97	8,24
	22	KDEO	-153,88	8,06
	22	FGMO	-152,65	8,00
	22	KDEO	-150,53	7,98
1A11	25	KDEO	-154,07	6,03
	25	KDEO	-153,75	5,04
	25	KDEO	-152,42	4,64
	25	KDEO	-148,95	4,21
	25	KDEO	-148,90	3,48
	25	KDEO	-146,56	3,18
	25	KDEO	-146,08	2,28
	25	FGMO	-144,78	2,26
	25	FGMO	-142,78	2,26
2LX0	32	FGMO	-228,98	10,53
	32	FGMO	-227,44	9,81
	32	FGMO	-226,56	8,87
	32	FGMO	-222,62	7,56
	32	KDEO	-219,89	6,80
	32	KDEO	-219,08	6,15
	32	KDEO	-217,55	5,58
	32	KDEO	-215,29	4,92
	32	FGMO	-211,90	4,58
2LVG	40	FGMO	-268,44	16,16
	40	FGMO	-264,13	14,36
	40	FGMO	-258,75	12,13
	40	FGMO	-257,56	8,15
	40	FGMO	-249,37	5,99
2KK7	52	FGMO	-343,28	15,95
	52	KDEO	-328,92	12,41
	52	KDEO	-324,60	6,45
	52	KDEO	-319,83	6,16

Proteína	Resíduos	Método	Energia de van der Waals	RMSD
2KK7	52	FGMO	-319,64	5,65
2KK7	52	FGMO	-318,01	5,22
2X43	67	FGMO	-436,70	11,17
2X43	67	FGMO	-421,67	8,76
2A3D	73	FGMO	-492,65	12,49
2A3D	73	FGMO	-487,65	10,31
2A3D	73	FGMO	-481,74	9,21
2A3D	73	FGMO	-465,22	8,05
2ZGG	92	FGMO	-577,37	18,17
2ZGG	92	FGMO	-571,37	15,00
2ZGG	92	FGMO	-570,84	12,60
2ZGG	92	FGMO	-551,59	12,30

5.1.3 Análises complementares

A Figura 5.15 mostra um resumo da quantidade de avaliações necessárias para atingir o critério de convergência para as 10% melhores soluções de cada método, segundo a energia de van der Waals. É possível verificar que a convergência da UNIO para as proteínas 1R8T e 2LX0 ocorre antes que a da KDEO e FGMO para as mesmas proteínas. Isso significa que, para a proteína 1R8T, a UNIO convergiu com menos de 700.000 avaliações, gastando as avaliações restantes até um milhão sem que mudanças significativas fossem produzidas no valor da energia de van der Waals. O mesmo ocorreu para a proteína 2LX0 que, utilizando a UNIO, convergiu com cerca de 750.000 avaliações. Embora a FGMO tenha收敛ido com cerca de 825.000 avaliações para a proteína 2LX0, a diferença dos valores de energia de van der Waals entre a FGMO e a UNIO foram significativas se for considerado todo processo evolutivo (Figura 5.7(e)). É possível perceber que para as proteínas maiores que 40 resíduos, houve uma tendência de crescimento para o número de avaliações da FGMO. Para as proteínas com mais de 40 resíduos, a FGMO precisou de mais de 900.000 avaliações para convergir, embora a KDEO tenha obtido pouca variabilidade a partir de cerca de 500.000 avaliações.

Sabe-se que as pontes de hidrogênio são favoráveis para a estabilização das moléculas. No entanto, utilizando somente a energia de van der Waals, o processo de minimização pode minimizar a energia de van der Waals além do esperado, evitando que conformações com menor energia de van der Waals tenham mais pontes de hidrogênio. A Figura 5.16 mostra a relação entre a quantidade de ligações de hidrogênio (calculadas utilizando o Stride (Heinig & Frishman, 2004)) com o RMSD, para as conformações de proteínas obtidas com base na melhor energia de van der Waals obtida para a UNIO, KDEO e FGMO. Observando as nove proteínas é possível verificar que a KDEO foi capaz de obter um melhor balanço entre a quantidade de ligações de hidrogênio com o RMSD,

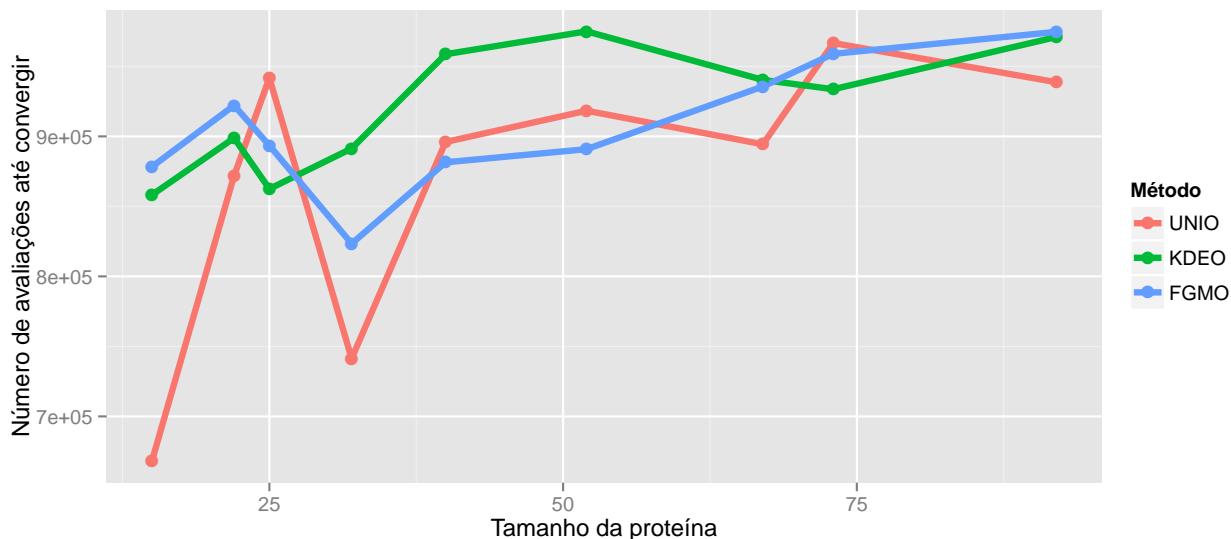


Figura 5.15: Dez porcento melhores soluções obtidas para a energia de van der Waals por cada método

com exceção das proteínas 2X43, 2A3D e 2ZGG. A UNIO obteve melhores valores para a quantidade de ligações de hidrogênio, especialmente para as proteínas 2KK7 e 2X43. Para a proteína 1A11, a quantidade de pontes de hidrogênio obtida pela UNIO foi significante, embora o RMSD seja considerado alto. Por fim, embora a FGMO tenha conseguido os melhores valores de energia de van der Waals e, na maioria dos casos, de RMSD, foi o que menos encontrou conformações com ligações de hidrogênio, entre os três modelos probabilísticos.

Foi avaliado também a relação entre a porcentagem de estruturas secundárias presente nas conformações preditas com o RMSD, uma vez que tal porcentagem é também um indicativo da qualidade das previsões. A Figura 5.17 mostra a porcentagem de estruturas secundárias calculada para cada conformação de proteína, utilizando o Stride. Devido a várias conformações de proteínas que forma preditas não possuir estrutura secundária, o valor da porcentagem de estruturas secundárias foi zero, especialmente para as proteínas 1R8T, 2LLR e 2LX0. Com exceção da proteína 2LX0, a UNIO foi capaz de encontrar a maior porcentagem de estruturas secundárias, principalmente para as maiores proteínas (2KK7, 2X43, 2A3D, 2ZGG). Apesar da porcentagem de estruturas secundárias da KDEO ter sido mais baixa que da UNIO, a KDEO foi capaz de manter um balanço entre esses dois aspectos. Por outro lado, a FGMO, responsável por encontrar os melhores valores de energia de van der Waals, encontrou poucos trechos de estruturas secundárias, mesmo tendo um RMSD relativamente baixo. Isso pode ser explicado pela mesma característica apresentada entre a relação entre RMSD e quantidade de ligações de hidrogênio, isto é, o processo de minimização da FGMO pode ir além do que se era esperado, utilizando somente a energia de van der Waals. Para obter uma energia global de certa conformação de proteína, acredita-se que a FGMO (ou outro método capaz de minimizar a energia de van der Waals além do esperado) pode comprometer a formação de pedaços de estruturas secundárias em favor de uma conformação com menor porcentagem de estruturas secundárias, porém com menor energia de van der Waals.

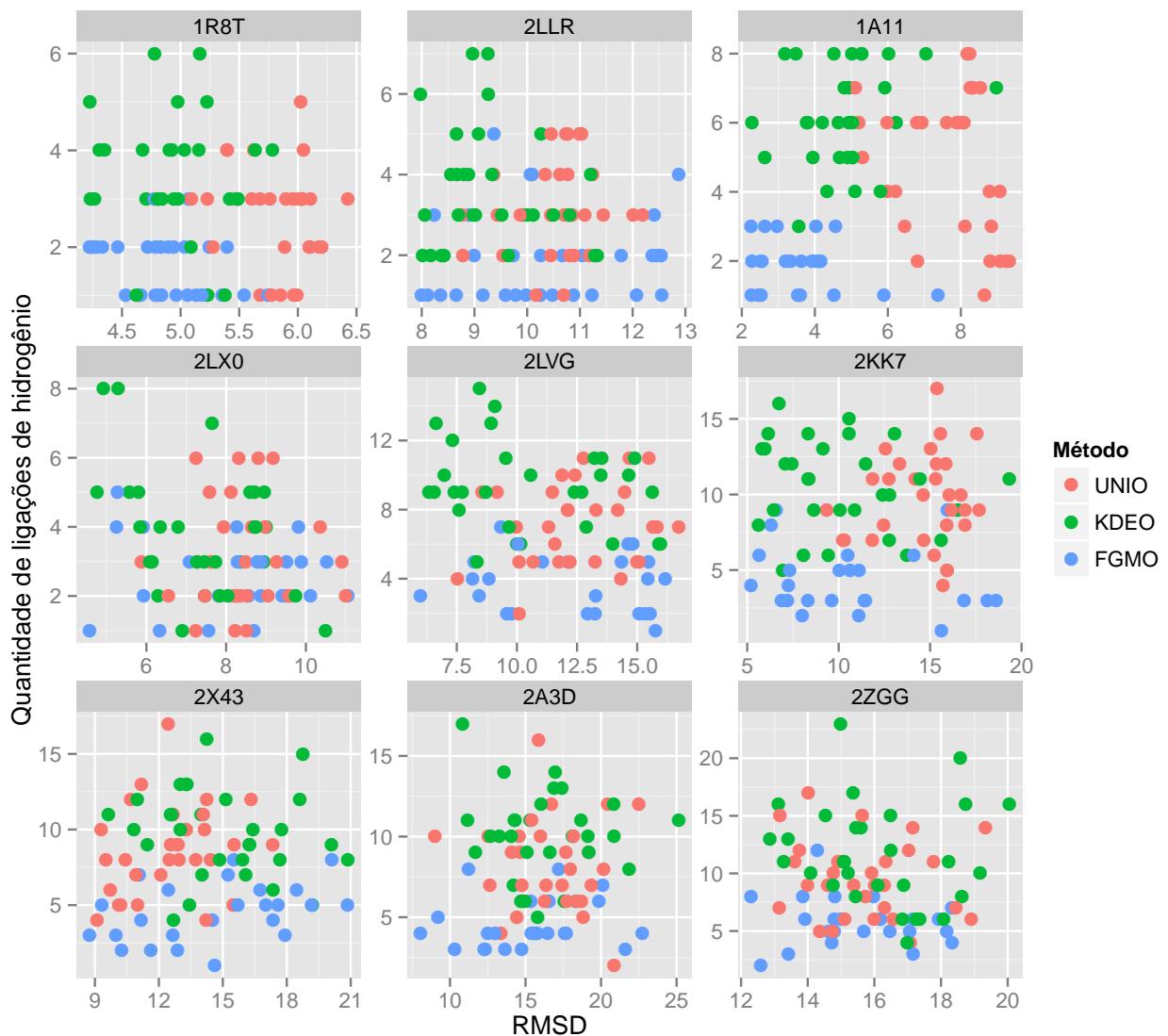


Figura 5.16: Relação entre a quantidade de ligações de hidrogênio e o RMSD para a UNIO, KDEO e FGMO.

É possível ver a distribuição da energia de van der Waals pelo RMSD para cada proteína no gráfico tridimensional mostrado pela Figura 5.18(a). Nesse gráfico é possível perceber uma tendência que existe entre a energia de van der Waals, RMSD e tamanho da proteína, pois conforme aumenta-se o tamanho da proteína o RMSD aumenta na mesma escala para os três modelos probabilísticos. Entretanto, com o aumento do tamanho da proteína é possível verificar que, além de haver uma mudança da escala no valor da energia de van der Waals (de -100 para -600), os EDAs tendem a distanciar-se, de forma que a FGMO mostra-se melhor que a KDEO e a KDEO, melhor que a UNIO. A Figura 5.18(b) mostra um gráfico tridimensional da porcentagem de estrutura secundária, energia de van der Waals e o tamanho da proteína. Nesse gráfico, é possível notar que a porcentagem de estruturas secundárias diminuiu com o aumento do tamanho da proteína, para todos os modelos probabilísticos. Além disso, embora a UNIO não tenha conseguido obter os melhores valores de energia de van der Waals e nem RMSD, foi capaz de encontrar uma maior

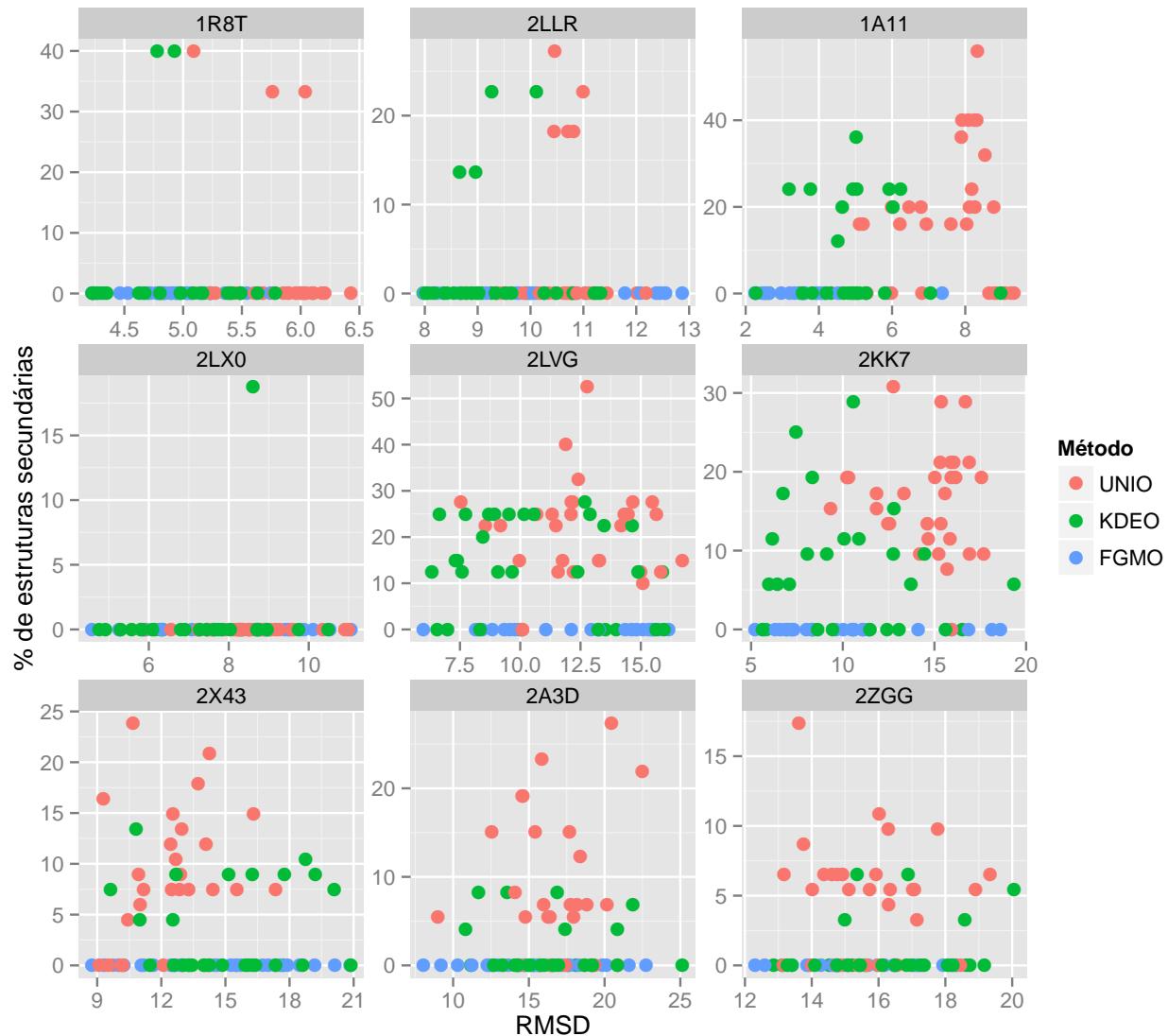


Figura 5.17: Relação entre a porcentagem de estruturas secundárias e o RMSD para a UNIO, KDEO e FGMO.

porcentagem de estruturas secundárias. Nesse caso, a formação das estruturas secundárias não foi favorecida pela menor energia de van der Waals, mas sim por conformações de proteínas com energia de van der Waals acima da menor energia (de van der Waals) encontrada pelos algoritmos que otimizaram mais (a KDEO e a FGMO).

Sabe-se que o uso de outras funções de energias na função de *fitness* de um problema em PSP (além da energia de van der Waals) pode contribuir para encontrar conformações de proteínas com menor RMSD. Isso pode implicar no aumento da energia de van der Waals devido ao efeito da interação de outras energias. Para isso, um experimento foi realizado utilizando a energia de van der Waals e de solvatação para compor o *fitness* da proteína (ver Apêndice C.1). A Figura 5.19 mostra a comparação entre os valores da energia de van der Waals e RMSD para as nove proteínas avaliadas. Nesta figura, é realizada uma comparação entre as execuções em que somente a energia de van der Waals é considerada com os próprios valores da energia de van der Waals, levando

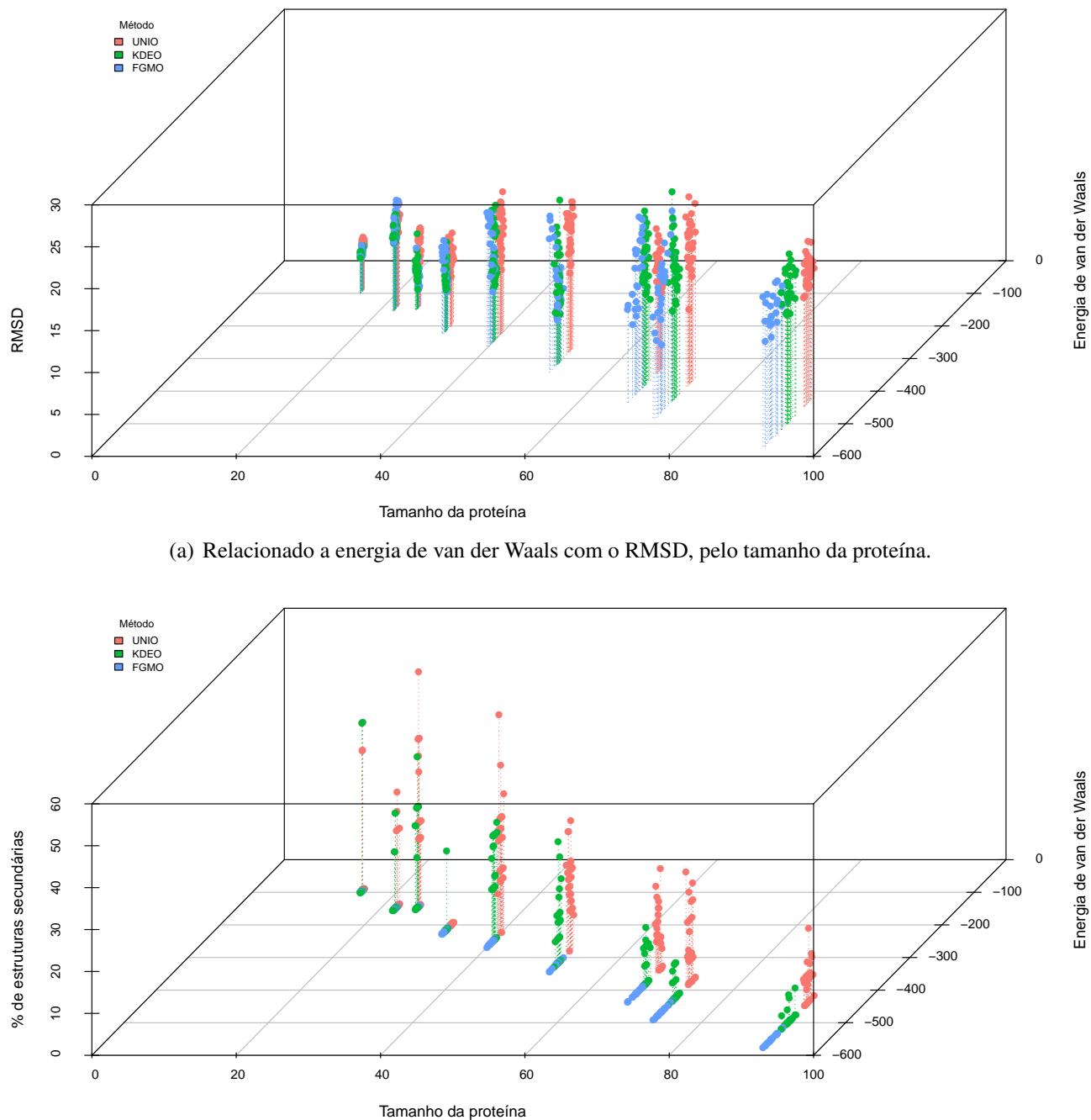


Figura 5.18: Energia de van der Waals pelo RMSD e pelo tamanho da proteína.

em consideração a energia de solvatação no processo de otimização. A Figura 5.19(a) mostra a comparação dos valores da energia de van der Waals para a UNIO. É possível verificar que o RMSD para as duas maiores proteínas é mais baixo quando utilizado a energia de solvatação e, consequentemente, os valores da energia de van der Waals são mais altos. A Figura 5.19(b) mostra a comparação entre os valores da energia de van der Waals para a KDEO. Os valores de RMSD para as duas maiores proteínas também foram menores quando foi utilizado energia de solvatação em sua predição, mostrando também que houve aumento da energia de van de Waals. A Figura 5.19(c)

mostra que a maior proteína obteve um RMSD mais baixo quando utilizado energia de solvatação com altos valores para a energia de van der Waals.

Por fim, a Figura 5.20 mostra que, com exceção à proteína 2A3D, melhores valores de RMSD podem ser obtidos quando é utilizado energia de solvatação em conjunto com a energia de van der Waals. Assim, evidencia-se que é possível melhorar a qualidade da estrutura predita utilizando outras funções de energia, mesmo utilizando um algoritmo mono-objetivo. No entanto, para que esses valores fossem obtidos foi necessário calibrar o peso atribuído à energia de solvatação para cada proteína e para cada método avaliado. No entanto, sabe-se que contribuição de cada função de energia pode variar de acordo com o processo evolutivo. Assim, embora o peso para a energia de solvatação tenha sido calibrado *a priori* (Seção C.1), manter o mesmo valor para todo o processo evolutivo pode não ser totalmente adequado, uma vez que a contribuição de cada energia conforme a proteína se enovelá podem mudar significativamente durante esse processo. Em Brasil et al. (2013) mostram-se resultados experimentais que corroboram com esse aspecto. Por exemplo, a Figura 5.21(i) mostra que a estrutura predita com menor RMSD é compacta demais, desfavorecendo a energia de van der Waals. A Figura 5.21 apresenta as estruturas preditas tanto utilizando somente a energia de van der Waals (em verde), como também uma ponderação entre as energias de van der Waals e solvatação (em amarelo), além do alinhamento delas com as estruturas nativas. Nesse caso, a conformação predita da proteína 2X43 utilizando van der Waals e solvatação obteve uma estrutura mais adequada.

Foi verificado também o efeito de se utilizar ou não o banco de dados de ângulos diedrais (ADB) para a geração da população inicial. A comparação para cada proteína pode ser vista na Figura 5.22, que mostra como os dados estão distribuídos, considerando a energia de van der Waals e RMSD. A Figura 5.22(a) mostra que a maioria dos resultados entre utilizar ou não ADB são semelhantes. A maioria dos *outliers* aparecem quando a população é inicializada sem o ADB, especialmente para a proteína 2LLR. A proteína 1A11 destacou-se por ser melhor minimizada sem ADB, porém, com RMSD relativamente mais alto. É possível perceber também que para as três maiores proteínas desse experimento (2X43, 2A3D e 2ZGG), o uso do ADB contribuiu para minimizar melhor a energia. O efeito oposto ocorre com o RMSD, apresentado na Figura 5.22(b), em que, utilizando ADB, o RMSD das maiores proteínas é mais alto do que não utilizar o ADB. O mesmo efeito ocorreu para a proteína 2LX0, ou seja, utilizando ADB, a energia de van der Waals foi mais alta, porém o RMSD foi melhor. Sem utilizar o ADB a energia de van der Waals foi mais baixa, porém com RMSD mais alto. O efeito “baixo RMSD com maior energia” (conformações de proteínas que possuem um baixo RMSD, mas alta energia) deste experimento contribui para mostrar a dificuldade que existe em tentar resolver o problema de PSP, pois nem sempre estruturas com baixo RMSD são também as que possuem energia mais baixa. Há diversos fatores que podem contribuir para que isso possa ocorrer como, por exemplo, interações na molécula não modelados pelos potenciais considerados.

Sabe-se que além da energia de van der Waals outras energias também atuam na estabilização da molécula. Assim, foi realizado o cálculo das energias não-covalentes: energia de van der

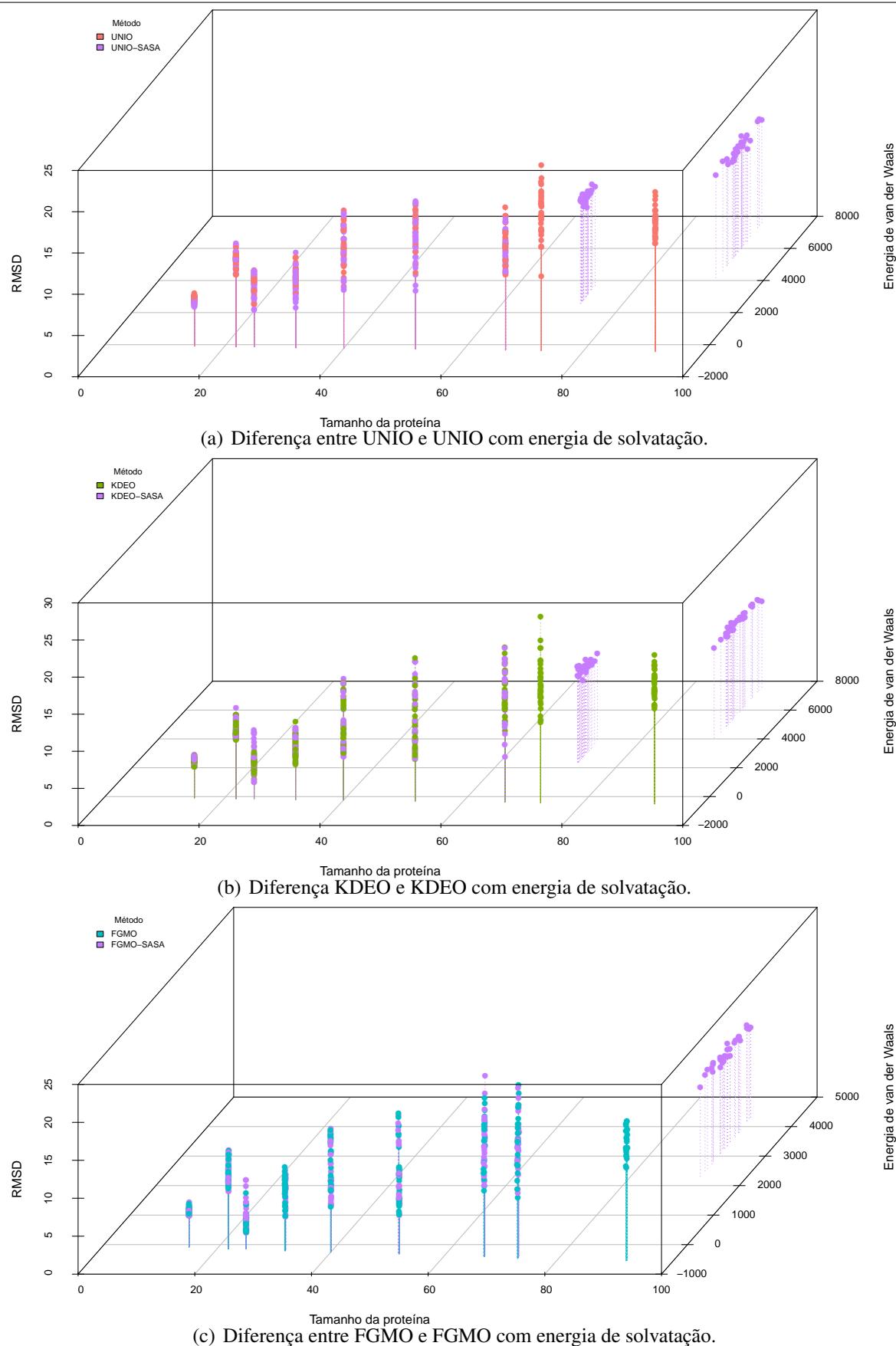


Figura 5.19: Influência da energia de solvatação na energia de van der Waals e no RMSD.

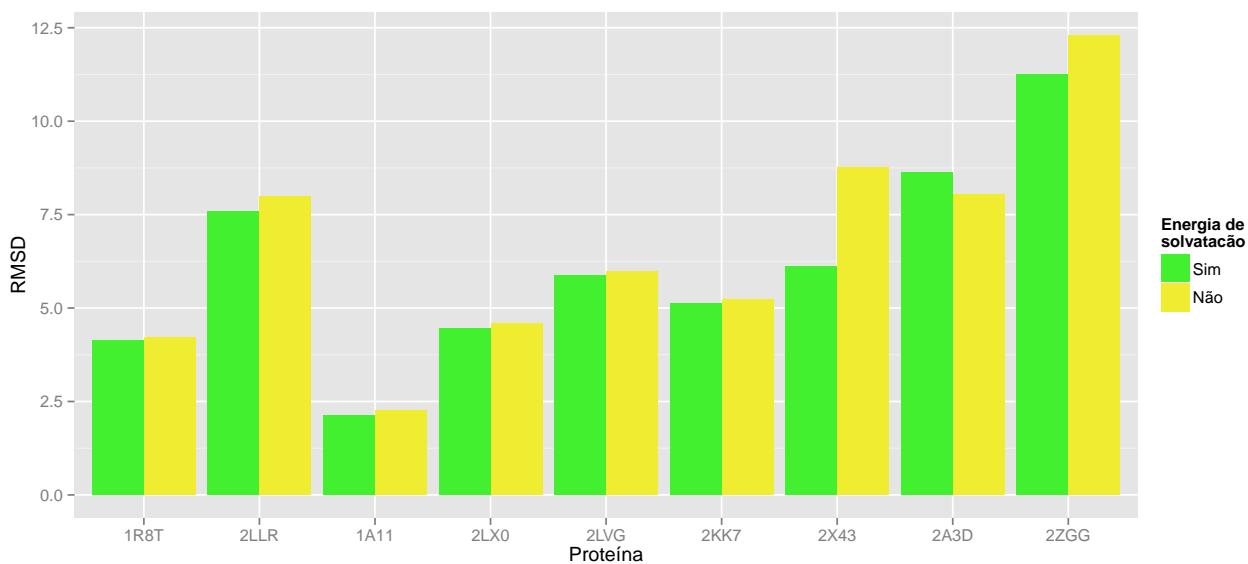


Figura 5.20: Conformação de proteína com melhor RMSD considerando o efeito da energia de solvatação.

Waals, eletrostática, solvatação e de ligações de hidrogênio das melhores estruturas preditas pelos EDAs propostos. Então, essas energias foram comparadas às correspondentes proteínas nativas. É importante destacar que as estruturas obtidas pelos métodos propostos foram otimizadas somente utilizando a energia de van der Waals na função de *fitness*. Para calcular a energia das proteínas nativas a partir do PDB, as coordenadas Cartesianas XYZ foram convertidas em ângulos diedrais utilizando uma rotina do pacote de modelagem molecular VMD (Humphrey et al., 1996), pois sabe-se que tal conversão não é trivial de se obter. Os ângulos diedrais das proteínas nativas são utilizados como entrada para uma chamada da função de avaliação do ProtPred-EDA, produzindo os valores de cada tipo de energia.

A Figura 5.23 mostra a comparação das energias e seus valores para as nove proteínas utilizadas com a conformação da proteína predita com melhor energia de van der Waals. Para todas as proteínas preditas, a energia de van der Waals encontrada foi mais baixa do que a energia de van der Waals das proteínas nativas. No entanto, a energia eletrostática das proteínas nativas é mais baixa do que das proteínas preditas. Assim, ao tentar diminuir a energia eletrostática das proteínas preditas, a energia de van der Waals provavelmente iria aumentar. O mesmo efeito também ocorreu com a energia de solvatação, pois esta energia das proteínas preditas também foi superior às estruturas nativas (com exceção da proteína 2LX0). Ao contrário das energias de solvatação e eletrostática, a energia de ligações de hidrogênio das proteínas preditas foi melhor do que das proteínas nativas. Esse resultado é outro indicativo de que um algoritmo multi-objetivo é importante para tratar esse problema e obter estruturas de proteínas com RMSD mais baixo, conforme foi mostrado em Brasil et al. (2013).

As principais características dos três EDAs propostos avaliados nesta seção podem ser sumarizadas conforme segue:

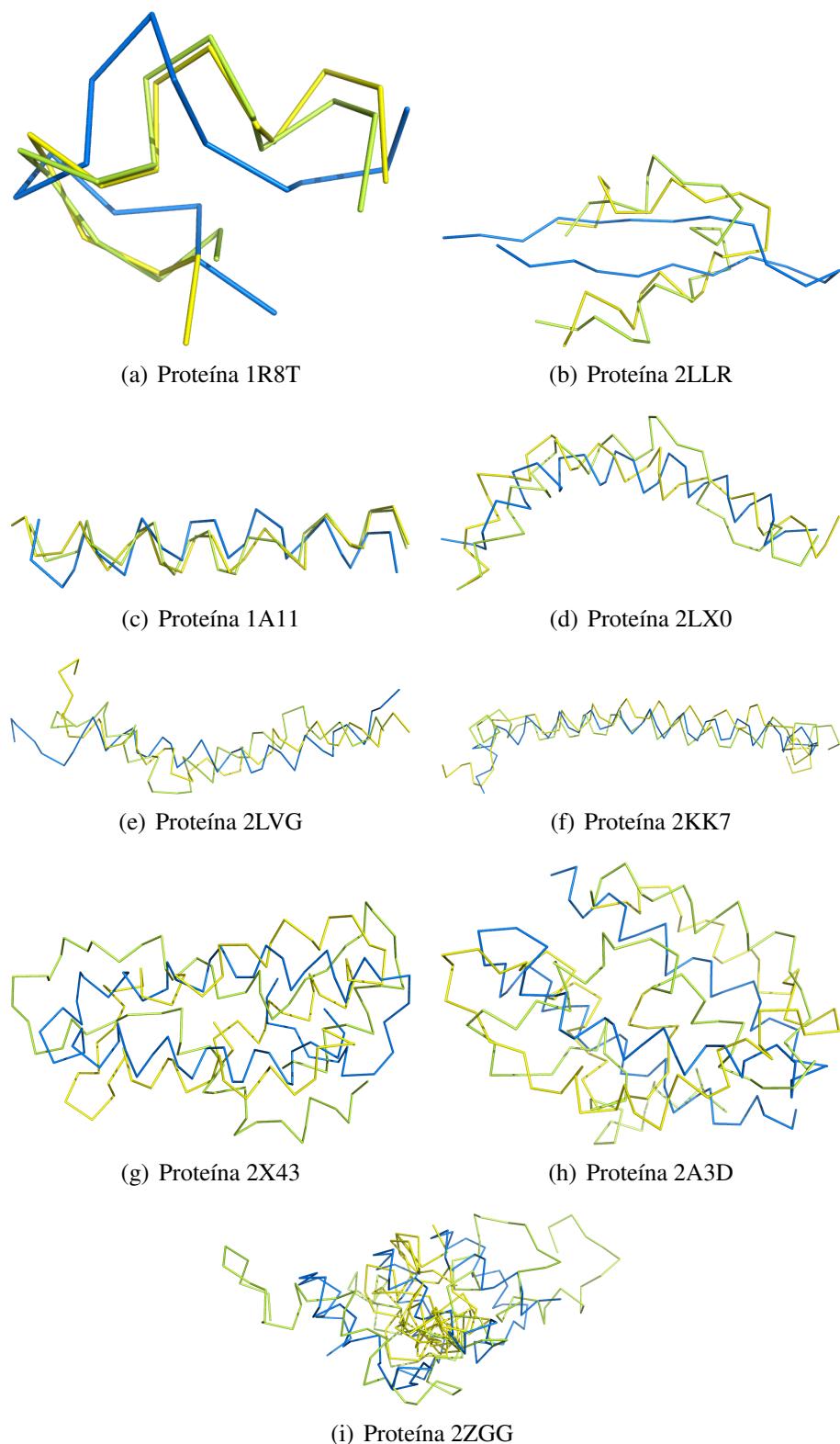
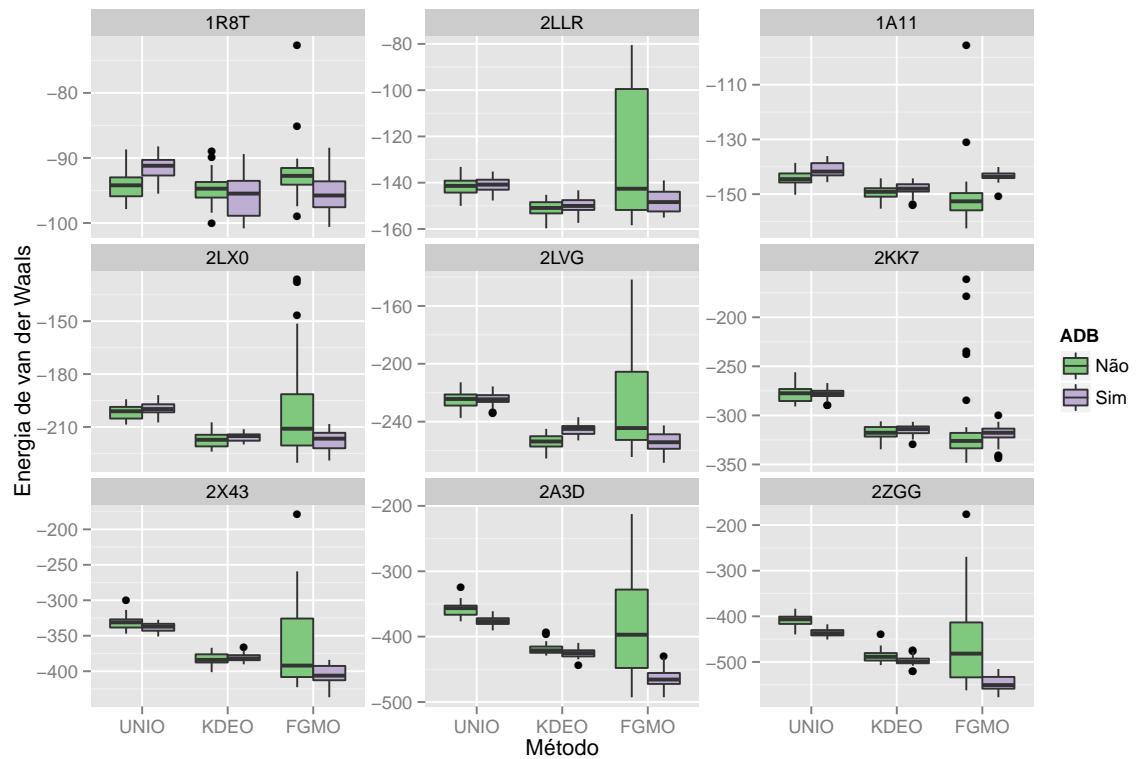
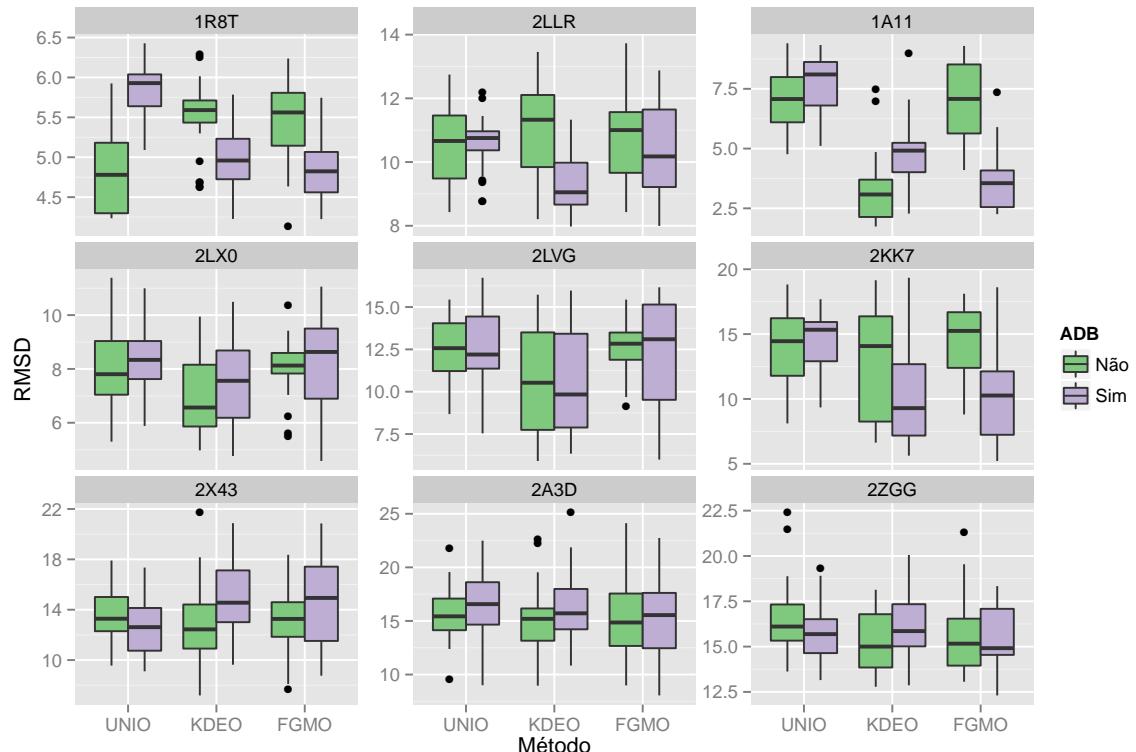


Figura 5.21: Estruturas de proteína preditas utilizando o ProtPred-EDA com energia de van der Waals. As proteínas nativas são representadas pela cor azul. As proteínas preditas utilizando somente energia de van der Waals (em verde) e utilizando van der Waals e solvatação (em amarelo) todas alinhadas à estrutura nativa.



(a) Comparação do valor da energia de van der Waals para as nove proteínas.



(b) Comparação do RMSD para as nove proteínas.

Figura 5.22: Experimentos comparando previsões com e sem o uso do ADB para gerar a população inicial.

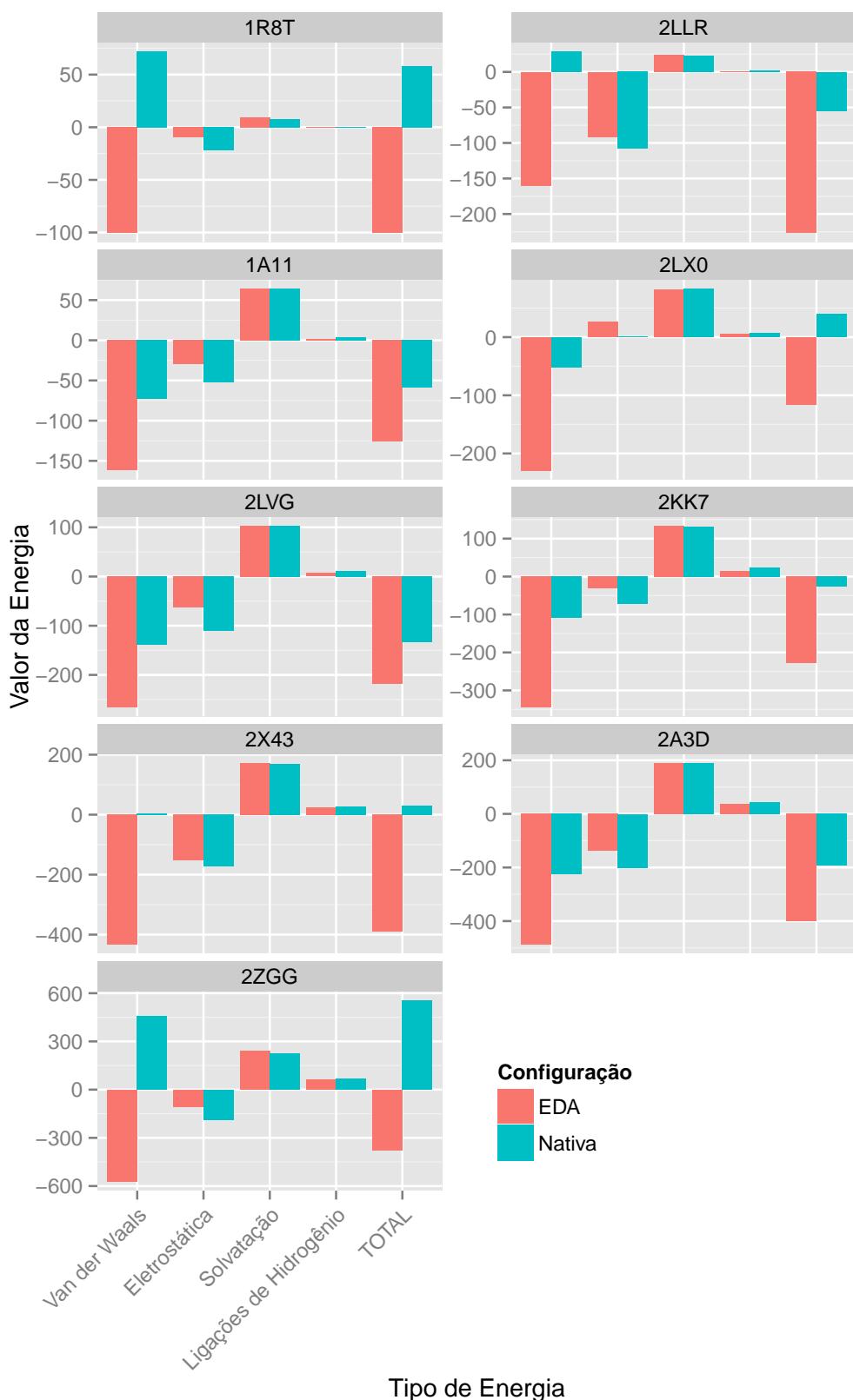


Figura 5.23: Comparação das energias de van der Waals, eletrostática, de solvatação e de ligações de hidrogênio para a proteína com menor energia de van der Waals predita. A Tabela 5.4 mostra qual método encontrou a menor energia de van der Waals para cada proteína.

