Universidade Estadual do Ceará

Francisco Flávio de Assunção Rabelo

HEURÍSTICA HÍBRIDA PARA O PROBLEMA DA PREDIÇÃO DA ESTRUTURA DE PROTEÍNAS UTILIZANDO O MODELO HIDROFÓBICO-POLAR

Fortaleza - Ceará

2013

Universidade Estadual do Ceará

Francisco Flávio de Assunção Rabelo

HEURÍSTICA HÍBRIDA PARA O PROBLEMA DA PREDIÇÃO DA ESTRUTURA DE PROTEÍNAS UTILIZANDO O MODELO HIDROFÓBICO-POLAR

Dissertação apresentada ao Curso de Mestrado Acadêmico em Ciência da Computação do Centro de Ciências e Tecnologia da Universidade Estadual do Ceará, como requisito parcial para a obtenção do grau de mestre em Ciência da Computação. Área de concentração: Sistemas de Computação.

Orientador: Prof. Dr. Gerardo Valdísio Rodrigues Viana

Fortaleza - Ceará

2013

Dados Internacionais de Catalogação na Publicação
Universidade Estadual do Ceará
Biblioteca Central Prof. Antônio Martins Filho
Bibliotecário(a) Responsável - Thelma Marylanda Silva de Melo CRB-3 / 623

R114h Rabelo, Francisco Flávio de Assunção

Heurística híbrida para o problema da predição da estrutura de proteínas utilizando o modelo Hidrofóbico-Polar / Francisco Flávio de Assunção Rabelo. — 2013.

CD-ROM. 61f :il. (algumas color.); 4 ¾ pol.

"CD-ROM contendo o arquivo no formato PDF do trabalho acadêmico, acondicionado em caixa de DVD Slim (19cm x 14cm x 7mm)".

Dissertação (Mestrado) – Universidade Estadual do Ceará, Centro de Ciências e Tecnologia, Curso de Mestrado Acadêmico em Ciência da Computação, Fortaleza, 2013.

Área de Concentração: Sistemas de Computação.

Orientação: Prof. Dr. Gerardo Valdísio Rodrigues Viana.

1. Biologia Computacional 2. Heurística 3. Proteínas .I. Título.

CDD: 001.4

FRANCISCO FLÁVIO DE ASSUNÇÃO RABELO

HEURÍSTICA HÍBRIDA PARA O PROBLEMA DA PREDIÇÃO DA ESTRUTURA DE PROTEÍNAS UTILIZANDO O MODELO HIDROFÓBICO-POLAR

Dissertação apresentada ao Curso de Mestrado Acadêmico em Ciência da Computação do Centro de Ciências e Tecnologia da Universidade Estadual do Ceará, como requisito parcial para a obtenção do grau de mestre em Ciência da Computação. Área de concentração: Sistemas de Computação.

Aprovada em: 29/07/2013

Banca Examinadora

Prof. Dr. Gerardo Valdísio Rodrigues Viana (Orientador)
Universidade Estadual do Ceará – UECE

Prof. Dr. Gustavo Augusto Lima de Campos
Universidade Estadual do Ceará – UECE

Prof. Dr. José Hélio Costa
Universidade Federal do Ceará – UFC

Agradecimentos

Gostaria de agradecer a todos aqueles que acreditaram e contribuíram de alguma forma para a realização deste trabalho.

Em especial, à minha esposa Marília Alves da Silva, pelo apoio e paciência nos momentos mais difíceis e ao amigo Paulo Henrique Silva por revisar este trabalho.

Ao meu orientador professor Gerardo Valdísio Rodrigues Viana pela orientação, atenção, paciência e principalmente por ter me dado a oportunidade de cursar o mestrado.

Ao professor Gustavo Augusto Lima de Campos pelos valiosos conselhos.

Aos amigos, colegas e professores do MACC que fizeram parte dessa empreitada.

RESUMO

O problema da predição da estrutura de proteínas utilizando o modelo Hidrofóbico-Polar pode ser definido como: dada uma sequência de aminoácidos hidrofóbicos e polares, encontrar uma conformação com o maior número de contatos hidrofóbicos entre vizinhos topológicos. Esse problema pertence à classe de problemas NP-completo e diversas abordagens têm sido propostas. Neste trabalho, apresentamos uma heurística GRASP híbrida para o problema em questão. O problema é abordado como um problema de otimização combinatória e uma formulação é proposta. A heurística foi testada com várias instâncias de referências e os resultados mostram que o método proposto apresenta desempenho semelhante ao de outros métodos disponíveis na literatura.

Palavras-chaves: Biologia Computacional. Heurística. Proteínas.

ABSTRACT

The protein structure prediction problem using Hydrophobic-Polar model can be defined as follows: given a sequence of hydrophobic and polar amino acids, find a conformation with the largest number of hydrophobic contacts between topological neighbors. This problem belongs to the class of NP-complete problems and several approaches have been proposed. In this paper, we present a hybrid GRASP heuristic for the problem concerned. The problem is approached as a combinatorial optimization problem and a formulation is proposed. The heuristic was tested with multiple instances of references and the results show that the proposed method has similar performance to the other methods available in the literature.

Key-words: Computacional Biology. Heuristic. Proteins.

SUMÁRIO

Lista de figuras **11**
Lista de quadros e tabelas **12**
1 Introdução **14**
2 Introdução às proteínas **16**
2.1 Aminoácidos 16
2.2 Peptídeos e proteínas 18
2.3 Estrutura das proteínas 18
2.3.1 Planaridade e rigidez da ligação peptídica 20
2.3.2 Estrutura primária 21
2.3.3 Estrutura secundária 22
2.3.4 Estrutura terciária e quaternária 23
2.4 Estabilidade estrutural 24
2.5 Enovelamento 24
2.6 Determinação da estrutura tridimensional de proteínas 25
2.6.1 Cristalografia por difração de raios X 25
2.6.2 Espectroscopia por ressonância magnética nuclear 26
3 Conceitos básicos sobre otimização **27**
3.1 Problemas de otimização 27
3.2 Vizinhança 28
3.3 Soluções ótimas 28
3.4 Classes de complexidade 29
3.4.1 A classe P 29
3.4.2 A classe NP 29
3.4.3 Problemas tratáveis e intratáveis 29
3.4.4 Problema NP-difícil e NP-completo 30
3.5 Métodos de otimização 30
3.5.1 Métodos heurísticos 31
3.6 *Greedy Randomized Adaptive Search Procedure* 32
3.7 *Variable Neighborhood Descent* 33
4 Predição da estrutura tridimensional de proteínas **35**
4.1 Modelagem por homologia 35
4.2 Reconhecimento do enovelamento (*Threading*) 36
4.3 Modelagem livre 36
4.4 Modelos protéicos em grade 37
4.4.1 O modelo Hidrofóbico-Polar 38
4.4.2 Problema da predição da estrutura protéica no modelo HP 39

4.4.3 Trabalhos relacionados . . . 39
5 A heurística proposta . . . 41
5.1 Representação da solução . . . 41
5.2 Função de energia . . . 42
5.3 Formulação do problema . . . 43
5.4 O procedimento principal . . . 43
5.5 Fase de construção . . . 43
5.6 Fase de busca local . . . 45
5.6.1 *Pull moves* . . . 46
5.6.2 Determinando as posições P_L e P_Q . . . 47
5.6.3 Estrutura de vizinhança . . . 48
5.7 Experimentos computacionais . . . 50
6 Resultados e discussões . . . 53
6.1 Modelo HP em grade bidimensional quadrangular . . . 53
6.2 Modelo HP em grade tridimensional cúbica . . . 55
7 Conclusões . . . 58
Referências Bibliográficas . . . 59

Lista de abreviaturas e/ou símbolos

DNA	Ácido dexorribonucléico
RNA	Ácido ribonucléico
COOH	Grupo carboxila
NH_2	Grupo amino
C_α	Carbono alfa
R	Cadeia lateral
H	Átomo de hidrogênio
OH	Hidroxila
u	Unidade de massa atômica
α	Letra grega alfa
β	Letra grega beta
G	Energia livre de Gibbs
C	Átomo de carbono
N	Átomo de nitrogênio
ϕ	Letra grega phi
ψ	Letra grega psi
ω	Letra grega omega
Å	Angstron
RMN	Ressonância magnética nuclear
F	Átomo de flúor
P	Átomo de fósforo
$\mathbb{R}$	Conjunto dos números reais
$\mathbb{Z}$	Conjunto dos números inteiros
$\mathbb{B}$	Conjunto dos números binários