• UNIO

- Vantagens
 - * Baixo custo computacional;
 - * Minimiza bem a energia para proteínas com até 40 resíduos;
 - * Obteve porcentagens de estruturas secundárias superiores à KDEO e FGMO;
 - * Possibilita formação de pontes de hidrogênio;
 - * Não produz muita variação na energia de van der Waals quando o ADB é utilizado;
 - * Pode ser diretamente estendido a outros problemas do mundo real;
- Desvantagens
 - * Obteve os maiores valores de energia em todos os casos avaliados;
 - * Com exceção de uma proteína avaliada (2X43), obteve os maiores valores de RMSD;
 - * Não apareceu na Fronteira de Pareto entre energia de van der Waals e RMSD;
 - * Requer muitas avaliações, especialmente para proteínas com mais de 50 resíduos;
 - * Não trata a distribuição dos dados de forma circular;

• KDEO

- Vantagens
 - * Minimiza bem a energia de van der Waals, especialmente para proteínas com até 40 resíduos;
 - * Produz conformações de proteínas com baixo RMSD;
 - * Possibilita formação de pontes de hidrogênio;
 - * Não produz muita variação na energia de van der Waals quando o ADB é utilizado;
 - * Encontrou uma certa quantidade de segmentos de estruturas secundárias superior à FGMO;
- Desvantagens
 - * Alto custo computacional;
 - * Não trata a distribuição dos dados de forma circular;

• FGMO

- Vantagens
 - * Otimiza melhor a energia de van der Waals;
 - * Requer relativamente poucas avaliações para atingir valores promissores de energia de van der Waals;
 - * Encontrou valores de RMSD promissores;

- * Baixo custo computacional;
- Desvantagens
 - * Dificuldade em estimar o número de componentes de mistura K (Seção 4.2.3);
 - * O uso do mesmo valor de K para todo o processo evolutivo;
 - * Alta variação da energia de van der Waals quando o ADB não é utilizado;
 - * Não favoreceu a formação de pontes de hidrogênio e nem de estruturas secundárias;
 - * Não trata a distribuição dos dados de forma circular;

5.2 Hierárquico

Esta seção apresenta os experimentos e resultados obtidos com as extensões hierárquicas dos EDAs UNIO, KDEO e FGMO propostas: hUNIO, hKDEO e hFGMO (Seção 3.4). Para esses testes foi utilizado o mesmo conjunto de nove proteínas (Tabela 5.1). Antes da execução dos experimentos foi realizada uma calibração para os parâmetros adicionados dos EDAs hierárquicos: α , que determina a quantidade de variáveis que serão sobrepostas e m , a quantidade de subproblemas. A calibração dos parâmetros e o resultado da calibração é mostrada na Seção 5.2.1 e os experimentos com os parâmetros já calibrados são mostrados na Seção 5.2.2.

5.2.1 Calibração

Após concluir os experimentos com os EDAs propostos UNIO, KDEO e FGM foi iniciado a calibração do EDA hierárquico. Os mesmos parâmetros utilizados nos modelos probabilísticos propostos foram utilizados na extensão hierárquica. Nesse caso, os parâmetros adicionais que precisaram ser calibrados foram o número de variáveis que foram sobrepostas (α) e o número de subproblemas (m). Na verdade, a quantidade de subproblemas é melhor determinada pelo tamanho da proteína, pois estruturas de proteínas muito pequenas podem não ser adequadas para serem decompostas em dois ou mais subproblemas. Acredita-se que existe uma relação entre a quantidade de motivos (dobramentos isolados, ver Seção 2.1) de uma estrutura terciária com a quantidade de subproblemas. É esperado que, ao dividir o problema original em subproblemas utilizando a extensão hierárquica proposta, este seja capaz de criar subproblemas que correspondam, em partes, aos motivos das estruturas de proteínas.

Para a calibração, foi utilizada a proteína de tamanho intermediário dos experimentos, a proteína 2LVG, conforme também realizado para calibração apresentada na Seção 5.1. O parâmetro α foi calibrado primeiramente, escolhendo-se quatro valores diferentes ($\alpha = [0; 1; 2; 3]$). Para cada α foi utilizado nove sementes diferentes, produzindo $4 \times 9 = 36$ execuções dos EDAs propostos hUNIO, hKDEO e hFGMO. A Figura 5.24 mostra o resultado da calibração do α para a extensão hierárquica. Utilizando a UNIO, o parâmetro $\alpha = 0$ obteve a melhor média, porém, $\alpha = 2$ obteve

valores de energia de van der Waals mais baixos. O pior caso para a UNIO é para $\alpha = 3$. Para a KDEO, a média de energia de van der Waals do $\alpha = 2$ foi melhor que os outros três casos, no entanto, o melhor valor de energia de van der Waals ocorreu quando $\alpha = 0$. Utilizando a hFGMO o melhor valor para α foi, tanto na média (para a energia de van der Waals) quanto o melhor valor, obtido para $\alpha = 2$. Dessa, foi considerado que $\alpha = 2$ produziu o melhor resultado nesta fase de calibração, e assim utilizado nos demais experimentos.

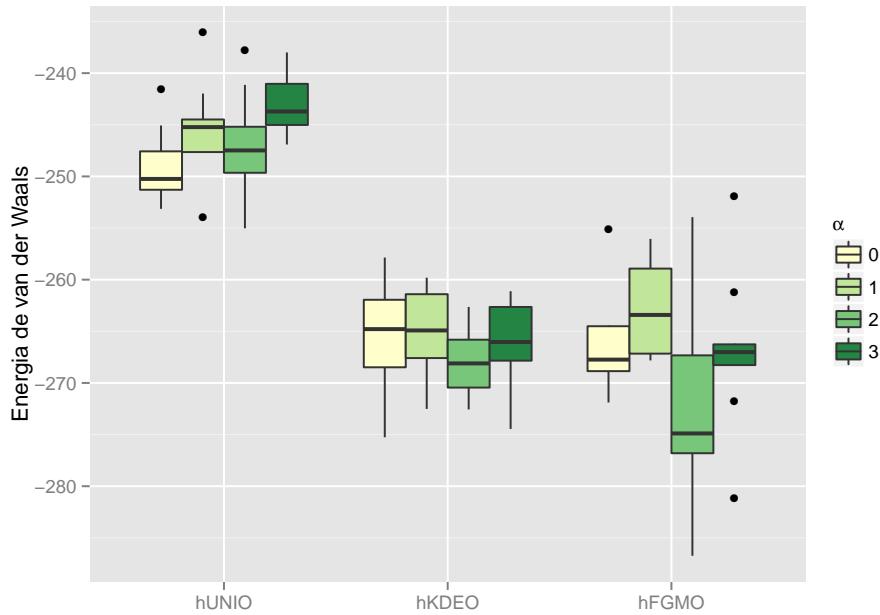


Figura 5.24: Calibração do parâmetro α para os EDAs hierárquicos usando cada um dos três modelos probabilísticos propostos.

Outra execução com a mesma proteína foi realizada comparando os valores da energia de van der Waals, RMSD e tempo de execução entre os EDAs propostos e suas extensões hierárquicas com dois e três subproblemas. Assim, é possível fazer uma comparação preliminar entre os EDA (UNIO, KDEO e FGMO) com os EDAs hierárquicos propostos (hUNIO, hKDEO e hFGMO) bem como determinar a influência de tratar a proteína 2LVG com dois ou três subproblemas. A Figura 5.25 apresenta os resultados de 30 execuções para cada configuração de parâmetros. A Figura 5.25(a) mostra a comparação entre os valores de energia de van der Waals, e evidencia que o uso de dois ou três subproblemas beneficia o processo de minimização da energia de van der Waals.

Utilizando três subproblemas, os resultados foram melhores para a hFGMO. Ao comparar o RMSD pela Figura 5.25(b), pode-se verificar que a hUNIO e hKDEO, com $m = 2$ foi melhor que a UNIO e KDEO, e que a hFGMO com $m = 3$ foi pior que a FGMO. Uma das desvantagens dos EDA hierárquicos é o tempo de execução é maior do que o tempo dos EDAs não hierárquicos, conforme mostra a Figura 5.25(c). Isso faz sentido, de certa forma, pois as extensões hierárquicas precisam de mais avaliações que o correspondente não hierárquico em cada execução.

Outro aspecto importante é que a redução da energia de van der Waals da abordagem hierárquica com $m = 2$ para $m = 3$ não foi significativa. Dessa forma, considerando que a extensão hierárquica com três subproblemas requer mais tempo de computação do que com dois subproblemas e que ambos encontraram valores semelhantes de energia de van der Waals, foi determinado que a extensão hierárquica com dois subproblemas deveria ser utilizada para a realização de novos experimentos, para os demais testes com os EDAs hierárquicos propostos.

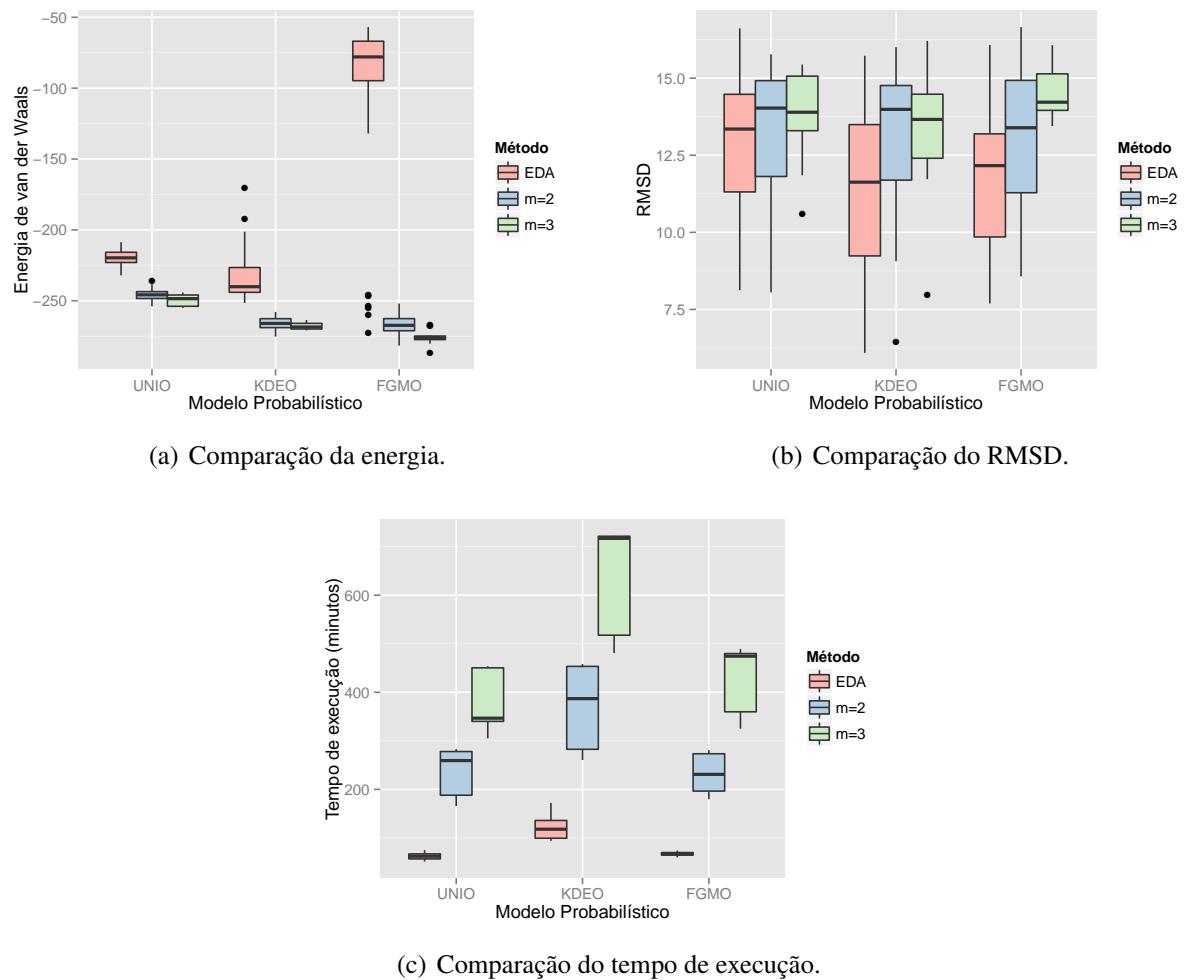


Figura 5.25: Comparação dos EDAs com os EDAs hierárquicos para a proteína 2LVG com $\alpha = 2$ e variando o número de subproblemas: $m = 2$ e $m = 3$ para a proteína 2LVG.

5.2.2 Resultados com EDAs hierárquicos

Ponderando os três aspectos, energia, RMSD e tempo de execução, um novo experimento com as nove proteínas (Tabela 5.1) foi realizado. Esta seção mostra os resultados obtidos a partir 30 execuções com sementes diferentes para o gerador de números aleatórios (o mesmo número de execuções utilizados nos experimentos com a UNIO, KDEO e FGMO) utilizando os parâmetros $m = 2$ e $\alpha = 2$.

As Figuras 5.26 e 5.27 mostram uma comparação entre os EDAs hierárquicos e não hierárquicos. Para todas as proteínas avaliadas, a média da energia de van der Waals da extensão hierárquica foi melhor do que do EDA, para os respectivos modelos probabilísticos. Para a menor proteína do conjunto, 1R8T, a KDEO e a FGMO encontraram valores de energia de van der Waals semelhantes em relação as respectivas extensões hierárquicas, pois é pequena demais para ser dividida em dois subproblemas. A partir de proteínas com o tamanho da 2LLR, com 22 resíduos, a extensão hierárquica foi superior em praticamente todos os casos, levando em consideração a energia de van der Waals. É interessante observar que, a hUNIO para a proteína 1A11 foi capaz de minimizar a energia de van der Waals melhor do que a KDEO ou FGMO. É possível concluir também que, conforme aumenta-se o tamanho da proteína, as extensões hierárquicas em geral apresentam melhor desempenho. Por exemplo, a hFGMO, que foi o único capaz de encontrar um valor de energia de van der Waals abaixo de $-600,00$ kcal/mol para a proteína 2ZGG.

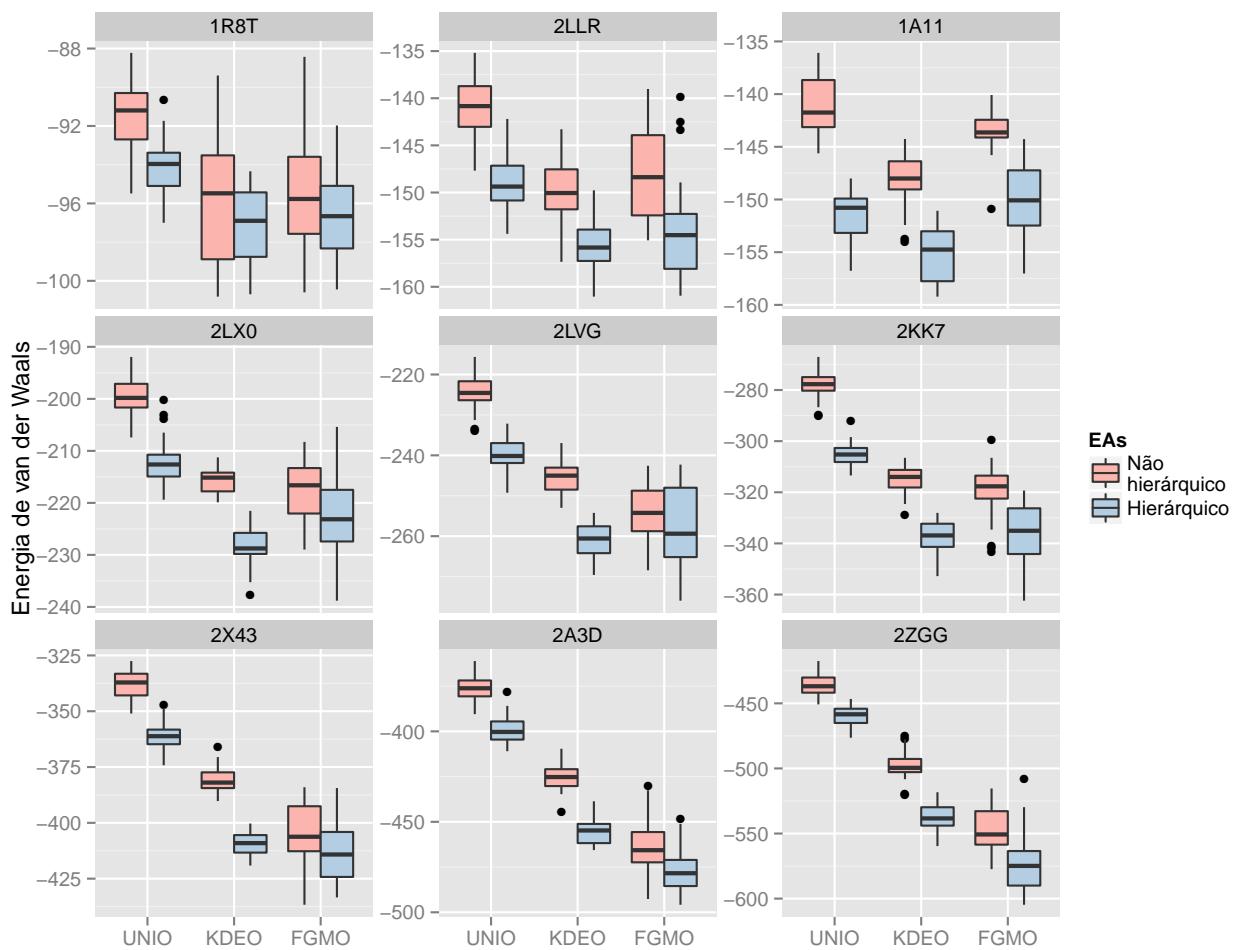


Figura 5.26: Comparação da energia de van der Waals entre os EDAs propostos e suas extensões hierárquicas com $m = 2$ e $\alpha = 2$.

Diferentemente dos valores da energia de van der Waals, a qualidade das proteínas preditas (Figura 5.27) não apresentou um padrão definido. Observando as três maiores proteínas (2X43, 2A3D e 2ZGG) é possível verificar que a mediana do RMSD para o EDA hierárquico foi mais

baixa do que para os não hierárquicos. Embora a mediana do RMSD para as proteínas 2LX0 e 2KK7 seja maior para os EDAs hierárquicos, os melhores valores encontrados foram obtidos pelos EDAs hierárquicos. É interessante observar também que para as três menores proteínas (1R8T, 2LLR e 1A11) a hUNIO obteve uma redução de RMSD considerável, sendo que o menor RMSD ocorre para a proteína 2LLR.

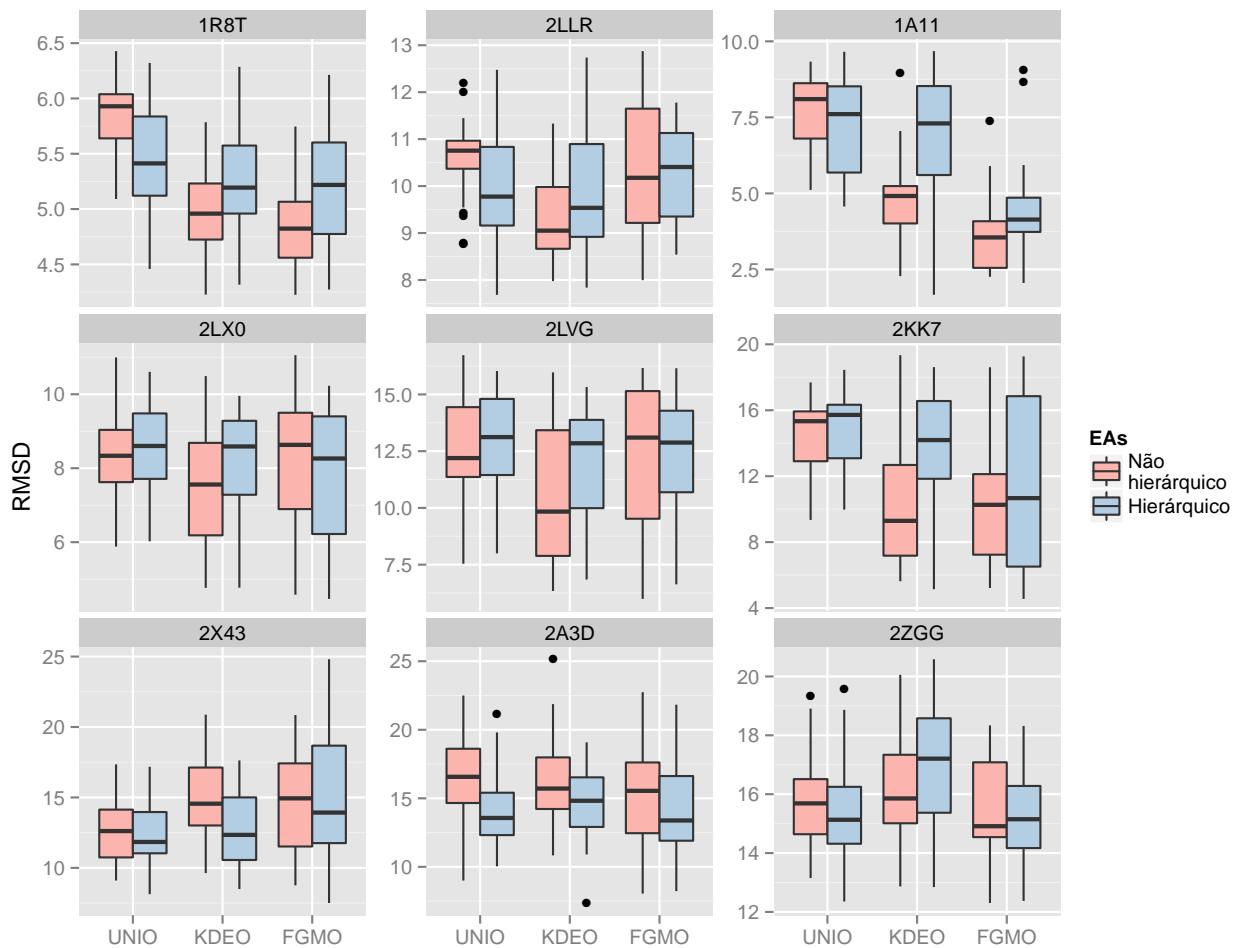


Figura 5.27: Comparação do RMSD entre os EDAs propostos e suas extensões hierárquicas com $m = 2$ e $\alpha = 2$.

A Figura 5.28 mostra um resumo da comparação entre cada método e suas extensões hierárquicas ($m = 2$ e $\alpha = 2$) de acordo com o tamanho da proteína, para 10% das melhores soluções segundo a energia de van der Waals. É possível notar que na Figura 5.28(a), as extensões hierárquicas foram superiores para todos os tamanhos de proteínas. Para proteínas relativamente pequenas, o ganho dos EDA hierárquicos em relação aos não hierárquicos é relativamente baixo. No entanto, é possível notar que, para proteínas acima de 50 resíduos, o ganho dos hierárquicos tende a ser cada vez mais significativo. O RMSD dos EDAs hierárquicos seguiu padrões semelhantes aos não hierárquicos (Figura 5.28(b)). Por exemplo, considerando da menor para a maior proteína, a UNIO seguiu um padrão para todos os tamanhos de proteína. A FGMO também seguiu, mas foi relativamente melhor para as proteínas entre 50 e 75 resíduos. A KDEO também seguiu um

padrão até proteínas com 30 resíduos. Entre 30 até cerca de 60 resíduos, a KDEO foi melhor do que a hKDEO. No entanto, houve uma redução considerável no valor do RMSD da hKDEO para as proteínas acima de 60 resíduos.

Por fim, o tempo computacional (Figura 5.28(c)) mostrou ser uma das desvantagens das extensões hierárquicas, especialmente para a hKDEO. Por outro lado, a hFGMO foi melhor (considerando o tempo de execução) do que a hUNIO em todos os casos, mesmo para as três proteínas maiores, para as quais a FGMO precisou de mais tempo de computação que a UNIO.

As principais vantagens e desvantagens das extensões hierárquicas propostas podem ser sumarizadas conforme segue:

- **EDAs hierárquicos**

- Vantagens
 - * Minimizam bem a energia de van der Waals, independente do modelo probabilístico;
 - * A capacidade de otimização melhora com o aumento do tamanho da proteína;
 - * Produz conformações de proteínas com RMSD relativamente baixo;
- Desvantagens
 - * Custo computacional maior, especialmente para a hKDEO;
 - * É menos adequado para proteínas com número de resíduos inferior a 40;
 - * Seria importante um método para estimar automaticamente a quantidade de sub-problemas m .

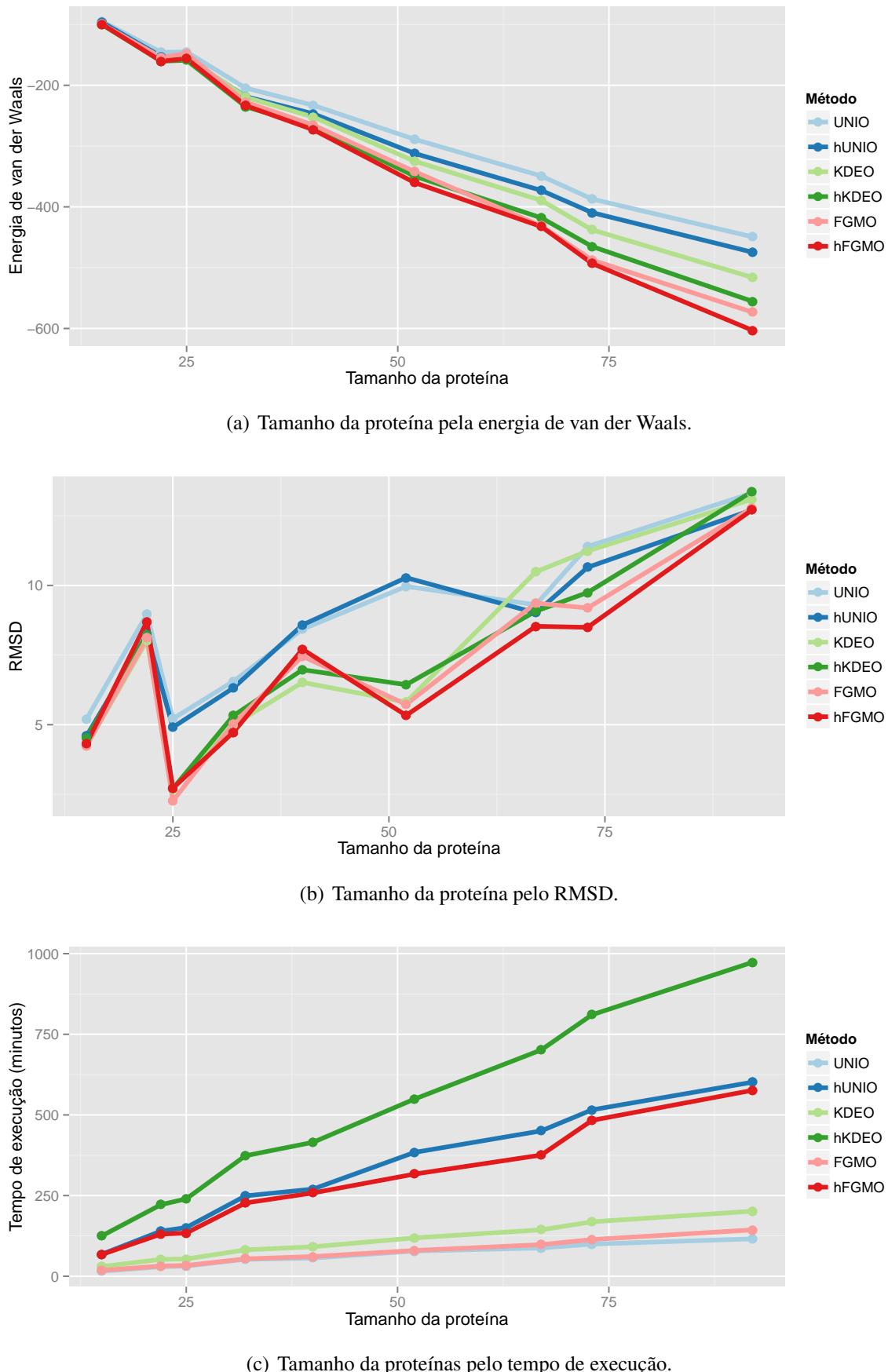


Figura 5.28: Síntese das 10% das melhores soluções obtidas pelo critério energia de van der Waals para os EDAs propostos e suas extensões hierárquicas com $m = 2$.

5.3 Comparação com outros métodos

Esta seção apresenta os experimentos realizados com outros métodos (RW, MC, GA e DE, Seção 4.1) utilizados como referências de desempenho para avaliação da FGMO e hFGMO. Baseado nos experimentos apresentados na Seção 5.2 verificou-se que a FGMO e a hFGMO possuem os melhores desempenho de uma forma geral e, assim, foram escolhidas para serem comparadas com os algoritmos de referência. Os testes consideraram as mesmas nove proteínas (Tabela 5.1) utilizadas nas previsões das Seções 5.1 e 5.2. Antes da execução dos experimentos foi realizada a calibração dos parâmetros utilizados pelas metaheurísticas de referência. As calibrações dos parâmetros e os resultados das calibrações são mostradas na Seção 5.3.1 e os experimentos com os parâmetros já calibrados são mostrados na Seção 5.3.2. Por fim, A Seção 5.3.3 mostra um experimento entre as metaheurísticas e a quantidade de conhecimento *a priori* utilizado para geração da população inicial.

5.3.1 Calibração

Os parâmetros dos algoritmos de referência (RW, MC, GA e DE) foram calibrados utilizando a proteína 2LVG, assim como realizado na calibração dos demais algoritmos nas Seções 5.1.1 e 5.2.1. Os parâmetros de cada algoritmo e os respectivos valores considerados são mostrados na Tabela 5.5. Todos os algoritmos foram calibrados com 36 execuções, determinado ou pela combinação de vários parâmetros ou pela utilização de sementes diferentes para o gerador de números aleatórios. Por exemplo, o único parâmetro da RW é o tamanho da população. Nesse caso, a RW foi executada utilizando a mesma população (500 indivíduos) porém com 36 sementes diferentes. Para o caso do MC, foi utilizado quatro tamanhos de população e três tamanhos do s_{mc} (Seção 4.1.2) e, para cada uma dessas combinações foi utilizado três sementes diferentes, produzindo as 36 execuções. Tanto o GA quanto a DE tiveram quatro tamanhos de população diferentes, três taxas de recombinação e três de mutação, produzindo as 36 execuções para cada um dos algoritmos.

Tabela 5.5: Tabela de parâmetros utilizados na calibração dos algoritmos de referência. Cada execução da calibração é dada pela combinação dos parâmetros.

População	s_{mc} (MC)	Taxa de Recombinação (GA)	Taxa de Mutação (GA)	Taxa de Recombinação (DE)	Taxa de Mutação (DE)
200	0,1	0,3	0,1	0,2	0,2
500	0,5	0,5	0,5	0,4	0,4
1000	1,0	0,9	0,9	0,8	0,8
2000	-	-	-	-	-

A Figura 5.29 mostra a contagem do tamanho da população das oito melhores execuções com base na menor energia de van der Waals encontrada, para os quatro algoritmos. A RW utilizou somente população com tamanho 500 e, por isso, obteve maior contagem. O mesmo tamanho de população, 500, foi o melhor tamanho para o GA. O MC e a DE tiveram uma contagem maior para uma população de tamanho 200 e, por isso, foi utilizado este número para o novo experimento.

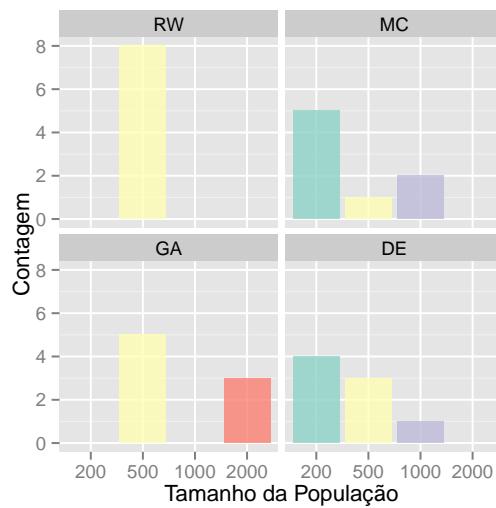


Figura 5.29: Calibração do tamanho da população para os métodos de referência.

Os parâmetros específicos de cada algoritmo são mostrados na Figura 5.30. Para o MC, o parâmetro s_{mc} , que obteve melhor contagem entre as oito melhores execuções, foi $s_{mc} = 1, 0$. A Tabela 5.6 mostra a combinação dos parâmetros para a calibração dos métodos de referência e pode auxiliar na escolha do parâmetro da taxa de recombinação do GA, em que os valores 0, 3 e 0, 5 obtiveram a mesma contagem pela Figura 5.30(b). Devido ao valor 0, 3 ter aparecido nas três primeiras vezes, foi escolhido como taxa de recombinação do GA. A taxa de mutação do GA foi definida em 0, 9. Por último, as taxas de recombinação e de mutação para a DE foram de 0, 2, pois estavam presentes em cinco das oito melhores execuções.

Após a calibração desses métodos, novos experimentos com os parâmetros já calibrados foram executados para as nove proteínas (Tabela 5.1), com 30 sementes diferentes para o gerador de números aleatórios. A Seção 5.3.2 apresenta os resultados obtidos pelos métodos de referência em comparação com a FGMO e hFGMO.

5.3.2 Comparação entre FGMO, hFGMO e métodos de referência

As Figuras 5.31-5.39 mostram os resultados dos experimentos realizados com os métodos de referência e dos EDAs FGMO e hFGMO. Para cada uma das nove proteínas avaliadas, foi comparado a energia de van der Waals, o RMSD, o tempo de execução e a relação entre energia de van der Waals com o RMSD, destacando a Fronteira de Pareto obtida para as soluções de todos os métodos testados nesta seção. Os valores referentes a Fronteira de Pareto, mostrando as solu-

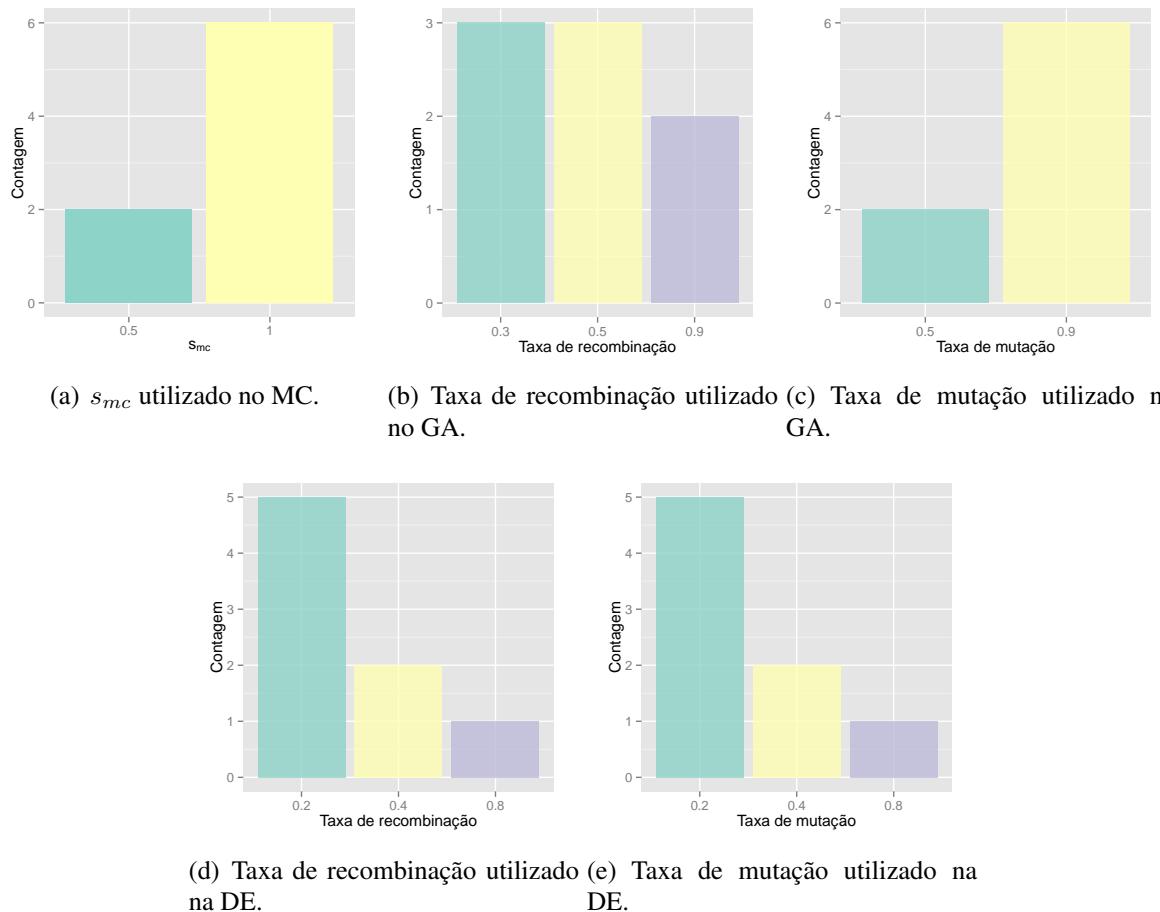


Figura 5.30: Calibração dos parâmetros mostrando a contagem das 8 melhores execuções para os algoritmos de busca de referência.

ções mais eficientes obtidas considerando a energia de van der Waals e RMSD, são mostrados na Tabela 5.7.

A Figura 5.31 mostra os resultados obtidos para a menor proteína do conjunto dos experimentos, a proteína 1R8T. A FGMO e a hFGMO foram capazes de minimizar a energia de van der Waals melhor do que os métodos de referência, sendo que a maioria de seus valores de energia de van der Waals estão abaixo de $-90,0 \text{ kcal/mol}$. Além disso, a mediana do RMSD da FGMO foi a menor em relação aos outros algoritmos avaliados. Na Fronteira de Pareto mostrada é possível perceber que o menor RMSD obtido foi pela RW. Apesar disso, o valor de energia de van der Waals desse ponto é relativamente mais alto do que qualquer outro valor obtido pelos outros métodos. Enquanto que a maioria das metaheurísticas precisaram de cerca de 10 a 20 minutos para predizer essa estrutura, a hFGMO precisou de mais de uma hora.

Para a segunda proteína 2LLR, apresentada na Figura 5.32, é possível perceber também que os melhores valores de energia de van der Waals ocorrem para a FGMO e hFGMO, ou seja, que os algoritmos propostos foram capazes de reduzir mais a energia de van der Waals do que os métodos de referência. Os valores do RMSD obtidos por estas metaheurísticas para a proteína 2LLR foram relativamente altos, considerando o tamanho da proteína. Nesse caso, o melhor RMSD foi obtido

Tabela 5.6: Parâmetros das oito melhores execuções (baseando-se na energia de van der Waals) da RW, MC, GA e DE para a proteína 2LVG.

Método	População	s_{mc}	c_r	m_r	m_f	Energia de van der Waals	RMSD	Tempo (minutos)
RW	500					-11,0	10,7	61
RW	500					-8,8	7,5	85
RW	500					-8,2	12,1	62
RW	500					-7,6	12,6	61
RW	500					-6,7	7,7	68
RW	500					-5,5	9,3	85
RW	500					-5,1	11,2	61
RW	500					-4,4	11,9	85
MC	200	0,5				-153,4	12,1	61
MC	200	1				-149,7	10,7	56
MC	200	1				-137,2	10,8	43
MC	200	0,5				-133,1	14,3	57
MC	1000	1				-132,9	12,9	64
MC	500	1				-131,1	12,0	66
MC	200	1				-121,6	13,7	68
MC	1000	1				-120,4	12,8	67
GA	500		0,5	0,9	0,1	-188,6	13,6	83
GA	500		0,9	0,9	0,1	-188,6	14,6	106
GA	500		0,3	0,9	0,1	-188,6	13,9	106
GA	2000		0,5	0,9	0,1	-179,0	17,8	86
GA	2000		0,9	0,9	0,1	-179,0	17,8	80
GA	2000		0,3	0,9	0,1	-179,0	18,0	76
GA	500		0,3	0,5	0,1	-178,3	14,9	86
GA	500		0,5	0,5	0,1	-178,3	14,9	106
DE	500		0,4			-275,6	9,6	67
DE	1000		0,2			-272,4	11,1	55
DE	200		0,4			-267,0	15,7	39
DE	200		0,2			-260,7	13,4	50
DE	200		0,2			-249,9	13,0	101
DE	200		0,8			-249,2	10,9	65
DE	500		0,2			-235,4	7,8	50
DE	500		0,2			-223,0	14,4	63

utilizando o MC, presente na Fronteira de Pareto. Os outros pontos de Fronteira de Pareto pertencem à FGMO e hFGMO. Ao analisar o tempo de execução é possível notar que, novamente, a hFGMO foi a mais lenta de todos e a DE a mais rápida. Entretanto, é interessante observar também que a FGMO consumiu aproximadamente o mesmo tempo das metaheurísticas mais simples.

Os resultados da proteína 1A11 são mostrados na Figura 5.33. Nesse caso, os valores da energia de van der Waals para o GA, DE e FGMO foram relativamente semelhantes. A hFGMO foi capaz de reduzir mais a energia de van der Waals e também de obter o melhor RMSD. Assim, todos os pontos da Fronteira de Pareto ocorreram para a hFGMO. O tempo computacional da proteína 1A11 seguiu o mesmo padrão das outras duas proteínas descritas anteriormente. Apesar disso, o tempo de execução da FGMO foi aproximadamente o mesmo que o do MC. Isso reforça a ideia de que um algoritmo adequado para PSP puramente *ab initio* com modelo *full-atom* pode ser eficiente, no

sentido de obter estruturas promissoras com tempo de execução semelhante a uma metaheurística relativamente mais simples.

Para a proteína 2LX0, os resultados são mostrados na Figura 5.34. Os melhores valores da energia de van der Waals foram obtidos por meio da FGMO e hFGMO, enquanto que o GA e a DE encontraram valores na faixa de -200,0 kcal/mol apenas. Com exceção da RW, houve uma tendência para valores mais baixos de RMSDs obtidos, sendo que os melhores valores foram encontrados pela FGMO e hFGMO. Na verdade, todos os pontos da Fronteira de Pareto pertencem à hFGMO. Os tempos computacionais também seguiram o mesmo padrão das proteínas 1R8T, 2LLR e 1A11, sendo que a DE foi o mais rápido, a FGMO foi semelhante às outras metaheurísticas e a hFGMO foi o método mais lento.

Para a proteína intermediária 2LVG com 40 resíduos, os resultados são mostrados na Figura 5.35. Considerando a energia de van der Waals, a FGMO e hFGMO foram as únicas a encontrarem valores inferiores a -250,0 kcal/mol, isto é, foram melhores que todos os métodos de referência. Com relação ao RMSD, a RW e a DE tiveram as menores medianas. No entanto, o valor do RMSD mais baixo foi encontrado pela FGMO. Assim, a Fronteira de Pareto é formada somente por pontos pertencentes à FGMO e hFGMO. O tempo de execução mostrou o mesmo comportamento das proteínas descritas anteriormente, em que a hFGMO foi mais lenta e a DE, mais rápida.

Para a proteína 2KK7, os resultados foram ainda mais significativos (Figura 5.36), pois os menores valores da energia de van der Waals obtidos pelos métodos de referência podem ser comparados aos piores valores obtidos pela FGMO. Sendo assim, a maior concentração dos valores da energia de van der Waals para a FGMO e hFGMO são menores do que -300,0 kcal/mol. O RMSD da FGMO e hFGMO também foram relativamente melhores, obtendo uma mediana em torno de 10,0 Å e sendo o melhor RMSD encontrado pela hFGMO menor do que 5,0 Å. Ao comparar os métodos de referência entre eles mesmos, é possível verificar que a RW obteve valores de RMSD relativamente melhores do que o MC, GA e DE. Isso pode ocorrer devido a algumas estruturas que são geradas aleatoriamente que podem ter certa similaridade com a conformação da proteína nativa. No entanto, a energia de van der Waals das estruturas obtidas por meio da RW é significativamente maior do que as energias obtidas pelas outras metaheurísticas. Assim, todos os pontos da Fronteira de Pareto pertencem à hFGMO, dominando a RW, especialmente no aspecto energia de van der Waals (Figura 5.36(d)). O tempo de execução dessa proteína é particularmente interessante, pois a FGMO foi, em geral, mais rápida do que a RW, MC e GA. Por outro lado, a hFGMO que forma a Fronteira de Pareto precisou também de mais tempo de execução.

A Figura 5.37 apresenta os resultados para a proteína 2X43. A partir de proteínas deste tamanho (67 resíduos ou maiores) fica ainda mais expressivo os ganhos obtidos pela FGMO e hFGMO. Observando a Figura 5.37(a) é possível verificar que nem mesmo os melhores valores da energia de van der Waals do GA e da DE foram capazes de superar os piores valores encontrados pela FGMO e a hFGMO. Além disso, a maior concentração dos valores da energia de van der Waals deles inferiores a -400,0 kcal/mol. Apesar de o GA ter conseguido a melhor mediana para o

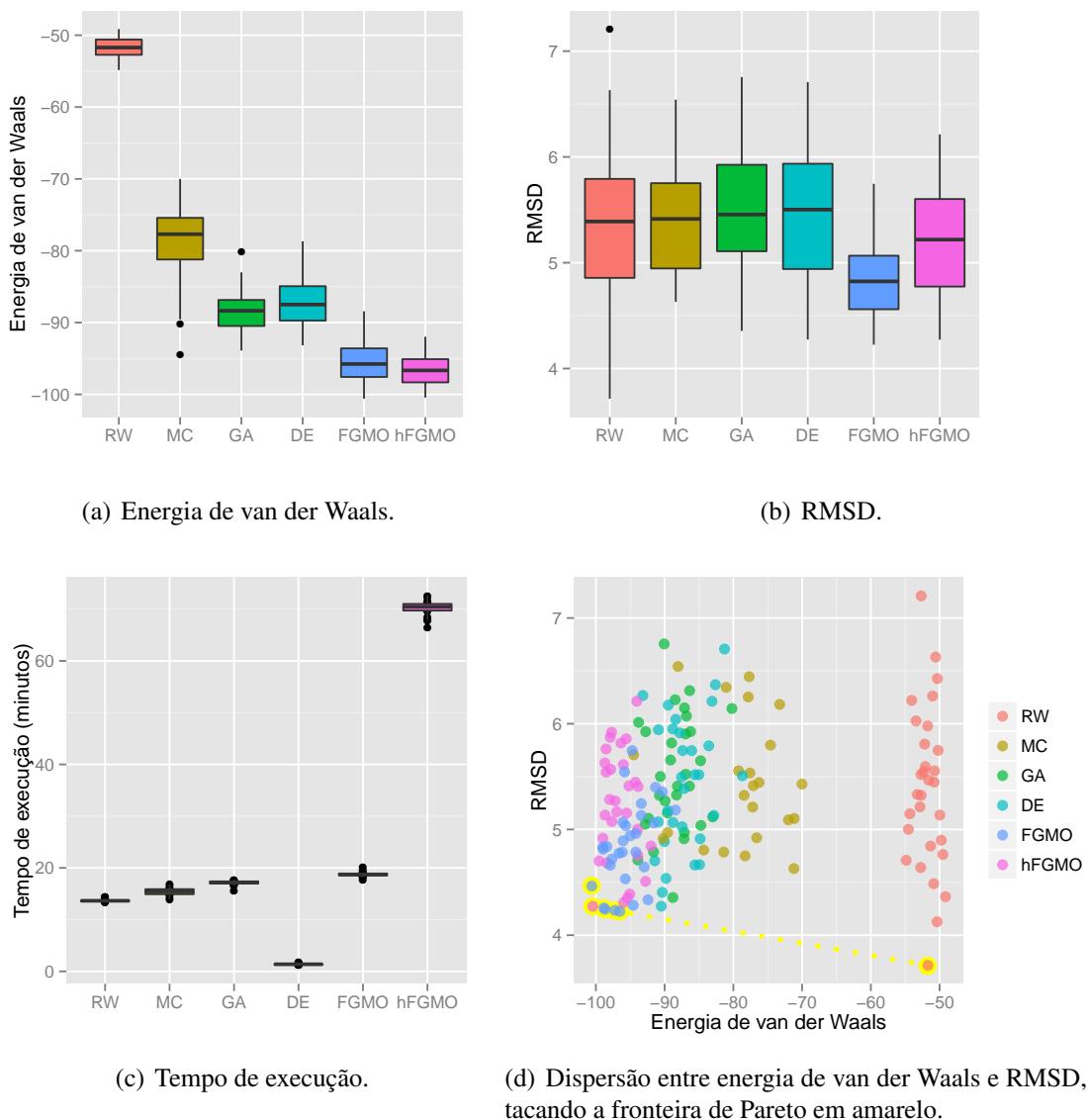


Figura 5.31: Execução dos métodos de referência em comparação com a FGMO e hFGMO para a proteína 1R8T com relação a melhor solução da última geração.