Ω Letra grega Ômega

PO Pesquisa operacional

GRASP *Greedy Randomized Adaptive Search Procedure*

LRC Lista restrita de candidatos

VND *Variable Neighborhood Descent*

PSP *Protein Structure Prediction*

CASP *Critical Assessment of Techniques for Structure Prediction*

PDB *Protein Data Bank*

HP Hidrofóbico-Polar

AG Algoritmo genético

AE Algoritmo evolucionário

HZ *Hydrophobic Zipper*

CI *Contact Interations*

CG *Core-directed chain Growth*

MCE Monte Carlo evolucionário

BT Busca Tabu

ACO *Ant Colony Optimization*

REMC *Replica Exchange Monte Carlo*

ϵ Letra grega épsilon

Lista de figuras

Figura 1 – Estrutura comum a todos os alfa-aminoácidos exceto a prolina. O grupo R ou cadeia lateral é diferente em cada aminoácido. Fonte: Lehninger, Nelson e Cox (2008) . . . 16
Figura 2 – Os vinte aminoácidos mais comuns encontrados nas proteínas. As cadeias laterais estão destacadas. Fonte: Lehninger, Nelson e Cox (2008) 17
Figura 3 – Formação da ligação peptídica. Fonte: Lehninger, Nelson e Cox (2008). 18
Figura 4 – Os níveis estruturais de uma proteína. Fonte: Marieb (2001). 19
Figura 5 – Planaridade e rigidez da ligação peptídica. Fonte: deconhecida. 20
Figura 6 – Gráfico de Ramachandram para o resíduo L-alanina. Fonte: Lehninger, Nelson e Cox (2008). 21
Figura 7 – A estrutura de uma alfa-hélice. Fonte: Alberts et al. (2007). 22
Figura 8 – A estrutura de uma folha-beta. Fonte: Alberts et al. (2007) 23
Figura 9 – Exemplo de uma conformação. As esferas pretas representam os resíduos hidrofóbicos (H) e as brancas representam os resíduos polares (P). . . . 38
Figura 10 – Exemplo de uma conformação para a sequência `HPPHPPHPPH`. O primeiro resíduo está numerado com 1. 42
Figura 11 – Exemplo de conformações diferentes, mas que apresentam o mesmo valor energético. Fonte: Krasnogor et al. (1999) 42
Figura 12 – Exemplo de solução parcial onde todas as posições adjacentes (horizontal e vertical) ao resíduo $i-1$ estão ocupadas, impedindo o crescimento da solução. 45
Figura 13 – Exemplo movimentos para resíduos não finais. 46
Figura 14 – Exemplo de movimentos para resíduos finais. 47
Figura 15 – As posições P_L e P_Q para um resíduo i em uma grade bidimensional quadrangular. 48
Figura 16 – As posições P_L e P_Q para um resíduo i em uma grade tridimensional cúbica. 48
Figura 17 – Comparação entre os melhores valores conhecidos e os melhores valores encontrados pelo algoritmo HG para o modelo HP em grade bidimensional quadrangular. 53
Figura 18 – Comparação entre os melhores valores conhecidos e os melhores valores encontrados pelo algoritmo HG para o modelo HP em grade tridimensional cúbica. 57

Lista de quadros e tabelas

Tabela 1 – Taxa de crescimento de algumas funções. 30
Tabela 2 – Instâncias de referência para a modelo HP em grade bidimensional quadrangular. E - Melhor valor de energia conhecido. 51
Tabela 3 – Instâncias de referência para o modelo HP em grade tridimensional cúbica. E - Melhor valor de energia conhecido. 52
Tabela 4 – Valores e os respectivos tempos, em segundos, encontrados pelo algoritmo HG para cada valor do parâmetro α. 54
Tabela 5 – Comparativo entre os melhores valores encontrados pelo algoritmo HG e os valores encontrados por outros algoritmos para o modelo HP em grade bidimensional quadrangular. E - Melhor valor conhecido. AG - Algoritmo genético de Unger e Moult (1993b). EMC - Algoritmo evolucionário de Monte Carlo de Liang e Wong (2001). CF - Algoritmo de otimização por colônia de formigas de Shmygelska e Hoos (2005). . 55
Tabela 6 – Valores e os respectivos tempos, em segundos, encontrados pelo algoritmo HG para cada valor do parâmetro α. 56
Tabela 7 – Comparativo entre os melhores valores encontrados pelo algoritmo HG e os valores encontrados por outros algoritmos para o modelo HP tridimensional em grade cúbica. E - Melhor valor conhecido. MC - Algoritmo de Monte Carlo. HZ - Método *Hydrophobic Zipper* de Fiebig e Dill (1993). CG - Método exato *Core-directed chain Growth* de Beutler e Dill (1996). 57

"A natureza é, afinal, um sistema
de possibilidades ilimitadas, mas
de opções finitas."
(Arthur M. Lesk)

1 Introdução

As proteínas são macromoléculas biológicas formadas por aminoácidos e desempenham importantes funções nas células. A função biológica de uma proteína é determinada pela sua estrutura tridimensional nativa. Segundo Anfinsen (1973), toda a informação necessária para que uma proteína enovele está contida em sua sequência de aminoácidos.

Atualmente, existem dois métodos experimentais que são usados para determinar a estrutura tridimensional de uma proteína: cristalografia por difração de raios X e espectroscopia por ressonância magnética nuclear (RMN). Quase toda a informação estrutural disponível sobre as proteínas é consequência da aplicação destes dois métodos. Contudo, o número de estruturas tridimensionais conhecidas ainda é muito pequeno se comparado ao número de sequências de aminoácidos disponíveis. Isso se deve, principalmente, à limitação de recursos envolvidos durante a utilização destes dois métodos. Por esse motivo, pesquisadores de diversas áreas vêm tentando desenvolver métodos computacionais que sejam capazes de prever a estrutura nativa de proteínas.

O problema da predição da estrutura tridimensional nativa de proteínas é, atualmente, um dos problemas mais desafiantes da Bioquímica e da Biologia Computacional. Ele consiste em determinar a estrutura nativa de uma proteína utilizando apenas a informação contida em sua sequência de aminoácidos. A resolução desse problema traria avanços significativos para medicina, principalmente em relação ao tratamento de doenças relacionadas ao mau enovelamento de proteínas.

O problema da predição da estrutura protéica pode ser entendido como um problema de otimização cuja solução está em um espaço de busca que cresce exponencialmente de acordo com o tamanho da sequência de aminoácidos da proteína. Com o objetivo de diminuir a complexidade desse problema, diversos modelos protéicos simplificados têm sido propostos. Embora alguns modelos sejam bastante simplificados, como é o caso do modelo Hidrofóbico-Polar (LAU; DILL, 1989), eles apresentam, ainda assim, características importantes do processo de enovelamento.

O problema da predição da estrutura protéica no modelo Hidrofóbico-Polar pode ser difinido como: dada um sequências de resíduos hidrofóbicos (H) e polares (P), encontrar uma conformação com o maior número de contatos hidrofóbicos (HH) entre vizinhos topológicos. Esse problema pertence à classe de problemas NP-completo (CRESCENZI et al., 1998; BERGER; LEIGHTON, 1998) e vários métodos computacionais (UNGER; MOULT, 1993a; FIEBIG; DILL, 1993; PATTON et al., 1995; TOMA; TOMA, 1996; BEUTLER; DILL, 1996; KRASNOGOR et al., 1999) têm sido propostos. Alguns destes métodos têm fornecido *insights* importantes sobre o enovelamento protéico. Outros, têm sido

particularmente úteis em pesquisas sobre o mau enovelamento da beta-amilóide (URBANC LUIS CRUZ; STANLEY, 2006), proteína encontrada em portadores de Alzheimer.