RMSD, a FGMO e hFGMO conseguiram os valores mais baixos. Assim, a Fronteira de Pareto é formada por apenas dois pontos, um pertencente à FGMO e outro à hFGMO. O tempo de execução da FGMO manteve o comportamento semelhante ao das outras proteínas já descritas.

A proteína 2A3D (Figura 5.38) sustenta ainda mais a eficiência da FGMO e hFGMO para essa classe de proteínas, mostrando que foram significativamente superiores do que os métodos de referência, considerando o aspecto energia de van der Waals. A Figura 5.38(b) mostra que o MC, GA e hFGMO obtiveram um valor médio do RMSD semelhante, porém, os melhores valores foram obtidos pela FGMO e hFGMO. Assim, a Fronteira de Pareto foi constituída somente pelos pontos pertencentes às execuções referentes à FGMO e hFGMO. O tempo de execução de cada algoritmo também seguiu o padrão das proteínas anteriores, em que a hFGMO foi significativamente mais lenta do que RW, MC, GA e FGMO, enquanto que a DE foi a mais rápida.

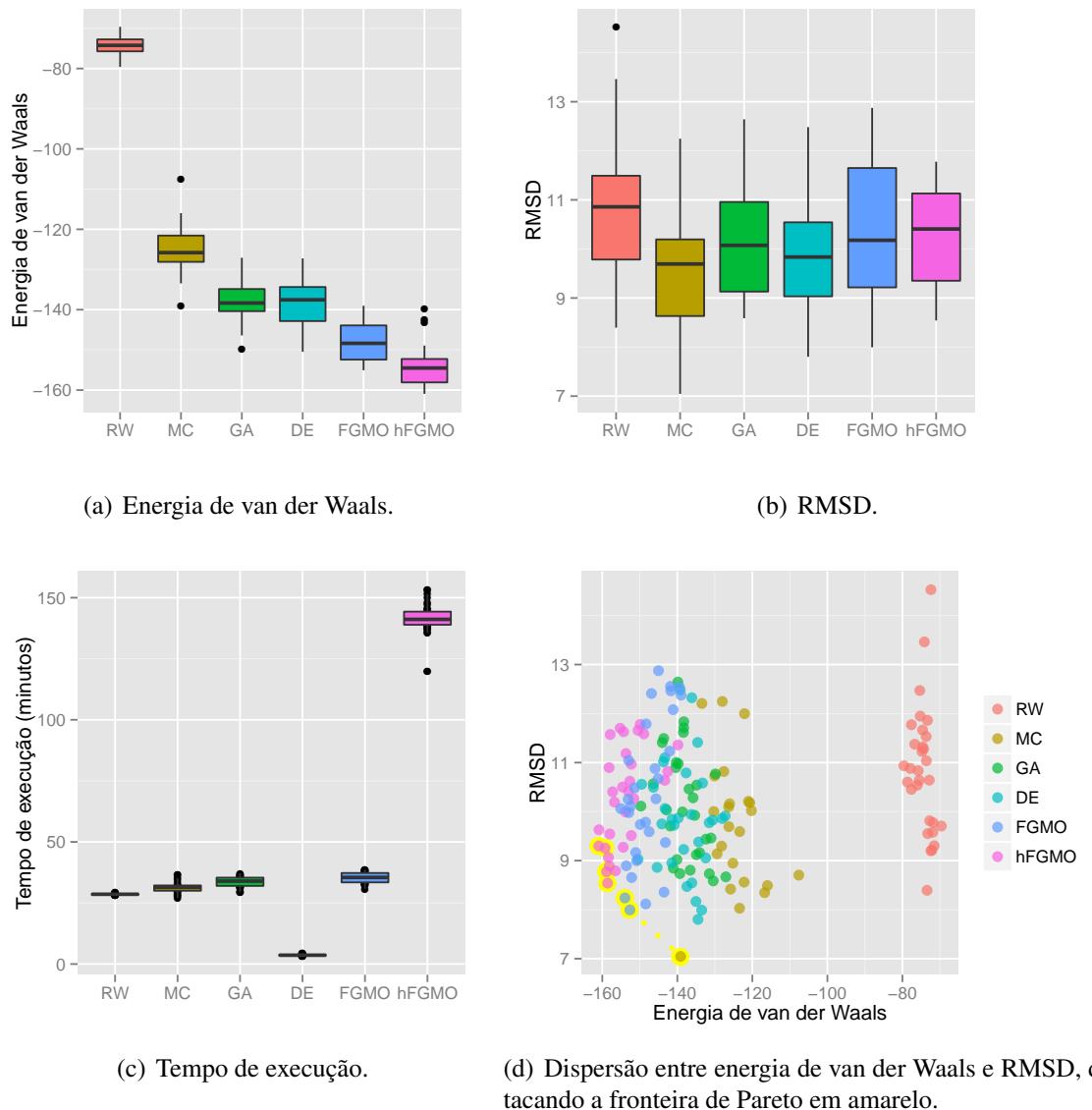


Figura 5.32: Execução dos métodos de referência em comparação com a FGMO e hFGMO para a proteína 2LLR com relação a melhor solução da última geração.

Para a última proteína do conjunto de testes 2ZGG, Figura 5.39, mostra mais uma vez, que a FGMO e hFGMO foram melhores do que a RW, MC, GA e DE, considerando a energia de van der Waals. Para essa proteína o RMSD não foi capaz de descrever adequadamente a qualidade das soluções, pois dobramentos mais compactos para esta proteína tendem a ter RMSD mais baixo, devido a estrutura da proteína nativa 2ZGG possuir uma estrutura compacta de seis α -hélices. O GA obteve o melhor RMSD, sendo parte de um dos pontos da Fronteira de Pareto. Os outros pontos da Fronteira de Pareto para a proteína 2ZGG ocorrem para a hFGMO. Por fim, o tempo de execução entre a FGMO, RW, MC e GA são semelhantes, enquanto que a DE foi a mais rápida e a hFGMO, mais lenta.

Foi realizada uma comparação entre a quantidade de ligações de hidrogênio encontrada nas conformações preditas com os métodos de referência e com os métodos FGMO e hFGMO pro-

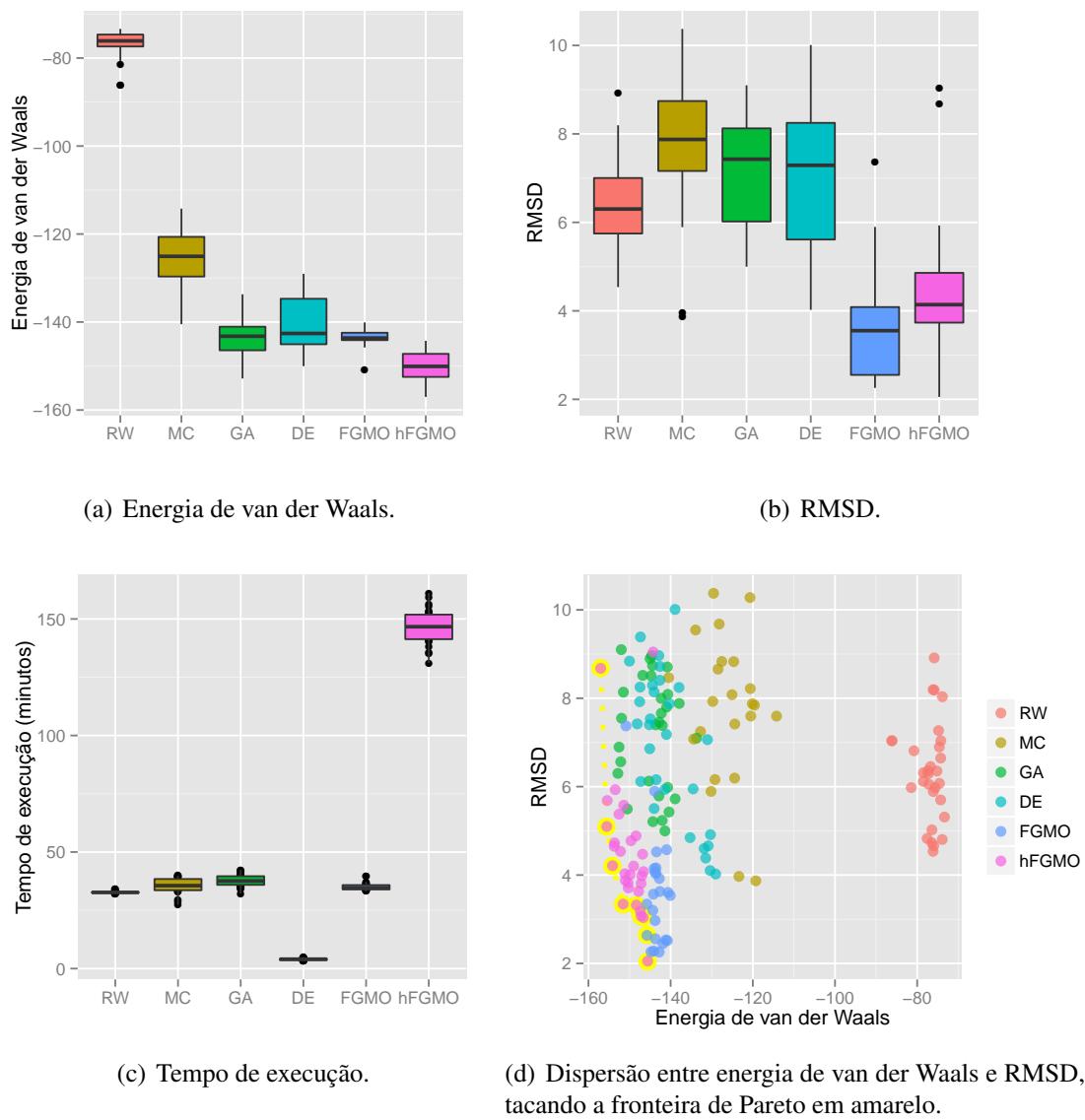


Figura 5.33: Execução dos métodos de referência em comparação com a FGMO e hFGMO para a proteína 1A11 com relação a melhor solução da última geração.

postos, mostrada na Figura 5.40. A quantidade de ligações de hidrogênio encontrada pela RW e GA obteve ganhos mais significativos que as outras metaheurísticas. O GA foi ainda capaz de encontrar uma quantidade significativa de ligações de hidrogênio e com menor RMSD, além do menor RMSD, especialmente para as maiores proteínas (2X43, 2A3D e 2ZGG). Para as demais proteínas (com exceção a proteína 2LLR), o RMSD do GA e MC foram bem distribuídos. Embora o MC e DE tenham obtido RMSD superior a FGMO e hFGMO (com exceção à proteína 2LLR), o número de ligações de hidrogênio encontrados pelo MC e DE foi relativamente superior que a obtida pela FGMO e hFGMO. Este pode ser um outro indicativo de que a energia de van der Waals isoladamente não favoreceu a formação de ligações de hidrogênio, pois as conformações encontradas com menor energia, não revelaram ser as que possuem mais ligações de hidrogênio.

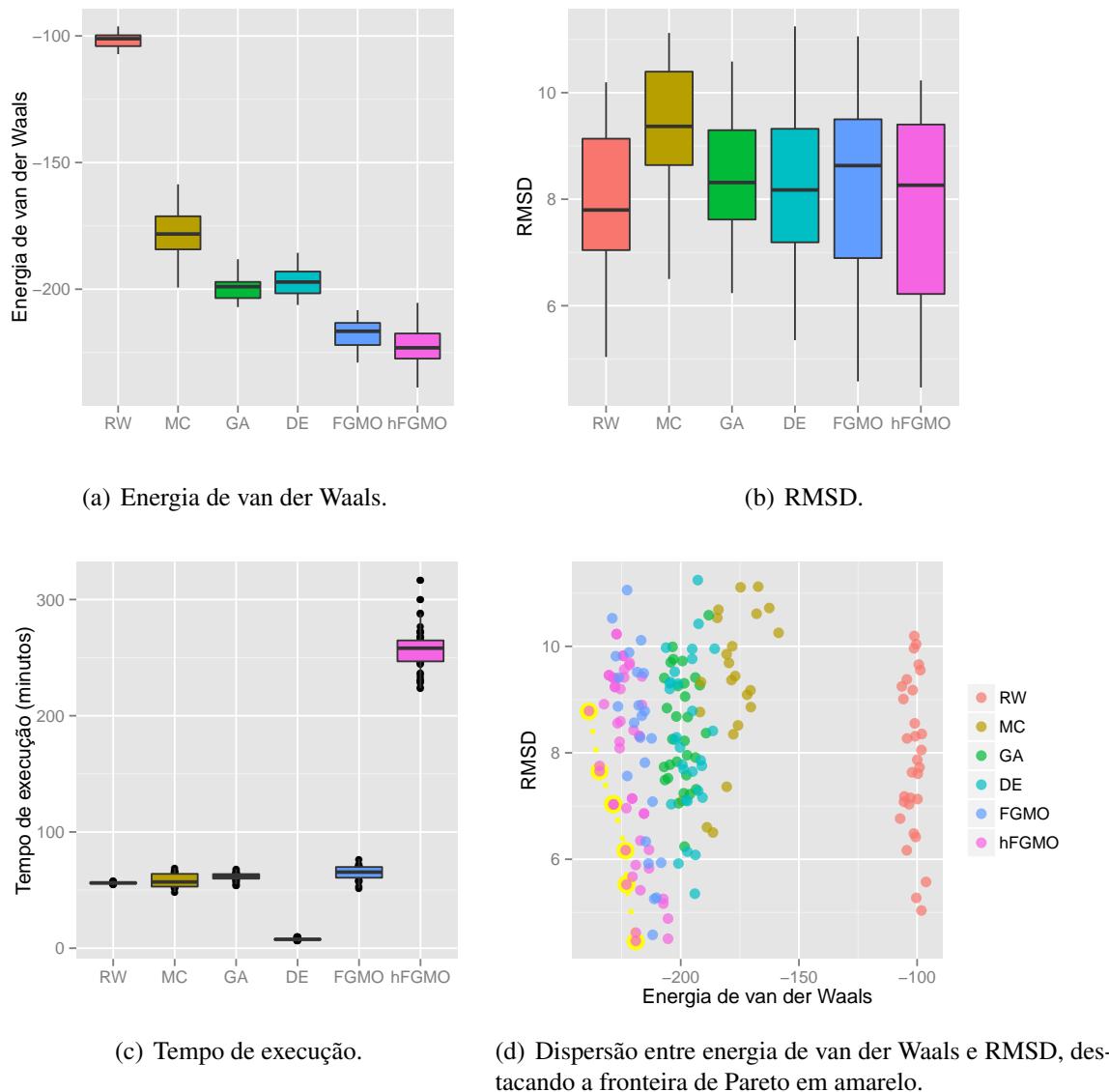


Figura 5.34: Execução dos métodos de referência em comparação com a FGMO e hFGMO para a proteína 2LX0 com relação a melhor solução da última geração.

Na comparação entre a porcentagem de estruturas secundárias com o RMSD, a Figura 5.41 apresenta resultados que corroboram com os mostrados na Figura 5.40, em que o GA destacou-se pelo alto número de porcentagem de estruturas secundárias sem ter sido o algoritmo que obteve os melhores valores de energia de van der Waals. A RW também se destacou nesse caso, pois a porcentagem de estruturas secundárias obtidas para proteínas como, por exemplo, a 1A11, 2LX0 e 2KK7 chegou próximo a 60% para alguns pontos. O MC e DE encontraram em até cerca de 40%, enquanto que a maioria das porcentagens de estruturas secundárias obtidas pela FGMO e hFGMO foram de 0%.

Utilizando a medida de qualidade denominada hipervolume (Auger et al., 2009), para determinar a qualidade de soluções no espaço dos objetivos em problemas multi-objetivos (Deb, 2001), foi realizada uma comparação da medida de hipervolume para os algoritmos avaliados nesta seção uti-

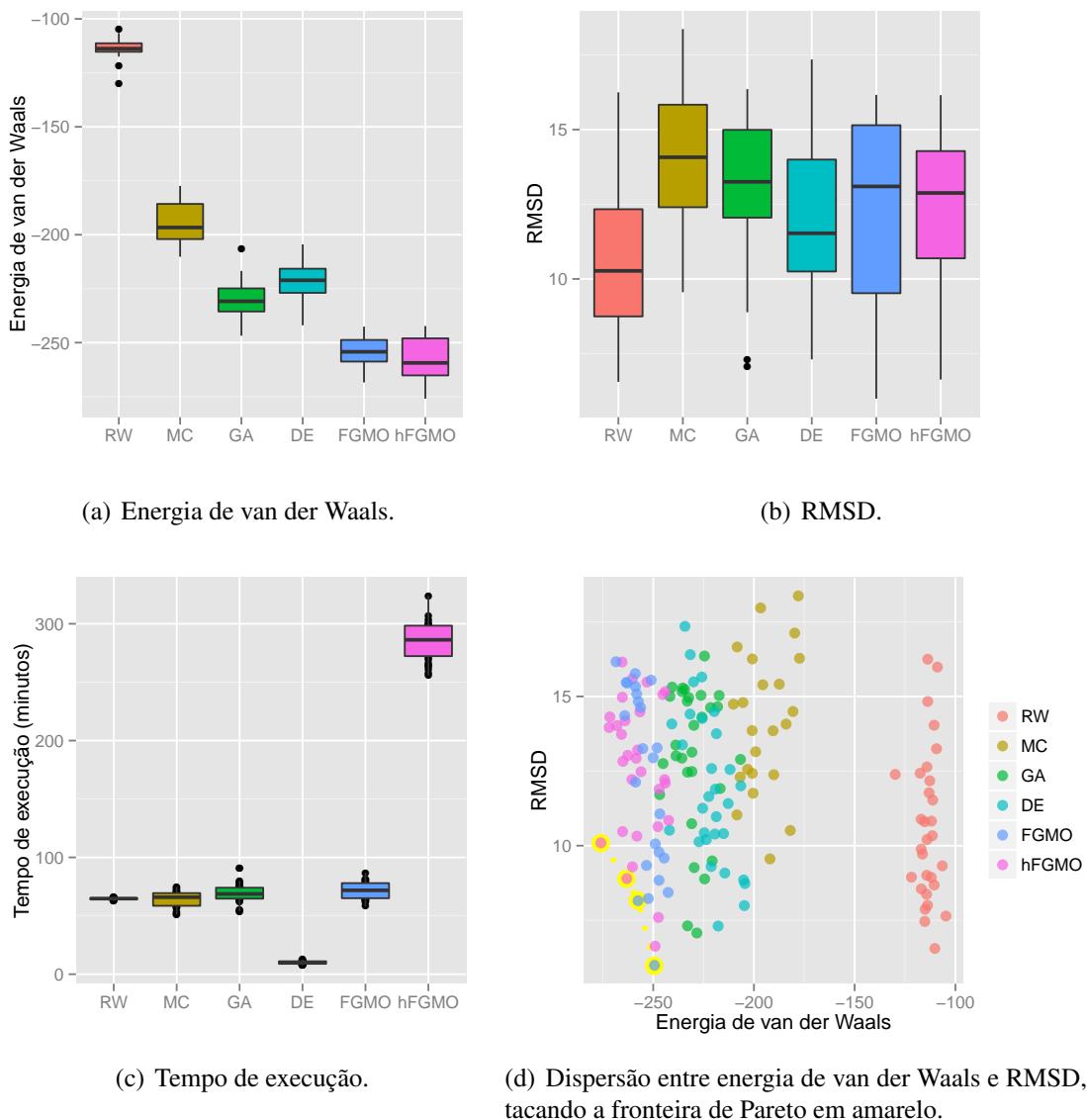


Figura 5.35: Execução dos métodos de referência em comparação com a FGMO e hFGMO para a proteína 2LVG com relação a melhor solução da última geração.

lizando os indicadores energia de van der Waals e RMSD (mostradas nas Figuras 5.31(e)-5.39(e)). A Figura 5.42 mostra a comparação do hipervolume para as nove proteínas avaliadas. A RW obteve o menor hipervolume para todas as proteínas, isto é, em nenhum dos casos a RW obteve pontos de energia de van der Waals e/ou RMSD que dominasse significativamente as outras metaheurísticas. Para a menor proteína, 1R8T, o hipervolume entre as metaheurísticas é relativamente similar (com exceção da RW). Conforme aumenta-se o tamanho da proteína é possível verificar que há uma tendência no aumento do hipervolume. A FGMO e hFGMO foram superiores em todos os casos, mais precisamente a hFGMO. Com exceção à proteína 2A3D, o GA e a DE foram praticamente equivalentes, enquanto que o MC foi inferior ao GA e a DE para as proteínas 2LVG, 2KK7, 2X43 e 2ZGG. Foi verificado que entre as proteínas com 73 e 92 resíduos (2A3D e 2ZGG) houve um comportamento diferente no valor do hipervolume que, ao invés de seguir a tendência e aumentar,

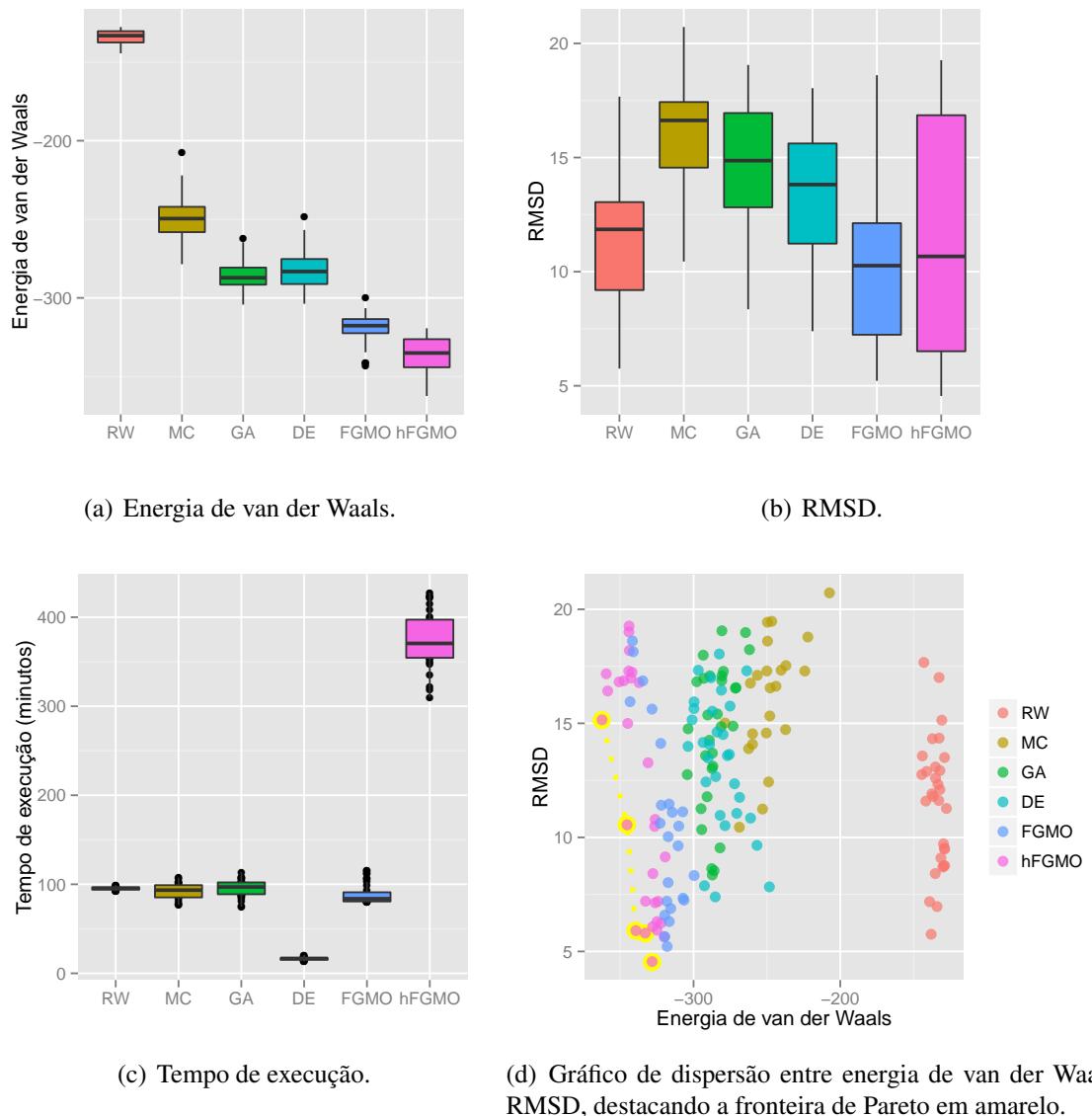


Figura 5.36: Execução dos métodos de referência em comparação com a FGMO e hFGMO para a proteína 2KK7 com relação a melhor solução da última geração.

diminuiu. Isso pode ter ocorrido pelo fato de não ter sido possível obter valores de RMSD para a proteína 2ZGG equivalentes a proteína 2A3D.

As estruturas de proteínas preditas com menor energia de van der Waals e menor RMSD, presentes nas Fronteiras de Pareto, mostradas pela Tabela 5.7, foram comparadas com as estruturas nativas. As estruturas preditas foram alinhadas com as proteínas nativas e exibidas na Figura 5.44.

A Figura 5.43 mostra um resumo da comparação entre os métodos de referência e as técnicas propostas neste trabalho, mostrando a média de 10% das melhores execuções considerando a energia de van der Waals. Observando a Figura 5.43(a) é possível perceber que a RW foi menos favorável se levar em consideração a energia de van der Waals, pelo qual a otimização ocorreu, pois não utiliza nenhum mecanismo para tentar evoluir as soluções. O MC foi superior a RW, pois encontrou valores de energia de van der Waals significativamente melhores. Considerando apenas

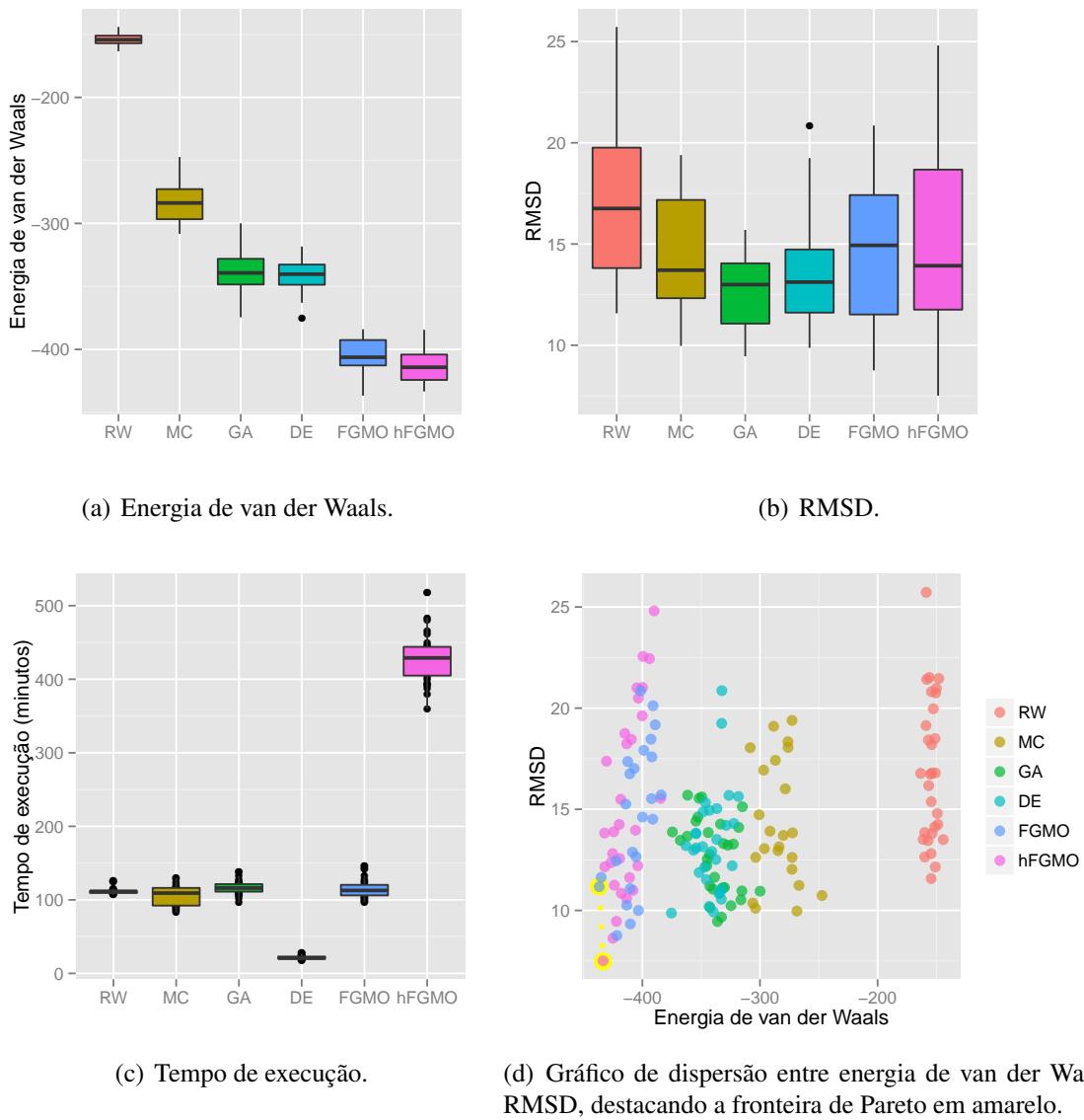


Figura 5.37: Execução dos métodos de referência em comparação com a FGMO e hFGMO para a proteína 2X43 com relação a melhor solução da última geração.

o aspecto energia de van der Waals, o GA e a DE obtiveram desempenhos parecidos para quase todos os tamanhos de proteínas. No entanto, ao considerar também o tempo de execução entre o GA e a DE, pode-se concluir que a DE foi superior ao GA, devido ao baixo custo computacional. Além disso, é possível verificar que os EDAs propostos, ou mais diretamente o modelo probabilístico FGM utilizado nesses EDAs (hierárquico ou não), apresentou resultados superiores para todas as classes de proteínas avaliadas. Para proteínas relativamente pequenas, com até 25 resíduos, observou-se que o ganho da FGMO ou mesmo da hFGMO foram menos significativos em relação aos métodos de referência. No entanto, para proteínas acima de 50 resíduos, a FGMO e hFGMO foram significativamente melhores do que os algoritmos de referência, mostrando que são promissores para essa classe de proteínas. É interessante observar novamente que isso é um indicativo de que o relacionamento de variáveis (ϕ, ψ), de fato, foi capaz de melhorar a exploração do espaço de

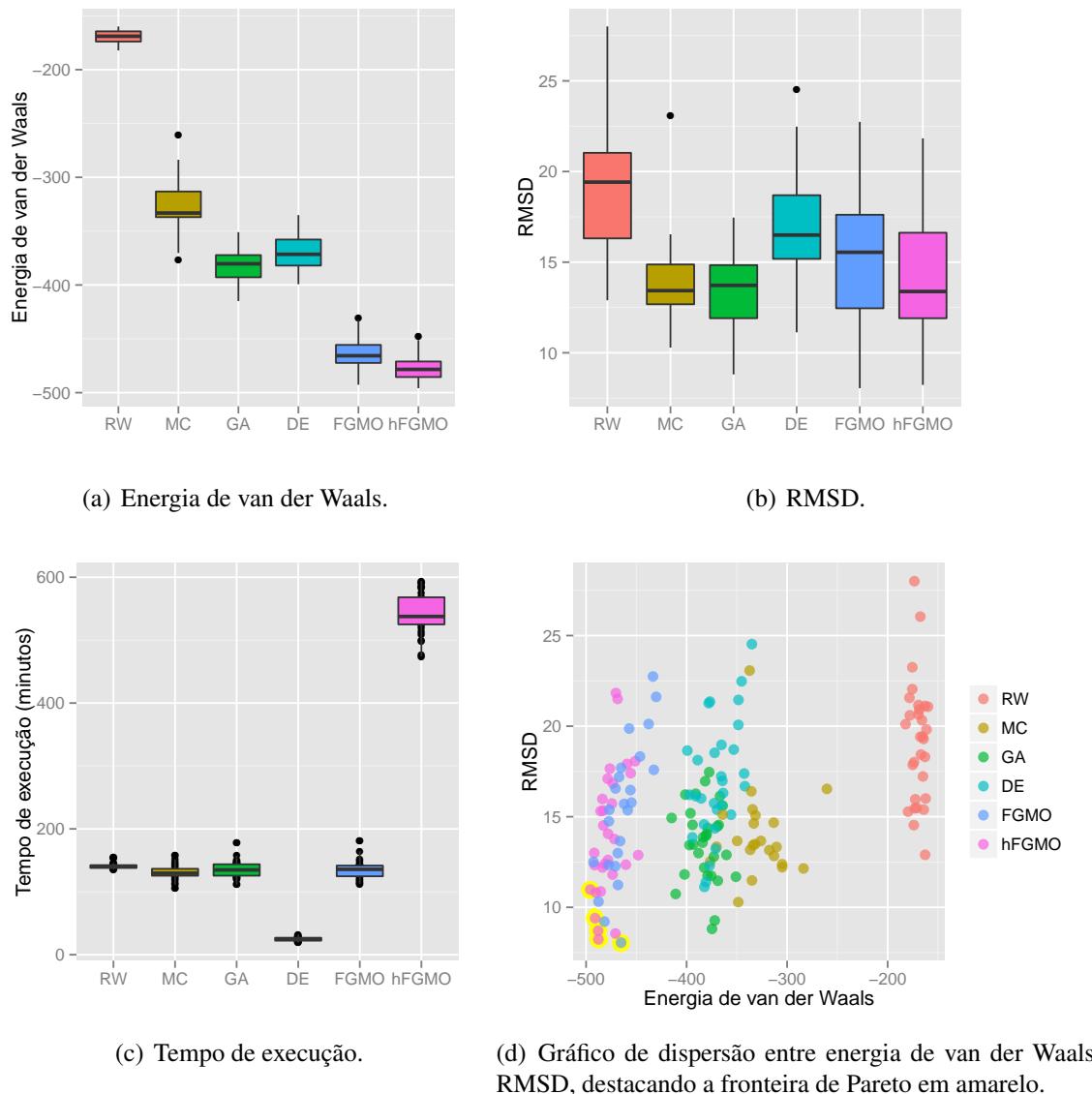


Figura 5.38: Execução dos métodos de referência em comparação com a FGMO e hFGMO para a proteína 2A3D com relação a melhor solução da última geração.

busca. Ao observar o resumo das metaheurísticas com relação ao RMSD (Figura 5.43(b)), pode-se concluir também que a FGMO e hFGMO apresentaram desempenho superior para quase todos os tamanhos de proteínas, com exceção das proteínas 2LLR (22 resíduos) e 2LVG (40 resíduos). Os tempos de execução resumidos de cada metaheurística são apresentados na Figura 5.43(c), em que é possível notar que a DE foi o algoritmo mais rápido. Um dos motivos disso é que a convergência da DE ocorreu antes de atingir um milhão de avaliações, devido ao desvio padrão do *fitness* da população ser inferior ao valor pré-definido. As metaheurísticas RW, MC, GA e FGMO mostraram ter um tempo de execução relativamente semelhante, sendo a FGMO até mais rápido do que a RW para as proteínas acima de 50 resíduos. Esse é um novo indicativo importante para a FGMO, indicando que, no aspecto tempo de execução, a FGMO pode ser semelhante a outras metaheurísticas mais simples. Assim, a FGMO pode ser também considerado eficiente, pois obteve

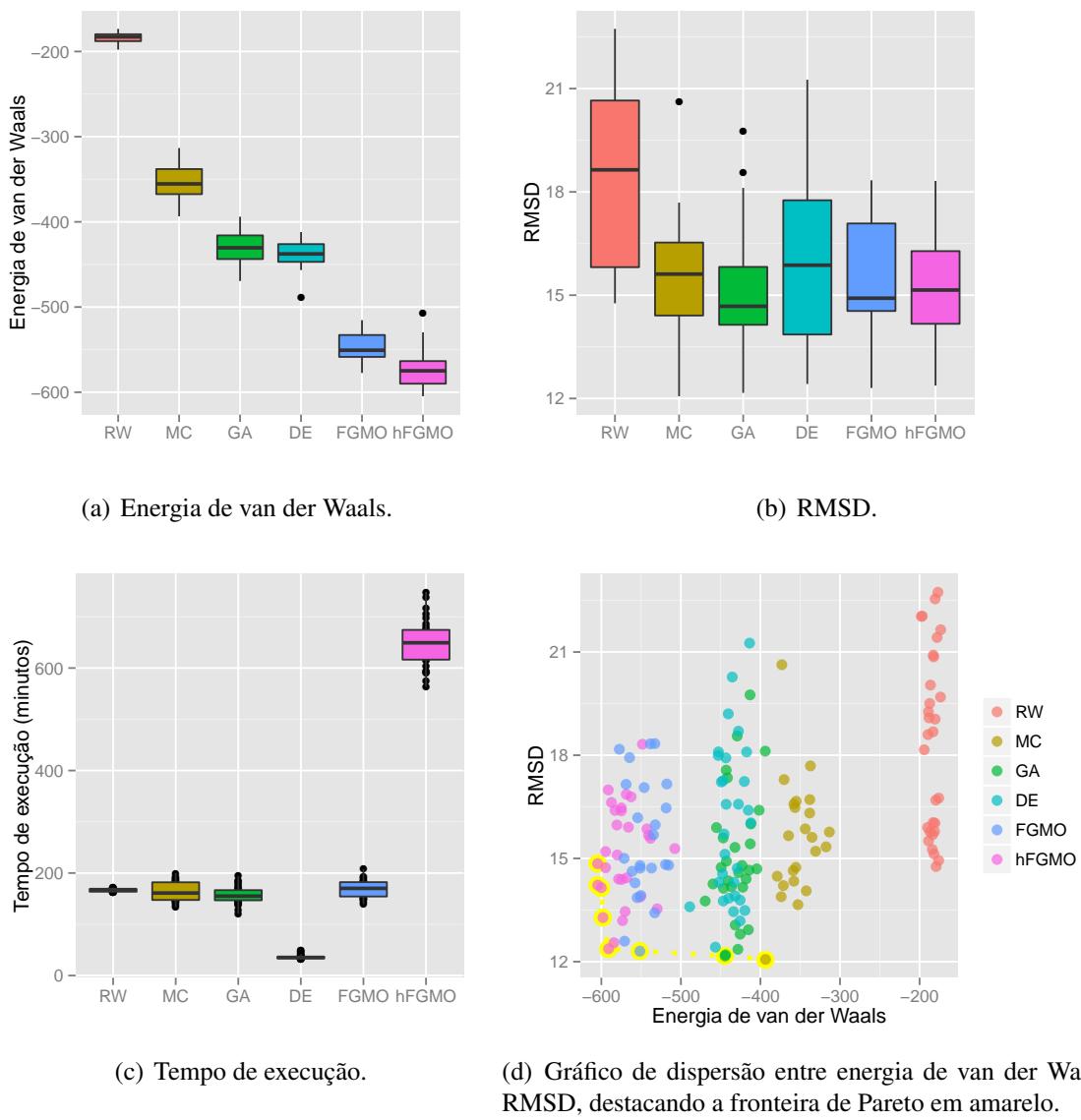


Figura 5.39: Execução dos métodos de referência em comparação com a FGMO e hFGMO para a proteína 2ZGG com relação a melhor solução da última geração.

valores de energia de van der Waals significativamente melhores do que os métodos de referência, com RMSD apropriados e com tempo de computação semelhante às metaheurísticas mais simples. No entanto, embora a hFGMO tenha sido melhor do que a FGMO, o tempo computacional da hFGMO ainda permanece alto. Assim, ao predizer de forma puramente *ab initio* novas estruturas de proteínas tanto a FGMO, quanto a hFGMO mostram-se adequadas. Além disso, a DE também mostrou, de certa forma, ser eficiente para o problema de PSP, pois foi capaz de encontrar valores de energia de van der Waals promissores com um tempo de computação relativamente baixo.

A Figura 5.44 mostra as estruturas das proteínas preditas com menor energia de van der Waals e com melhor RMSD, isto é, os dois extremos da Fronteira de Pareto, para cada proteína avaliada. Os valores das energias e RMSDs das proteínas utilizadas na Figura 5.44 encontram-se na Tabela 5.7. As proteínas preditas com menor energia de van der Waals e menor RMSD para a proteína 1R8T

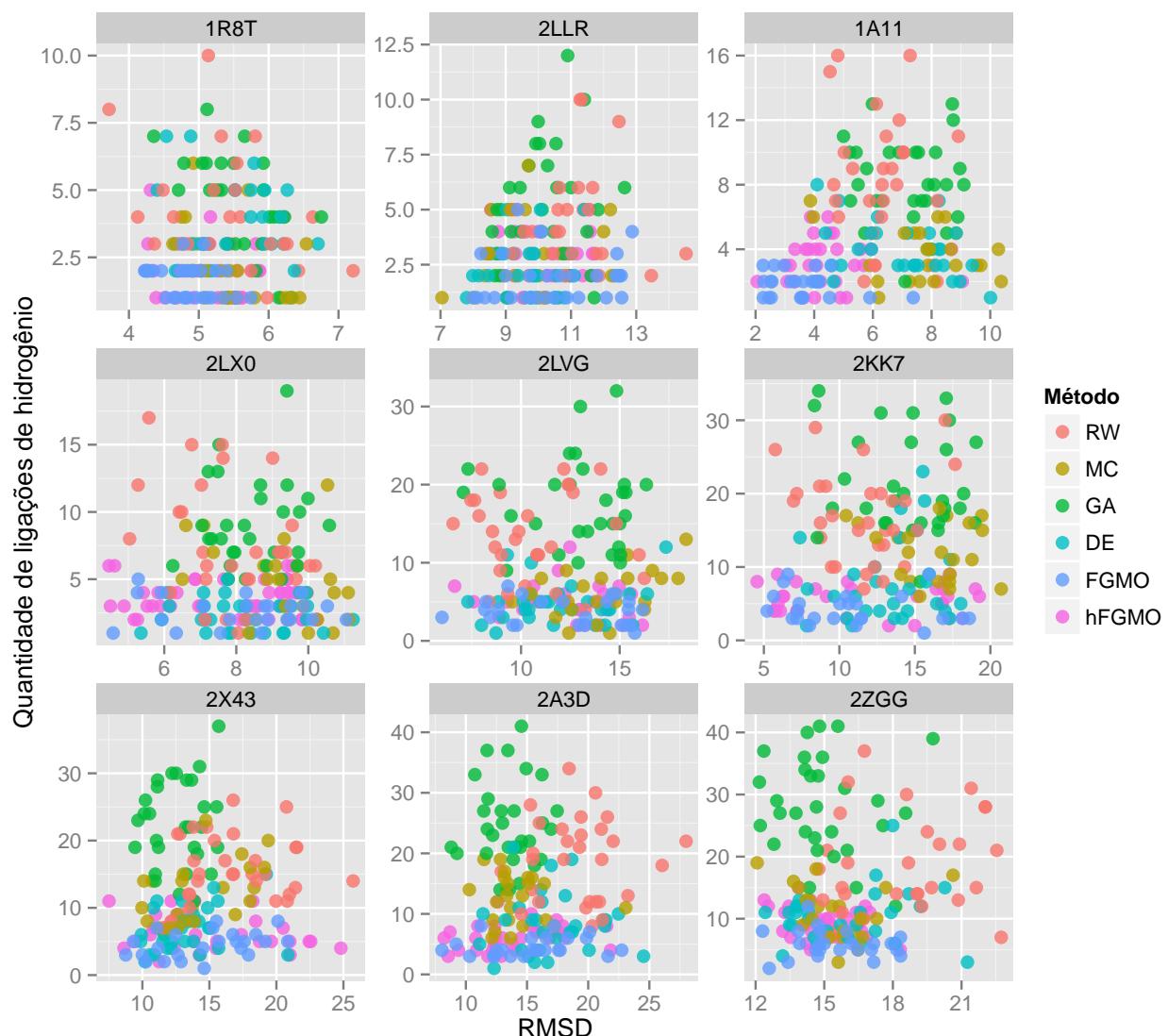


Figura 5.40: Comparação entre a quantidade de pontes de hidrogênio de cada conformação com o RMSD.

mostram que estruturas relativamente semelhantes (estruturas verde e vermelha) podem ter valores de energia de van der Waals significativamente diferentes. As estruturas preditas para a proteína 2LLR não foram capazes de estabilizar-se na forma de uma folha- β . Acredita-se que utilizando a energia de ligações de hidrogênio a predição de folhas- β tenha mais chances de ocorrer, pois seria capaz de modelar a interação entre os átomos $O - H$ das cadeias principais de modo que favorecesse a formação de folhas. De fato, esse tipo de relação entre predição de folhas- β e outras energias como a de ligações de hidrogênio é relacionado no trabalho de Brasil et al. (2013). Mesmo assim, a proteína 1A11 obteve duas conformações interessantes, sendo a estrutura com menor RMSD relativamente semelhante à proteína nativa.

Para a proteína 2LX0 (Figura 5.44(d)), a estrutura predita com menor RMSD ajustou-se melhor à estrutura nativa com a hFGMO do que com a FGMO (Figura 5.14(d)). A estrutura predita com menor energia de van der Waals para a proteína 2LVG reduziu a energia de van der Waals de

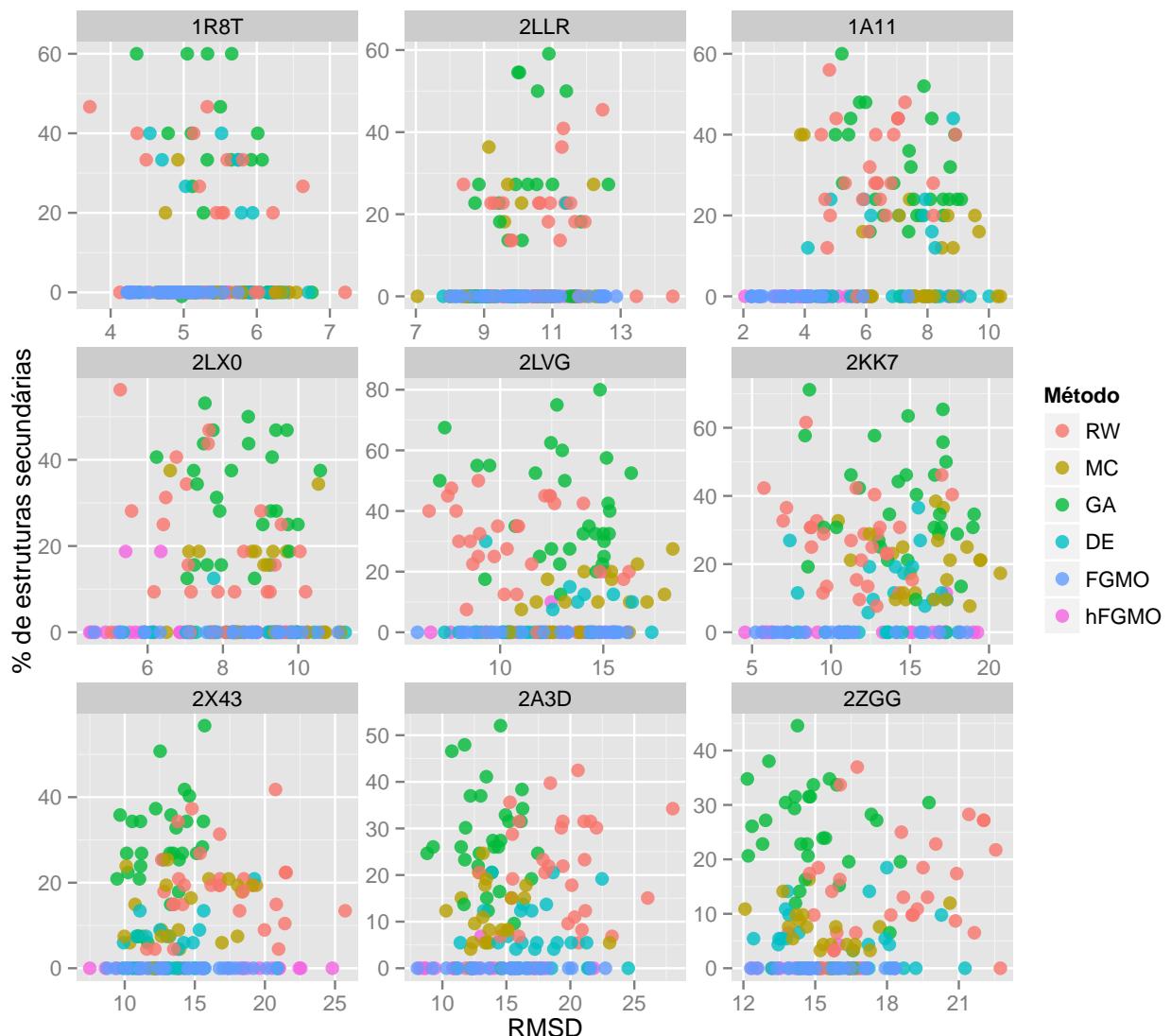


Figura 5.41: Comparação entre a porcentagem de estruturas secundárias de cada conformação com o RMSD.

modo que possibilitasse a interação entre hélices, enquanto que a estrutura nativa é formada por apenas uma hélice. Para a proteína 2KK7, ambas as estruturas preditas são interessantes, porém a estrutura com menor RMSD consegue sobrepor-se adequadamente à estrutura nativa, enquanto que a proteína com menor energia de van der Waals encontrou hélices a mais. A proteína 2X43 pode ser considerada um dos casos mais interessantes mostrados na Figura 5.44(d), pois, diferentemente da estrutura com menor RMSD encontrado pela FGMO (que não resultou na formação adequada das hélices, Figura 5.14(g)), a estrutura com menor RMSD encontrada pela hFGMO (Figura 5.44(g)) foi mais apropriada, pois as hélices entre a estrutura predita e nativa estão relativamente bem alinhadas. Todos os métodos utilizados encontraram dificuldade para predizer adequadamente a estrutura 2A3D, com três α -hélices bem definidas. Nesse caso, a estrutura com menor energia de van der Waals ficou melhor alinhada à proteína nativa do que a estrutura com menor RMSD. Por

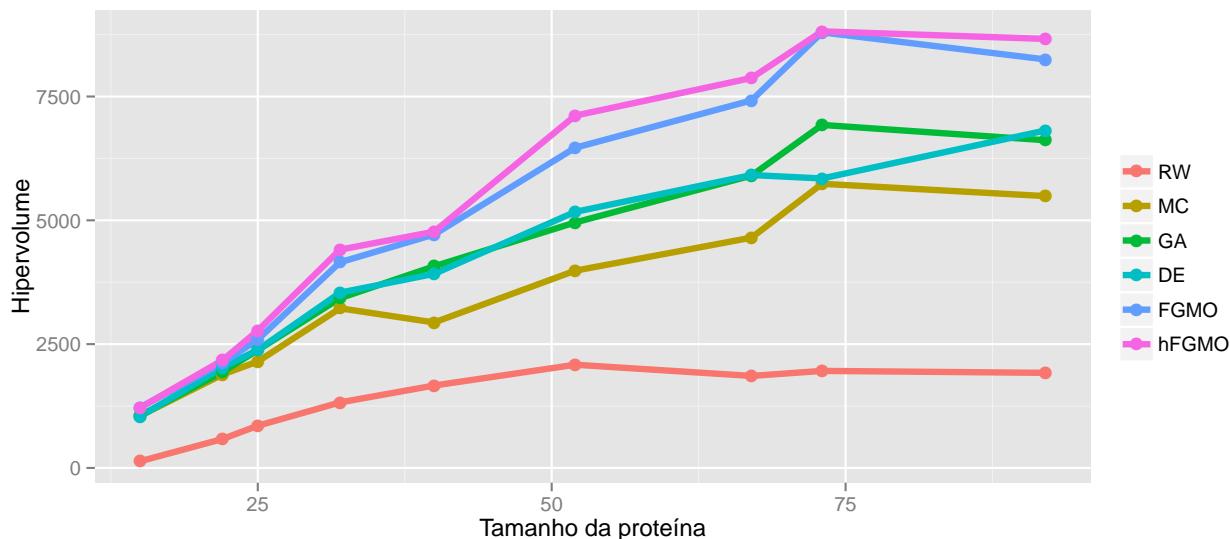


Figura 5.42: Tamanho da proteína pelo hipervolume.

Tabela 5.7: Soluções na Fronteira de Pareto encontradas considerando todas as metaheurísticas.

Proteína	Método	Energia de van der Waals	RMSD	Proteína	Método	Energia de van der Waals	RMSD
1R8T	FGMO	-100,60	4,46	2LX0	hFGMO	-234,33	7,66
1R8T	hFGMO	-100,44	4,27	2LX0	hFGMO	-228,44	7,03
1R8T	FGMO	-98,83	4,26	2LX0	hFGMO	-223,31	6,17
1R8T	FGMO	-98,67	4,25	2LX0	hFGMO	-222,95	5,53
1R8T	FGMO	-97,24	4,24	2LX0	hFGMO	-219,03	4,47
1R8T	FGMO	-96,52	4,22	2LVG	hFGMO	-275,96	10,09
1R8T	RW	-51,70	3,71	2LVG	hFGMO	-263,08	8,90
2LLR	hFGMO	-160,94	9,30	2LVG	FGMO	-257,56	8,15
2LLR	hFGMO	-159,25	9,26	2LVG	FGMO	-249,37	5,99
2LLR	hFGMO	-158,84	8,77	2KK7	hFGMO	-362,41	15,15
2LLR	hFGMO	-158,58	8,54	2KK7	hFGMO	-345,34	10,55
2LLR	FGMO	-153,97	8,24	2KK7	hFGMO	-339,34	5,91
2LLR	FGMO	-152,65	8,00	2KK7	hFGMO	-332,96	5,80
2LLR	MC	-139,12	7,05	2KK7	hFGMO	-328,23	4,55
1A11	hFGMO	-157,02	8,68	2X43	FGMO	-436,70	11,17
1A11	hFGMO	-155,55	5,10	2X43	hFGMO	-433,41	7,51
1A11	hFGMO	-154,15	4,21	2A3D	hFGMO	-495,84	10,98
1A11	hFGMO	-151,56	3,34	2A3D	hFGMO	-491,01	9,40
1A11	hFGMO	-148,35	3,32	2A3D	hFGMO	-487,98	8,70
1A11	hFGMO	-147,36	3,18	2A3D	hFGMO	-487,73	8,23
1A11	hFGMO	-147,19	3,07	2A3D	FGMO	-465,22	8,05
1A11	hFGMO	-146,65	3,04	2ZGG	hFGMO	-604,71	10,20
1A11	FGMO	-145,76	2,63	2ZGG	hFGMO	-590,88	10,19
1A11	hFGMO	-145,59	2,05	2ZGG	hFGMO	-584,09	9,43
2LX0	hFGMO	-238,80	8,79	2ZGG	GA	-443,89	9,28

fim, nenhuma das estruturas sobrepostas à estrutura nativa da proteína 2ZGG foram capazes de mostrar algum padrão.

A Figura 5.45 mostra o efeito que se tem nos valores da energia de van der Waals e no RMSD quando é utilizado uma biblioteca de ângulos diedrais (ADB, ver Seção 2.4) para inicializar a população inicial. Na maioria dos casos, o fato de utilizar ADB contribuiu para que fossem obtidos valores mais baixos da energia de van der Waals, de acordo com o teste de comparação mostrado na Tabela E.36. Entretanto, algumas exceções ocorreram como, por exemplo, não houve variação significativa no valor da energia de van der Waals com ou sem ADB para a DE, somente para as proteínas 1R8T e 2LLR. O mesmo ocorreu com a hFGMO para as proteínas 1R8T, 2LLR e 2LVG e para as proteínas 2LX0 e 2KK7 com a FGMO. A Figura 5.45 também mostra que metaheurísticas mais simples que utilizam o ADB foram equivalentes ou piores do que metaheurísticas mais sofisticadas sem o ADB. Utilizando a proteína 2LVG como exemplo, é possível verificar que os valores da energia de van de Waals obtidos pelo GA e DE com ADB foram, em média, mais altos do que a FGMO sem ADB. Isso contribui para mostrar que os EDAs são capazes de explorar adequadamente o espaço de busca mesmo quando nenhum conhecimento *a priori* é utilizado, contribuindo para que sejam utilizados para PSP puramente *ab initio*. Outra característica interessante da Figura 5.45 é que a hFGMO com ou sem ADB foi capaz de reduzir significativamente a energia de van der Waals para as proteínas pequenas. Entretanto, para as três maiores proteínas (2X43, 2A3D e 2ZGG) a hFGMO sem ADB pode ser comparado à DE, enquanto que a hFGMO com ADB foi melhor que todas as outras metaheurísticas.

O efeito que se tem em utilizar ou não ADB mostrou pouca influência no RMSD (Figura 5.46). Segundo o teste de comparação mostrado na Tabela E.37 houve uma diferença pouco significativa entre o fato de utilizar ou não ADB. As proteínas em que a FGMO obteve uma diferença mais significativa foram as proteínas 1A11 e a 2KK7. Acredita-se que isso ocorreu pelo fato de ambas as duas proteínas serem uma α -hélice. A mesma diferença ocorreu com a hFGMO, que utilizando ADB obteve valores de RMSD significativamente melhores. Na verdade, com exceção a RW, a diferença entre usar ou não ADB para a proteína 1A11 foi significativa para todas as metaheurísticas. As proteínas 2LX0, 2A3D e 2ZGG também mostraram uma diferença significativa no valor do RMSD para a hFGMO. É interessante observar também que em alguns casos como, por exemplo, o MC para as proteínas 2LVG e 2X43, a DE para a proteína 2A3D e a FGMO para as proteínas 2X43 e 2ZGG obtiveram valores de RMSD melhores quando não utilizaram o ADB.

5.3.3 Metaheurísticas e a quantidade de conhecimento *a priori*

Após concluir os experimentos entre os métodos de referência e os EDAs propostos, foi realizado um novo experimento para determinar a relação que existe entre quantidade de conhecimento *a priori*, ou seja, o tamanho do espaço de busca em PSP, e a capacidade de sucesso de uma metaheurística (Figura 5.47). Para este novo experimento foi utilizado a proteína 1A11 e os resultados são apresentados em um gráfico que relaciona o sucesso das metaheurísticas com a quantidade de conhecimento *a priori* (Figura 5.47(a)). Cada metaheurística foi capaz de encontrar a estrutura correta da proteína 1A11 utilizando um certo nível de conhecimento *a priori*, chamados de P1, P2,

P3, P4 e P5. Sabe-se que a média dos ângulos diedrais (ϕ, ψ) da proteína 1A11 estão concentrados na região $[\phi; \psi] = [-65, 0; -35, 0]$ (Figura 5.1). Utilizando essa informação como base, o espaço de busca foi sucessivamente dividido em 50 retângulos, em que cada retângulo com lados de 5,0 graus na direção de ϕ e 5,0 graus na de ψ (Figura 5.47(b)). Isso produziu um total de 50 níveis do conhecimento *a priori* para que cada método utilizasse para gerar suas populações iniciais, partindo da mais tendenciosa, isto é, do menor retângulo mais próximo de $[-65, 0; -35, 0]$, até a região menos tendenciosa, isto é, todo o espaço de busca.

Observando a RW pela Figura 5.47(a), é possível verificar que mesmo a metaheurística mais simples foi capaz de encontrar a solução correta (estrutura em azul, nível de conhecimento *a priori* que corresponde ao P5 na Figura 5.47(b)). Porém, foi necessário utilizar muito conhecimento *a priori*, produzindo uma predição tendenciosa, isto é, foi praticamente dado a resposta ao algoritmo antes de sua execução. Isso também significa que para uma sequência de aminoácidos diferente da proteína 1A11, a RW possivelmente irá encontrar a mesma estrutura. O método de MC, mais sofisticado que a RW foi capaz de encontrar a solução correta utilizando menos conhecimento *a priori*, isto é, aumentando o espaço de busca para P4, utilizado para gerar a população inicial. A partir de P4, a RW não é mais capaz de encontrar a solução correta, pois o espaço de busca torna-se relativamente grande para a RW (observar estruturas em amarelo, para as quais a RW falha se com outros níveis de conhecimento *a priori* forem utilizados). O GA, conseguiu encontrar a solução correta utilizando menos conhecimento *a priori* do que o MC, possibilitando aumentar o espaço de busca ainda mais para P3. Nesse caso, tanto a RW quanto o MC não foram capazes de encontrar a solução correta. A DE obteve a estrutura correta com um espaço de busca maior (P2), enquanto que o GA, MC e RW falham ao obter a estrutura correta com tal nível de conhecimento *a priori*. Por fim, o EDA proposto (a KDEO) foi capaz de encontrar a solução correta utilizando todo o espaço de busca (P1), isto é, sem utilizar conhecimento *a priori*, caracterizando uma predição puramente *ab initio*. Utilizando tal espaço de busca, todas as outras metaheurísticas falharam ao tentar obter a solução correta.

Sabe-se que uma das principais características que diferenciam uma metaheurística de outra é a capacidade que elas têm em estimar e amostrar novas soluções. A RW, por sua vez, não utiliza informação de outras conformações para estimar e amostrar novas soluções, isto é, novas soluções são geradas utilizando nenhum indivíduo, pois novas soluções são geradas aleatoriamente. O método de MC utiliza apenas um indivíduo para amostrar uma nova solução. O GA utiliza em geral 2 indivíduos para compor uma nova solução, enquanto que a DE utiliza o princípio três indivíduos. Uma das características interessantes dos EDAs é que podem utilizar 1, 2, 3 indivíduos ou mesmo um número de soluções bem maior (como a metade de uma população de indivíduos) para estimar e amostrar novas soluções. Os EDAs utilizam um conjunto de indivíduos promissores (os selecionados, que podem ser relativamente grandes) para extrair estatísticas relevantes e assim amostrar novos indivíduos. Com base nessas informações e na Figura 5.47(a), acredita-se que há uma relação entre qualidade da metaheurística e quantidade de indivíduos utilizados para gerar uma nova solução. Além disso, os EDAs propostos mostram-se especialmente adequados para proteínas rela-

tivamente pequenas (até 100 resíduos) e para sequências com baixa similaridade. Com um método com a capacidade de predição de estruturas puramente *ab initio*, pode-se focar em estruturas de proteínas que não possíveis de se determinar utilizando métodos experimentais, ou pelo tamanho, ou pelo fato de que algumas proteínas não poderem ser cristalizadas e possivelmente para proteínas com regiões de baixa similaridade na sequência.

Por fim, as principais vantagens e desvantagens dos métodos de referência e dos EDAs propostos podem ser sumarizadas da seguinte maneira:

- **RW**

- Vantagens
 - * Baixo custo computacional;
 - * Possibilita formação de ligações de hidrogênio e formação de algumas estruturas secundárias;
- Desvantagens
 - * Não otimiza adequadamente a energia de van der Waals;
 - * Altos valores de RMSD;
 - * Não possui herança para compor novas soluções;
 - * Sensível ao uso de ADB;

- **MC**

- Vantagens
 - * Baixo custo computacional;
 - * Possibilita formação de algumas ligações de hidrogênio e formação de algumas estruturas secundárias;
 - * Otimiza a energia de van der Waals melhor do que a RW;
 - * Requer população relativamente pequena;
- Desvantagens
 - * Altos valores de energia de van der Waals em comparação com o GA, DE e EDA;
 - * Altos valores de RMSD;
 - * Sensível ao uso de ADB;

- **GA**

- Vantagens
 - * Baixo custo computacional;
 - * Possibilita formação de ligações de hidrogênio e formação de estruturas secundárias;

- * Otimiza a energia de van der Waals melhor do que o MC;

- Desvantagens

- * Não optimiza a energia de van der Waals tão bem quanto os EDAs;
- * Valores de RMSD mais altos que os obtidos pelo EDA;
- * Sensível ao uso de ADB;

- **DE**

- Vantagens

- * Baixo custo computacional;
- * Convergência rápida para soluções promissoras
- * Otimiza a energia de van der Waals melhor do que o MC;

- Desvantagens

- * Não optimiza a energia de van der Waals tão bem quanto o EDA;
- * Sensível aos parâmetros de reprodução;
- * Valores de RMSD mais altos que os obtidos pelo EDA;
- * Sensível ao uso de ADB;

- **FGMO**

- Vantagens

- * Otimizou a energia de van der Waals melhor do que os algoritmos de referência;
- * Baixo custo computacional;
- * Convergência rápida para soluções promissoras;
- * Consegue obter valores de RMSD relativamente baixos;

- Desvantagens

- * Sensível a quantidade de componentes de mistura;
- * Baixa porcentagem de estruturas secundárias;

- **hFGMO**

- Vantagens

- * Obteve os melhores valores de energia de van der Waals e valores de RMSD;

- Desvantagens

- * Alto custo computacional;
- * Sensível a quantidade de componentes de mistura;
- * Baixa porcentagem de estruturas secundárias;
- * Sensível ao uso de ADB para proteínas com mais de 67 resíduos;

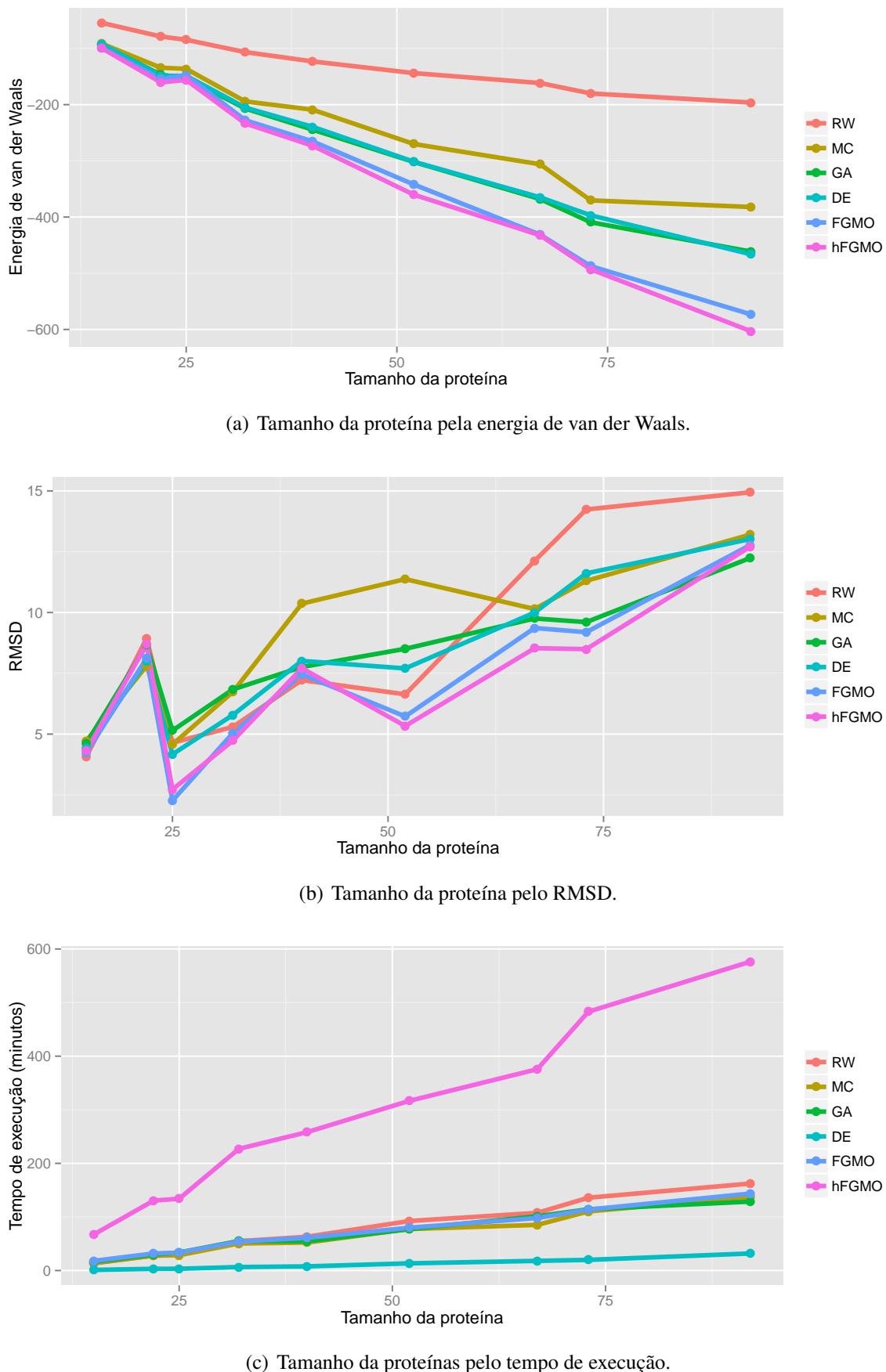


Figura 5.43: Síntese das 10% melhores soluções em relação a energia de van der Waals obtidas pelos métodos de referência e pelos FGMO e hFGMO.

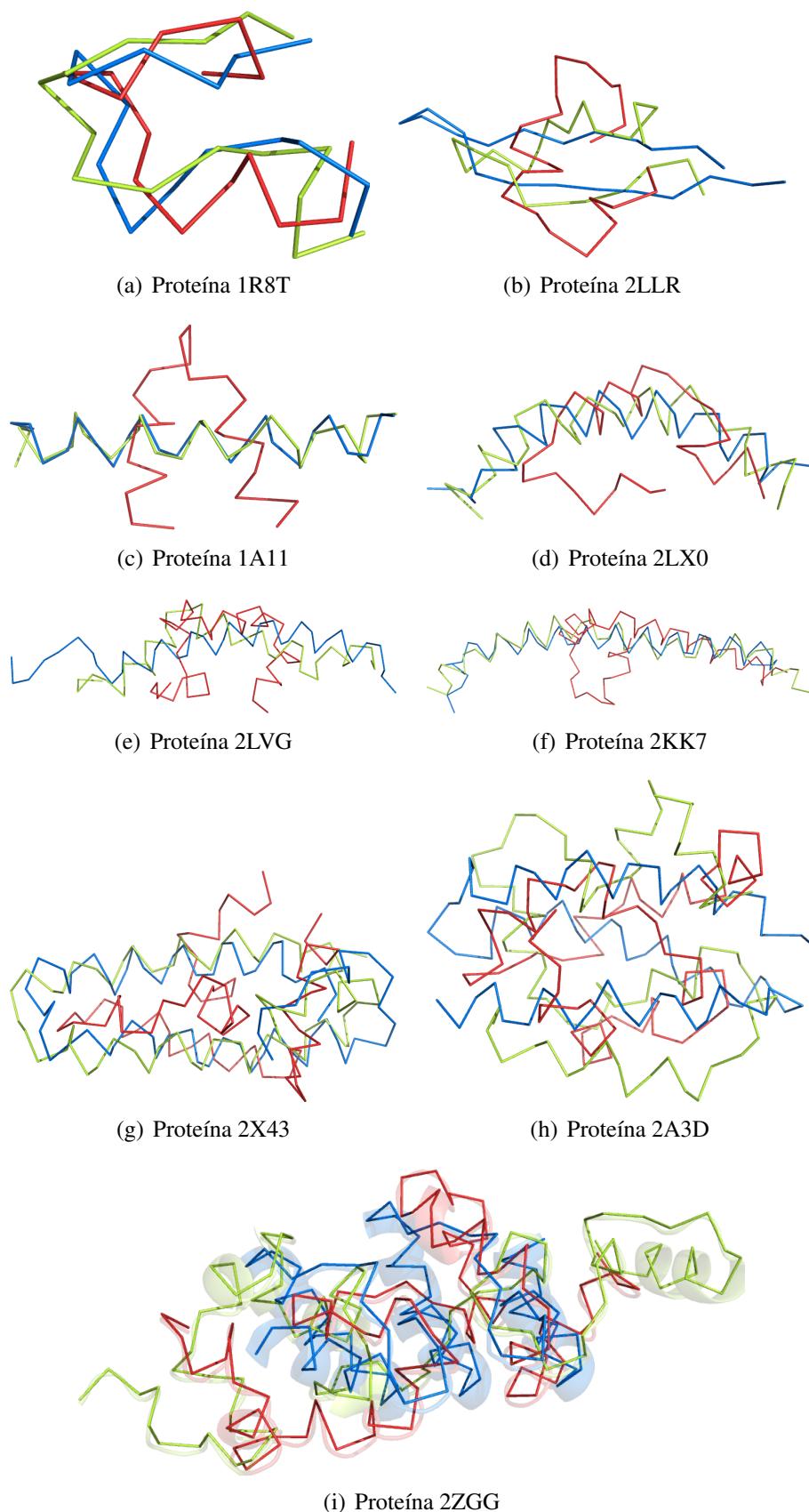


Figura 5.44: Estruturas das melhores proteínas preditas entre os métodos de referência e os EDAs propostos. As proteínas nativas são representadas pela cor azul e as proteínas preditas com menor energia de van der Waals (em vermelho) e menor RMSD (em verde) estão alinhadas a proteína nativa. A Tabela 5.7 mostra qual método corresponde a estrutura com menor energia de van der Waals e RMSD.

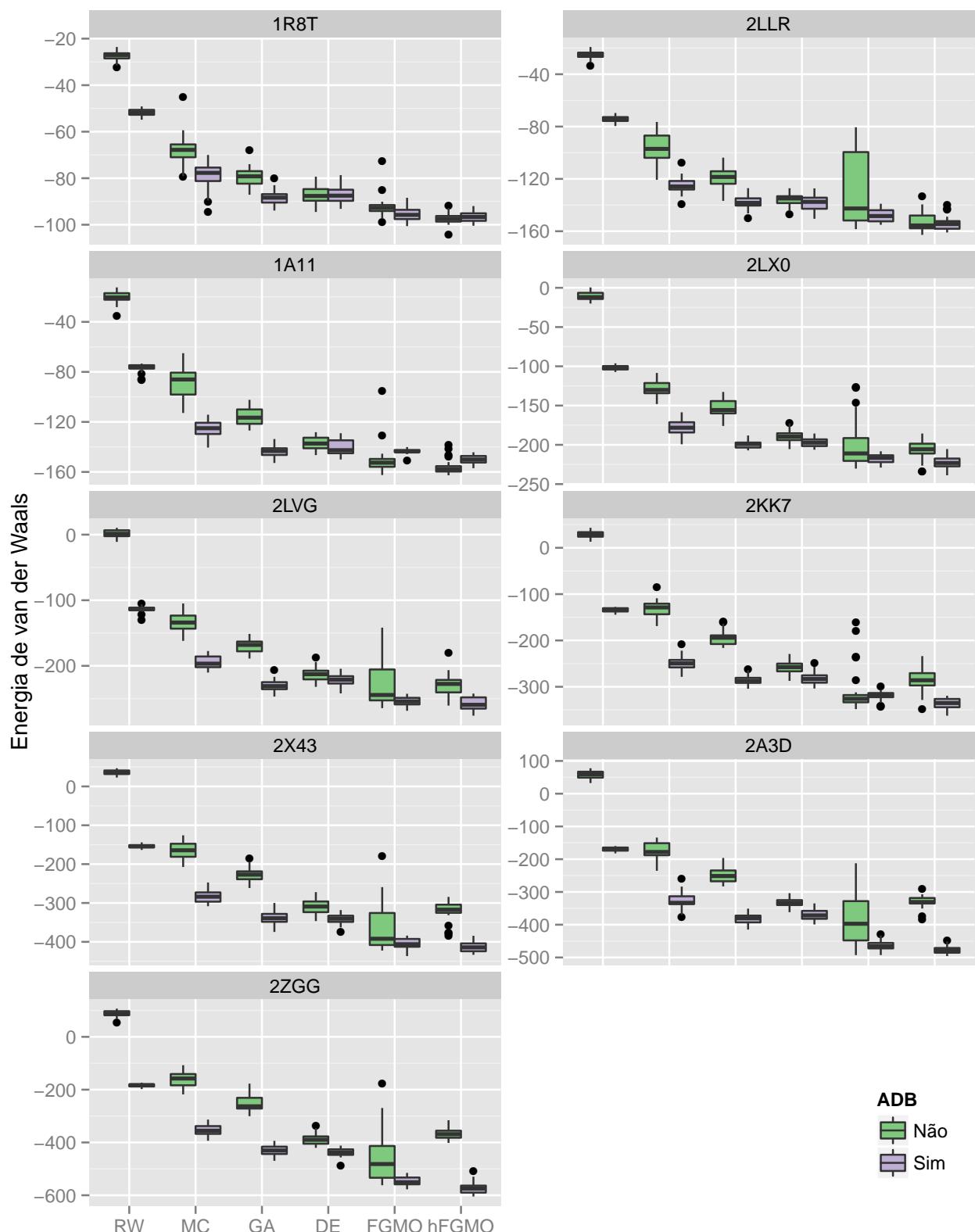


Figura 5.45: Experimento com/sem ADB para gerar a população inicial, mostrando a diferença entre energia de van der Waals.

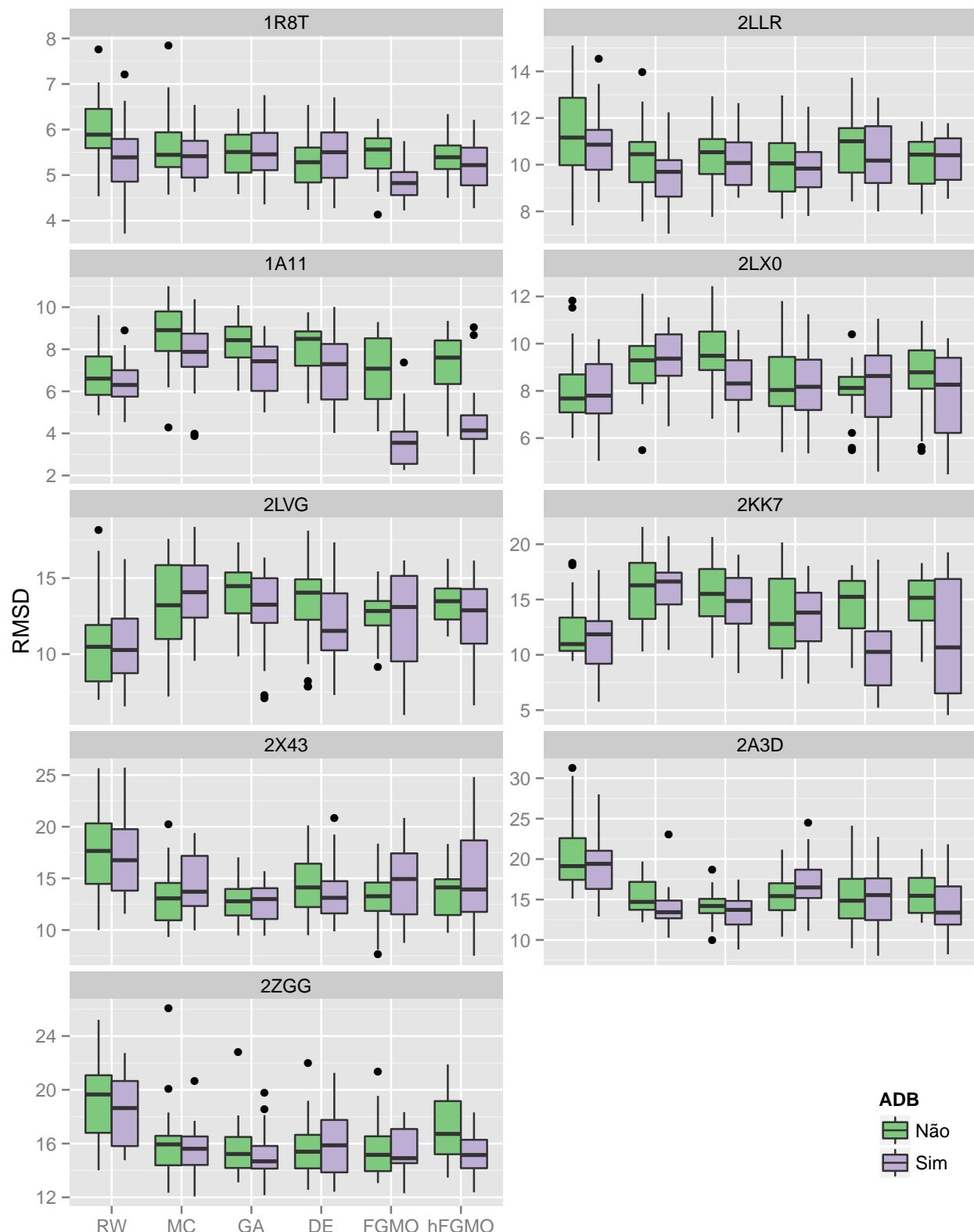
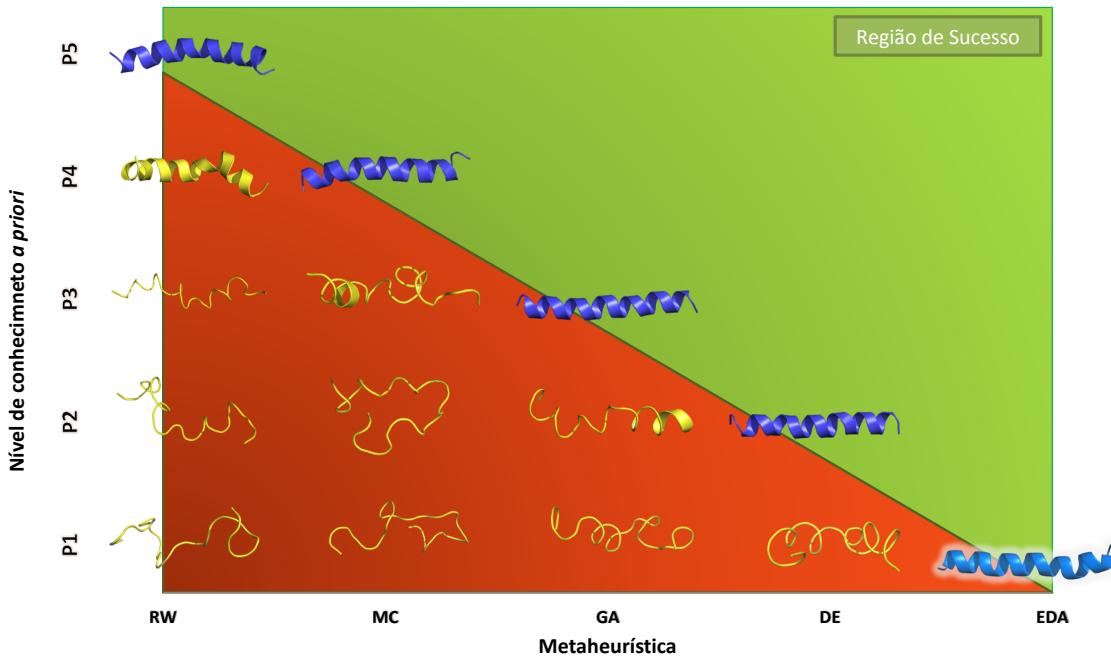
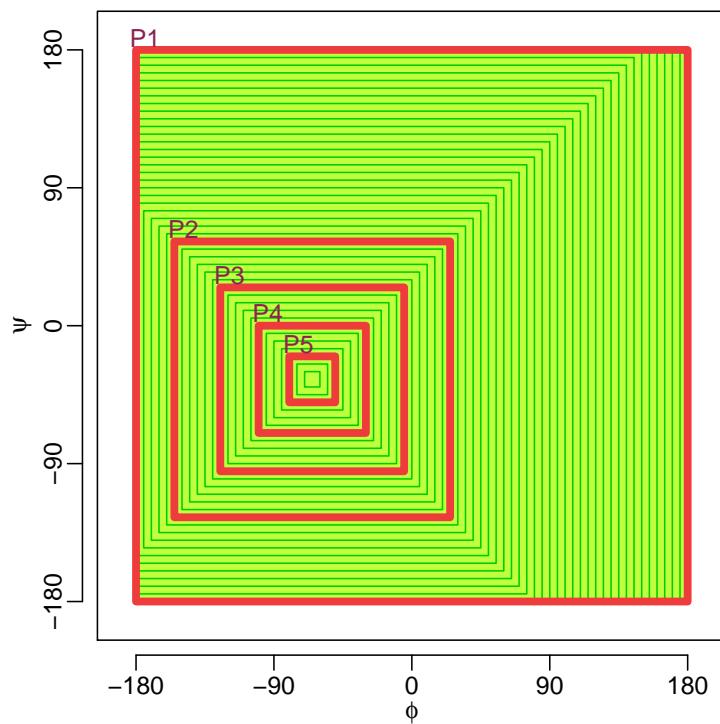


Figura 5.46: Experimento com/sem ADB para gerar a população inicial, mostrando a diferença entre RMSD.



(a) Metaheurística e quantidade de conhecimento *a priori* utilizado para gerar a população inicial. A região em vermelho mostra que os algoritmos podem falhar e a região em verde as regiões que o algoritmo prediz corretamente. As proteínas em amarelo mostram as estruturas preditas incorretas em que as metaheurísticas falharam com o nível de conhecimento *a priori* utilizado, enquanto que as estruturas em azul correspondem aos casos em que as metaheurísticas obtiveram sucesso.



(b) Níveis de conhecimento *a priori* utilizado em todas as metaheurísticas, destacando a quantidade mínima que cada metaheurística precisou para obter sucesso na predição. P1, P2, P3, P4 e P5 correspondem respectivamente ao EDA (KDEO), DE, GA, MC e RW, respectivamente.

Figura 5.47: Nível de conhecimento *a priori* necessário para predizer corretamente a estrutura da proteína 1A11 de acordo com a metaheurística utilizada.

5.4 Síntese dos resultados

Com a realização de todos os experimentos referentes aos EDAs propostos e dos métodos de referência, foi elaborado uma síntese dos melhores e piores algoritmos. Assim, a Tabela 5.8 destaca os algoritmos que obtiveram os melhores e piores desempenhos, considerando os aspectos energia de van der Waals, RMSD e tempo de execução. Para cada proteína é destacado em negrito os melhores/piores valores por proteína.

A Tabela 5.8 apresenta também os resultados que não foram mostrados nas Seção 5.3, como a hUNIO e a hKDEO. Considerando o aspecto energia, a FGMO e a hFGMO foram superiores do que as demais metaheurísticas da literatura analisadas. Para as proteínas pequenas 2LLR e 1A11, a hKDEO foi capaz de minimizar a energia de van der Waals melhor do que a hFGMO (mostrado nos resultados da Seção 5.2.2). Com isso, pode-se concluir que a KDEO é interessante para proteínas relativamente pequenas. É possível também verificar que os métodos de referência foram superiores à UNIO em quase todos os casos. Isso mostra que, mesmo um EDA com um modelo probabilístico capaz de estimar a distribuição dos dados para o problema de PSP, tal EDA pode obter resultados semelhantes a metaheurísticas mais simples. No entanto, o fato de tratar os ângulos diedrais (ϕ, ψ) de um mesmo aminoácido como variáveis relacionadas resultou em uma melhoria no desempenho do EDA, que é superior a todas as outras metaheurísticas. Os valores da pior energia de van der Waals e RMSD pertencem aos métodos de referência, enquanto que os piores tempos ocorrem para as extensões hierárquicas do EDA proposto.

Tabela 5.10: Hipervolume considerando os critérios energia de van der Waals e RMSD, para as nove proteínas.

Linha	Proteína	Resíduos	Método	Hipervolume
1	1R8T	15	RW	135,7
2	1R8T	15	MC	1052,3
3	1R8T	15	GA	1056,5
4	1R8T	15	DE	1041,0
5	1R8T	15	UNIO	1061,3
6	1R8T	15	KDEO	1228,6
7	1R8T	15	FGMO	1223,0
8	1R8T	15	hUNIO	1126,5
9	1R8T	15	hKDEO	1219,6
10	1R8T	15	hFGMO	1217,5
11	2LLR	22	RW	584,1
12	2LLR	22	MC	1885,6
13	2LLR	22	GA	1938,3
14	2LLR	22	DE	2030,7
15	2LLR	22	UNIO	1885,8
16	2LLR	22	KDEO	2163,6
17	2LLR	22	FGMO	2116,8
18	2LLR	22	hUNIO	2129,7

Linha	Proteína	Resíduos	Método	Hipervolume
19	2LLR	22	hKDEO	2253,4
20	2LLR	22	hFGMO	2174,2
21	1A11	25	RW	851,9
22	1A11	25	MC	2148,8
23	1A11	25	GA	2378,9
24	1A11	25	DE	2383,9
25	1A11	25	UNIO	2203,1
26	1A11	25	KDEO	2682,5
27	1A11	25	FGMO	2592,7
28	1A11	25	hUNIO	2507,7
29	1A11	25	hKDEO	2867,3
30	1A11	25	hFGMO	2772,6
31	2LX0	32	RW	1323,6
32	2LX0	32	MC	3223,7
33	2LX0	32	GA	3427,5
34	2LX0	32	DE	3539,0
35	2LX0	32	UNIO	3475,8
36	2LX0	32	KDEO	3961,6
37	2LX0	32	FGMO	4153,8
38	2LX0	32	hUNIO	3736,8
39	2LX0	32	hKDEO	4338,5
40	2LX0	32	hFGMO	4407,9
41	2LVG	40	RW	1664,1
42	2LVG	40	MC	2937,5
43	2LVG	40	GA	4071,4
44	2LVG	40	DE	3919,3
45	2LVG	40	UNIO	3739,9
46	2LVG	40	KDEO	4405,4
47	2LVG	40	FGMO	4713,3
48	2LVG	40	hUNIO	3982,8
49	2LVG	40	hKDEO	4648,8
50	2LVG	40	hFGMO	4770,9
51	2KK7	52	RW	2084,2
52	2KK7	52	MC	3983,2
53	2KK7	52	GA	4957,0
54	2KK7	52	DE	5170,2
55	2KK7	52	UNIO	4446,6
56	2KK7	52	KDEO	6222,1
57	2KK7	52	FGMO	6465,7
58	2KK7	52	hUNIO	4745,2
59	2KK7	52	hKDEO	6810,2
60	2KK7	52	hFGMO	7113,8
61	2X43	67	RW	1857,8
62	2X43	67	MC	4650,4
63	2X43	67	GA	5903,5
64	2X43	67	DE	5914,2

Linha	Proteína	Resíduos	Método	Hipervolume
65	2X43	67	UNIO	5675,5
66	2X43	67	KDEO	6220,0
67	2X43	67	FGMO	7421,6
68	2X43	67	hUNIO	6425,1
69	2X43	67	hKDEO	7204,9
70	2X43	67	hFGMO	7874,4
71	2A3D	73	RW	1959,4
72	2A3D	73	MC	5737,8
73	2A3D	73	GA	6926,1
74	2A3D	73	DE	5845,4
75	2A3D	73	UNIO	6397,5
76	2A3D	73	KDEO	6736,5
77	2A3D	73	FGMO	8794,7
78	2A3D	73	hUNIO	6469,7
79	2A3D	73	hKDEO	8485,8
80	2A3D	73	hFGMO	8815,7
81	2ZGG	92	RW	1920,4
82	2ZGG	92	MC	5488,8
83	2ZGG	92	GA	6617,4
84	2ZGG	92	DE	6813,7
85	2ZGG	92	UNIO	5964,3
86	2ZGG	92	KDEO	7098,0
87	2ZGG	92	FGMO	8249,4
88	2ZGG	92	hUNIO	6664,4
89	2ZGG	92	hKDEO	7661,2
90	2ZGG	92	hFGMO	8663,0

5.5 Considerações finais

Este capítulo mostrou os resultados obtidos de experimentos realizados com as técnicas de otimização propostas neste trabalho. Primeiramente, é descrito o cenário em que os experimentos foram executados bem como as proteínas utilizadas. Embora as técnicas de otimização propostas otimizem apenas a energia (mais especificamente a energia de van der Waals) é avaliado também a qualidade das proteínas preditas (RMSD) e a eficiência computacional, medindo o tempo de execução. Para cada algoritmo de otimização foi realizado uma calibração dos parâmetros para determinar o conjunto mais adequado para os experimentos.

Os primeiros experimentos executados avaliaram os novos EDAs: UNIO, KDEO e FGMO. Os resultados mostram comparações entre a energia de van der Waals, RMSD, tempo de execução, energia de van der Waals por RMSD, destacando a Fronteira de Pareto e a evolução da energia de van der Waals de acordo com o número de avaliações. Todas essas comparações foram realizadas para as nove proteínas utilizadas (Tabela 5.1). Para cada uma dessas proteínas é apresentada uma

discussão sobre os resultados obtidos. Verificou-se que a KDEO é capaz de minimizar melhor a energia de van der Waals para proteínas relativamente pequenas, enquanto que a FGMO foi melhor para outras proteínas. No entanto, o tempo de execução da KDEO foi o mais alto de todos os EDAs propostos, em todos os casos. A UNIO pode ser considerado a mais rápida, mas a FGMO obteve o tempo de execução semelhante para todas as proteínas. Foi também mostrado a diferença entre utilizar ou não um banco de dados de ângulos diedrais (ADB, Seção 2.4) para gerar a população inicial com ângulos diedrais mais prováveis. Por último, são mostradas as configurações das proteínas preditas com menor RMSD e menor energia de van der Waals alinhadas à estrutura da proteína nativa. Os melhores resultados obtidos foram para estruturas de proteínas que contém apenas uma α -hélice.

Em seguida, foi realizado os experimentos comparando somente os EDAs propostos. A hUNIO foi melhor do que a UNIO, a hKDEO melhor do que a KDEO e a hFGMO foi melhor do que a FGMO. O mesmo efeito também ocorreu com o RMSD, mostrando que os EDAs hierárquicos são superiores nesse aspecto. No entanto, o tempo de execução dos EDAs hierárquicos é significativamente maior que dos não hierárquicos, chegando a ser cerca de quatro vezes maior. Isso ocorre porque, considerando dois subproblemas, o EDA hierárquico precisa fazer quatro chamadas ao seu correspondente não hierárquico (ver Seção 4.3). Acredita-se que mesmo que um EDA não hierárquico que utilize o mesmo número de avaliações hierárquico, ainda seria difícil de encontrar os mesmos valores para a energia de van der Waals encontrados pelo hierárquico. Observando os gráficos da evolução da energia de van der Waals da Seção 5.1.2 é possível verificar que a população já convergiu e, dificilmente, o EDA seria capaz de dar “saltos” para outros valores de energia significativamente diferentes.