Neste trabalho, apresentamos uma heurística GRASP híbrida para o problema da predição da estrutura protéica no modelo Hidrofóbico-Polar. O problema é abordado como um problema de otimização combinatória e uma formulação é proposta. O algoritmo proposto foi testado com várias instâncias de referência e os resultados obtidos mostram que a heurística apresenta desempenho semelhante aos de outros trabalhos. Não encontramos na literatura, nenhum trabalho relacionado ao tema que utilize uma heurística GRASP híbrida.

Este trabalho está dividido da seguinte forma: no capítulo 2, fazemos uma introdução à bioquímica das proteínas. Nas seções 2.1 e 2.2, mostramos a estrutura de um aminoácido e como eles se unem para formar peptídeos e proteínas. Na seção seguinte, mostramos os níveis estruturais de uma proteína e como estes níveis se relacionam. Nas seções 2.4 e 2.5, comentamos sobre as forças que atuam na estabilização da estrutura nativa e sobre o processo de enovelamento. Na seção 2.6, descrevemos, de forma resumida, os métodos experimentais utilizados na determinação da estrutura tridimensional de proteínas.

No capítulo 3, introduzimos alguns conceitos importantes sobre otimização. A definição de problema de otimização e suas principais características são dadas na seção 3.1. Nas seções 3.2 e 3.3, definimos vizinhança e solução ótima. Em seguida, mostramos como os problemas de otimização são classificados de acordo com sua complexidade. Na seção 3.5, mostramos como os métodos de otimização podem ser classificados. Em seguida, apresentamos a metaheurística GRASP (RESENDE; RIBEIRO, 2002) e o procedimento VND (HANSEN; MLADENOVIĆ, 2003).

No capítulo 4, definimos o problema da predição da estrutura tridimensional de proteínas e quais os métodos computacionais disponíveis. Na seção 4.4, introduzimos os modelos protéicos em grade. Nas subseções 4.4.1 e 4.4.2, apresentamos o modelo Hidrofóbico-Polar e definimos o problema estudado neste trabalho. Na última seção, mostramos algumas abordagens computacionais para o problema em questão.

No capítulo 5, apresentamos a heurística proposta e no capítulo 6, apresentamos os resultados dos experimentos computacionais para o modelo HP em grade bidimensional quadragular e tridimensional cúbica. No último capítulo, apresentamos as conclusões e perspectivas de trabalho futuro.

2 Introdução às proteínas

As proteínas são as macromoléculas biológicas mais abundantes nas células e apresentam uma grande variedade de tipos. Em uma única célula, por exemplo, podem ser encontrados milhares de tipos diferentes.

Todas as proteínas são construídas a partir do mesmo conjunto de 20 aminoácidos ligados covalentemente em uma sequência linear. Embora cada aminoácido tenha uma cadeia lateral com propriedades químicas diferentes, este conjunto de 20 moléculas pode ser considerado o alfabeto com o qual a linguagem da estrutura protéica é escrita.

As proteínas são encontradas em uma grande variedade de tamanhos, de peptídeos relativamente pequenos com apenas poucos resíduos a enormes polímeros. O mais notável é que as células são capazes de produzir proteínas com propriedades e funções estritamente diferentes apenas juntando os mesmos 20 aminoácidos em sequências de tamanhos e combinações diferentes. Enzimas, hormônios, anticorpos, fibras musculares, proteína do cristalino do olho, entre outras, são alguns exemplos de proteínas com funções biológicas diferentes. De todas as proteínas, as enzimas são as mais variadas e especializadas, sendo responsáveis pela catálise[1] de praticamente todas as reações celulares.

2.1 Aminoácidos

Vinte tipos diferentes de aminoácidos são geralmente encontrados nas proteínas. Os aminoácidos possuem uma estrutura comum, formada por um grupo carboxila (—COOH) e um grupo amino (—NH_2) ligados a um mesmo átomo de carbono, o carbono alfa (C_α). Por isso, os aminoácidos são chamados também de alfa-aminoácidos. Os aminoácidos diferem apenas em suas cadeias laterais (grupo R) que variam em estrutura, tamanho e carga elétrica. Esta última influencia bastante na solubilidade dos mesmos em água. No estado sólido, os aminoácidos existem como íons dipolares (Figura 1).