A partir dos resultados obtidos pelos métodos propostos neste trabalho, foram comparados com os métodos de referência para verificar o desempenho de outras técnicas de otimização em relação aos EDAs propostos neste trabalho. A FGMO e a hFGMO foram escolhidos para serem comparadas com os métodos de referência. Os resultados referentes a energia, RMSD e tempo de execução são apresentados separadamente para as nove proteínas. Para todas as proteínas a FGMO e hFGMO obtiveram os melhores valores de energia de van der Waals, mostrando que os modelos probabilísticos propostos foram adequados para o problema de PSP. Além disso, os melhores RMSD também foram obtidos utilizando as técnicas propostas. A desvantagem é que o tempo computacional dos EDAs hierárquicos foram superiores aos métodos de referência.

Foi realizado um novo experimento para tentar relacionar a quantidade de conhecimento *a priori* utilizada em uma predição com a qualidade da metaheurística. Por meio desse experimento foi mostrado que a qualidade de um metaheurística tem relações com a quantidade de indivíduos da população que são utilizados para se gerar novas soluções.

Foi também realizado uma comparação entre os valores das energias não-covalentes das melhores proteínas preditas pelos experimentos em comparação com os valores das energias referentes às proteínas nativas. Em todos os casos, as proteínas preditas utilizando as técnicas propostas foram capazes de reduzir a energia de van der Waals em níveis inferiores aos presentes nas estruturas

nativas. Assim, as proteínas preditas ficaram desbalanceadas com relação aos outros potenciais de energia, o que pode ter refletido em alguns RMSDs e em estruturas como, por exemplo, a formação de hélices para a proteína 2LLR.

Uma síntese dos resultados também é mostrada neste capítulo, destacando as melhores e piores execuções de cada proteína. Por fim, as considerações finais deste trabalho são apresentadas no Capítulo 6.

Tabela 5.8: Melhores e piores soluções para cada proteína considerando os fatores energia de van der Waals, RMSD e tempo de execução. Os melhores valores de cada fator de cada proteína são destacados em negrito. O mesmo ocorre para os piores valores encontrados, nas colunas dos piores. Proteína 1R8T, 2LLR, 1A11, 2LX0 e 2LVG.

Linha	Proteína	Método	Melhor Energia	Melhor Tempo	Melhor RMSD	Pior Energia	Pior Tempo	Pior RMSD
1	1R8T	RW	-54,85	13,35	3,71	-49,14	14,42	7,21
2	1R8T	MC	-94,51	13,89	4,63	-70,01	16,82	6,54
3	1R8T	GA	-93,90	15,61	4,36	-80,21	17,75	6,75
4	1R8T	DE	-93,17	1,18	4,27	-78,69	1,70	6,71
5	1R8T	UNIO	-95,49	15,42	5,09	-88,22	18,03	6,43
6	1R8T	KDEO	-100,82	29,81	4,23	-89,39	33,42	5,79
7	1R8T	FGMO	-100,60	17,65	4,22	-88,43	20,17	5,75
8	1R8T	hUNIO	-97,00	66,15	4,46	-90,66	77,63	6,32
9	1R8T	hKDEO	-100,70	124,22	4,32	-94,34	140,57	6,29
10	1R8T	hFGMO	-100,44	66,47	4,27	-91,98	72,62	6,21
11	2LLR	RW	-79,61	27,99	8,39	-69,63	29,30	14,53
12	2LLR	MC	-139,12	27,04	7,05	-107,63	36,64	12,25
13	2LLR	GA	-149,72	29,37	8,59	-127,08	37,17	12,64
14	2LLR	DE	-150,48	3,14	7,80	-127,24	4,31	12,48
15	2LLR	UNIO	-147,68	29,32	8,77	-135,19	33,96	12,19
16	2LLR	KDEO	-157,35	52,06	7,98	-143,29	57,37	11,33
17	2LLR	FGMO	-155,07	30,74	8,00	-139,02	38,63	12,88
18	2LLR	hUNIO	-154,38	137,80	7,68	-142,20	158,45	12,48
19	2LLR	hKDEO	-161,04	221,61	7,84	-149,77	251,67	12,74
20	2LLR	hFGMO	-160,94	119,72	8,54	-139,84	153,41	11,78
21	1A11	RW	-86,20	32,11	4,54	-73,37	34,05	8,91
22	1A11	MC	-140,49	27,34	3,87	-114,23	39,89	10,37
23	1A11	GA	-152,84	32,16	5,00	-133,69	42,39	9,10
24	1A11	DE	-150,03	3,24	4,02	-129,03	4,88	10,01
25	1A11	UNIO	-145,62	30,78	5,11	-136,09	39,53	9,33
26	1A11	KDEO	-154,07	53,16	2,28	-144,26	61,67	8,97
27	1A11	FGMO	-150,87	33,16	2,26	-140,09	39,66	7,37
28	1A11	hUNIO	-156,76	149,95	4,57	-148,01	180,29	9,66
29	1A11	hKDEO	-159,22	232,50	1,66	-151,07	289,29	9,68
30	1A11	hFGMO	-157,02	131,14	2,05	-144,27	161,15	9,04
31	2LX0	RW	-107,21	54,53	5,04	-96,25	57,70	10,20
32	2LX0	MC	-199,38	48,23	6,50	-158,61	68,93	11,12
33	2LX0	GA	-207,07	53,61	6,24	-188,16	68,12	10,59
34	2LX0	DE	-206,22	6,32	5,35	-185,64	9,95	11,24
35	2LX0	UNIO	-207,42	50,31	5,88	-191,94	62,63	11,00
36	2LX0	KDEO	-219,89	81,45	4,76	-211,25	98,86	10,49
37	2LX0	FGMO	-228,98	51,58	4,58	-208,28	76,69	11,06
38	2LX0	hUNIO	-219,39	243,41	6,02	-200,26	282,80	10,61
39	2LX0	hKDEO	-237,61	363,02	4,77	-221,53	417,92	9,96
40	2LX0	hFGMO	-238,80	223,76	4,47	-205,40	316,97	10,23
41	2LVG	RW	-129,83	62,84	6,55	-104,69	65,89	16,25
42	2LVG	MC	-210,20	51,12	9,56	-177,36	74,92	18,37
43	2LVG	GA	-246,80	53,84	7,07	-206,67	91,16	16,35
44	2LVG	DE	-241,96	7,66	7,31	-204,47	12,61	17,35
45	2LVG	UNIO	-233,99	54,98	7,54	-215,66	68,67	16,73
46	2LVG	KDEO	-252,99	90,56	6,34	-236,96	111,78	15,97
47	2LVG	FGMO	-268,44	58,29	5,99	-242,59	86,89	16,16
48	2LVG	hUNIO	-249,26	265,37	8,00	-232,16	314,41	16,03
49	2LVG	hKDEO	-269,59	412,33	6,84	-254,23	469,65	15,33
50	2LVG	hFGMO	-275,96	255,94	6,63	-242,30	324,13	16,15

Tabela 5.9: Continuação da Tabela 5.8. Proteínas 2KK7, 2X43, 2A3D e 2ZGG.

Linha	Proteína	Método	Melhor Energia	Melhor Tempo	Melhor RMSD	Pior Energia	Pior Tempo	Pior RMSD
51	2KK7	RW	-144,42	92,16	5,75	-127,64	99,04	17,67
52	2KK7	MC	-278,57	76,73	10,45	-207,37	107,50	20,72
53	2KK7	GA	-304,28	74,97	8,35	-261,88	113,72	19,06
54	2KK7	DE	-303,75	13,24	7,39	-248,30	20,42	18,04
55	2KK7	UNIO	-289,97	77,01	9,35	-267,06	90,87	17,69
56	2KK7	KDEO	-328,92	118,41	5,63	-306,53	146,13	19,34
57	2KK7	FGMO	-343,28	79,85	5,22	-299,64	115,26	18,61
58	2KK7	hUNIO	-313,42	380,57	9,97	-292,13	469,58	18,45
59	2KK7	hKDEO	-352,81	541,10	5,15	-328,06	687,23	18,62
60	2KK7	hFGMO	-362,41	309,91	4,55	-319,34	427,70	19,27
61	2X43	RW	-163,42	106,99	11,57	-144,12	125,86	25,72
62	2X43	MC	-308,30	82,65	9,97	-247,29	130,14	19,39
63	2X43	GA	-374,51	96,44	9,45	-299,93	137,67	15,70
64	2X43	DE	-375,31	17,23	9,87	-318,36	28,62	20,86
65	2X43	UNIO	-351,02	86,62	9,11	-327,51	111,30	17,35
66	2X43	KDEO	-390,23	140,24	9,63	-366,00	185,11	20,87
67	2X43	FGMO	-436,70	96,63	8,76	-384,04	146,75	20,85
68	2X43	hUNIO	-374,18	444,15	8,13	-347,37	514,90	17,18
69	2X43	hKDEO	-419,13	699,62	8,49	-400,30	894,24	17,63
70	2X43	hFGMO	-433,41	359,59	7,51	-384,41	518,58	24,81
71	2A3D	RW	-182,18	135,45	12,90	-159,88	154,24	28,00
72	2A3D	MC	-376,52	105,90	10,29	-260,43	157,66	23,07
73	2A3D	GA	-414,89	111,76	8,81	-350,92	178,03	17,46
74	2A3D	DE	-399,26	19,29	11,13	-335,06	32,24	24,53
75	2A3D	UNIO	-390,47	98,74	9,00	-361,13	113,76	22,50
76	2A3D	KDEO	-444,36	168,16	10,84	-409,60	206,98	25,14
77	2A3D	FGMO	-492,65	112,57	8,05	-430,27	181,31	22,74
78	2A3D	hUNIO	-410,95	509,72	10,04	-378,17	581,12	21,13
79	2A3D	hKDEO	-465,61	790,01	7,37	-438,64	934,59	19,08
80	2A3D	hFGMO	-495,84	474,20	8,23	-448,16	594,23	21,83
81	2ZGG	RW	-197,84	161,97	14,76	-173,41	171,28	22,74
82	2ZGG	MC	-393,57	133,22	12,07	-313,35	199,81	20,63
83	2ZGG	GA	-469,44	120,53	12,16	-393,96	194,40	19,75
84	2ZGG	DE	-488,90	31,66	12,42	-411,99	49,54	21,26
85	2ZGG	UNIO	-450,80	115,16	13,15	-417,51	133,77	19,34
86	2ZGG	KDEO	-520,64	197,01	12,87	-474,80	240,37	20,05
87	2ZGG	FGMO	-577,37	139,46	12,30	-515,38	207,72	18,34
88	2ZGG	hUNIO	-476,44	596,40	12,36	-446,64	674,73	19,57
89	2ZGG	hKDEO	-559,64	965,47	12,85	-518,28	1077,89	20,58
90	2ZGG	hFGMO	-604,71	563,29	12,38	-507,67	748,48	18,32

Considerações Finais

Pesquisadores de diversas áreas tem reunido esforços para tentar resolver e aprimorar os métodos de determinação e predição de estruturas terciárias de proteínas. No entanto, devido à alta complexidade do problema, encontrar estruturas terciárias de proteínas ainda é um dos maiores desafios da Biologia Molecular Celular. O uso de computadores tem ajudado os pesquisadores da área da biologia a encontrarem estruturas de proteínas, extraíndo estatísticas de estruturas já determinadas para inferir a estrutura de proteínas desconhecidas. Essa técnica baseia-se em conhecimento *a priori* de outras estruturas. No entanto, sabe-se que algumas estruturas de proteínas não podem ser determinadas por meio de métodos experimentais. Assim, os métodos de predição de estruturas de proteínas que se baseiam em conhecimento *a priori* podem ser tendenciosos as limitações dos métodos experimentais, pois somente serão capazes de predizer estruturas com informações geradas a partir dos métodos experimentais.

Dessa forma, este trabalho não utiliza conhecimento *a priori* de estruturas previamente determinadas, sendo caracterizado como *ab initio*. Predizer estruturas de proteínas utilizando modelagem *ab initio* é significativamente complicado, pois é necessário lidar com todo o espaço de busca. Além disso, utilizando a representação *full-atom* o problema fica ainda mais complicado, pois o processo de avaliar as conformações de proteínas precisa de mais recursos computacionais e de potenciais de energia que descrevam adequadamente o comportamento das proteínas, como estão presentes na natureza. Sabe-se que as proteínas na natureza são estabilizadas (encontradas no seu estado natural) com a menor energia. Assim, o problema de encontrar estruturas de proteínas pode ser entendido como um problema de otimização, em que se deseja obter conformações de proteína com a menor energia.

O problema de PSP (*Protein Structure Prediction*) com modelagem *ab initio* e representação *full-atom* não é trivial, pois, exige rotinas de manipulação de aminoácidos, conversão de ângulos diedrais para coordenadas Cartesianas, utilização de um conjunto de parâmetros para descrever a carga de cada átomo, rotinas eficientes para o cálculo das energias potenciais e um algoritmo de busca global. Todas essas características faziam parte do programa desenvolvido pelo próprio grupo de pesquisa do LCR ICMC-USP, chamado ProtPred. Nesta tese, foi proposto um Algoritmo de Estimação de Distribuição (EDA) que substituiu o Algoritmo Genético do ProtPred, produzindo o ProtPred-EDA.

Os EDAs têm ganhado atenção entre os pesquisadores da área de Computação Evolutiva devido ao sucesso que têm obtido. Os EDAs utilizam modelos probabilísticos para estimar os valores das variáveis e amostrar novas soluções, a partir de um conjunto de soluções promissoras. Isso tem mostrado ser um aspecto fundamental para a exploração adequada do espaço de busca de problemas combinatórios. Existem vários modelos probabilísticos que podem ser utilizados nos EDAs. O modelo probabilístico mais adequado a ser utilizado está relacionado com certas características do problema. Após a realização de uma análise do comportamento dos dados para o problema de PSP *full-atom* verificou-se que um EDA para este problema deveria ser capaz de tratar os aspectos multi-modalidade e multi-dimensionalidade. Isto é, deveria ser capaz de tratar distribuições multivariadas e não-paramétricas. Considerando esses aspectos e os EDAs presentes na literatura, decidiu-se desenvolver o próprio modelo probabilístico. Na verdade, foram desenvolvidos três modelos probabilísticos, o univariado (UNI), o bivariado que utiliza o método do kernel (KDE2D) e o multivariado que utiliza o modelo de misturas finitas (FGM), desenvolvido durante o estágio no exterior na Durham University, Inglaterra. Isso produziu três novos EDAs para PSP: UNIO (*Univariate model-based Optimization*), KDEO (*Kernel Density Estimation model-based Optimization*) e FGMO (*Finite Gaussian Mixture model-based Optimization*). Para a KDEO e FGMO foi estabelecido que os ângulos diedrais (ϕ, ψ) do mesmo aminoácido são tratados como variáveis correlacionadas. De acordo com os resultados obtidos, a relação (ϕ, ψ) estabelecida permitiu explorar melhor o espaço de busca. Além disso, considerando que as estruturas de proteínas possuem uma hierarquia foi também desenvolvido uma extensão hierárquica dos EDAs propostos. Essa extensão hierárquica do EDA, divide o problema original em subproblemas e tenta tratá-los de forma independente. Cada um dos modelos probabilísticos desenvolvidos podem também ser combinados com a extensão hierárquica proposta produzindo: a hUNIO (*hierarchical Univariate model-based Optimization*), a hKDEO (*hierarchical Kernel Density Estimation model-based Optimization*) e a hFGMO (*hierarchical Finite Gaussian Mixture model-based Optimization*)

Para avaliar o desempenho dos métodos propostos foi realizado experimentos para comparar a energia de van der Waals, o RMSD e o tempo de execução, para um conjunto de nove proteínas extraídas do PDB. Para todos os experimentos realizados, os parâmetros de entrada de cada método foram calibrados de acordo com o valor da melhor energia de van der Waals obtida. Os experimentos mostraram que a KDEO foi a melhor para proteínas com até 25 resíduos para o aspecto energia de van der Waals. Para proteínas maiores que 25 resíduos a FGMO foi capaz de minimizar

melhor a energia de van der Waals. Considerando apenas o RMSD, a KDE2D foi semelhante ou superior à FGMO para proteínas com até 50 resíduos. A UNIO apenas foi melhor que a FGMO e KDEO no aspecto tempo computacional. No entanto, para proteínas com até 50 resíduos o tempo computacional da FGMO foi semelhante à UNIO. Isso mostra que mesmo um EDA para PSP mais sofisticado, como o caso da FGMO, pode ter um custo computacional relativamente semelhante em comparação com a UNIO (mais simples). Considerando um panorama geral para os aspectos energia de van der Waals, RMSD e tempo de execução das nove proteínas avaliadas, pode-se concluir que entre a UNIO, KDEO e FGMO, a FGMO foi a mais adequada.

Foi também utilizado os mesmos aspectos para avaliar a qualidade entre os EDAs não hierárquicos e os hierárquicos. Foi verificado que, na maioria dos casos, os EDAs hierárquicos foram superiores aos não hierárquicos, para os aspectos energia de van der Waals e RMSD. O tempo computacional dos EDAs hierárquicos ainda é significativamente superior aos não hierárquicos, pois utiliza um maior número de avaliações. Pode-se concluir que os EDAs FGMO e sua extensão hierárquica, a hFGMO, foram as mais adequadas para todos os tamanhos de proteínas. Assim, a FGMO e hFGMO foram escolhidas para serem comparadas com os algoritmos de referência.

Para avaliar o desempenho dos métodos propostos (FGMO e hFGMO) foi realizada uma comparação entre algumas metaheurísticas da literatura, chamadas de algoritmos de referência, neste trabalho. Os resultados mostraram que a FGMO e hFGMO foram superiores a todos os algoritmos de referência para todas as nove proteínas avaliadas, para o aspecto energia de van der Waals. Na verdade, o indicador de hipervolume mostrou que a FGMO e hFGMO superou todos os algoritmos de referência considerando os critérios energia de van der Waals e RMSD. Foi verificado também que a FGMO é capaz de convergir rapidamente para regiões promissoras utilizando um número relativamente pequeno de avaliações.

Do ponto de vista da Computação Evolutiva, a FGMO e hFGMO mostraram ser as melhores metaheurísticas para o problema de PSP *full-atom* e *ab initio*. No entanto, com certos ajustes, tais métodos poderiam também ser estendidos para outros problemas difíceis do mundo real, que utilizem variáveis contínuas. Isso porque a FGMO mostrou uma convergência rápida para regiões promissoras do espaço de busca e, além disso, tem o desempenho semelhante ao método mais simples (UNIO).

Em um experimento puramente *ab initio* realizado para a proteína 1A11, que comparou a quantidade de conhecimento *a priori* com a qualidade da metaheurística, mostrou que mesmo uma metaheurística relativamente simples como, por exemplo, a RW, pode encontrar a solução esperada quando uma quantidade adequada de conhecimento *a priori* é fornecida. Foi verificado que o sucesso de uma determinada metaheurística, tem certa relação com a quantidade de soluções candidatas que são utilizadas para amostrar novas soluções. Por exemplo, o método de MC utiliza um indivíduo para gerar uma nova solução e, por isso, precisa de menos conhecimento *a priori* para encontrar a solução esperada do que a RW precisa, pois a RW gera novos indivíduos sem utilizar informação de nenhum indivíduo, ou seja, não há herança de características de soluções candidatas. Esse padrão manteve-se também para o GA, que foi capaz de encontrar a solução esperada

utilizando menos conhecimento *a priori* do que o MC precisou para obter sucesso. Isso porque o GA utiliza informação de dois indivíduos para amostrar a nova solução. O DE, que utiliza informação de três indivíduos para gerar uma nova solução, conseguiu encontrar a solução esperada utilizando menos conhecimento *a priori* do que o GA precisou. Assim, foi verificado a existência de um certo padrão entre (1) a quantidade de informação de indivíduos da população utilizada para gerar novas soluções e (2) a quantidade de conhecimento *a priori*, pois as metaheurísticas RW, MC, GA e DE utilizaram informações de 0, 1, 2 e 3 soluções candidatas, respectivamente, para gerar novas soluções. Sabe-se que os EDA, ao invés de utilizarem informação de 0, 1, 2 ou 3 indivíduos para gerar novas soluções, utilizam um conjunto de soluções, definido pelo tamanho do conjunto dos selecionados. Assim, as estatísticas extraídas do conjunto de soluções promissoras pelos EDAs são, em geral, mais adequadas para amostrar novas soluções. Isso foi mostrado ao obter a estrutura correta da proteína 1A11 executando o EDA sem conhecimento *a priori*, ou seja, utilizando todo o espaço de busca e desconsiderando o diagrama de Ramachandran. Dessa forma, o EDA foi capaz de encontrar a estrutura de proteína correta a partir de uma predição puramente *ab initio*. Isso pode ser especialmente interessante, pois tal predição foi realizada inteiramente livre de tendência, apoiando a ideia de que os EDAs são adequados para o problema de PSP, mesmo quando não é utilizada nenhuma estrutura homóloga na predição.

A maior contribuição deste trabalho está no aspecto da minimização da energia de van der Waals das conformações de proteínas, pois, ao utilizar somente a energia de van der Waals, as estruturas de proteínas são preditas favorecendo apenas a energia de van der Waals e, consequentemente, a formação de hélices. Foi mostrado que os métodos propostos são capazes de minimizar a energia de van der Waals além da energia de van der Waals das estruturas nativas. No entanto, foi mostrado também que adicionando uma segunda energia ao cálculo do *fitness* os valores do RMSD podem diminuir. Isso foi mostrado em um experimento em que foi considerado a ponderação entre as energias de van der Waals e solvatação. Por outro lado, ao utilizar mais de uma energia no cálculo do *fitness*, o problema poderia ser melhor tratado utilizando algoritmos multi-objetivos.

Em um trabalho colaborativo com o pesquisador Horálio Perez-Sanches, em paralelo a tese, foi verificado que, de fato, é possível utilizar GPUs para melhorar o tempo computacional da energia de solvatação. Isso indica que tal ideia poderia ser estendida para outros potenciais de energia, melhorando a eficiência global de um algoritmo de otimização para PSP.

De forma geral, o objetivo principal deste trabalho, isto é, o desenvolvimento de um EDA para PSP *ab initio* e com representação *full-atom*, foi alcançado, produzindo resultados significantes. Assim, foi gerado contribuições para a área da Computação Evolutiva, mostrando que é possível melhorar a otimização do problema de PSP *ab initio* e *full-atom* utilizando EDAs, para a área de Bioinformática, mostrando que estruturas relevantes podem ser obtidas com as técnicas propostas, mesmo quando nenhum conhecimento *a priori* é utilizado e, por fim, para a área de Computação Paralela, mostrando que a função de energia pode deixar de ser o gargalo utilizando técnicas eficientes e GPUs.

Por fim, devido ao menor número de pesquisas sobre EDA para PSP com modelagem puramente *ab initio* espera-se que este trabalho de doutorado seja também visto como um referencial teórico e um incentivo para trabalhos futuros relacionados bem como estimular novas pesquisas na área. A seguir, é apresentado os potenciais trabalhos futuros gerados a partir desta pesquisa.

6.1 Trabalhos futuros

Este trabalho permitiu também que novas pesquisas possam ser elaboradas, podendo produzir contribuições nos valores de energia, RMSD ou mesmo para o tempo de execução. Os trabalhos futuros podem ser divididos em itens de (1) a (13). Para melhorar o aspecto da minimização da energia e também diminuir o RMSD é possível: (1) elaborar um algoritmo eficiente para estimativação do número de componentes de mistura do FGM; (2) utilização de modelos que relacionem os ângulos diedrais (ϕ, ψ) de mais de um aminoácido, precisando de modelos probabilísticos $2d_r$ -variados, onde d_r seria o número de resíduos considerado no modelo probabilístico; (3) substituição da distribuição normal do FGM por uma distribuição circular von Mises; (4) utilizar um algoritmo multi-objetivo para tratar as funções de energia e; (5) melhorar o EDA hierárquico. Para melhorar o aspecto RMSD, poderia: (6) utilizar outros campos de força e (7) realizar melhorias nas funções de energia. Para melhorar o tempo computacional, poderia: (8) implementar outros cálculos de energia como, por exemplo, van der Waals e eletrostática utilizando GPUs; (9) melhorar a distribuição de probabilidade conjunta do KDE2D, (10) elaborar um banco de dados de componentes de mistura para amostrar novas soluções e; (11) realizar previsões com menos avaliações. Por fim, ainda poderia (12) melhorar a conversão entre ângulos diedrais e coordenadas Cartesianas e; (13) ser realizadas melhorias na interface do ProtPred-EDA.

Com relação ao desenvolvimento do Item (1), poderia ser utilizado um dos métodos existentes para encontrar o número ideal de componentes de mistura da literatura (Wang et al., 2004). O problema, é que, em geral, os métodos precisam calcular a máxima verossimilhança com vários componentes de mistura para depois determinar qual o número mais adequado. Isso poderia garantir que o melhor número de componentes de mistura fosse encontrado, porém, com um tempo computacional significativamente alto. É esperado, então, que seja utilizado ou desenvolvido um algoritmo capaz de manter um equilíbrio entre a estimativa da quantidade de componentes de mistura e o custo computacional. Para o Item (2), a modelagem de relacionamentos entre mais ângulos diedrais vizinhos, além de (ϕ, ψ), pode melhorar a exploração do espaço de busca de um algoritmo de otimização global para PSP. Basicamente, isso seria semelhante ao que o Quark (Xu & Zhang, 2012b) faz. O Quark utiliza combinações de 1 – 20 resíduos para predizer proteínas. Um modelo probabilístico capaz de manipular vários resíduos de uma vez, poderá contribuir para explorar melhor o espaço de busca. No entanto, conforme aumenta a dimensão do modelo como, por exemplo, da FGMO, a estimação do número adequado de componentes de mistura torna-se também mais complicada. Portanto, o desenvolvimento do Item (2) seria melhor aproveitado após o desenvolvimento do Item (1). Espera-se também que substituindo a distribuição normal pela von

Mises no FGM (Item 3) seja possível lidar melhor com os valores que são gerados fora do intervalo $[-180, 0; +180, 0]$. Assim, poderia realizar comparações entre utilizar-se o FGM com distribuição normal e von Mises.

Uma das etapas fundamentais para a melhoria do ProtPred-EDA é, certamente, o desenvolvimento do Item (4). Com uma versão multi-objetivo do ProtPred-EDA, pode ser possível adicionar outras energias potenciais relevantes à estabilização da molécula. O próprio grupo de pesquisa já obteve sucesso na versão anterior do ProtPred ao utilizar algoritmos multi-objetivos (Brasil et al., 2013). Assim, espera-se que com a implementação do algoritmo multi-objetivo do ProtPred antigo para o novo ProtPred-EDA, ocorra um ganho de desempenho significativo tanto para os valores de energia quanto para o RMSD.

Espera-se também que com melhorias nos EDAs hierárquicos (Item 5) seja possível obter estruturas com energia e valores de RMSD mais baixos. Por exemplo, pode-se utilizar o conhecimento das estruturas secundárias das proteínas como hélices e folhas e defini-las como subproblemas. De acordo com a quantidade de estruturas secundárias de uma conformação, poderia realizar certas mudanças no valor do *fitness* do indivíduo, isto é, quanto mais segmentos de estruturas secundárias um determinado indivíduo possuir, melhor será o *fitness* de tal indivíduo.

Em todos os experimentos realizados neste trabalho foi utilizado o campo de força *charmm27*. Ao utilizar outros conjuntos de parâmetros de campos de força (Item 6) pode ser possível predizer estruturas diferentes. Assim, comparações entre o uso de diferentes campos de força poderiam ser realizadas para determinar quais campos de força são mais apropriados para o problema de PSP com modelagem *ab initio*. A realização de melhorias nas funções de energia (Item 7) também poderá contribuir para encontrar estruturas mais relevantes, como mostrado por Brasil et al. (2013).

O cálculo das energias em GPU e de forma eficiente (Item 8) pode reduzir significativamente o tempo global de execução do ProtPred-EDA. O cálculo da energia de van der Waals em GPU já está sendo desenvolvido no trabalho de mestrado do Guilherme Oliveira Quintino (Quintino, 2014). O Item (9) poderia ser realizado a partir de, ao invés de utilizar um mapa de densidade bidualimensional para estimação do par de variáveis (ϕ, ψ) , poderia ser utilizado somente a distribuição referente ao valor que está sendo gerado condicional. O Item (10) sugere que fosse utilizado um banco de dados de estimativa de componentes de misturas. Por exemplo, poderia estimar os componentes de mistura entre os ângulos (ϕ, ψ) de duas Alaninas vizinhas de modo *off-line* e depois utilizar tal estimador para gerar novas soluções. Isso poderia ser realizado para várias combinações de aminoácidos como, por exemplo, de 1 – 20, semelhante ao Quark. Isso poderia acelerar o processo de estimativa e amostragem de novas soluções, pois os dados já foram estimados *a priori* da execução do processo evolutivo. A partir dos gráficos de convergência dos valores da energia de van der Waals da FGMO é possível perceber que a FGMO pode ser capaz de convergir rapidamente para soluções promissoras. Assim, utilizando menos avaliações (Item 11) como, por exemplo 100.000 ou 200.000, seria possível obter estruturas de proteínas promissoras utilizando um tempo computacional significativamente menor.

Um dos problemas recorrentes para o problema de PSP *full-atom* e a necessidade de manipulação de uma conformação de proteína ora utilizando ângulos diedrais e ora utilizando as coordenadas Cartesianas. Entretanto, sabe-se que a conversão entre ambos os lados é difícil de se obter, pois há vários fatores envolvidos. Assim, com o desenvolvimento do Item (12), poderia ser realizado um estudo entre os métodos de conversão das coordenadas Cartesianas para ângulos diedrais (e vice-versa) e disponibilizar tal método para os pesquisadores da área de PSP.

Por fim, com o objetivo de permitir que outros pesquisadores sejam capazes de utilizar o ProtPred-EDA como uma ferramenta de predição, seria necessário o desenvolvimento do Item (13). Assim, uma interface precisaria ser elaborada, baseada no ProtPred-EDA *web*, a fim de permitir que os usuários entrassem com a sequência de aminoácido, escolhessem os parâmetros e o resultado fosse exibido na tela.

Experimento preliminar com o rBOA

Foi realizado um experimento preliminar com a proteína 1A11 para tentar determinar se o rBOA (Seção 3.3.3) poderia ser adequado para o problema de PSP. Para isso, utilizou-se o rBOA proposto por (Ahn et al., 2004)¹. Para tornar possível o uso do rBOA com o problema de PSP foi necessário adicionar uma função de *fitness* ao rBOA. Essa função de *fitness* é a mesma utilizada pelo ProtPred-EDA. Além disso, a manipulação de variáveis também foi adequada para que o rBOA fosse capaz de representar os ângulos diedrais corretamente na faixa de valores de ângulos possíveis $[-180, 0; +180, 0]$.

Foi executado um experimento com a proteína 1A11 e o rBOA para verificar o comportamento da estrutura de relacionamento das variáveis, mostrado na Figura A.1. O tamanho da população utilizado neste experimento foi de 5.000 indivíduos e pressão de seleção definida em 0,5. Nesta figura, é mostrado o grafo de relacionamento de variáveis da última geração do rBOA, em que cada variável é representada por um nó do grafo e as arestas indicam os relacionamentos entre as variáveis. Cada resíduo é representado por um nó com cor diferente e identificado por um número entre 1 a 25, referente a cada resíduo da proteína. Os ângulos diedrais ϕ 's são representados por F1-F25, ψ 's por P1-P25 e os χ 's apenas pelos números 1-25. Os resíduos que são vizinhos na sequência de aminoácidos possuem tons de cor semelhantes (Figura A.1(a)), as mesmas cores correspondentes no grafo da Figura A.1(b). É possível notar que cores semelhantes tendem a estar agrupadas no grafo da Figura A.1(b), pelo modelo de relação de variáveis estabelecido pelo algoritmo de aprendizado de relacionamento de variáveis do rBOA.

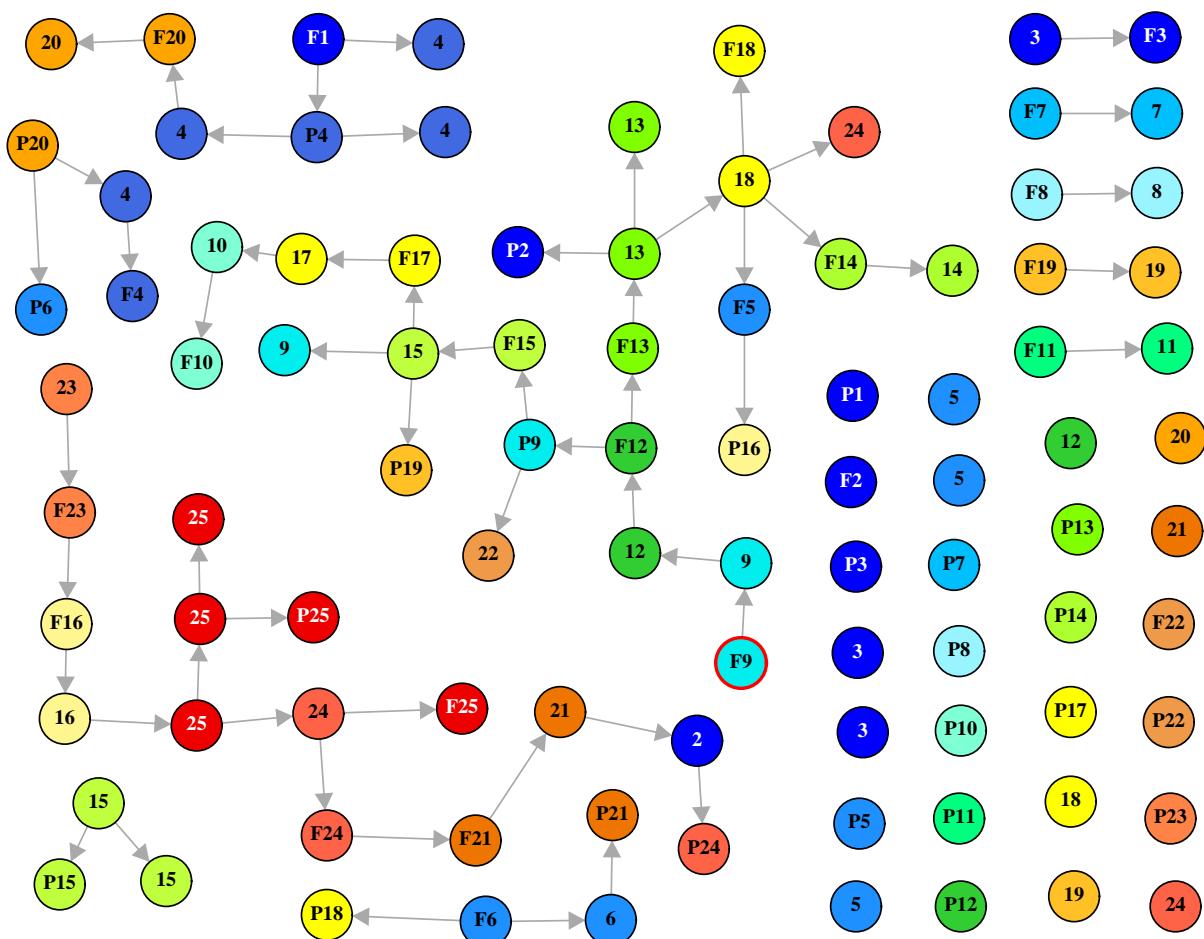
¹Disponível para download em: http://www.evolution.re.kr/bbs/view.php?id=down&page=1&sn1=&divpage=1&sn=off&ss=on&sc=on&select_arrange=headnum&desc=asc&no=3.

Sabe-se que a interação entre um átomo pode ser mais forte na medida que se aproxima de outro átomo, devido as cargas magnéticas dos átomos. Isso pode causar restrições de ângulos diedrais, especialmente para aqueles que estão mais próximos entre si. Por conta disso, já era esperado que os ângulos diedrais do mesmo resíduo tivessem relações, pois, devido à proximidade entre eles é bem provável que ocorresse esse tipo de interação.

Assim, um EDA capaz de estabelecer relacionamentos entre os ângulos diedrais (ϕ, ψ) do mesmo resíduo poderia economizar recursos computacionais, ao invés de tentar aprender o grafo de interação de variáveis a cada geração. Além disso, o grafo da Figura A.1(b) mostra que não foi encontrado relacionamento para várias variáveis, sendo que essas variáveis, de certa forma, possuem relacionamentos.

GSEKMS*T*ASVLLA**QAVF**LLLTSQR

(a) Estrutura primária da proteína 1A11.



(b) Grafo de relacionamento encontrado na última geração pelo rBOA para a proteína 1A11.

Figura A.1: Grafo da interação de variáveis encontrado pelo rBOA na última geração para a proteína 1A11.

Etapas preliminares ao desenvolvimento do ProtPred-EDA

Antes de iniciar o desenvolvimento do EDA algumas etapas preliminares foram realizadas para permitir uma melhor escalabilidade e desempenho geral do algoritmo. A versão original do ProtPred foi projetada para lidar com proteínas pequenas somente, pois a alocação das variáveis de maior parte do algoritmo era realizada estaticamente. Além disso, algumas variáveis não estavam sendo liberadas da memória após serem alocadas, exigindo uma alta quantidade de memória e podendo causar estouro de memória mesmo para proteínas muito pequenas. Assim, a primeira melhoria realizada no ProtPred foi a substituição da alocação de memória estática por alocação dinâmica de todo o algoritmo. Isso tornou possível o uso do ProtPred-EDA para proteínas de qualquer tamanho, pelo aspecto quantidade de memória necessária. A Figura B.1 mostra um exemplo da quantidade de memória necessária para dois tamanhos de população diferentes após a melhoria do gerenciamento de memória. É possível perceber que mesmo para proteínas relativamente grandes (5.000 resíduos) a quantidade de memória necessária ainda é baixa, especialmente no caso em que a população é igual a 500.

Para melhorar o entendimento da estrutura do código-fonte, o ProtPred-EDA foi dividido em bibliotecas específicas para cada atividade que executa. No total, 10 bibliotecas foram criadas, conforme mostra a Figura B.2. As bibliotecas são: (1) *base*, que contém funções matemáticas, estatísticas e manipula matrizes e vetores; (2) *em*, contém o algoritmo *Expectation-Maximization* utilizado no modelo probabilístico de Misturas Gaussianas Finitas (Seção 4.2.3); (3) *exec*, contém os binários para a execução do ProtPred-EDA; (4) *iniparser*, tem as funções utilizadas para a leitura dos parâmetros de entrada (adaptado de Devillard (2012)); (5) *mks*, contém as funções da primeira

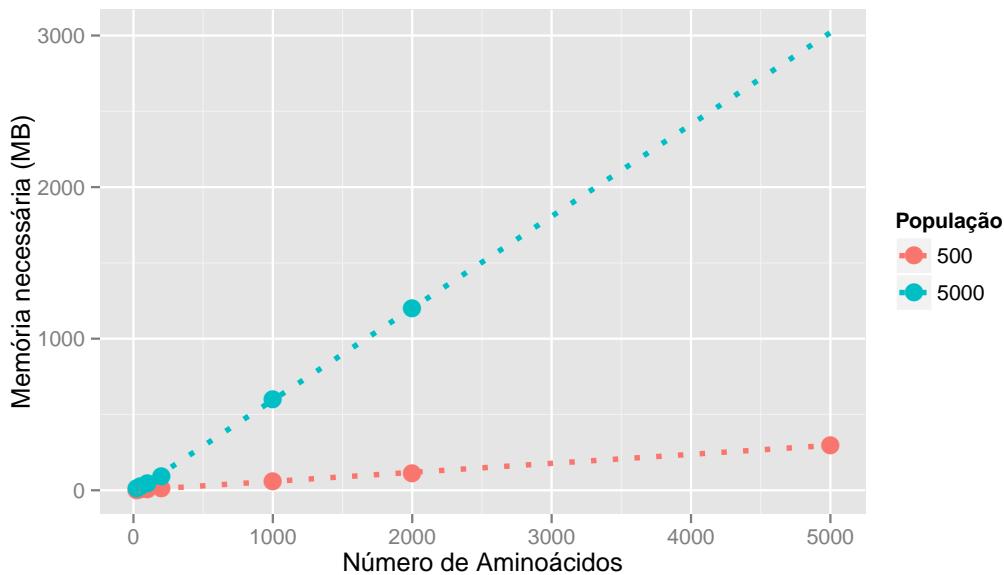


Figura B.1: Quantidade de memória necessária para o ProtPred-EDA. Os pontos representam os valores de memória obtidos pelo gerenciador de memória do Sistema Operacional e as linhas tracejadas representam o ajuste linear.

versão do kernel multivariado implementado (não utilizado nos experimentos); (6) *msasa*, cálculo da energia de solvatação em GPU; (7) *msasacpu*, cálculo da energia de solvatação em CPU; (8) *optimizer* possui todos os algoritmos de otimização, modelos probabilísticos e suas funções como, por exemplo, os algoritmos de otimização de referência e o EDA proposto; (9) *protpred* contém as funções de manipulação de conformações de proteínas bem como as funções de avaliação das conformações; (10) *stats* contém funções estatísticas mais específicas como, por exemplo, o agrupamento hierárquico (Seção 3.4). Várias funções da biblioteca *base* e *stats* foram convertidas da Linguagem R para a Linguagem C, para manter o mesmo desempenho das outras funções do ProtPred-EDA já implementadas em C.

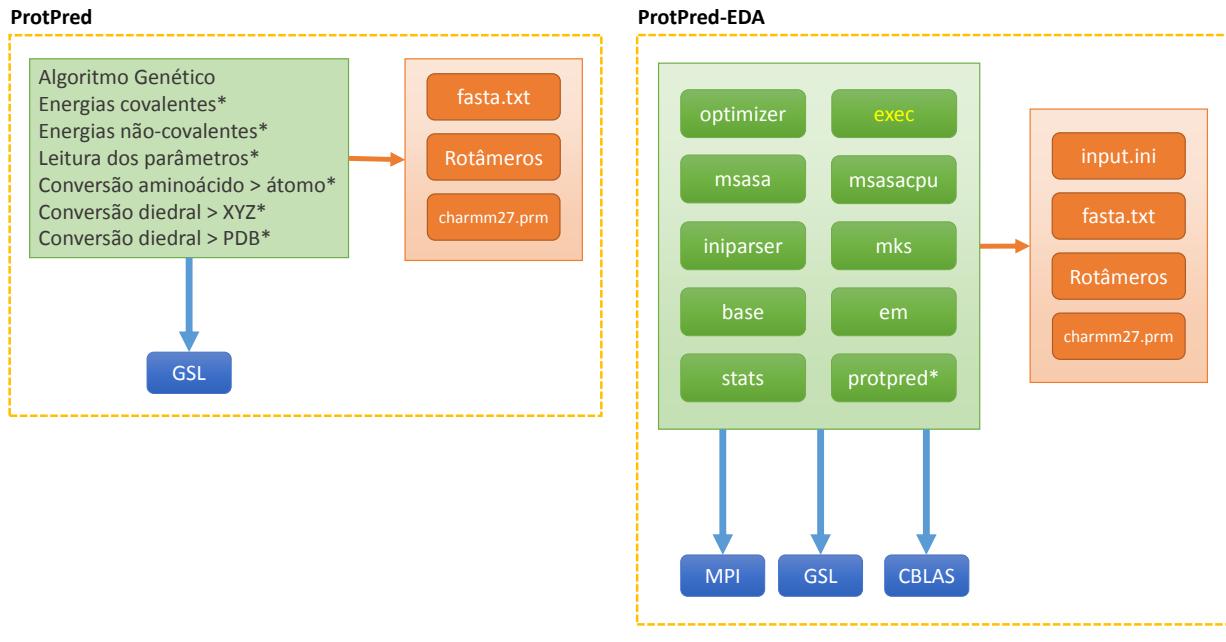
A maneira com que os parâmetros são passados para o ProtPred-EDA também foi alterada. Ao invés de passar todos os argumentos pela linha de comando, foi elaborado um arquivo de inicialização (*.ini*) que contém todas as informações necessárias para uma determinada execução, utilizando a biblioteca *iniparser*. Assim, a execução de experimentos repetidos e de novos experimentos tornou-se mais flexível, pois é possível executar versões diferentes do ProtPred-EDA modificando apenas as propriedades desejadas. Além disso, é possível combinar múltiplas variações de parâmetros para que um único arquivo de entrada produza múltiplas execuções do ProtPred-EDA.

Tendo em vista que múltiplas instâncias do ProtPred-EDA podem ser executadas, especialmente para realizar experimentos com proteínas de diferentes tamanhos ou sementes (*seeds*), foi também adicionado uma opção para executar o ProtPred-EDA em modo paralelo. Para isso, foi implementado rotinas que utilizam MPI (*Message Passing Interface*) para distribuir cada configuração nos respectivos processadores, mapeada utilizando escalonamento *round-robin*.

APÊNDICE B. ETAPAS PRELIMINARES AO DESENVOLVIMENTO DO PROTPRED-EDA

Foram também realizadas melhorias no código-fonte do ProtPred-EDA. Alguns arquivos que continham várias centenas de linhas foram reduzidos para algumas dezenas de linhas, por meio de novas estruturas de dados e definição de variáveis categóricas. O código-fonte também foi dividido em novos arquivos, separando cada um deles pela função principal que executa. Isso contribuiu para um melhor entendimento do sistema como um todo e também para aumentar a eficiência de correção ou adição de novos recursos ao ProtPred-EDA. O uso de bibliotecas de alto desempenho como, por exemplo, GSL e CBLAS (M. Galassi et al, 2009) também contribuiu para simplificar e aumentar o desempenho do ProtPred-EDA.

A Figura B.2 mostra um esquema da versão anterior do ProtPred e da nova versão, o ProtPred-EDA. Na Figura B.2(a) todas as funções pertencem ao mesmo binário, ou seja, há uma forte relação entre as partes do algoritmo. A leitura da sequência de aminoácidos é descrita no arquivo *fasta.txt*, a biblioteca de ângulos diedrais está representada pela caixa *Rotâmeros* e o arquivo que contém o conjunto de parâmetros utilizados no cálculo do *fitness* está representado por *charmm27.prm*. É destacado com um asterisco as funções do ProtPred que foram transformadas na biblioteca *protpred*, utilizada no ProtPred-EDA. A Figura B.2(b) mostra que no ProtPred-EDA as bibliotecas são específicas para cada tarefa que desempenham. Nesta figura é destacado a biblioteca *exec*, que contém o binário e é responsável por fazer a ligação de todas as outras bibliotecas.



(a) Versão anterior do ProtPred.

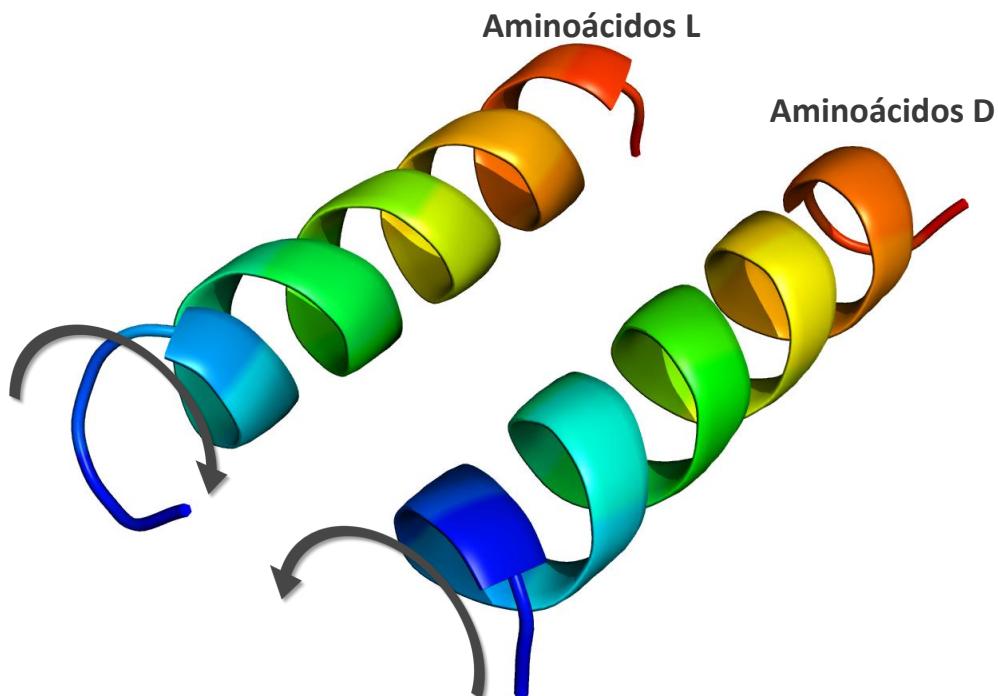
(b) Versão desenvolvida neste trabalho.

Figura B.2: Versões do ProtPred. O código-fonte é representado pela cor verde, as bibliotecas de terceiros em azul e em laranja estão representados os parâmetros que podem ser personalizados.

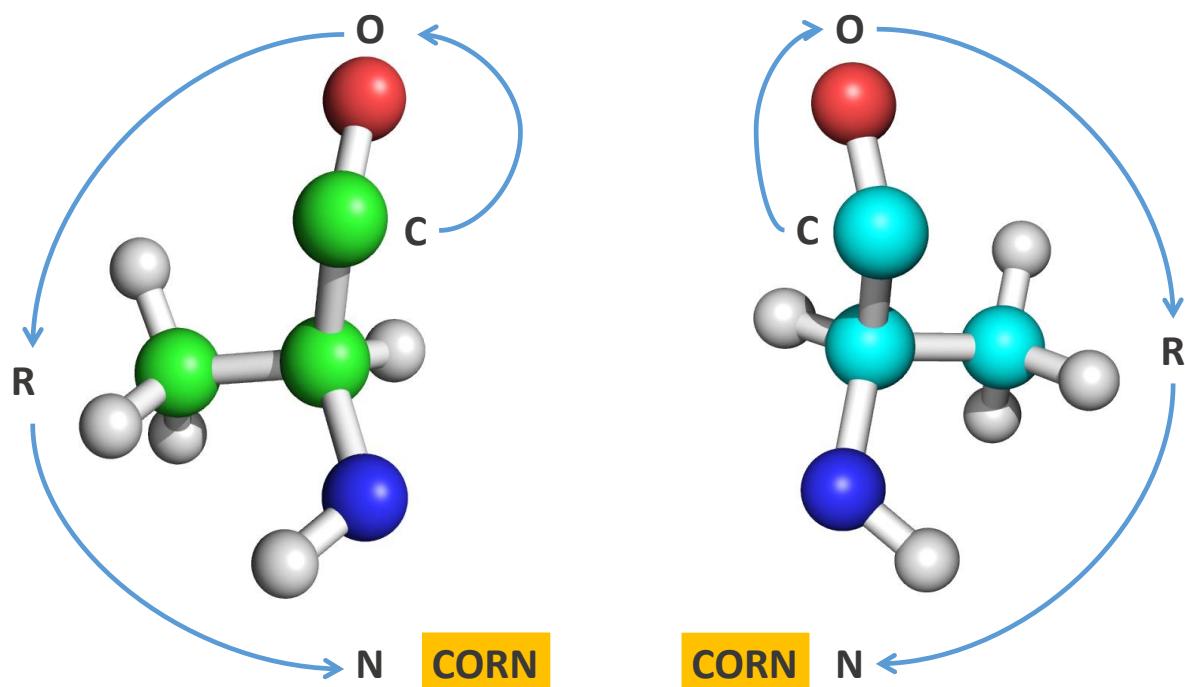
O diagrama de Ramachandran (Figura 2.8) descreve os valores prováveis que os ângulos diedrais (ϕ, ψ) podem assumir. No entanto, para avaliar a qualidade do EDA proposto, foi adicionado um parâmetro ao ProtPred-EDA que permite adicionar ou remover o diagrama de Ramachandran

para gerar a população inicial. Caso utilize, os ângulos diedrais (ϕ, ψ) de cada aminoácido assumem valores baseados em uma biblioteca de valores de ângulos diedrais e, caso contrário, são gerados com distribuição uniforme no intervalo $[-180; +180]$. Dessa forma, pode-se chamar as previsões que não fazem o uso do diagrama de Ramachandran de puramente *ab initio*.

Nos primeiros experimentos realizados com o ProtPred-EDA sem o uso do Ramachandran, verificou-se que as configurações das proteínas preditas tendiam a gerar hélices de mão esquerda (*left-handed*), mesmo para proteínas que já se conheciam que eram de mão direita (*right-handed*). Após uma inspeção nos arquivos de saída do ProtPred-EDA e, em conjunto com as rotinas utilizadas para converter a sequência de aminoácidos em coordenadas Cartesianas, descobriu-se, por padrão, que as rotinas estavam configuradas para utilizar aminoácidos de mão esquerda, chamados aminoácidos D. Tal efeito não era percebido se a população inicial fosse gerada utilizando o diagrama de Ramachandran. No entanto, sabe-se que a maioria das proteínas na natureza possuem aminoácidos com orientação de mão direita, chamados aminoácidos L. Dessa forma, foi adicionado um parâmetro ao ProtPred-EDA que define a orientação dos aminoácidos, sendo do tipo L ou D. Por padrão, todas as previsões apresentadas neste trabalho possuem aminoácidos L, com exceção à Figura B.3, que mostra uma comparação entre as duas orientações de aminoácidos L e D, e o resultado final da previsão de uma proteína hipotética, uma sequência de 20 Alaninas. Nesta figura, é possível notar que utilizando aminoácidos L as hélices são formadas no sentido horário, enquanto que os para aminoácidos D as hélices são orientadas no sentido anti-horário.



(a) Proteína predita com o ProtPred-EDA utilizando aminoácidos L (hélice de mão direita) e D (hélice de mão esquerda).



(b) Aminoácido L utilizado na predição da estrutura mostrada em (a) produzindo a hélice de mão direita.

(c) Aminoácido D utilizado na predição da estrutura mostrada em (a) produzindo a hélice de mão esquerda.

Figura B.3: Orientações L e D dos aminoácidos. Pode-se identificar a orientação dos aminoácidos verificando se a palavra “CORN” se forma no sentido horário, isto é, aminoácido L, ou se forma no sentido anti-horário, aminoácido D. Para “CORN” lê-se Carbono-Oxigênio-Radical-Nitrogênio.

Energia de solvatação

Este capítulo mostra os resultados obtidos utilizando energia de van der Waals e energia de solvatação para compor a função de *fitness* do ProtPred-EDA. Assim, a função de *fitness* é formada pela soma ponderada da energia de van der Waals, com peso fixo, e pela energia de solvatação, com peso variável. Este capítulo mostra resultados obtidos por meio de dois experimentos. O primeiro, foi realizado para tentar determinar os pesos da energia de solvatação para que fosse verificar a influência da energia de solvatação no RMSD para as proteínas utilizadas nos experimentos do (Capítulo 5, Tabela 5.1), mostrado na Seção C.1. O segundo experimento, realizado em colaboração com Horálio Pérez-Sánchez, Espanha, focou no aspecto do tempo de execução. Os resultados com relação a melhoria do tempo de execução da energia de solvatação são apresentados na Seção C.2.

C.1 Influência da energia de solvatação no RMSD

Esta seção descreve o experimento realizado para determinar a influência da energia de solvatação no ProtPred-EDA. O ProtPred-EDA possui um conjunto de energias que podem ser utilizadas para compor o *fitness*. Em aproximadamente todos os resultados mostrados no Capítulo 5 apenas a energia de van der Waals foi considerada. Dessa forma, para tentar mostrar a influência da energia de solvatação na qualidade das estruturas preditas, isto é, no RMSD, um novo experimento foi realizado adicionando a energia de solvatação no cálculo do *fitness*.

Em primeiro lugar, foi realizado um experimento para determinar o peso ideal da energia de solvatação. Considere o peso da energia de van der Waals e solvatação como w_1 e w_2 , respectivamente. A função de *fitness* $f(x)$ pode ser escrita na forma $f(x) = w_1E_{vdw} + w_2E_{sasa}$, em que

E_{vdw} é a energia de van der Waals e E_{sasa} é a energia de solvatação. Nesse sentido, o valor de w_1 foi fixado em 1,0 e o valor de w_2 foi variado em 0,001; 0,005; 0,01; 0,05; 0,1; 0,5 e 1,0, em que 1,0 representa maior influência e 0,001 menor influência da energia de solvatação no *fitness*.

As Figuras C.1, C.2 e C.3 mostram uma comparação do RMSD utilizando diferentes pesos para a energia de solvatação, com os três EDAs propostos (UNIO, KDEO e FGMO) e para nove proteínas. Para cada proteína e para cada EDA proposto foi encontrado um peso responsável por ter produzido os valores de RMSD mais baixos. Os valores dos pesos que obtiveram os valores de RMSD considerados mais baixos são mostrados na Tabela C.1. Essa tabela sugere não haver um padrão entre o peso adequado para cada caso. Com exceção para as duas maiores proteínas (2A3D e 2ZGG). Nesse caso, o maior peso para a energia de solvatação pode privilegiar conformações de proteínas com menor área, evitando que cadeias com vários resíduos, de forma semelhante a estrutura desnaturalizada, sejam formadas.

Tabela C.1: Pesos escolhidos por proteína.

	UNIO	KDEO	FGMO
1R8T	0,010	0,010	0,010
2LLR	0,010	0,005	0,050
1A11	0,010	0,005	0,001
2LX0	0,001	0,005	0,100
2LVG	0,001	0,001	0,001
2KK7	0,010	0,005	0,005
2X43	0,010	0,010	0,001
2A3D	0,500	0,500	0,001
2ZGG	1,000	1,000	0,500

Após determinar o melhor peso para cada EDA proposto e proteína, foi realizado um novo experimento com tais pesos. A Figura C.4 mostra o resultado de 30 execuções com os pesos calibrados em relação a execuções sem energia de solvatação ($w_2 = 0,0$). Na maioria dos casos, o uso da energia de solvatação com o peso adequado foi capaz de reduzir o RMSD. Embora o teste estatístico (mostrado na Tabela E.38) mostre que na maioria dos casos não houve diferença significativa entre utilizar ou não a energia de solvatação, ao observar a Figura C.4 é possível notar que, para todos os casos o efeito da energia de solvatação contribuiu para melhorar a mediana do RMSD ou mesmo encontrando o valor de RMSD mais baixo. Por exemplo, observando o FGM para a proteína 1R8T é possível ver que a mediana dos valores do RMSD utilizando energia de solvatação foi menor que somente van der Waals e, além disso, também encontrou o RMSD mais baixo. A proteína 1A11 não mostrou muita diferença em utilizar ou não a energia de solvatação. Acredita-se que a energia de van der Waals pode ter uma influência maior para tal proteína. Para a proteína 2LVG, mesmo utilizando um peso baixo para a energia de solvatação relativamente baixo (0,001), as execuções da KDEO e FGMO que utilizaram a energia de solvatação foram capazes de reduzir significativamente a mediana do RMSD. A KDEO utilizando a energia de solvatação obteve o melhor valor de RMSD para a proteína 2X43 (cerca de 7,5 Å) enquanto que os melhores

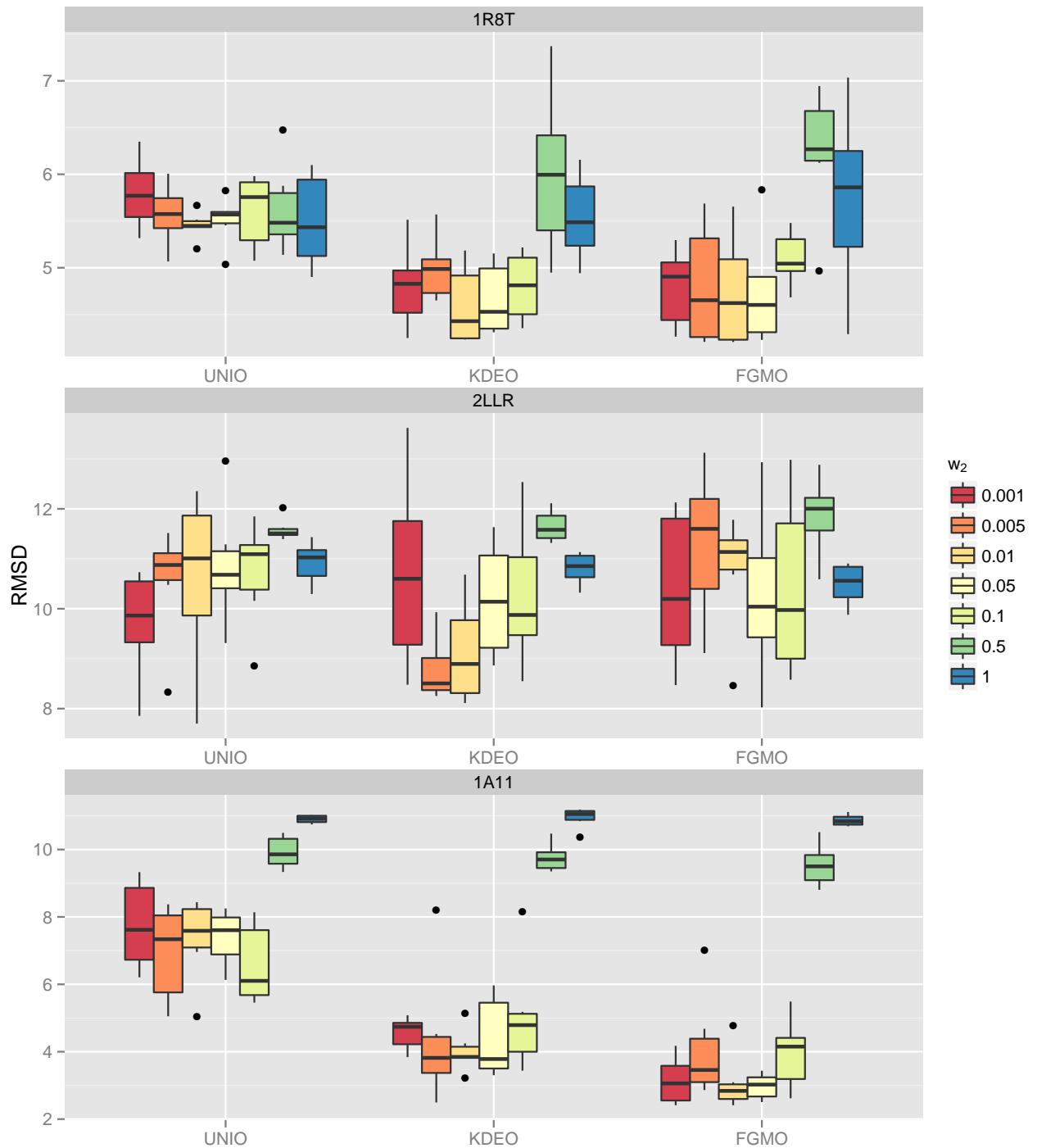


Figura C.1: Energia de solvatação com diferentes pesos, mostrando a distribuição do RMSD de três proteínas

RMSD encontrados pelas outras execuções estão em cerca de 10,0 Å. Para as duas maiores proteínas, os melhores valores médios de RMSD apareceram com pesos altos. A execução com energia de solvatação para a proteína 2A3D com a FGMO manteve a média próxima a execução utilizando somente energia de van der Waals, pois o peso utilizado nesse caso foi de 0,001. Nos casos em

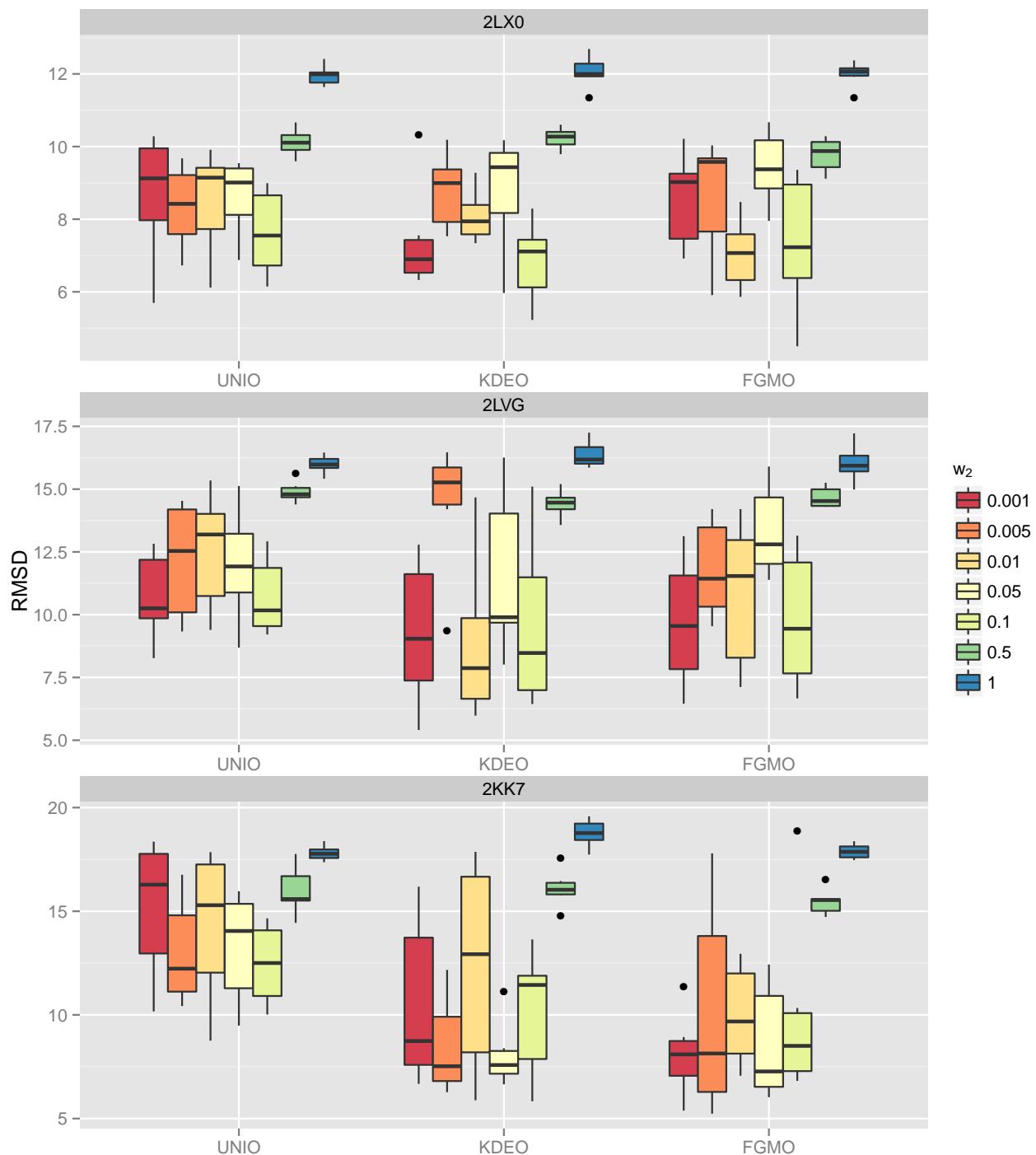


Figura C.2: Energia de solvatação com diferentes pesos, mostrando a distribuição do RMSD de três proteínas

que o peso da energia de solvatação foi de 0, 5 ou 1, 0 houve uma redução significativa no valor médio do RMSD.

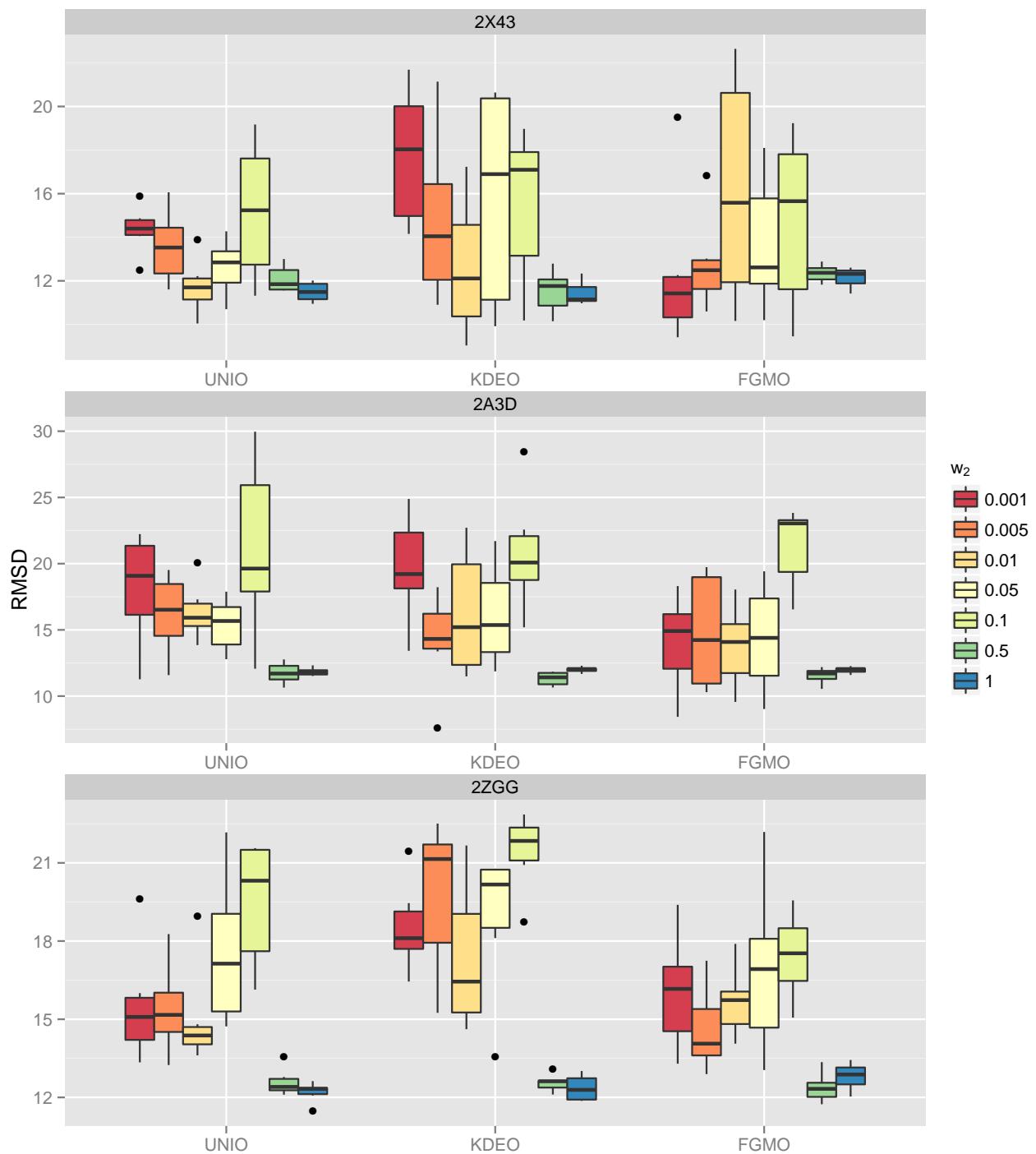


Figura C.3: Energia de solvatação com diferentes pesos, mostrando a distribuição do RMSD de três proteínas

C.2 Energia de solvatação em GPU

Nesta seção é mostrado um experimento realizado para analisar o tempo de execução do cálculo da energia de solvatação. Com exceção à proteína 1A11, as proteínas utilizadas nesta seção também são diferentes das apresentadas nas seções anteriores (Tabela 5.1). As proteínas escolhidas

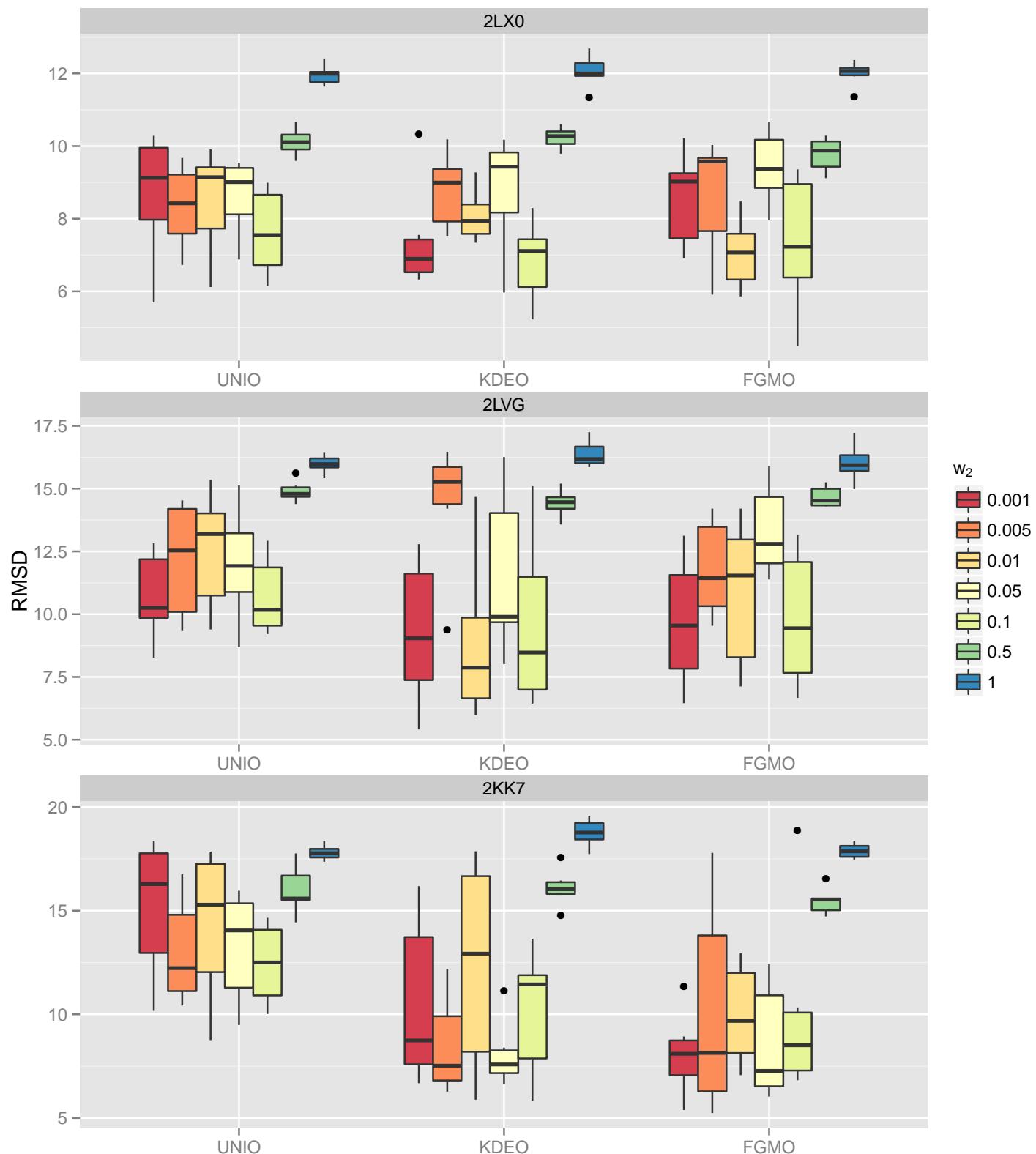


Figura C.4: Energia de solvatação com diferentes pesos, mostrando a distribuição do RMSD de três proteínas

para avaliar o cálculo da energia de solvatação em GPU foram selecionadas com base no tamanho, para determinar a escalabilidade da técnica. A Tabela C.2 mostra as proteínas utilizadas neste experimento bem como o tamanho de cada proteína.

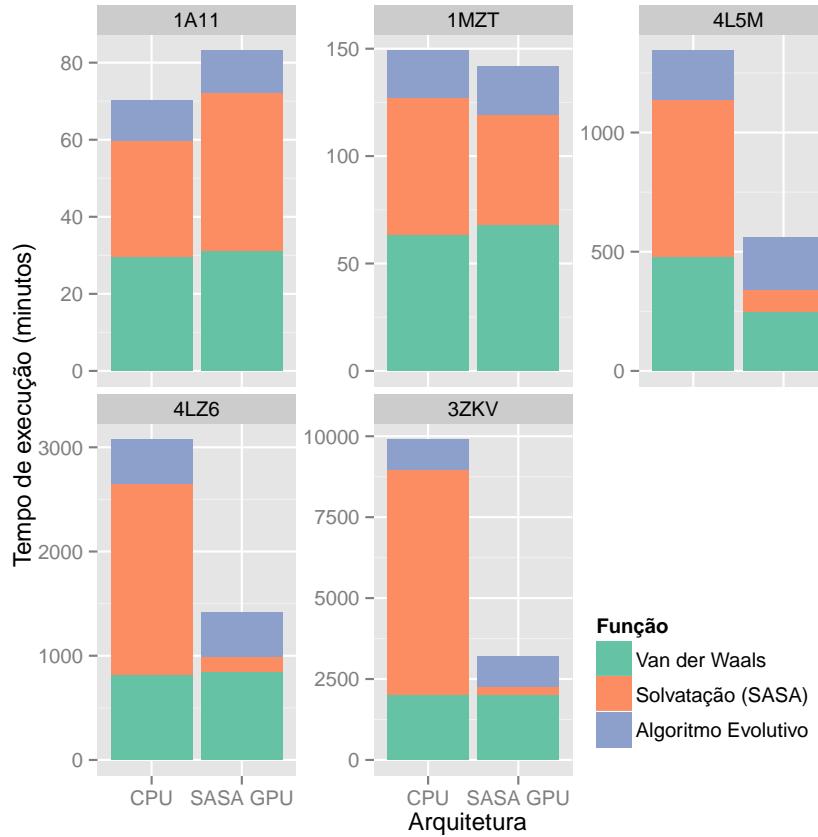
Tabela C.2: Proteínas utilizadas nos experimentos mostrando a quantidade de resíduos, quantidade de variáveis do problema e a quantidade de átomos de cada proteína.

Proteína	Resíduos	Variáveis	Átomos
1A11	25	95	390
1MZT	50	185	745
4L5M	217	880	3.471
4LZ6	446	1681	7.026
3ZKV	971	3911	15.642

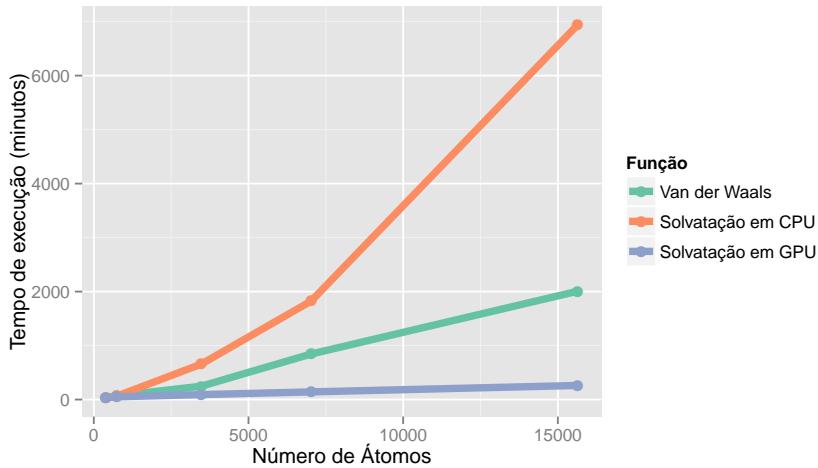
Para cada uma das cinco proteínas foi executado uma instância da FGMO utilizando CPU, isto é, todas as tarefas do Algoritmo Evolutivo (EA), o cálculo da energia de van der Waals e solvatação em CPU, e outra instância, sendo todas as tarefas do EA e o cálculo da energia de van der Waals em CPU e a energia de solvatação em GPU. A Figura C.5 mostra que o tempo de execução da energia de van der Waals e o tempo de execução do EA permaneceram praticamente constantes para todas as proteínas. Ao observar os tempos de computação da energia de solvatação (SASA) em CPU e GPU para a menor proteína 1A11 é possível perceber que o tempo de computação da SASA foi maior em GPU. No entanto, a partir da proteína 1MZT, com 50 resíduos, o cálculo da energia de solvatação começa a ser vantajoso. Quanto maior a proteína, maior é ganho do tempo de execução do SASA em GPU. Observando a proteína 3ZKV da Figura C.5(a) é possível ver que a energia de van der Waals e o EA foram o gargalo no tempo de execução total. Isso está de acordo com a lei de Amdahl (Grama et al., 2003), que diz que o *speedup* máximo que um algoritmo pode atingir está relacionado a porção paralela de determinado algoritmo. Dessa forma, a partir de proteínas com o tamanho semelhante a proteína 3ZKV, não será possível melhorar a eficiência do ProtPred-EDA aumentando a quantidade de processadores utilizados no cálculo da energia de solvatação.

Sabe-se que uma versão sequencial do cálculo da energia de solvatação é mais lenta do que o cálculo da energia de van der Waals. No entanto, a Figura C.5(b) mostra como o tempo de execução da energia de solvatação pode ser reduzido e ainda ser mais rápido do que a energia de van der Waals. O ganho torna-se mais significativo de acordo com o tamanho da proteína. Espera-se que, em uma instância futura, possa ser desenvolvido uma versão da energia de van der Waals em GPU para que deixe de ser o gargalo do ProtPred-EDA, permitindo que a execução do ProtPred-EDA para proteínas com mais de 100 resíduos seja mais eficiente.

A Figura C.6 faz a comparação do RMSD de três proteínas diferentes. Para cada proteína foi atribuído um peso referente à energia de solvatação. Nesta figura é possível perceber a dificuldade que existe em tentar encontrar um peso adequado para o cálculo da energia de solvatação. Para a proteína 1A11 o peso mais adequado, isto é, que obteve a melhor média de RMSD foi com o valor



(a) Tempo total necessário pela energia de van der Waals, energia de solvatação em CPU e GPU e o tempo restante do algoritmo como, por exemplo, as rotinas do EA.



(b) Tempo de execução da energia de solvatação das versões em CPU e em GPU, em comparação com o tempo de execução da energia de van der Waals.

Figura C.5: Comparaçao do tempo de execucao da energia de solvatacao em CPU e em GPU.

0, 5. A Figura C.7 mostra a configuração da estrutura da proteína 1A11 predita utilizando o peso ideal (a), peso muito alto (1, 0) (b) e peso muito baixo (c). Para a proteína 1MZT o melhor peso foi 0, 05, no entanto, o RMSD mais baixo foi obtido com o peso sendo 0, 5. A Figura C.8 mostra a estrutura da proteína predita utilizando peso 0, 5 (a), peso muito alto (1, 0) (b) e peso muito baixo

(0,001) (c). É interessante observar que a Figura C.8(c), isto é, com baixo efeito da energia de solvatação, foi apenas capaz de formar hélices por conta do efeito da energia de van der Waals. Em seguida, para a maior das três proteínas, a 4L5M teve um comportamento diferente das outras duas proteínas. O melhor peso para a proteína 4L5M foi 1,0, ou seja, o que era considerado muito alto para as outras duas proteínas é o ideal para a proteína 4L5M. A Figura C.9 mostra as estruturas preditas utilizando o peso ideal (a) e os pesos mais baixos. É interessante observar na Figura C.9 que não foi possível realizar uma compactação adequada para a proteína utilizando um peso para a energia de solvatação menor do que 1,0.

Por conta disso, estimar o peso adequado para cada uma das energias é um problema complicado em si para um algoritmo mono-objetivo. No entanto, isso pode ser melhor tratado com algoritmos multi-objetivos. Assim, é possível explorar melhor o espaço de busca de outras energias além das energias de van der Waals e de solvatação como, por exemplo, energia eletrostática e de pontes de hidrogênio. Espera-se que a versão multi-objetivo capaz de lidar com isso seja desenvolvida em uma instância futura.

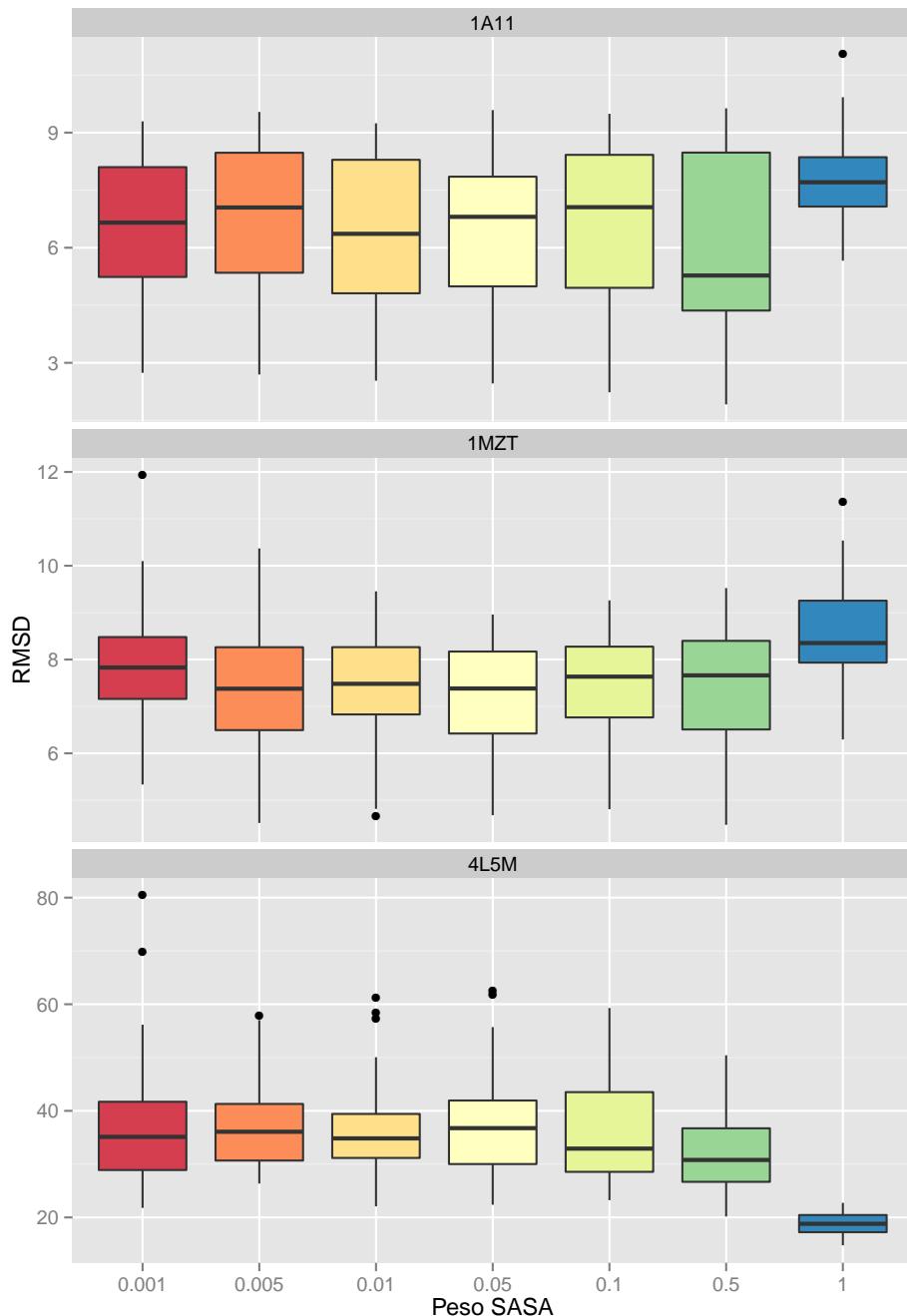


Figura C.6: Energia de solvatação com diferentes pesos, mostrando a distribuição do RMSD de três proteínas

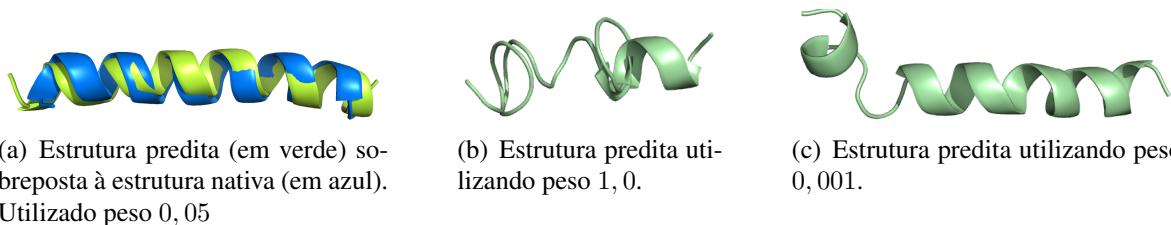
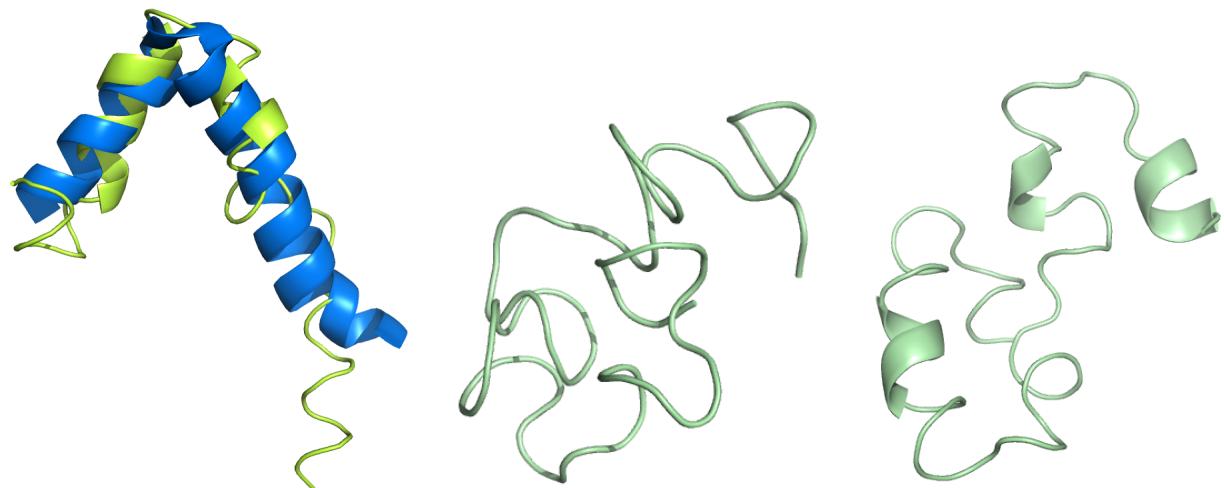
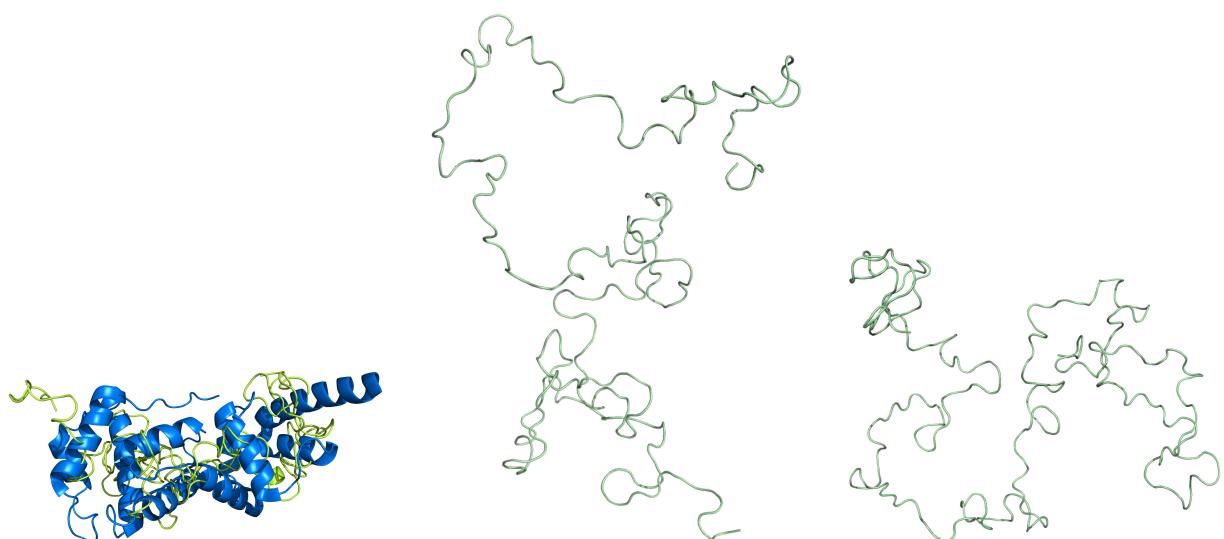


Figura C.7: Estrutura da proteína 1A11 predita utilizando energia de van der Waals e solvatação.



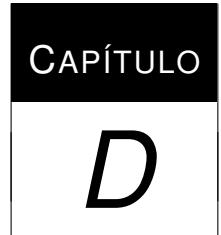
(a) Estrutura predita (em verde) sobreposta à estrutura nativa (em azul). Utilizado peso 0,5.
(b) Estrutura predita utilizando peso 1,0.
(c) Estrutura predita utilizando peso 0,001.

Figura C.8: Estrutura da proteína 1MZT predita utilizando energia de van der Waals e solvatação.



(a) Estrutura predita (em verde) sobreposta à estrutura nativa (em azul). Utilizado peso 0,5.
(b) Estrutura predita utilizando peso 0,5.
(c) Estrutura predita utilizando peso 0,001.

Figura C.9: Estrutura da proteína 4L5M predita utilizando energia de van der Waals e solvatação.



Executando o ProtPred-EDA

Este capítulo mostra uma documentação de como executar o ProtPred-EDA proposto nesta tese. Basicamente, existem duas maneiras para executar o ProtPred-EDA. A primeira, é a execução do ProtPred-EDA em CPU do próprio usuário (aplicativo cliente, Seção D.1) e a segunda maneira é a execução do ProtPred-EDA diretamente no servidor do LCR, por meio de uma interface *web* (Seção D.2).

D.1 ProtPred-EDA local

Esta seção mostra como executar o ProtPred-EDA no computador cliente. Em outras palavras, mostra os procedimentos necessários para que o ProtPred-EDA seja executado na própria máquina do usuário, utilizando recursos próprios. O ProtPred-EDA está escrito em código C/C++ e pode ser obtido na forma:

D.1.1 Código-fonte

- **Repositório GIT** O código-fonte da versão mais atualizada do ProtPred-EDA está disponível no repositório `gitolite@lcrserver.icmc.usp.br:src.git`¹;
- **Arquivo compactado** O arquivo compactado com o código-fonte da versão do ProtPred-EDA (*release 344*) pode ser obtido no endereço <http://lcrserver.icmc.usp.br/~daniel/protpred-eda.r344.7z> (10 MB).

¹Enviar pedido de solicitação para `dbonetti@icmc.usp.br` ou `acbd@icmc.usp.br` caso o acesso seja negado.

O ProtPred-EDA funciona tanto nos Sistemas Operacionais Microsoft Windows quanto Linux. Os requisitos necessários para compilar no Linux são:

- **Biblioteca GSL** Pode ser obtida pelo endereço <http://www.gnu.org/software/gsl/> ou, dependendo da distribuição do Linux é possível instalar o GSL utilizando o comando: `sudo apt-get install gsl`;
- **MPI** Nos experimentos realizados foi utilizado o OpenMPI, que pode ser obtido por meio do endereço <http://www.open-mpi.org/> ou, dependendo da distribuição do Linux é possível instalar o OpenMPI utilizando o comando: `sudo apt-get install openmpi`;

Após obter todo o código-fonte é necessário ir até a pasta *protpred/exp*s e editar o arquivo *Makefile*. Dentro do *Makefile* deve-se localizar a linha *INC = -I/usr/lib/openmpi/include* e substituir pelo caminho do OpenMPI instalado no computador local. Após editar o arquivo *Makefile* deve-se ir até a pasta *protpred/exp*s e digitar *make*. Com isso, um binário chamado *protpred* será gerado na mesma pasta.