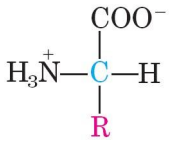


Figura 1 – Estrutura comum a todos os alfa-aminoácidos exceto a prolina. O grupo R ou cadeia lateral é diferente em cada aminoácido. Fonte: Lehninger, Nelson e Cox (2008)

Em todos os aminoácidos, o carbono alfa está ligado a quatro grupos diferentes: um grupo carboxila, um grupo animo, um grupo R e um átomo de hidrogênio. A única

[1] Mudança de velocidade (geralmente aumento) de uma reação química devido à adição de um catalizador (enzima).

exceção é a glicina onde o grupo R é substituído por um outro átomo de hidrogênio.

Os aminoácidos podem ser agrupados em 5 classes diferentes, de acordo com as propriedades químicas das cadeias laterais, em especial a polaridade[2]. Esta varia grandemente de não polar (hidrofóbico) à polar (hidrofílico). A estrutura dos vinte aminoácidos mais comuns pode ser vista na figura 2.

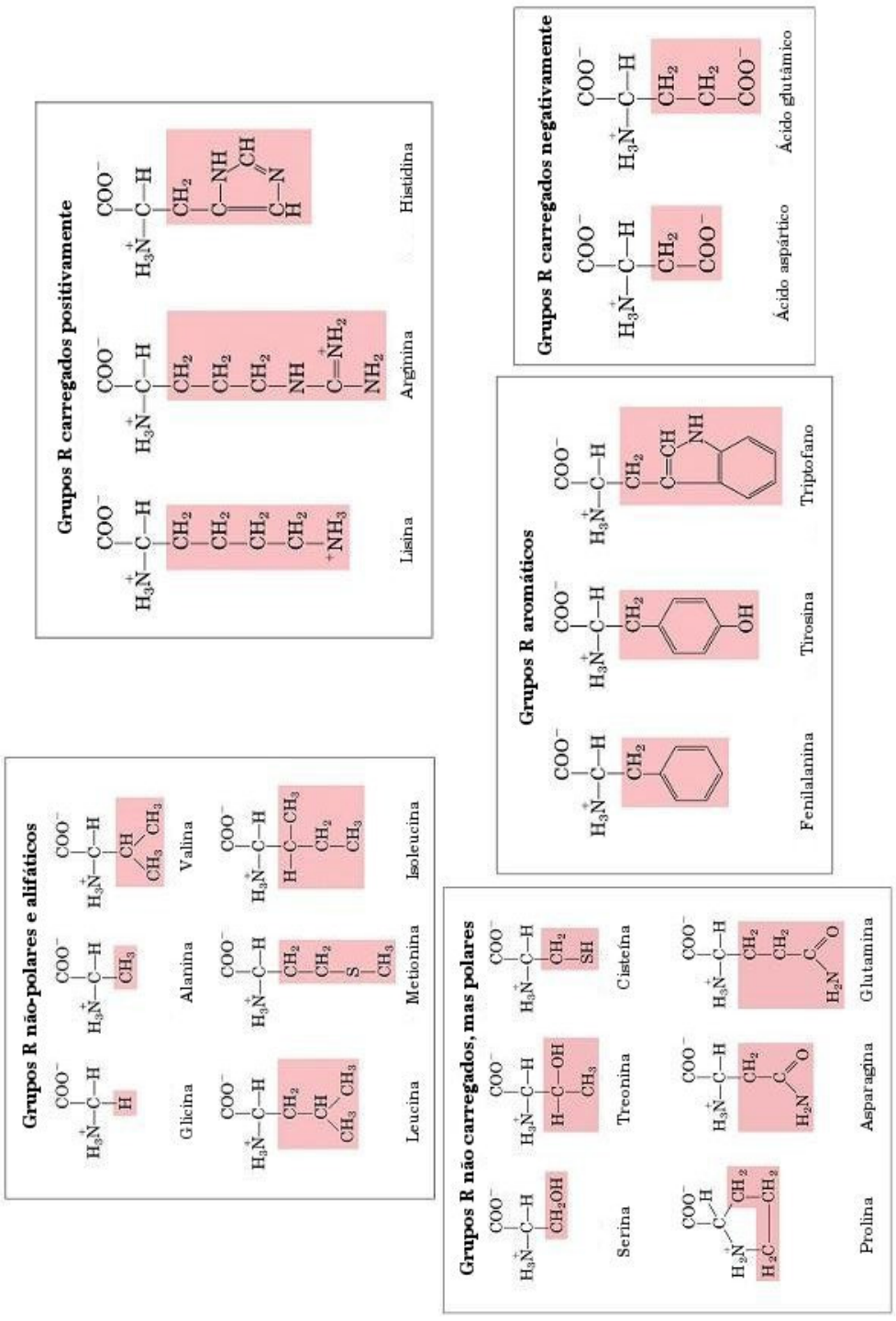


Figura 2 – Os vinte aminoácidos mais comuns encontrados nas proteínas. As cadeias laterais estão destacadas. Fonte: Lehninger, Nelson e Cox (2008)

[2] Tendência em interagir com a água em pH biológico (pH $\cong$ 7).

2.2 Peptídeos e proteínas

Os aminoácidos podem estar ligados covalentemente através de ligações peptídicas dando origem a peptídeos e proteínas. A ligação peptídica é formada a partir do grupo carboxila de um aminoácido e do grupo amino de outro. Durante a formação dessa ligação, um aminoácido perde um átomo de hidrogênio (H) e o outro uma hidroxila (—OH) , sendo liberada uma molécula de água (reação de condensação) (Figura 3). Após a formação da

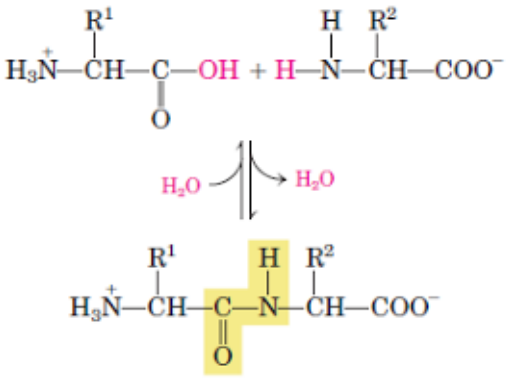


Figura 3 – Formação da ligação peptídica. Fonte: Lehninger, Nelson e Cox (2008).

ligação peptídica, os aminoácidos recebem a denominação de resíduos de aminoácidos, ou simplesmente, resíduos.