Os requisitos para compilar o ProtPred-EDA no Microsoft Windows são:

- **Microsoft Visual Studio 2012** O projeto com o código-fonte para Windows está em Visual Studio 2012. É necessário obter uma licença do Visual Studio 2012 para compilar o projeto ou utilizar a versão de avaliação;
- **Biblioteca GSL** Pode ser obtida pelo endereço <http://www.gnu.org/software/gsl/> ou, a versão compilada para o Visual Studio 2012: <http://lcrserver.icmc.usp.br/~daniel/gslvs2012.rar>;
- **MPI** O Microsoft *High Performance Computing R2* pode ser baixado pelo endereço <http://www.microsoft.com/en-us/download/details.aspx?id=44950>

Para compilar o projeto no Visual Studio 2012 (VS) é necessário selecionar o projeto chamado *experiments* e definir como projeto inicial. Em seguida, escolher o modo de execução (*Debug*, *Release* ou *MPI*) e selecionar a opção para Compilar.

D.1.2 Binários

Os binários do ProtPred-EDA podem ser obtidos diretamente pelos endereços:

- **ProtPred-EDA r344 32 bits para Linux** A versão binária para Linux do ProtPred-EDA pode ser obtida pelo endereço
<http://lcrserver.icmc.usp.br/~daniel/protpred-eda-bin-linux.r344.7z>
- **ProtPred-EDA r344 32 bits para Windows (sem suporte à MPI)** A versão binária para Windows do ProtPred-EDA pode ser obtida pelo endereço <http://lcrserver.icmc.usp.br/~daniel/protpred-eda-bin-win.r344.7z>

D.1.3 Executando

O ProtPred-EDA pode ser executado pela linha de comando. Os comandos são iguais tanto para o Windows quanto para o Linux. Em primeiro lugar, deve-se localizar a pasta que contém o binário do ProtPred-EDA. Nesta pasta, se apenas for chamado o binário `./protpred` será realizada uma execução do ProtPred-EDA utilizando o arquivo de entrada padrão, chamado `input.ini` localizado na mesma pasta do binário. O ProtPred-EDA também pode ser chamado passando o arquivo de entrada como parâmetro como, por exemplo, `./protpred inputs/inC1-HEDA-1a11.ini` que irá executar o ProtPred-EDA com o arquivo de configuração `inC1-HEDA-1a11.ini` (o mesmo utilizado para executar os experimentos com o EDA hierárquico para a proteína 1A11, mostrado na Seção 5.2.2).

D.2 ProtPred-EDA servidor

A versão do ProtPred-EDA *web* (versão servidor), baseado no Framework Galaxy (Goecks et al., 2010) está sendo desenvolvida em colaboração com o mestrande Alexandre Defelicibus, como parte do objetivo de seu trabalho de mestrado (Defelicibus, 2014). O objetivo de seu trabalho é elaborar um ambiente capaz de disponibilizar ferramentas para predição, integração e análise de estruturas de proteínas. O ProtPred-EDA *web* está hospedado na *cloud* do LCR e pode ser acessado pelo endereço: `http://200.144.255.42:8004/`. Embora o projeto ainda esteja em fase de experimentação e testes, é possível executar instâncias do ProtPred-EDA, personalizando os parâmetros de entrada.

A Figura D.1 mostra a tela inicial do Servidor Galaxy do ICMC-USP. A ferramenta correspondente a esta tese de doutorado chama-se *ProtPred-EDA Tools*. Ao clicar nesta opção será exibido as ferramentas disponíveis para o *ProtPred-EDA Tools*. Neste caso, há somente a ferramenta *ProtPred-EDA*. Ao clicar nela é exibido no painel central várias opções de parametrização, de acordo com o método a ser utilizado. A Figura D.2 mostra a tela inicial da ferramenta ao selecionar o *ProtPred-EDA*. Dentro do campo *Optim Method* é possível selecionar todas as metaheurísticas utilizadas neste trabalho. Os parâmetros seguintes, dependem de qual metaheurística foi inicialmente selecionada.

A Figura D.3 mostra um exemplo de como executar o EDA proposto utilizando FGMO para a proteína 1A11, com os mesmos parâmetros utilizados nos experimentos da Seção 5.1.2. Primeiramente, deve-se alterar o *Optim Method* para *EDA - Estimation of Distribution Algorithm* e o campo *Probabilistic Model* para *FGM - Finite Gaussian Mixtures*. Em seguida, ajustar os valores *Population Size*, *Offspring Size* e *Size of Selected* para 200. Em *Source of amino acid sequence* deve-se colocar a sequência de aminoácidos da proteína 1A11: GSEKMSTAISVLLAQAVFLLTSQR, colocar o e-mail para o qual será notificado o término da execução e clicar no botão *Execute*. Isso irá executar um processo no servidor com os parâmetros ajustados e os resultados armazenados serão disponibilizados para o usuário assim que a execução terminar.

A sequência de aminoácidos das nove proteínas utilizadas nos experimentos do Capítulo 5 são apresentadas a seguir:

- **1R8T**
RCCHPQCGAAYSCRK
- **2LLR**
RGCYTRCWVGRNGRVCMRVCT
- **1A11**
GSEKMSTAISVLLAQAVFLLLTSQR
- **2LX0**
KKHTIWEVIAGLVALLTFLAFGFWLFKYLQKK
- **2LVG**
ASRAALIEEGQRIAEMLKSKIQGLLQQASKQAQDIQPAMQ
- **2KK7**
MAVKLMGVVDKIJKSILDDAKAEANKIISEAEAEEKAKILEKAKEAEKRKAEI
- **2X43**
MDQEYRDQMKNAAAEEAKDNVHDK1QELKDDVGNKAAEVRAVSSTVESIKDKLSGGSSSRASSYTL
- **2A3D**
MGSWAEFKQRLAAIKTRLQALGGSEELAAFEKEIAAFESELQAYKGKGNPEVEALRKEEAAIRDELQAYRH
- **2ZGG**
HMGSIDLGKKLLEARAGQDDEVRLMANGADVAAKDKNGSTPLHLAARNGHLEVVKLLLEAGADVNAQDKFGKTAFDISIDNGNEDIAEILQ



Figura D.1: Tela inicial do Servidor Galaxy do ICMC-USP. O painel à esquerda mostra as ferramentas já desenvolvidas e o painel central mostra aonde as ferramentas serão exibidas.

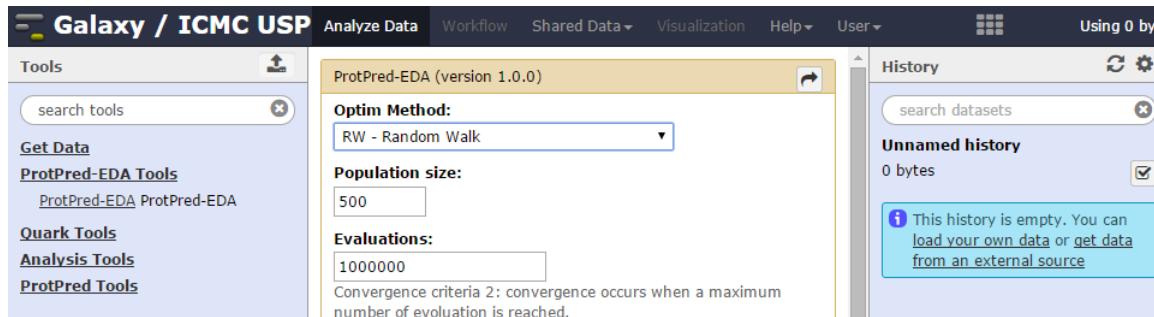


Figura D.2: Exibição inicial da ferramenta ProtPred-EDA.

(a) Exemplo de parâmetros para o EDA com o modelo probabilístico FGM (FGMO).

Optim Method: EDA - Estimation of Distribution Algorithm

Population size: 200

Offspring size: 200

Probabilistic Model: FGM - Finite Gaussian Mixtures

Number of mixture components: 10

Threshold: 1.5

Lambda: 0.9

Selection Method: Tournament

Size of selected: 200

Tournament size: 2

Evaluations: 1000000

Source of amino acid sequence: An upload file or your history as an input or you can insert Own Sequence

Sequence File: (highlighted in red)

(b) Selecionando a opção para entrar com a própria sequência de aminoácidos.

Optim Method: RW - Random Walk

Population size: 500

Evaluations: 1000000

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

Threshold: 0.0001

Source of amino acid sequence: Own Sequence

Your own amino acids sequence: GSEKMSTAISVLLAQAVFLLTSQR

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

User's Email Adress: dbonetti@icmc.usp.br

Email address to which the notice will be sent when the run ends

Execute

(c) Após terminar de ajustar os parâmetros é necessário clicar em Execute.

Use Amino Acid Database (ADB):

Use L amino acids: L is common in nature

User's Email Adress: dbonetti@icmc.usp.br

Email address to which the notice will be sent when the run ends

Execute

Figura D.3: Exemplo de ajuste de parâmetros para a proteína 1A11.

Análises estatísticas

Este capítulo mostra as análises estatísticas realizadas para os experimentos realizados neste trabalho. Devido aos resultados apresentados com relação a energia de van der Waals, RMSD e tempo de execução não apresentarem uma distribuição normal, foi utilizado o teste de Wilcoxon dois-a-dois (Hollander & Wolfe, 1973) para comparar se houve variação significativa entre os métodos avaliados. Se a hipótese nula H_0 for aceita, isto é, quando as amostras não são significativamente diferentes, o p-valor obtido é maior que 0,05 (utilizando um nível de significância de 95%). Caso a hipótese nula seja rejeitada, isto é, há diferença significativa entre as amostras, o p-valor obtido é menor ou igual que 0,05. Assim, é possível quantificar o quanto um método é diferente em relação ao outro.

As Tabelas E.1-E.9 mostram o p-valor para o teste de Wilcoxon para a energia de van der Waals das proteínas 1R8T, 2LLR, 1A11, 2LX0, 2LVG, 2KK7, 2X43, 2A3D e 2ZGG. É destacado em negrito os valores em que a hipótese nula foi rejeitada, ou seja, não houve diferença significativa, segundo o teste de Wilcoxon.

As Tabelas E.10-E.18 mostram o p-valor para o teste de Wilcoxon os valores de RMSD das proteínas 1R8T, 2LLR, 1A11, 2LX0, 2LVG, 2KK7, 2X43, 2A3D e 2ZGG. É destacado em negrito os valores em que a hipótese nula foi rejeitada, ou seja, não houve diferença significativa, segundo o teste de Wilcoxon.

As Tabelas E.19-E.27 mostram o p-valor para o teste de Wilcoxon para o tempo total de execução das metaheurísticas para as proteínas 1R8T, 2LLR, 1A11, 2LX0, 2LVG, 2KK7, 2X43, 2A3D e 2ZGG. É destacado em negrito os valores em que a hipótese nula foi rejeitada, ou seja, não houve diferença significativa, segundo o teste de Wilcoxon.

Tabela E.1: P-valores para a proteína 1R8T da comparação dois-a-dois do teste de Wilcoxon para os valores de energia de van der Waals utilizando o teste de Wilcoxon. Os valores acima de 0,05 estão destacados em negritos.

	RW	MC	GA	DE	UNIO	KDEO	FGMO	hUNIO	hKDEO
MC	1,1e-13								
GA	7,6e-16	1,8e-05							
DE	7,6e-16	8,8e-05	1,0e+00						
UNIO	7,2e-07	7,1e-02	2,1e-01	9,1e-01					
KDEO	2,4e-08	1,2e-03	1,0e+00	5,2e-01	4,6e-03				
FGMO	2,8e-09	1,2e-03	1,0e+00	5,7e-01	8,9e-03	1,0e+00			
hUNIO	7,6e-16	1,5e-10	9,3e-11	8,5e-14	1,1e-09	4,0e-02	2,1e-02		
hKDEO	1,2e-14	2,3e-12	1,2e-14	1,2e-14	2,0e-10	1,8e-04	4,8e-06	6,8e-07	
hFGMO	7,6e-16	4,8e-12	6,1e-14	2,8e-15	3,8e-11	4,8e-04	1,8e-05	3,0e-04	1,0e+00

Tabela E.2: P-valores para a proteína 2LLR da comparação dois-a-dois do teste de Wilcoxon para os valores de energia de van der Waals utilizando o teste de Wilcoxon. Os valores acima de 0,05 estão destacados em negritos.

	RW	MC	GA	DE	UNIO	KDEO	FGMO	hUNIO	hKDEO
MC	1,1e-13								
GA	7,6e-16	2,5e-09							
DE	7,6e-16	3,6e-09	1,0e+00						
UNIO	1,7e-04	1,0e+00	6,6e-04	2,0e-04					
KDEO	1,7e-05	4,2e-03	1,0e+00	1,0e+00	1,7e-05				
FGMO	1,0e+00	1,0e+00	5,3e-01	3,2e-01	1,0e+00	5,3e-01			
hUNIO	7,6e-16	1,1e-13	4,6e-12	3,7e-10	1,0e-14	4,1e-06	2,0e-06		
hKDEO	7,6e-16	1,1e-13	7,6e-16	4,5e-15	3,9e-15	9,6e-13	1,0e-11	1,4e-10	
hFGMO	7,6e-16	1,1e-13	1,8e-12	1,1e-11	1,8e-14	1,5e-10	3,7e-10	1,3e-05	1,0e+00

Tabela E.3: P-valores para a proteína 1A11 da comparação dois-a-dois do teste de Wilcoxon para os valores de energia de van der Waals utilizando o teste de Wilcoxon. Os valores acima de 0,05 estão destacados em negritos.

	RW	MC	GA	DE	UNIO	KDEO	FGMO	hUNIO	hKDEO
MC	9,9e-14								
GA	7,6e-16	2,5e-12							
DE	7,6e-16	8,0e-09	4,1e-01						
UNIO	1,6e-02	1,3e-01	4,5e-10	2,3e-07					
KDEO	1,9e-04	4,5e-01	4,4e-04	6,3e-03	6,3e-03				
FGMO	7,0e-04	4,5e-01	3,2e-07	1,6e-03	4,9e-02	8,9e-01			
hUNIO	7,6e-16	9,9e-14	1,7e-05	7,5e-14	3,9e-15	9,9e-13	1,2e-14		
hKDEO	7,6e-16	9,9e-14	1,2e-11	7,6e-16	3,9e-15	1,6e-14	5,9e-15	4,8e-06	
hFGMO	7,6e-16	9,9e-14	7,3e-05	3,8e-09	4,6e-15	3,8e-10	1,7e-14	4,5e-01	1,6e-06

Tabela E.4: P-valores para a proteína 2LX0 da comparação dois-a-dois do teste de Wilcoxon para os valores de energia de van der Waals utilizando o teste de Wilcoxon. Os valores acima de 0,05 estão destacados em negritos.

	RW	MC	GA	DE	UNIO	KDEO	FGMO	hUNIO	hKDEO
MC	9,9e-14								
GA	7,3e-16	2,5e-10							
DE	7,3e-16	2,4e-09	6,1e-01						
UNIO	1,8e-04	6,1e-01	4,0e-06	1,1e-03					
KDEO	1,8e-04	1,0e+00	1,3e-01	1,3e-01	1,0e+00				
FGMO	9,8e-01	2,0e-01	3,2e-02	3,2e-02	9,8e-01	2,6e-01			
hUNIO	7,3e-16	9,9e-14	2,2e-11	1,7e-12	8,5e-15	1,4e-04	1,4e-04		
hKDEO	7,3e-16	9,9e-14	7,3e-16	7,3e-16	3,6e-15	3,6e-15	8,0e-14	7,3e-16	
hFGMO	2,5e-12	2,0e-10	5,0e-12	2,8e-12	8,2e-22	2,8e-17	7,3e-15	1,8e-07	1,1e-03

Tabela E.5: P-valores para a proteína 2LVG da comparação dois-a-dois do teste de Wilcoxon para os valores de energia de van der Waals utilizando o teste de Wilcoxon. Os valores acima de 0,05 estão destacados em negritos.

	RW	MC	GA	DE	UNIO	KDEO	FGMO	hUNIO	hKDEO
MC	9,9e-14								
GA	7,6e-16	1,0e-12							
DE	7,6e-16	1,5e-10	1,4e-02						
UNIO	2,5e-04	2,5e-01	1,0e-09	2,3e-03					
KDEO	1,8e-04	8,6e-02	1,0e+00	7,6e-01	9,6e-06				
FGMO	5,3e-04	1,4e-01	1,0e+00	7,6e-01	2,9e-05	7,6e-01			
hUNIO	7,6e-16	9,9e-14	3,5e-05	2,8e-10	4,1e-15	1,7e-02	6,3e-01		
hKDEO	7,6e-16	9,9e-14	7,6e-16	7,6e-16	3,6e-15	3,6e-15	4,3e-12	7,6e-16	
hFGMO	7,6e-16	9,9e-14	5,2e-14	7,6e-16	3,6e-15	9,5e-13	9,7e-09	1,5e-12	1,0e+00

Tabela E.6: P-valores para a proteína 2KK7 da comparação dois-a-dois do teste de Wilcoxon para os valores de energia de van der Waals utilizando o teste de Wilcoxon. Os valores acima de 0,05 estão destacados em negritos.

	RW	MC	GA	DE	UNIO	KDEO	FGMO	hUNIO	hKDEO
MC	9,3e-14								
GA	7,6e-16	4,5e-12							
DE	7,6e-16	2,7e-09	1,0e+00						
UNIO	6,3e-04	1,2e-01	2,2e-11	2,7e-09					
KDEO	1,9e-04	1,0e+00	1,1e-01	1,2e-01	5,0e-03				
FGMO	5,7e-04	1,0e+00	2,8e-02	5,3e-02	7,4e-03	1,0e+00			
hUNIO	7,6e-16	9,3e-14	2,2e-12	2,2e-12	3,4e-15	7,4e-03	5,1e-04		
hKDEO	7,6e-16	9,3e-14	7,6e-16	7,6e-16	3,4e-15	3,4e-15	2,2e-13	7,6e-16	
hFGMO	7,6e-16	9,3e-14	7,6e-16	7,6e-16	3,4e-15	8,1e-15	2,6e-13	7,6e-16	1,0e+00

Tabela E.7: P-valores para a proteína 2X43 da comparação dois-a-dois do teste de Wilcoxon para os valores de energia de van der Waals utilizando o teste de Wilcoxon. Os valores acima de 0,05 estão destacados em negritos.

	RW	MC	GA	DE	UNIO	KDEO	FGMO	hUNIO	hKDEO
MC	9,3e-14								
GA	7,6e-16	1,5e-12							
DE	7,6e-16	9,3e-14	4,1e-01						
UNIO	6,9e-04	1,6e-01	3,1e-07	1,7e-08					
KDEO	1,7e-04	3,9e-01	9,4e-02	9,4e-02	2,2e-02				
FGMO	5,3e-04	5,2e-02	1,1e-01	1,2e-01	6,3e-08	6,5e-05			
hUNIO	7,6e-16	9,3e-14	4,7e-07	1,7e-08	4,1e-15	3,4e-02	3,1e-01		
hKDEO	7,6e-16	9,3e-14	7,6e-16	7,6e-16	3,4e-15	3,4e-15	1,6e-08	7,6e-16	
hFGMO	7,6e-16	9,3e-14	7,6e-16	7,6e-16	3,4e-15	4,5e-15	6,4e-09	7,6e-16	3,9e-01

Tabela E.8: P-valores para a proteína 2A3D da comparação dois-a-dois do teste de Wilcoxon para os valores de energia de van der Waals utilizando o teste de Wilcoxon. Os valores acima de 0,05 estão destacados em negritos.

	RW	MC	GA	DE	UNIO	KDEO	FGMO	hUNIO	hKDEO
MC	9,3e-14								
GA	7,6e-16	2,5e-10							
DE	7,6e-16	1,9e-07	8,6e-02						
UNIO	1,0e+00	1,2e-01	4,5e-09	1,1e-05					
KDEO	1,0e+00	2,6e-01	2,4e-02	4,4e-02	1,0e+00				
FGMO	5,3e-04	5,7e-02	1,3e-01	9,2e-02	1,1e-10	2,0e-08			
hUNIO	7,6e-16	9,3e-14	1,6e-05	1,1e-10	6,3e-15	9,2e-03	2,6e-01		
hKDEO	2,6e-15	2,6e-13	2,6e-15	2,6e-15	1,9e-14	2,2e-14	1,8e-04	2,6e-15	
hFGMO	7,6e-16	9,3e-14	7,6e-16	7,6e-16	3,4e-15	3,4e-15	2,3e-11	7,6e-16	8,3e-08

Tabela E.9: P-valores para a proteína 2ZGG da comparação dois-a-dois do teste de Wilcoxon para os valores de energia de van der Waals utilizando o teste de Wilcoxon. Os valores acima de 0,05 estão destacados em negritos.

	RW	MC	GA	DE	UNIO	KDEO	FGMO	hUNIO	hKDEO
MC	9,3e-14								
GA	7,6e-16	9,3e-14							
DE	7,6e-16	9,3e-14	7,8e-01						
UNIO	1,0e+00	2,5e-01	2,9e-06	1,1e-07					
KDEO	1,0e+00	7,8e-01	2,2e-02	1,5e-02	4,6e-01				
FGMO	9,1e-01	4,6e-01	1,1e-01	8,9e-02	4,6e-02	8,9e-02			
hUNIO	7,6e-16	9,3e-14	9,2e-10	7,5e-09	5,7e-15	9,9e-03	2,6e-02		
hKDEO	7,6e-16	9,3e-14	7,6e-16	7,6e-16	3,4e-15	3,5e-15	1,7e-06	7,6e-16	
hFGMO	7,6e-16	9,3e-14	7,6e-16	7,6e-16	3,4e-15	3,7e-15	2,8e-12	7,6e-16	2,9e-07

Tabela E.10: P-valores para a proteína 1R8T da comparação dois-a-dois do teste de Wilcoxon para os valores de RMSD utilizando o teste de Wilcoxon. Os valores acima de 0,05 estão destacados em negritos.

	RW	MC	GA	DE	UNIO	KDEO	FGMO	hUNIO	hKDEO
MC	1,0e+00								
GA	1,0e+00	1,0e+00							
DE	1,0e+00	1,0e+00	1,0e+00						
UNIO	1,0e+00	1,0e+00	1,0e+00	1,0e+00					
KDEO	6,9e-01	3,5e-01	5,4e-03	1,4e-01	4,5e-13				
FGMO	4,1e-02	1,1e-02	1,0e-04	6,1e-03	1,7e-15	5,0e-01			
hUNIO	1,0e+00	1,0e+00	1,0e+00	1,0e+00	1,0e+00	3,9e-02	1,4e-03		
hKDEO	1,0e+00	1,0e+00	1,0e+00	1,0e+00	7,3e-03	1,0e+00	8,1e-02	1,0e+00	
hFGMO	1,0e+00	1,0e+00	1,0e+00	1,0e+00	7,6e-03	1,0e+00	1,7e-01	1,0e+00	1,0e+00

Tabela E.11: P-valores para a proteína 2LLR da comparação dois-a-dois do teste de Wilcoxon para os valores de RMSD utilizando o teste de Wilcoxon. Os valores acima de 0,05 estão destacados em negritos.

	RW	MC	GA	DE	UNIO	KDEO	FGMO	hUNIO	hKDEO
MC	1,3e-01								
GA	1,0e+00	1,0e+00							
DE	1,3e-01	1,0e+00	1,0e+00						
UNIO	1,0e+00	1,5e-02	6,9e-01	8,1e-03					
KDEO	3,0e-03	1,0e+00	1,0e+00	1,0e+00	2,6e-06				
FGMO	1,0e+00	7,2e-01	1,0e+00	7,5e-01	1,0e+00	5,4e-03			
hUNIO	4,6e-01	1,0e+00	1,0e+00	1,0e+00	1,3e-01	1,0e+00	1,0e+00		
hKDEO	1,4e-01	1,0e+00	1,0e+00	1,0e+00	5,7e-02	1,0e+00	1,0e+00	1,0e+00	
hFGMO	1,0e+00	1,0e+00	1,0e+00	1,0e+00	1,0e+00	2,8e-01	1,0e+00	1,0e+00	1,0e+00

Tabela E.12: P-valores para a proteína 1A11 da comparação dois-a-dois do teste de Wilcoxon para os valores de RMSD utilizando o teste de Wilcoxon. Os valores acima de 0,05 estão destacados em negritos.

	RW	MC	GA	DE	UNIO	KDEO	FGMO	hUNIO	hKDEO
MC	1,5e-02								
GA	2,7e-01	1,0e+00							
DE	1,0e+00	1,0e+00	1,0e+00						
UNIO	1,5e-04	1,0e+00	1,0e+00	4,8e-01					
KDEO	1,2e-02	7,1e-04	1,5e-04	9,9e-03	1,8e-12				
FGMO	3,2e-05	4,2e-05	4,3e-06	2,2e-05	3,1e-15	4,0e-04			
hUNIO	4,8e-01	1,0e+00	1,0e+00	1,0e+00	1,0e+00	3,0e-04	5,1e-06		
hKDEO	8,8e-01	1,0e+00	1,0e+00	1,0e+00	1,0e+00	8,2e-03	3,0e-04	1,0e+00	
hFGMO	1,4e-05	4,3e-06	4,0e-08	9,5e-06	3,2e-11	3,5e-01	1,0e+00	1,9e-07	6,0e-05

Tabela E.13: P-valores para a proteína 2LX0 da comparação dois-a-dois do teste de Wilcoxon para os valores de RMSD utilizando o teste de Wilcoxon. Os valores acima de 0,05 estão destacados em negritos.

	RW	MC	GA	DE	UNIO	KDEO	FGMO	hUNIO	hKDEO
MC	0,048								
GA	1,000	0,627							
DE	1,000	0,551	1,000						
UNIO	1,000	1,000	1,000	1,000					
KDEO	1,000	0,030	1,000	1,000	0,371				
FGMO	1,000	1,000	1,000	1,000	1,000	0,547			
hUNIO	1,000								
hKDEO	1,000	0,231	1,000	1,000	1,000	1,000	1,000	1,000	
hFGMO	1,000	0,041	1,000	1,000	1,000	1,000	0,649	1,000	1,000

Tabela E.14: P-valores para a proteína 2LVG da comparação dois-a-dois do teste de Wilcoxon para os valores de RMSD utilizando o teste de Wilcoxon. Os valores acima de 0,05 estão destacados em negritos.

Tabela E.15: P-valores para a proteína 2KK7 da comparação dois-a-dois do teste de Wilcoxon para os valores de RMSD utilizando o teste de Wilcoxon. Os valores acima de 0,05 estão destacados em negritos.

Tabela E.16: P-valores para a proteína 2X43 da comparação dois-a-dois do teste de Wilcoxon para os valores de RMSD utilizando o teste de Wilcoxon. Os valores acima de 0,05 estão destacados em negritos.

	RW	MC	GA	DE	UNIO	KDEO	FGMO	hUNIO	hKDEO
MC	3,1e-01								
GA	2,4e-04	1,0e+00							
DE	4,8e-03	1,0e+00	1,0e+00						
UNIO	9,0e-07	1,0e+00	1,0e+00	1,0e+00					
KDEO	1,1e-01	1,0e+00	3,5e-01	1,0e+00	1,4e-02				
FGMO	2,7e-02	1,0e+00	1,0e+00	1,0e+00	1,1e-01	1,0e+00			
hUNIO	1,2e-04	7,2e-01	1,0e+00	1,0e+00	1,0e+00	1,0e-01		2,2e-01	
hKDEO	6,0e-04	1,0e+00	1,0e+00	1,0e+00	1,0e+00	3,8e-01	7,0e-01	1,0e+00	
hFGMO	1,0e+00	1,0e+00	1,0e+00	1,0e+00	7,0e-01	1,0e+00	1,0e+00	4,8e-01	1,0e+00

Tabela E.17: P-valores para a proteína 2A3D da comparação dois-a-dois do teste de Wilcoxon para os valores de RMSD utilizando o teste de Wilcoxon. Os valores acima de 0,05 estão destacados em negritos.

	RW	MC	GA	DE	UNIO	KDEO	FGMO	hUNIO	hKDEO
MC	1,3e-06								
GA	9,0e-09	1,0e+00							
DE	7,1e-01	6,1e-03	7,1e-04						
UNIO	6,3e-06	1,0e+00	1,0e+00	1,3e-01					
KDEO	5,4e-06	1,0e+00	1,0e+00	3,7e-02	1,0e+00				
FGMO	6,1e-05	1,0e+00	1,0e+00	3,2e-01	1,0e+00	1,0e+00			
hUNIO	2,1e-06	1,0e+00	1,0e+00	1,8e-02	1,0e+00	1,0e+00	1,0e+00		
hKDEO	6,6e-06	1,0e+00	1,0e+00	1,7e-01	1,0e+00	1,0e+00	1,0e+00	1,0e+00	
hFGMO	1,8e-05	1,0e+00	1,0e+00	5,0e-02	1,0e+00	1,0e+00	1,0e+00	1,0e+00	1,0e+00

Tabela E.18: P-valores para a proteína 2ZGG da comparação dois-a-dois do teste de Wilcoxon para os valores de RMSD utilizando o teste de Wilcoxon. Os valores acima de 0,05 estão destacados em negritos.

	RW	MC	GA	DE	UNIO	KDEO	FGMO	hUNIO	hKDEO
MC	3,5e-03								
GA	6,0e-06	1,0e+00							
DE	2,0e-02	1,0e+00	1,0e+00						
UNIO	2,8e-07	4,0e-01	1,0e+00	8,9e-02					
KDEO	1,3e-03	1,0e+00	1,0e+00	1,0e+00	1,0e+00				
FGMO	5,6e-08	1,9e-01	1,0e+00	1,8e-02	1,0e+00	7,9e-01			
hUNIO	4,0e-04	1,0e+00	1,0e+00	1,0e+00	5,5e-01	1,0e+00	1,7e-01		
hKDEO	1,0e+00	4,3e-01	2,3e-02	1,0e+00	2,5e-04	1,6e-01	2,5e-05	1,4e-01	
hFGMO	7,0e-05	1,0e+00	1,0e+00	1,0e+00	9,7e-01	1,0e+00	4,7e-01	1,0e+00	3,0e-02

Tabela E.19: P-valores para a proteína 1R8T da comparação dois-a-dois do teste de Wilcoxon para os valores de energia de tempo de execuçãp utilizando o teste de Wilcoxon. Os valores acima de 0,05 estão destacados em negritos.

	RW	MC	GA	DE	UNIO	KDEO	FGMO	hUNIO	hKDEO
MC	1,2e-12								
GA	7,6e-16	2,6e-11							
DE	7,6e-16	5,5e-14	7,6e-16						
UNIO	3,6e-15	5,4e-11	1,6e-04	3,6e-15					
KDEO	3,6e-15	8,0e-13	3,6e-15	3,6e-15	3,8e-12				
FGMO	3,6e-15	8,0e-13	3,6e-15	3,6e-15	2,6e-07	3,8e-12			
hUNIO	7,6e-16	5,5e-14	7,6e-16	7,6e-16	3,6e-15	3,6e-15	3,6e-15		
hKDEO	6,6e-15	3,8e-13	6,6e-15	6,6e-15	5,8e-14	5,8e-14	5,8e-14	6,6e-15	
hFGMO	7,6e-16	5,5e-14	7,6e-16	7,6e-16	3,6e-15	3,6e-15	3,6e-15	3,1e-04	6,6e-15

Tabela E.20: P-valores para a proteína 2LLR da comparação dois-a-dois do teste de Wilcoxon para os valores de energia de tempo de execuçãp utilizando o teste de Wilcoxon. Os valores acima de 0,05 estão destacados em negritos.

	RW	MC	GA	DE	UNIO	KDEO	FGMO	hUNIO	hKDEO
MC	2,0e-05								
GA	7,6e-16	5,5e-04							
DE	7,6e-16	4,8e-14	7,6e-16						
UNIO	3,1e-15	2,4e-07	3,3e-05	3,1e-15					
KDEO	3,1e-15	8,8e-13	3,1e-15	3,1e-15	7,5e-11				
FGMO	3,1e-15	1,5e-11	7,4e-11	3,1e-15	5,5e-04	2,3e-10			
hUNIO	7,6e-16	4,8e-14	7,6e-16	7,6e-16	3,1e-15	3,1e-15	3,1e-15		
hKDEO	7,6e-16	4,8e-14	7,6e-16	7,6e-16	3,1e-15	3,1e-15	3,1e-15	7,6e-16	
hFGMO	7,6e-16	4,8e-14	7,6e-16	7,6e-16	3,1e-15	3,1e-15	3,1e-15	1,5e-05	7,6e-16

Tabela E.21: P-valores para a proteína 1A11 da comparação dois-a-dois do teste de Wilcoxon para os valores de energia de tempo de execuçãp utilizando o teste de Wilcoxon. Os valores acima de 0,05 estão destacados em negritos.

	RW	MC	GA	DE	UNIO	KDEO	FGMO	hUNIO	hKDEO
MC	9,6e-06								
GA	4,7e-12	5,2e-02							
DE	7,6e-16	6,1e-14	7,6e-16						
UNIO	1,0e-13	5,7e-07	2,2e-06	3,2e-15					
KDEO	3,2e-15	1,0e-12	3,2e-15	3,2e-15	3,3e-07				
FGMO	6,4e-15	8,7e-07	5,3e-06	4,1e-15	1,0e-01	2,8e-06			
hUNIO	7,6e-16	6,1e-14	7,6e-16	7,6e-16	3,2e-15	2,4e-13	4,1e-15		
hKDEO	7,6e-16	6,1e-14	7,6e-16	7,6e-16	3,2e-15	3,2e-15	4,1e-15	7,6e-16	
hFGMO	7,6e-16	6,1e-14	7,6e-16	7,6e-16	5,7e-15	4,3e-10	5,8e-14	2,4e-10	7,6e-16

As Tabelas E.28-E.30 mostram o teste de comparação de Wilcoxon dois-a-dois para os EDAs hierárquicos utilizando quatro valores de α diferentes. Neste experimento foi utilizado a proteína

Tabela E.22: P-valores para a proteína 2LX0 da comparação dois-a-dois do teste de Wilcoxon para os valores de energia de tempo de execução utilizando o teste de Wilcoxon. Os valores acima de 0,05 estão destacados em negritos.

	RW	MC	GA	DE	UNIO	KDEO	FGMO	hUNIO	hKDEO
MC	7,8e-01								
GA	4,3e-09	3,3e-02							
DE	7,1e-16	7,4e-14	7,1e-16						
UNIO	1,4e-10	3,3e-07	6,6e-05	3,3e-15					
KDEO	3,3e-15	1,4e-12	3,3e-15	3,3e-15	3,2e-07				
FGMO	7,4e-14	3,8e-10	4,5e-11	4,3e-15	1,7e-03	3,9e-06			
hUNIO	7,1e-16	7,4e-14	7,1e-16	7,1e-16	3,3e-15	2,3e-13	4,3e-15		
hKDEO	7,1e-16	7,4e-14	7,1e-16	7,1e-16	3,3e-15	3,3e-15	4,3e-15	7,1e-16	
hFGMO	1,6e-12	1,1e-10	1,6e-12	1,6e-12	5,8e-21	8,8e-17	1,4e-20	1,7e-03	1,6e-12

Tabela E.23: P-valores para a proteína 2LVG da comparação dois-a-dois do teste de Wilcoxon para os valores de energia de tempo de execução utilizando o teste de Wilcoxon. Os valores acima de 0,05 estão destacados em negritos.

	RW	MC	GA	DE	UNIO	KDEO	FGMO	hUNIO	hKDEO
MC	6,7e-01								
GA	3,2e-03	7,1e-02							
DE	7,6e-16	6,4e-14	7,6e-16						
UNIO	3,0e-06	6,6e-06	7,0e-05	3,2e-15					
KDEO	3,2e-15	1,3e-12	3,2e-15	3,2e-15	3,7e-07				
FGMO	4,7e-12	9,8e-10	9,0e-10	4,3e-15	7,1e-02	5,6e-06			
hUNIO	7,6e-16	6,4e-14	7,6e-16	7,6e-16	7,8e-15	1,8e-10	1,1e-14		
hKDEO	7,6e-16	6,4e-14	7,6e-16	7,6e-16	3,2e-15	3,2e-15	4,3e-15	7,6e-16	
hFGMO	7,6e-16	6,4e-14	7,6e-16	7,6e-16	2,6e-14	2,3e-10	4,2e-14	6,7e-01	7,6e-16

Tabela E.24: P-valores para a proteína 2KK7 da comparação dois-a-dois do teste de Wilcoxon para os valores de energia de tempo de execução utilizando o teste de Wilcoxon. Os valores acima de 0,05 estão destacados em negritos.

	RW	MC	GA	DE	UNIO	KDEO	FGMO	hUNIO	hKDEO
MC	6,0e-01								
GA	6,0e-01	6,0e-01							
DE	7,6e-16	7,7e-14	7,6e-16						
UNIO	3,2e-03	1,4e-04	1,4e-04	3,2e-15					
KDEO	3,2e-15	1,6e-12	3,2e-15	3,2e-15	5,3e-07				
FGMO	5,3e-06	4,1e-06	1,3e-06	4,3e-15	6,0e-01	8,0e-06			
hUNIO	7,6e-16	7,7e-14	7,6e-16	7,6e-16	1,7e-12	1,6e-09	2,4e-12		
hKDEO	7,6e-16	7,7e-14	7,6e-16	7,6e-16	3,2e-15	3,2e-15	4,3e-15	7,6e-16	
hFGMO	7,6e-16	7,7e-14	7,6e-16	7,6e-16	3,1e-11	2,9e-09	7,7e-11	1,4e-04	7,6e-16

2LVG e, de acordo com o teste de Wilcoxon, não houve diferença significativa entre o valor do α utilizado para o KDE2D e FGM. Apesar disso, os experimentos com o EDA hierárquico Capítulo 5 foram executados com $\alpha = 2$, pois encontrou valores de energia mais baixos.

Tabela E.25: P-valores para a proteína 2X43 da comparação dois-a-dois do teste de Wilcoxon para os valores de energia de tempo de execuçãp utilizando o teste de Wilcoxon. Os valores acima de 0,05 estão destacados em negritos.

	RW	MC	GA	DE	UNIO	KDEO	FGMO	hUNIO	hKDEO
MC	4,3e-01								
GA	3,7e-02	3,7e-02							
DE	7,6e-16	7,7e-14	7,6e-16						
UNIO	2,1e-03	2,3e-05	2,0e-03	3,2e-15					
KDEO	3,2e-15	1,5e-12	3,2e-15	3,2e-15	7,3e-07				
FGMO	5,7e-09	8,7e-09	4,1e-08	4,3e-15	1,8e-01	2,4e-05			
hUNIO	7,6e-16	7,7e-14	7,6e-16	7,6e-16	1,6e-10	2,6e-09	1,0e-12		
hKDEO	7,6e-16	7,7e-14	7,6e-16	7,6e-16	3,2e-15	3,2e-15	4,3e-15	7,6e-16	
hFGMO	7,6e-16	7,7e-14	7,6e-16	7,6e-16	9,8e-10	2,6e-09	1,6e-10	3,9e-08	7,6e-16

Tabela E.26: P-valores para a proteína 2A3D da comparação dois-a-dois do teste de Wilcoxon para os valores de energia de tempo de execuçãp utilizando o teste de Wilcoxon. Os valores acima de 0,05 estão destacados em negritos.

	RW	MC	GA	DE	UNIO	KDEO	FGMO	hUNIO	hKDEO
MC	1,4e-03								
GA	1,3e-01	7,7e-01							
DE	7,6e-16	7,7e-14	7,6e-16						
UNIO	3,8e-02	5,3e-02	3,8e-02	7,3e-15					
KDEO	7,3e-15	2,8e-12	9,8e-15	7,3e-15	1,1e-04				
FGMO	6,6e-07	6,6e-08	1,2e-08	5,3e-15	1,0e+00	1,3e-05			
hUNIO	7,6e-16	7,7e-14	7,6e-16	7,6e-16	2,3e-06	5,7e-01	8,8e-11		
hKDEO	2,5e-15	2,1e-13	2,5e-15	2,5e-15	3,5e-14	3,5e-14	2,7e-14	2,5e-15	
hFGMO	7,6e-16	7,7e-14	7,6e-16	7,6e-16	3,9e-06	5,3e-01	1,4e-10	1,0e+00	2,5e-15

Tabela E.27: P-valores para a proteína 2ZGG da comparação dois-a-dois do teste de Wilcoxon para os valores de energia de tempo de execuçãp utilizando o teste de Wilcoxon. Os valores acima de 0,05 estão destacados em negritos.

	RW	MC	GA	DE	UNIO	KDEO	FGMO	hUNIO	hKDEO
MC	1,0e+00								
GA	1,2e-02	1,0e+00							
DE	7,6e-16	9,6e-14	7,6e-16						
UNIO	1,0e+00	1,0e+00	1,0e+00	1,1e-12					
KDEO	1,1e-12	8,6e-11	1,1e-12	1,1e-12	5,3e-05				
FGMO	1,9e-08	3,5e-07	5,4e-10	2,1e-14	1,3e-09	1,0e+00			
hUNIO	7,6e-16	9,6e-14	7,6e-16	7,6e-16	1,1e-12	1,1e-12	2,0e-08		
hKDEO	7,6e-16	9,6e-14	7,6e-16	7,6e-16	1,1e-12	1,1e-12	2,1e-14	7,6e-16	
hFGMO	7,6e-16	9,6e-14	7,6e-16	7,6e-16	1,1e-12	1,1e-12	1,3e-08	1,0e+00	7,6e-16

As Tabelas E.31-E.33 mostram o teste de comparação de Wilcoxon dois-a-dois entre o EDA não hierárquico e hierárquico com dois e três subproblemas. A Tabela E.31 mostra a comparação

Tabela E.28: P-valores do teste de Wilcoxon para o EDA hierárquico com o modelo probabilístico UNI, avaliando três valores para α , considerando o aspecto energia de van der Waals para a proteína 2LVG. Os valores acima de 0,05 estão destacados em negritos.

	$\alpha = 0$	$\alpha = 1$	$\alpha = 2$
$\alpha = 1$	0,454		
$\alpha = 2$	0,636	0,636	
$\alpha = 3$	0,017	0,636	0,044

Tabela E.29: P-valores do teste de Wilcoxon para o EDA hierárquico com o modelo probabilístico KDE2D, avaliando três valores para α , considerando o aspecto energia de van der Waals para a proteína 2LVG. Os valores acima de 0,05 estão destacados em negritos.

	$\alpha = 0$	$\alpha = 1$	$\alpha = 2$
$\alpha = 1$	1,00		
$\alpha = 2$	0,99	0,66	
$\alpha = 3$	1,00	1,00	1,00

Tabela E.30: P-valores do teste de Wilcoxon para o EDA hierárquico com o modelo probabilístico FGM, avaliando três valores para α , considerando o aspecto energia de van der Waals para a proteína 2LVG. Os valores acima de 0,05 estão destacados em negritos.

	$\alpha = 0$	$\alpha = 1$	$\alpha = 2$
$\alpha = 1$	1,00		
$\alpha = 2$	0,78	0,25	
$\alpha = 3$	1,00	1,00	0,48

entre o p-valor considerando a energia de van der Waals, a Tabela E.32 mostra a comparação entre os valores de RMSD e a Tabela E.33 a comparação entre os tempos de execução.

Tabela E.31: P-valores do teste de Wilcoxon entre os EDAs sem hierarquia e com hierarquia em que $m = 2, 3$, avaliando a energia de van der Waals, para a proteína 2LVG. Os valores acima de 0,05 estão destacados em negritos.

Tabela E.32: P-valores do teste de Wilcoxon entre os EDAs sem hierarquia e com hierarquia em que $m = 2, 3$, avaliando o RMSD, para a proteína 2LVG. Os valores acima de 0,05 estão destacados em negritos.

	UNIO	KDEO	FGMO	hUNIO	$m = 2$	hFGMO	$m = 3$	
				hUNIO	hKDEO		hUNIO	hKDEO
KDEO	1,0000							
FGMO	1,0000	1,0000						
hUNIO $m = 2$	1,0000	0,7244	0,9125					
hKDEO $m = 2$	1,0000	0,9700	1,0000	1,0000				
hFGMO $m = 2$	1,0000	0,7244	1,0000	1,0000	1,0000			
hUNIO $m = 3$	1,0000	0,6747	0,4963	1,0000	1,0000	1,0000		
hKDEO $m = 3$	1,0000							
hFGMO $m = 3$	0,9700	0,0285	0,0077	1,0000	1,0000	1,0000	1,0000	1,0000

Tabela E.33: P-valores do teste de Wilcoxon entre os EDAs sem hierarquia e com hierarquia em que $m = 2, 3$, avaliando o tempo de execução, para a proteína 2LVG. Os valores acima de 0,05 estão destacados em negritos.

	UNIO	KDEO	FGMO	hUNIO	$m = 2$	hFGMO	$m = 3$	
				hUNIO	hKDEO		hUNIO	hKDEO
KDEO	9,2e-15							
FGMO	4,0e-03	6,4e-17						
hUNIO $m = 2$	4,5e-07	1,1e-06	7,7e-08					
hKDEO $m = 2$	4,5e-07	4,5e-07	7,7e-08	6,6e-04				
hFGMO $m = 2$	4,5e-07	4,5e-07	7,7e-08	3,4e-01	1,2e-03			
hUNIO $m = 3$	4,5e-07	4,5e-07	7,7e-08	6,6e-04	3,4e-01	6,6e-04		
hKDEO $m = 3$	4,5e-07	4,5e-07	7,7e-08	6,6e-04	6,6e-04	6,6e-04	6,6e-04	
hFGMO $m = 3$	4,5e-07	4,5e-07	7,7e-08	6,6e-04	3,1e-01	6,6e-04	3,4e-01	1,2e-03

A Tabela E.34 mostra o teste de comparação para os valores da energia de van der Waals para as nove proteínas com os três EDAs propostos. Os valores dessa tabela representam se houve diferença significativa entre o fato de utilizar ou não o banco de dados de ângulos diedrais (ADB) para a geração da população inicial. A Tabela E.35 faz a mesma comparação, porém, para os valores de RMSD.

Tabela E.34: P-valores do teste de Wilcoxon para os EDAs UNIO, KDEO e FGMO, com e sem ADB, para nove proteínas, considerando o aspecto energia de van der Waals. Os valores acima de 0,05 estão destacados em negritos.

	1R8T	2LLR	1A11	2LX0	2LVG	2KK7	2X43	2A3D	2ZGG
UNIO	2,1e-10	4,7e-09	7,2e-13	4,5e-10	6,4e-08	8,8e-09	3,8e-06	1,9e-03	3,1e-03
KDEO	1,7e-03	7,8e-10	1,9e-11	7,1e-11	6,8e-11	1,4e-10	1,0e-09	1,4e-06	8,3e-07
FGMO	4,0e-02	2,7e-03	2,6e-11	2,7e-05	3,4e-01	7,4e-07	6,0e-02	9,8e-01	1,1e-03

Tabela E.35: P-valores do teste de Wilcoxon para os EDAs UNIO, KDEO e FGMO, com e sem ADB, para nove proteínas, considerando o aspecto RMSD. Os valores acima de 0,05 estão destacados em negritos.

	1R8T	2LLR	1A11	2LX0	2LVG	2KK7	2X43	2A3D	2ZGG
UNIO	4,3e-10	7,7e-01	1,8e-02	7,8e-02	8,4e-01	8,7e-01	8,0e-02	6,5e-02	4,2e-05
KDEO	5,8e-06	3,1e-05	1,6e-08	8,4e-03	8,1e-01	4,7e-02	2,5e-02	2,5e-01	3,8e-01
FGMO	1,6e-05	3,1e-01	4,3e-06	8,5e-02	5,2e-01	3,1e-04	3,2e-01	6,5e-01	3,4e-03

A Tabela E.36 mostra a mesma comparação realizada pela Tabela E.34, porém, considerando todas as metaheurísticas avaliadas, ou seja, o teste de comparação de Wilcoxon para as metaheurística levando em consideração o fato de utilizar ou não ADB. Da mesma maneira, a Tabela E.37 faz uma comparação similar a Tabela E.35, porém, avaliando os valores do RMSD.

Tabela E.36: P-valores do teste de Wilcoxon para os métodos RW, MC, GA, DE, FGMO e hFGMO com e sem ADB, para nove proteínas, considerando o aspecto energia de van der Waals. Os valores acima de 0,05 estão destacados em negritos.

	1R8T	2LLR	1A11	2LX0	2LVG	2KK7	2X43	2A3D	2ZGG
RW	1,7e-17	1,7e-17	1,7e-17	1,7e-17	5,2e-15	1,7e-17	1,7e-17	1,7e-17	1,7e-17
MC	2,7e-06	3,3e-11	4,9e-13	4,9e-13	2,9e-12	4,9e-13	4,9e-13	4,9e-13	4,9e-13
GA	3,9e-13	3,5e-14	1,7e-17	1,7e-17	5,2e-15	1,7e-17	1,7e-17	1,7e-17	1,7e-17
DE	9,8e-01	1,3e-01	3,9e-02	5,4e-05	8,4e-05	2,4e-08	5,8e-11	6,3e-12	3,3e-15
FGMO	4,0e-02	2,7e-03	2,6e-11	2,7e-05	3,4e-01	7,4e-07	6,0e-02	9,8e-01	1,1e-03
hFGMO	1,5e-01	7,5e-01	6,2e-07	1,7e-10	6,5e-01	4,9e-11	1,7e-17	1,7e-17	1,7e-17

A Tabela E.38 faz uma comparação com o teste de Wilcoxon para os valores de RMSD produzidos, entre uma função de *fitness* que utiliza somente a energia de van der Waals e outra execução

Tabela E.37: P-valores do teste de Wilcoxon para os métodos RW, MC, GA, DE, FGMO e hFGMO com e sem ADB, para nove proteínas, considerando o aspecto RMSD. Os valores acima de 0,05 estão destacados em negritos.

	1R8T	2LLR	1A11	2LX0	2LVG	2KK7	2X43	2A3D	2ZGG
RW	6,0e-03	2,1e-01	2,4e-01	8,9e-01	6,5e-01	6,9e-01	4,6e-01	5,8e-01	2,0e-01
MC	5,2e-01	2,2e-01	2,7e-02	6,3e-01	7,7e-03	6,8e-01	2,5e-01	1,9e-02	7,6e-01
GA	9,2e-01	3,0e-01	1,0e-03	2,4e-03	7,3e-02	2,4e-01	7,6e-01	4,7e-01	2,0e-01
DE	3,8e-01	6,2e-01	4,3e-03	9,7e-01	9,6e-02	9,7e-01	1,6e-01	5,7e-02	7,7e-01
FGMO	1,6e-05	3,1e-01	4,3e-06	8,5e-02	5,2e-01	3,1e-04	3,2e-01	6,5e-01	3,4e-03
hFGMO	2,1e-01	7,8e-01	5,9e-09	8,0e-03	5,4e-02	5,7e-02	4,1e-01	4,0e-02	9,2e-04

em que a função de *fitness* sejam composta pela soma da energia de van der Waals e energia de solvatação.

Tabela E.38: P-valores do teste de Wilcoxon, com e sem energia de solvatação, para nove proteínas, considerando o aspecto RMSD. Os valores acima de 0,05 estão destacados em negritos.

	UNIO	KDEO	FGMO
1R8T	1,5e-04	7,7e-02	1,5e-01
2LLR	8,8e-01	7,6e-01	6,6e-01
1A11	3,7e-01	2,7e-01	8,6e-02
2LX0	2,7e-01	5,6e-01	8,8e-01
2LVG	4,6e-01	7,6e-02	2,6e-02
2KK7	6,6e-02	1,3e-01	7,5e-01
2X43	8,4e-01	5,9e-01	6,6e-01
2A3D	1,5e-09	5,5e-08	9,7e-01
2ZGG	3,2e-16	3,2e-16	5,0e-11

Referências Bibliográficas

AHN, C.; RAMAKRISHNA, R.; GOLDBERG, D. Real-coded bayesian optimization algorithm: Bringing the strength of boa into the continuous world. In: DEB, K., ed. *Genetic and Evolutionary Computation GECCO 2004*, v. 3102 de *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, p. 840–851, 2004.

Disponível em http://dx.doi.org/10.1007/978-3-540-24854-5_86

AHN, C. W.; RAMAKRISHNA, R. S. On the scalability of real-coded bayesian optimization algorithm. *IEEE Transactions On Evolutionary Computation*, v. 12, n. 3, 2008.

ALBERTS, B.; BRAY, D.; HOPKIN, K.; JOHNSON, A.; LEWIS, J.; RAFF, M.; ROBERTS, K.; WALTER, P. *Fundamentos da biologia celular*. Artmed Editora, 2007.

ANFINSEN, C. B. Studies on the principles that govern the folding of protein chains. *Nobel Lecture*, p. 103–119, 1972.

ARNOLD, K.; BORDOLI, L.; KOPP, J.; SCHWEDE, T. The swiss-model workspace: A web-based environment for protein structure homology modelling. *Bioinformatics*, , n. 22, p. 195–201, 2006.

AUGER, A.; BADER, J.; BROCKHOFF, D.; ZITZLER, E. Theory of the hypervolume indicator: Optimal mu-distributions and the choice of the reference point. In: *Proceedings of the Tenth ACM SIGEVO Workshop on Foundations of Genetic Algorithms*, FOGA '09, New York, NY, USA: ACM, 2009, p. 87–102 (FOGA '09,).

Disponível em <http://doi.acm.org/10.1145/1527125.1527138>

BACARDIT, J.; STOUT, M.; HIRST, J. D.; SASTRY, K.; LLORÀ, X.; KRASNOKOR, N. Automated alphabet reduction method with evolutionary algorithms for protein structure prediction. In: *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, GECCO '07, New York, NY, USA: ACM, 2007, p. 346–353 (GECCO '07,).

Disponível em <http://doi.acm.org/10.1145/1276958.1277033>

- BALUJA, S. *Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning.* Relatório Técnico CMU-CS-94-163, Computer Science Department, Pittsburgh, PA, 1994.
- BAXEVANIS, A. D.; OUELLETTE, B. F. F. *Bioinformatics 2th edition: A practical guide to the analysis of genes and proteins.* Wiley-Interscience, 2001.
- BÄCK, T. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms.* Oxford University Press, USA, 1996.
- BECKER, O. M.; JR., A. D. M.; ROUX, B.; WATANABE, M. *Computational biochemistry and biophysics.* CRC, 2001.
- BECKER, R. A.; CHAMBERS, J. M.; WILKS, A. R. *The new s language : a programming environment for data analysis and graphics.* Pacific Grove, Calif. : Wadsworth & Brooks/Cole Advanced Books & Software, includes index, 1988.
- BERENBOYM, I.; AVIGAL, M. Genetic algorithms with local search optimization for protein structure prediction problem. In: *Proceedings of the 10th annual conference on Genetic and evolutionary computation*, GECCO '08, New York, NY, USA: ACM, 2008, p. 1097–1098 (GECCO '08,).
Disponível em <http://doi.acm.org/10.1145/1389095.1389296>
- BERG, J.; TYMOCZKO, J.; STRYER, L. *Biochemistry 5th ed.* W. H. Freeman, 894 p., 2002.
- BERGERON, B. *Bioinformatics computing.* Prentice Hall, 2002.
- BERMAN, H. M.; WESTBROOK, J.; FENG, Z.; GILLILAND, G.; BHAT, T. N.; WEISSIG, H.; SHINDYALOV, I. N.; BOURNE, P. E. The protein data bank. 2000.
Disponível em <http://www.rcsb.org/pdb/home/home.do> (Acessado em 05/02/2009)
- BONDI, A. van der waals volumes and radii. *The Journal of Physical Chemistry*, v. 68, n. 3, p. 441–451, 1964.
Disponível em <http://dx.doi.org/10.1021/j100785a001>
- BONETTI, D.; PEREZ-SANCHEZ, H.; DELBEM, A. An efficient solvent accessible surface area calculation applied in ab initio protein structure prediction. In: *International Work-Conference on Bioinformatics and Biomedical Engineering*, Granda, Spain, 2014.
- BONETTI, D. R.; DELBEM, A. C.; TRAVIESO, G.; DE SOUZA, P. S. L. Enhanced van der waals calculations in genetic algorithms for protein structure prediction. *Concurrency and Computation: Practice and Experience*, v. 25, n. 15, p. 2170–2186, 2013.
Disponível em <http://dx.doi.org/10.1002/cpe.2913>

- BONETTI, D. R. F. *Aumento da eficiência do cálculo da energia de van der waals em algoritmos genéticos para predição de estruturas de proteínas.* Dissertação de Mestrado, USP - ICMC, 2010.
- BONETTI, D. R. F.; DELBEM, A. C. B.; ANDRADE, F. B. Improving the efficiency of the van der waals energy function used in the genetic algorithms for protein structure prediction. *Biomat 2010 10h International Symposium on Mathematical and Computational Biology*, 2010a.
- BONETTI, D. R. F.; DELBEM, A. C. B.; TRAVIESO, G.; DE SOUZA, P. S. L. Optimizing van der waals calculi using cell-lists and mpi. In: *Evolutionary Computation (CEC), 2010 IEEE Congress on*, 2010b, p. 1 –7.
- BONNEAU, R.; BAKER, D. Ab initio protein structure prediction: Progress and prospects. *Annual Review on Biophys and Biomolecular Structures*, v. 30, p. 173–189, 2001.
- BOSMAN, P.; THIERENS, D. *Continuous iterated density estimation evolutionary algorithms within the idea framework.* Relatório Técnico, Utrecht University, 2000.
- BOSMAN, P. A.; THIERENS, D. Advancing continuous ideas with mixture distributions and factorization selection metrics. In: *Proceedings of the Optimization by Building and Using Probabilistic Models OBUPM Workshop at the Genetic and Evolutionary Computation Conference GECCO-2001*, Morgan Kaufmann, 2001, p. 208–212.
- BRANDEN, C.; TOOZE, J. *Introduction to protein structure.* 2a ed. New York, NY, USA: Garland Publishing, Inc., 410, 1999.
- BRASIL, C. R. S.; DELBEM, A. C. B.; DA SILVA, F. L. B. Multiobjective evolutionary algorithm with many tables for purely ab initio protein structure prediction. *Journal of Computational Chemistry*, v. 34, n. 20, p. 1719–1734, 2013.
Disponível em <http://dx.doi.org/10.1002/jcc.23315>
- BROOKS, B.; BRUCCOLERI, R.; OLAFSON, B. States, dj; swaminathan, s.; karplus, m. *J. Comput. Chem*, v. 4, n. 2, p. 187–217, 1983.
- BUJNICKI, J. M. *Prediction of protein structures, functions, and interactions.* West Sussex: Wiley, 302 p., 2009.
- CASE, D.; DARDEN, T.; CHEATHAM, T.; III; SIMMERLING, C.; WANG, J.; DUKE, R.; LUO, R.; MERZ, K.; WANG, B.; PEARLMAN, D.; CROWLEY, M.; BROZELL, S.; TSUI, V.; GOHLKE, H.; MONGAN, J.; HORNAK, V.; CUI, G.; BEROZA, P.; SCHAFMEISTER, C.; CALDWELL, J.; ROSS, W.; KOLLMAN, P. *Amber 8.* <http://ambermd.org/>, university of California, San Francisco., 2004.

- CAVASOTTO, C. N.; PHATAK, S. S. Homology modeling in drug discovery: current trends and applications. *Drug Discovery Today*, v. 14, n. 13-14, p. 676–683, 2009.
Disponível em <http://www.sciencedirect.com/science/article/pii/S1359644609001469>
- CHEN, C.-H.; CHEN, Y.-P. Real-coded ecga for economic dispatch. In: *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*, GECCO '07, New York, NY, USA: ACM, 2007, p. 1920–1927 (GECCO '07,).
Disponível em <http://doi.acm.org/10.1145/1276958.1277343>
- CHEN, C.-H.; LIU, W.-N.; CHEN, Y.-P. Adaptive discretization for probabilistic model building genetic algorithms. In: *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, GECCO '06, New York, NY, USA: ACM, 2006, p. 1103–1110 (GECCO '06,).
Disponível em <http://doi.acm.org/10.1145/1143997.1144174>
- CHIANG, Y.-S.; GELFAND, T. I.; KISTER, A. E.; GELFAND, I. M. New classification of supersecondary structures of sandwich-like proteins uncovers strict patterns of strand assemblage. *Proteins: Structure, Function, and Bioinformatics*, v. 68, n. 4, p. 915–921, 2007.
Disponível em <http://dx.doi.org/10.1002/prot.21473>
- CHIVIAN, D.; ROBERTSON, T.; BONNEAU, R.; BAKER, D. Ab initio methods. *Structural Bioinformatics*, v. 27, p. 547–557, 2003.
- COOPER, G. F.; HERSKOVITS, E. A bayesian method for the induction of probabilistic networks from data. *Machine Learning*, v. 9, p. 309–347, 10.1007/BF00994110, 1992.
Disponível em <http://dx.doi.org/10.1007/BF00994110>
- CORNELL, W.; CIEPLAK, P.; BAYLY, C.; GOULD, I.; MERZ, K.; FERGUSON, D.; SPELLMEYER, D.; FOX, T.; CALDWELL, J.; KOLLMAN, P. A second generation force field for the simulation of proteins, nucleic acids, and organic molecules. *Journal of the American Chemical Society*, v. 117, n. 19, p. 5179–5197, 1995.
- CUI, Y.; CHEN, R. S.; WONG, W. H. Protein folding simulation with genetic algorithm and supersecondary structure constraints. *Proteins: Structure, Function, And Genetics*, v. 31, p. 247–257, 1998.
- DEB, K. *Multi-objective optimization using evolutionary algorithms*. New York: John Wiley & Sons, 2001.
- DEB, K.; PRATAP, A.; AGARWAL, S.; MEYARIVAN, T. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions On Evolutionary Computation*, v. 6, n. 2, p. 182–197, 2002.

- DEFELICIBUS, A. Sistema computacional via web integrando preditores de estruturas de proteínas, 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, 2014.
- DEVILLARD, N. iniparser: stand-alone ini parser library in ansi c. 2012.
Disponível em <http://ndevilla.free.fr/iniparser/>
- EINBECK, J.; HINDE, J. A note on npml estimation for exponential family regression models with unspecified dispersion parameter. *Austrian Journal of Statistics*, v. 35, p. 233–243, 2006.
- EISENBERG, D.; MCLACHLAN, A. D. Solvation energy in protein folding and binding. *Nature*, v. 319, n. 6050, p. 199–203, 1986.
- ExPASY PROTEOMICS SERVER 2009.
Disponível em <http://www.expasy.ch/sprot/> (Acessado em 04/02/2009)
- FRIEDMAN, H. L. Electrolyte solutions at equilibrium. *Annual Review of Physical Chemistry*, v. 32, n. 1, p. 179–204, 1981.
Disponível em <http://dx.doi.org/10.1146/annurev.pc.32.100181.001143>
- FRIESNER, R. A. *Computational methods for protein folding: Advances in chemical physics*. Wiley, 2002.
- GABRIEL, P. H. R.; MELO, V. V. D.; DELBEM, A. C. B. Algoritmos evolutivos e modelo hp para predição de estruturas de proteínas. *Sba Controle & Automação*, v. 23, n. 1, p. 25–37, 2012.
- GASPAR-CUNHA, A.; TAKAHASHI, R.; ANTUNES, C. H. *Manual de computação evolutiva e metaheurística*. Imprensa da Universidade de Coimbra, Coimbra, 2012.
- GAUDIO, A. C.; TAKAHATA, Y. Calculation of molecular surface area with numerical factors. *Computers & Chemistry*, v. 16, n. 4, p. 277 – 284, 1992.
Disponível em <http://www.sciencedirect.com/science/article/pii/0097848592800474>
- GEHAN, E. A generalized wilcoxon test for comparing arbitrarily singly-censored samples. *Biometrika*, v. 52, n. 1-2, p. 203–223, 1965.
- GIBAS, C.; JAMBECK, P. *Developing bioinformatics computer skills*. O'Reilly & Associates, Inc., 2001.
- GLASSER, O. Wilhelm conrad rontgen and the early history of the roentgen rays. *The American Journal of the Medical Sciences*, v. 187, n. 4, p. 566, 1934.

- GOECKS, J.; NEKRUTENKO, A.; TAYLOR, J.; TEAM, T. G. Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biology*, v. 11, n. 8, p. R86, 2010.
Disponível em <http://genomebiology.com/2010/11/8/R86>
- GOLDBERG, D. E. *The design of innovation: Lessons from and for competent genetic algorithms*. Norwell, USA: Kluwer Academic Publishers, 248 p., 2002.
- GRAMA, A.; GUPTA, A.; KARYPIS, G.; KUMAR, V. *Introduction to parallel computing*. Addison-Wesley, 2003.
- GREINER, W. *Quantum mechanics: An introduction*. Springer, 485 p., 2001.
- HARIK, G. Linkage learning via probabilistic modeling in the ecga. *Urbana*, v. 51, n. 61, p. 801, 1999.
- HARIK, G. R.; LOBO, F. G.; GOLDBERG, D. E. The compact genetic algorithm. *IEEE Transactions on Evolutionary Computation*, v. 3, n. 4, p. 287–297, 1998.
- HART, J. *Nonparametric smoothing and lack-of-fit tests*. Springer New York, 1997.
- HARTIGAN, J. A. *Clustering algorithms*. 99th ed. New York, NY, USA: John Wiley & Sons, Inc., 1975.
- HAUSCHILD, M.; PELIKAN, M. A survey of estimation of distribution algorithms. 2011.
- HECKERMAN, D.; GEIGER, D.; CHICKERING, D. M. Learning bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, v. 20, p. 197–243, 10.1007/BF00994016, 1995.
Disponível em <http://dx.doi.org/10.1007/BF00994016>
- HEINIG, M.; FRISHMAN, D. Stride: a web server for secondary structure assignment from known atomic coordinates of proteins. *Nucl. Acids Res.*, v. 32, n. W500-2, 2004.
- HERMANN, R. B. Theory of hydrophobic bonding. ii. correlation of hydrocarbon solubility in water with solvent cavity surface area. 1972.
- HERPER, M. The cost of creating a new drug now 5 billion, pushing big pharma to change. 2013.
Disponível em <http://www.forbes.com/sites/matthewherper/>
- HILBERT, M.; BÖHM, G.; JAENICKE, R. Structural relationships of homologous proteins as a fundamental principle in homology modeling. *Proteins: Structure, Function, and Genetics*, v. 17, n. 2, p. 138–151, 1993.
- HOLLAND, J. H. *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control and artificial intelligence*. University of Michigan Press, 1975.

- HOLLANDER, M.; WOLFE, D. *Nonparametric statistical methods.* Wiley Series in Probability and Statistics - Applied Probability and Statistics Section. Wiley, 1973.
Disponível em <http://books.google.es/books?id=ajxMAAAAMAAJ>
- HUANG, Y. J.; MAO, B.; ARAMINI, J. M.; MONTELIONE, G. T. Assessment of template-based protein structure predictions in casp10. *Proteins: Structure, Function, and Bioinformatics*, v. 82, p. 43–56, 2014.
Disponível em <http://dx.doi.org/10.1002/prot.24488>
- HUMPHREY, W.; DALKE, A.; SCHULTEN, K. VMD – Visual Molecular Dynamics. *Journal of Molecular Graphics*, v. 14, p. 33–38, 1996.
- JONES, D. T. Progress in protein structure prediction. *Current Opinion in Structural Biology*, v. 7, p. 377–387, 1997.
- JONES, D. T.; TAYLOR, W. R.; THORNTON, J. M. A new approach to protein fold recognition. *Letters to Nature*, v. 358, p. 86–89, 1992.
- JONES, J. E. On the determination of molecular fields. ii. from the equation of state of a gas. *Proceedings of the Royal Society of London. Series A*, v. 106, n. 738, p. 463–477, 1924.
Disponível em <http://rspa.royalsocietypublishing.org/content/106/738/463.short>
- JORDAN, I.; DIVISION, N. A. T. O. S. A. *Learning in graphical models: [proceedings of the nato advanced study institute... : Ettore mairola center, erice, italy, september 27-october 7, 1996]. Adaptive computation and machine learning.* Springer Netherlands, 1998.
Disponível em <http://books.google.com.br/books?id=7f61BBKdJ4EC>
- JORGENSEN, W.; TIRADO-RIVES, J. The opls [optimized potentials for liquid simulations] potential functions for proteins, energy minimizations for crystals of cyclic peptides and crambin. *Journal of the American Chemical Society*, v. 110, n. 6, p. 1657–1666, 1988.
- KAC, M.; KIEFER, J.; WOLFOWITZ, J. On tests of normality and other tests of goodness of fit based on distance methods. *The Annals of Mathematical Statistics*, v. 26, n. 2, p. 189–211, 1955.
Disponível em <http://dx.doi.org/10.1214/aoms/1177728538>
- KARCHIN, R.; DIEKHANS, M.; KELLY, L.; THOMAS, D.; PIEPER, U.; ESWAR, N.; HAUSSLER, D.; SALI, A. Ls-snp:large-scale annotation of coding non-synonymous snps based on multiple information sources. *Bioinformatics*, v. 15, n. 21, p. 2814–2820, 2005.
- KARPLUS, M.; LEVITT, M.; WARSHEL, A. The nobel prize in chemistry 2013. *The Royal Swedish Academy of Sciences*, 2013.

- Disponível em http://www.nobelprize.org/nobel_prizes/chemistry/laureates/2013/press.pdf
- KAUFMAN, L.; ROUSSEEUW, P. J. *Finding groups in data*, cap. Introduction John Wiley & Sons, Inc., p. 1–67, 2008.
- Disponível em <http://dx.doi.org/10.1002/9780470316801.ch1>
- KENDREW, J. C.; BODO, G.; DINTZIS, H. M.; PARRISH, R.; WYCKOFF, H.; PHILLIPS, D. A three-dimensional model of the myoglobin molecule obtained by x-ray analysis. *Nature*, v. 181, n. 4610, p. 662–666, 1958.
- KHIMASIA, M. M.; COVENEY, P. V. Protein structure prediction as a hard optimization problem: The genetic algorithm approach. *Molecular Simulation*, v. 19, n. 4, p. 205–226, 1997.
- Disponível em <http://dx.doi.org/10.1080/08927029708024151>
- KLAUDA, J. B.; WU, X.; PASTOR, R. W.; BROOKS, B. R. Long-range lennard-jones and electrostatic interactions in interfaces: Application of the isotropic periodic sum method. *The Journal of Physical Chemistry B*, v. 111, n. 17, p. 4393–4400, pMID: 17425357, 2007.
- Disponível em <http://dx.doi.org/10.1021/jp068767m>
- KOSLOFF, M.; KOLODNY, R. Sequence-similar, structure-dissimilar protein pairs in the pdb. *Proteins: Structure, Function, and Bioinformatics*, v. 71, n. 2, p. 891–902, 2008.
- Disponível em <http://dx.doi.org/10.1002/prot.21770>
- KRANE, D. E.; RAYMER, M. L. *Fundamental concepts of bioinformatics*. Pearson Education, 2003.
- LAGÜE, P.; PASTOR, R. W.; BROOKS, B. R. Pressure-based long-range correction for lennard-jones interactions in molecular dynamics simulations. *The Journal of Physical Chemistry B*, v. 108, n. 1, p. 363–368, 2004.
- Disponível em <http://dx.doi.org/10.1021/jp030458y>
- LARRANAGA, P.; ETXEVERRIA, R.; LOZANO, J.; PENA, J.; PE, J.; ET AL. Optimization by learning and simulation of bayesian and gaussian networks. 1999.
- LARRANAGA, P.; LOZANO, J. A. *Estimation of distribution algorithms: A new tool for evolutionary computation*, v. 2. New York: Springer, 382 p., 2002.
- LASKEY, R. A.; HONDA, B. M.; MILLS, A. D.; FINCH, J. T. Nucleosomes are assembled by an acidic protein which binds histones and transfers them to dna. *Nature*, v. 275, n. 15, p. 416–420, 1978.
- Disponível em <http://www.fasebj.org/content/9/15/1559.abstract>

- LAU, K. F.; DILL, K. A. A lattice statistical mechanics model of the conformational and sequence spaces of proteins. *Macromolecules*, v. 22, n. 10, p. 3986–3997, 1989.
Disponível em <http://dx.doi.org/10.1021/ma00200a030>
- LEE, J.; WU, S.; ZHANG, Y. Ab initio protein structure prediction. In: RIGDEN, D. J., ed. *From Protein Structure to Function with Bioinformatics*, Springer Netherlands, p. 3–25, 10.1007/978-1-4020-9058-51, 2009.
Disponível em <http://dx.doi.org/10.1007/978-1-4020-9058-51>
- LEWIS, P. O. A genetic algorithm for maximum-likelihood phylogeny inference using nucleotide sequence data. *Molecular Biology and Evolution*, v. 15, n. 3, p. 277–283, 1998.
Disponível em <http://mbe.oxfordjournals.org/content/15/3/277.abstract>
- LI, X.; YIN, M. Application of differential evolution algorithm on self-potential data. *PloS one*, v. 7, n. 12, p. e51199, 2012.
- DE LIMA, T.; GABRIEL, P.; DELBEM, A.; FACCIOLI, R.; DA SILVA, I. Evolutionary algorithm to ab initio protein structure prediction with hydrophobic interactions. In: *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, IEEE, 2008, p. 612–619.
- LIMA, T. W. *Algortimos evolutivos para predição de estruturas de proteínas*. Dissertação de Mestrado, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, SP, 2006.
- LIMA, T. W.; DELBEM, A. C. B. *Estruturas de dados eficientes para algoritmos evolutivos aplicados ao projeto de redes*. Relatório Técnico 301, Instituto de Ciências Matemáticas e de Computação - USP, 2007.
- DE LIMA, T. W.; FACCIOLI, R. A.; GABRIEL, P. H. R.; DELBEM, A. C. B.; DA SILVA, I. N. Evolutionary approach to protein structure prediction with hydrophobic interactions. In: *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, GECCO '07, New York, NY, USA: ACM, 2007, p. 425–425 (GECCO '07,).
Disponível em <http://doi.acm.org/10.1145/1276958.1277045>
- LODISH, H.; BERK, A.; MATSUDAIRA, P.; KAISER, C. A.; KRIEGER, M.; SCOTT, M. P.; ZIPURSKY, L.; DARNELL, J. *Molecular cell biology 5th edition*. W. H. Freeman, 2003.
- M. GALASSI ET AL Gnu scientific library reference manual (3rd ed.). ISBN 0954612078, 2009.
Disponível em <http://www.gnu.org/software/gsl/>
- MACKERELL JR, A. Empirical force fields for biological macromolecules: overview and issues. *Journal of computational chemistry*, v. 25, n. 13, p. 1584–1604, 2004.

- MACKERELL JR, A.; BANAVALI, N.; FOLOPPE, N. Development and current status of the charmm force field for nucleic acids. *Biopolymers*, v. 56, n. 4, p. 257–265, 2000.
- MANSOUR, N.; KANJI, F.; KHACHFE, H. Evolutionary algorithm for protein structure prediction. In: *Natural Computation (ICNC), 2010 Sixth International Conference on*, 2010, p. 3974 –3977.
- MARDIA, K. V.; JUPP, P. E. *Basic concepts and models* John Wiley & Sons, Inc., p. 25–56, 2008.
Disponível em <http://dx.doi.org/10.1002/9780470316979.ch3>
- MARDIA, K. V.; ZEMROCH, P. J. The von mises distribution function. *Journal of the Royal Statistical Society*, v. 24, n. 2, p. 268–272, 1975.
- MARTÍ-RENOM, M.; STUART, A.; FISER, A.; SÁNCHEZ, R.; MELO, F.; ŠALI, A. Comparative protein structure modeling of genes and genomes. *Annual review of biophysics and biomolecular structure*, v. 29, n. 1, p. 291–325, 2000.
- MARZZOCO, A.; TORRES, B. B. *Bioquímica básica*. Guanabara Koogan, 1999.
- MCLACHLAN, G.; PEEL, D. *Finite mixture models*. Wiley series in probability and statistics: Applied probability and statistics. Hoboken, NJ, USA: Wiley, 456 p., 2004.
Disponível em http://books.google.com.br/books?id=c2_fAox0DQoC
- METROPOLIS, N.; ULAM, S. The monte carlo method. *Journal of the American Statistical Association*, v. 44, n. 247, p. pp. 335–341, 1949.
Disponível em <http://www.jstor.org/stable/2280232>
- MIJAJLOVIC, M.; BIGGS, M.; DJURDJEVIC, D. On potential energy models for ea-based ab initio protein structure prediction. *Evolutionary Computation*, v. 18, n. 2, p. 255–275, 2010.
- MOON, T. The expectation-maximization algorithm. *Signal Processing Magazine, IEEE*, v. 13, n. 6, p. 47–60, 1996.
- MUEHLENBEIN, H.; MAHNIG, T. Convergence theory and applications of the factorized distribution algorithm. *Journal of Computing and Information Technology*, v. 7, 1999.
- MUEHLENBEIN, H.; PAASS, G. From recombination of genes to the estimation of distributions i. binary parameters. Springer-Verlag, 1996, p. 178–187.
- MÜHLENBEIN, H. The equation for response to selection and its use for prediction. *Evol. Comput.*, v. 5, n. 3, p. 303–346, 1997.
Disponível em <http://dx.doi.org/10.1162/evco.1997.5.3.303>
- NARAYANAN, K.; LAKSHMIKUTTY, B. *Stoichiometry and process calculations*. PHI Learning, 2006.
Disponível em <http://books.google.com.br/books?id=f-Vj0fiJodYC>

- NAUDTS, B.; NAUDTS, J. The effect of spin-flip symmetry on the performance of the simple ga. In: EIBEN, A.; BÄCK, T.; SCHOENAUER, M.; SCHWEFEL, H.-P., eds. *Parallel Problem Solving from Nature - PPSN V*, v. 1498 de *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, p. 67–76, 10.1007/BFb0056850, 1998.
Disponível em <http://dx.doi.org/10.1007/BFb0056850>
- NELSON, D. L.; COX, M. M. *Lehninger principles of biochemistry fourth edition.* w. H. Freeman, 2004.
- OGAWA, H.; TOYOSHIMA, C. Homology modeling of the cation binding sites of na+k+-atpase. *Proceedings of the National Academy of Sciences*, v. 99, n. 25, p. 15977–15982, 2002.
Disponível em <http://www.pnas.org/content/99/25/15977.abstract>
- OLIVER, J. Decision graphs-an extension of decision trees. 1993.
- ORENGO, C.; JONES, D.; THORNTON, J. *Bioinformatics: Genes, protein and computers.* BIOS Scientific Publishers, 2003.
- PEARSON, K. Note on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*, v. 58, n. 347-352, p. 240–242, 1895.
Disponível em <http://rspl.royalsocietypublishing.org/content/58/347-352/240.short>
- PEARSON, K. The random walk. *Nature*, v. 72, p. 294; 318; 342, 1905.
- PELIKAN, M. *Hierarchical bayesian optimization algorithm: Toward a new generation of evolutionary algorithms.* Springer, 184 p., 2005.
- PELIKAN, M.; GOLDBERG, D. E.; ; CANTU-PAZ, E. Boa: The bayesian optimization algorithm. *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-99*, 1999.
- PELIKAN, M.; GOLDBERG, D. E. The hierarchical bayesian optimization algorithm. 2003.
Disponível em <http://www.illigal.uiuc.edu/hboa/>
- PELIKAN, M.; GOLDBERG, D. E.; TSUTSUI, S. Getting the best of both worlds: Discrete and continuous genetic and evolutionary algorithms in concert. *Information Sciences*, v. 156, n. 3-4, p. 147 – 171, evolutionary Computation, 2003.
Disponível em <http://www.sciencedirect.com/science/article/pii/S0020025503001749>
- PELIKAN, M.; MUEHLENBEIN, H. The bivariate marginal distribution algorithm. In: ROY, R.; FURUHASHI, T.; CHAWDHRY, P., eds. *Advances in Soft Computing*, Springer London, p. 521–535, 1999.
Disponível em http://dx.doi.org/10.1007/978-1-4471-0819-1_39

PELIKAN, M.; PELIKAN, M.; GOLDBERG, D. E.; GOLDBERG, D. E. Escaping hierarchical traps with competent genetic algorithms. In: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO2001)*, Morgan Kaufmann, 2001, p. 511–518.

PELIKAN, M.; SASTRY, K.; GOLDBERG, D. E. Multiobjective hboa, clustering, and scalability. In: *Proceedings of the 2005 conference on Genetic and evolutionary computation*, GECCO '05, New York, NY, USA: ACM, 2005, p. 663–670 (GECCO '05,).
Disponível em <http://doi.acm.org/10.1145/1068009.1068122>

PENG, J.; XU, J. Low-homology protein threading. *Bioinformatics*, v. 26, n. 12, p. i294–i300, 2010.
Disponível em <http://bioinformatics.oxfordjournals.org/content/26/12/i294.abstract>

PIEPER, U.; ESWAR, N.; DAVIS, F. P.; BRABERG, H.; MADHUSUDHAN, M. S.; ROSSI, A.; MARTI-RENOM, M.; KARCHIN, R.; WEBB, B. M.; ERAMIAN, D.; SHEN, M.-Y.; KELLY, L.; MELO, F.; SALI, A. Modbase: a database of annotated comparative protein structure models and associated resources. *Nucleic Acids Research*, v. 34, n. suppl 1, p. D291–D295, 2006.

Disponível em http://nar.oxfordjournals.org/content/34/suppl_1/D291.abstract

QUINTINO, G. O. *Otimização do cálculo da energia eletrostática em algoritmos genéticos para predição de estruturas de proteínas (em andamento)*. Dissertação de Mestrado, Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo, 2014.

QUIÑONERO, E. J. C.; PÉREZ-SÁNCHEZ, H.; CECILIA, J.; GARCÍA, J. Murcia: Fast parallel solvent accessible surface area calculation on gpus and application to drug discovery and molecular visualization. 2011, p. 52–55.

R CORE TEAM *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria, 2014.

Disponível em <http://www.R-project.org/>

RABI, I.; ZACHARIAS, J.; MILLMAN, S.; KUSCH, P. A new method of measuring nuclear magnetic moment. *Phys. Rev.*, v. 53, p. 318–318, 1938.

Disponível em <http://link.aps.org/doi/10.1103/PhysRev.53.318>

RAMACHANDRAN, G.; RAMAKRISHNAN, C.; SASISEKHARAN, V. Stereochemistry of polypeptide chain configurations. *Journal of Molecular Biology*, v. 7, n. 1, p. 95 – 99, 1963.

Disponível em <http://www.sciencedirect.com/science/article/pii/S0022283663800236>

- RCSB, P. Looking at structures: Methods for determining atomic structures. 2014.
Disponível em http://www.rcsb.org/pdb/101/static101.do?p=education_discussion/Looking-at-Structures/methods.html
- RIPLEY, B.; CORPORATION, E. *Stochastic simulation*, v. 21. Wiley Online Library, 1987.
- RIPLEY, B. D.; VENABLES, W. N. *Modern applied statistics with s-plus*. Springer-Verlag New York, NY, 1994.
- ROCCA, P.; OLIVERI, G.; MASSA, A. Differential evolution as applied to electromagnetics. *Antennas and Propagation Magazine, IEEE*, v. 53, n. 1, p. 38–49, 2011.
- ROHL, C. A.; STRAUSS, C. E.; MISURA, K. M.; BAKER, D. Protein structure prediction using rosetta. In: BRAND, L.; JOHNSON, M. L., eds. *Numerical Computer Methods, Part D*, v. 383 de *Methods in Enzymology*, Academic Press, p. 66 – 93, 2004.
Disponível em <http://www.sciencedirect.com/science/article/pii/S0076687904830040>
- ROSENBLATT, M. Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, p. 832–837, 1956.
- RUDLOF, S.; KÖPPEN, M. Stochastic hill climbing with learning by vectors of normal distributions. 1996, p. 60–70.
- RUSKA, E. The development of the electron microscope and of electron microscopy. *Nobel lecture*, v. 20, 1986.
- RUSSELL, S. J.; NORVIG, P. *Artificial intelligence: A modern approach*. 2 ed. Pearson Education, 2003.
- SAITOU, N.; NEI, M. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, v. 4, n. 4, p. 406–425, 1987.
Disponível em <http://mbe.oxfordjournals.org/content/4/4/406.abstract>
- SALI, A.; BLUNDELL, T. L. Comparative protein modelling by satisfaction of spatial restraints. *J. Mol. Biol.*, , n. 234, p. 779–815, 1993.
- SANTANA, R.; LARRAÑAGA, P.; LOZANO, J. Protein folding in 2-dimensional lattices with estimation of distribution algorithms. In: BARREIRO, J.; MARTÍN-SÁNCHEZ, F.; MAJO, V.; SANZ, F., eds. *Biological and Medical Data Analysis*, v. 3337 de *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, p. 388–398, 2004.
Disponível em http://dx.doi.org/10.1007/978-3-540-30547-7_39

- SANTANA, R.; LARRANAGA, P.; LOZANO, J. A. Protein folding in simplified models with estimation of distribution algorithms. *IEEE Transactions On Evolutionary Computation*, v. 12, n. 4, p. 418–438, 2008.
- SAWILOWSKY, S.; FAHOOME, G. *Statistics through monte carlo simulation with fortran*. JMASM, 2002.
Disponível em <http://books.google.com.br/books?id=AUTCAwAACAAJ>
- SCHMITT, L. M. Theory of genetic algorithms. *Theoretical Computer Science*, v. 259, n. 1-2, p. 1–61, 2001.
Disponível em <http://www.sciencedirect.com/science/article/pii/S0304397500004060>
- SCHMOLZE, D.; STANDLEY, C.; FOGARTY, K.; FISCHER, A. Advances in microscopy techniques. *Archives of Pathology & Laboratory Medicine*, v. 135, n. 2, p. 255–263, 2011.
- SCHRÖDINGER, LLC The PyMOL molecular graphics system, version 1.3r1, 2010.
- SCHWARZ, G. Estimating the dimension of a model. *The annals of statistics*, p. 461–464, 1978.
- SCOTT, W.; HÜNENBERGER, P.; TIRONI, I.; MARK, A.; BILLETER, S.; FENNEN, J.; TORDA, A.; HUBER, T.; KRÜGER, P.; VAN GUNSTEREN, W. The gromos biomolecular simulation program package. *The Journal of Physical Chemistry A*, v. 103, n. 19, p. 3596–3607, 1999.
- SEBAG, M.; DU COULOMBIER, A. Extending population-based incremental learning to continuous search spaces. In: EIBEN, A.; BÄCK, T.; SCHOENAUER, M.; SCHWEFEL, H.-P., eds. *Parallel Problem Solving from Nature - PPSN V*, v. 1498 de *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, p. 418–427, 10.1007/BFb0056884, 1998.
Disponível em <http://dx.doi.org/10.1007/BFb0056884>
- SETUBAL, J.; MEIDANIS, J. *Introduction to computational molecular biology*. PWS Publishing Company, 1997.
- SHAKHNOVICH, E.; FARZTDINOV, G.; GUTIN, A. M.; KARPLUS, M. Protein folding bottlenecks: A lattice monte carlo simulation. *Phys. Rev. Lett.*, v. 67, p. 1665–1668, 1991.
Disponível em <http://link.aps.org/doi/10.1103/PhysRevLett.67.1665>
- SHAO, M.; WANG, S.; WANG, C.; YUAN, X.; LI, S.; ZHENG, W.; BU, D. Incorporating ab initio energy into threading approaches for protein structure prediction. *BMC Bioinformatics*, v. 12, n. Suppl 1, p. S54, 2011.
Disponível em <http://www.biomedcentral.com/1471-2105/12/S1/S54>
- SILVERMAN, B. W. *Density estimation for statistics and data analysis*, v. 26. CRC press, 1986.

- SLABINSKI, L.; JAROSZEWSKI, L.; RODRIGUES, A. P.; RYCHLEWSKI, L.; WILSON, I. A.; LESLEY, S. A.; GODZIK, A. The challenge of protein structure determination—lessons from structural genomics. *Protein Sci.*, v. 16, n. 11, p. 2472–2482, 2007.
- SÁNCHEZ, R.; SALI, A. Large-scale protein structure modeling of the *saccharomyces cerevisiae* genome. *Proceedings of the National Academy of Sciences*, v. 95, n. 23, p. 13597–13602, 1998. Disponível em <http://www.pnas.org/content/95/23/13597.abstract>
- SOBHA, K.; KANAKARAJU, C.; YADAV, K. S. K. Is protein structure prediction still an enigma? *African Journal of Biotechnology*, v. 7, n. 25, p. 4687–4693, 2008. Disponível em <http://www.academicjournals.org/AJB>
- SRINIVAS, N.; DEB, K. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, v. 2, n. 3, p. 221–248, 1994.
- STORN, R.; PRICE, K. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, v. 11, p. 341–359, 10.1023/A:1008202821328, 1997. Disponível em <http://dx.doi.org/10.1023/A:1008202821328>
- SUGANTHAN, P.; HANSEN, N.; LIANG, J.; DEB, K.; CHEN, Y.; AUGER, A.; TIWARI, S. Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization. *KanGAL Report*, 2005.
- TAI, C.-H.; BAI, H.; TAYLOR, T. J.; LEE, B. Assessment of template-free modeling in casp10 and roll. *Proteins: Structure, Function, and Bioinformatics*, v. 82, p. 57–83, 2014. Disponível em <http://dx.doi.org/10.1002/prot.24470>
- TANTAR, A.-A.; MELAB, N.; TALBI, E.-G. A grid-based hybrid hierarchical genetic algorithm for protein structure prediction. In: *Parallel and Distributed Computational Intelligence*, p. 291–319, 2010.
- TOP500 SUPERCOMPUTER SITES 2014. Disponível em <http://www.top500.org/> (Acessado em 04/05/2014)
- TRUDEAU, R. *Introduction to graph theory*. Dover Books on Mathematics. Dover Publications, 2013. Disponível em <http://books.google.com.br/books?id=eRLEAgAAQBAJ>
- TSUTSUI, S.; PELIKAN, M.; GOLDBERG, D. Evolutionary algorithm using marginal histogram models in continuous domain. *IlliGAL Report*, v. 2001019, 2001.
- TUFFERY, P. Rotamer library. 2003.

- UNGER, R. The genetic algorithm approach to protein structure prediction. *Structure and Bonding*, v. 110, p. 153–175, 2004.
- UNGER, R.; MOULT, J. Genetic algorithms for protein folding simulations. *Journal of Molecular Biology*, v. 231, n. 1, p. 75 – 81, 1993.
Disponível em <http://www.sciencedirect.com/science/article/pii/S0022283683712581>
- VAN SICKEL, J.; LEE, K.; HEO, J. Differential evolution and its applications to power plant control. In: *Intelligent Systems Applications to Power Systems, 2007. ISAP 2007. International Conference on*, 2007, p. 1–6.
- VARGAS, D. V.; DELBEM, A. C. B.; DE MELO, V. V. Algoritmo filo-genético. 2010.
Disponível em <http://www.dep.uminho.pt/escolaeads2010/posters.html>
- WANG, H. X.; LUO, B.; BING ZHANG, Q.; WEI, S. Estimation for the number of components in a mixture model using stepwise split-and-merge {EM} algorithm. *Pattern Recognition Letters*, v. 25, n. 16, p. 1799 – 1809, 2004.
Disponível em <http://www.sciencedirect.com/science/article/pii/S0167865504001734>
- WARSHEL, A.; LEVITT, M. Theoretical studies of enzymic reactions: Dielectric, electrostatic and steric stabilization of the carbonium ion in the reaction of lysozyme. *Journal of Molecular Biology*, v. 103, n. 2, p. 227 – 249, 1976.
Disponível em <http://www.sciencedirect.com/science/article/pii/0022283676903119>
- WEBSTER, D. M. *Protein structure prediction: Methods and protocols*. Humana Press, 415 p., 2000.
- WOEGINGER, G. J. Exact algorithms for np-hard problems: A survey. *Lecture Notes in Computer Science*, v. 2570, p. 185–207, 2003.
- WOLFF, K.; VENDRUSCOLO, M.; PORTO, M. Efficient identification of near-native conformations in ab initio protein structure prediction using structural profiles. *Proteins: Structure, Function, and Bioinformatics*, v. 78, n. 2, p. 249–258, 2010.
Disponível em <http://dx.doi.org/10.1002/prot.22533>
- WONG, K.-C.; LEUNG, K.-S.; WONG, M.-H. Protein structure prediction on a lattice model via multimodal optimization techniques. In: *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, GECCO '10, New York, NY, USA: ACM, 2010, p. 155–162 (GECCO '10,).
Disponível em <http://doi.acm.org/10.1145/1830483.1830513>

- WU, S.; SKOLNICK, J.; ZHANG, Y. Ab initio modeling of small proteins by iterative tasser simulations. *BMC Biology*, v. 5, n. 1, p. 17, 2007.
Disponível em <http://www.biomedcentral.com/1741-7007/5/17>
- XU, D.; ZHANG, Y. Ab initio protein structure assembly using continuous structure fragments and optimized knowledge-based force field. *Proteins: Structure, Function, and Bioinformatics*, 2012a.
- XU, D.; ZHANG, Y. Ab initio protein structure assembly using continuous structure fragments and optimized knowledge-based force field. *Proteins*, v. 80, p. 1715–1735, 2012b.
- YANNAKAKIS, M. On the approximation of maximum satisfiability. In: *Proceedings of the third annual ACM-SIAM symposium on Discrete algorithms*, SODA '92, Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1992, p. 1–9 (SODA '92,).
Disponível em <http://dl.acm.org/citation.cfm?id=139404.139406>
- YU, T.-L.; SANTARELLI, S.; GOLDBERG, D. Military antenna design using a simple genetic algorithm and hboa. In: PELIKAN, M.; SASTRY, K.; CANTÚPAZ, E., eds. *Scalable Optimization via Probabilistic Modeling*, v. 33 de *Studies in Computational Intelligence*, Springer Berlin Heidelberg, p. 275–289, 2006.
Disponível em http://dx.doi.org/10.1007/978-3-540-34954-9_12
- ZAKI, M. J.; BYSTROFF, C. *Protein structure prediction 2th edition*. Troy, USA: Humana Press, 352 p., 2008.
- ZHANG, Q.; WANG, J.; GUERRERO, G. D.; CECILIA, J. M.; GARCÍA, J. M.; LI, Y.; PÉREZ-SÁNCHEZ, H.; HOU, T. Accelerated conformational entropy calculations using graphic processing units. *Journal of Chemical Information and Modeling*, v. 53, n. 8, p. 2057–2064, 2013.
Disponível em <http://pubs.acs.org/doi/abs/10.1021/ci400263t>
- ZHANG, Y. Progress and challenges in protein structure prediction. *Current Opinion in Structural Biology*, v. 18, n. 3, p. 342 – 348, nucleic acids / Sequences and topology, 2008.
Disponível em <http://www.sciencedirect.com/science/>
- ZHOU, A.; QU, B.-Y.; LI, H.; ZHAO, S.-Z.; SUGANTHAN, P. N.; ZHANG, Q. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, v. 1, n. 1, p. 32 – 49, 2011a.
Disponível em <http://www.sciencedirect.com/science/article/pii/S2210650211000058>
- ZHOU, R. Free energy landscape of protein folding in water: explicit vs. implicit solvent. *Proteins-Structure Function and Genetics*, v. 53, n. 2, p. 148–161, 2003.

- ZHOU, Y.; DUAN, Y.; YANG, Y.; FARAGGI, E.; LEI, H. Trends in template/fragment-free protein structure prediction. *Theoretical Chemistry Accounts: Theory, Computation, and Modeling (Theoretica Chimica Acta)*, v. 128, p. 3–16, 10.1007/s00214-010-0799-2, 2011b.
Disponível em <http://dx.doi.org/10.1007/s00214-010-0799-2>