Os peptídeos formados por dois, três e quatro aminoácidos recebem o nome de: dipeptídeo, tripeptídeo e tetrapeptídeo, respectivamente. De forma geral, quando poucos aminoácidos estão ligados, a estrutura formada é chamada de oligopeptídeo. Quando muitos aminoácidos estão ligados, um polipeptídeo é formado. Em um peptídeo, o resíduo com o grupo amino livre em uma das extremidades é chamado de resíduo amino-terminal ou N-terminal; já o resíduo da outra extremidade com o grupo carboxila livre é chamado de carboxi-terminal ou C-terminal.

As proteínas podem ter milhares de resíduos de aminoácidos. Embora os termos proteína e polipeptídeo possam, às vezes, ser usados como sinônimos, um polipeptídeo possui geralmente massa molecular inferior a 10.000 **u** (LEHNINGER; NELSON; COX, 2008).

2.3 Estrutura das proteínas

A disposição espacial dos átomos de uma proteína é chamada de conformação. Uma proteína pode assumir várias conformações. Por exemplo, uma conformação pode ser obtida rotacionando-se uma das várias ligações covalentes sem que ocorra a quebra dessa ligação. Em condições fisiológicas, uma ou mais comumente poucas conformações predominam. Tais conformações são aquelas termodinamicamente mais estáveis, ou seja,

aquelas com a menor energia livre de Gibbs (G) (ANFINSEN, 1973). Em seu estado funcional, essas conformações são chamadas de conformações nativas.

A estrutura de uma proteína pode ser dividida, conceitualmente, em quatro níveis:

- **Estrutura primária**: descreve todas as ligações covalentes (ligações peptídicas e dissulfetos) que unem os aminoácidos em uma cadeia polipeptídica. Seu elemento mais importante é a sequência de aminoácidos;
- **Estrutura secundária**: se refere a arranjos de aminoácidos, particularmente estáveis, que dão origem a padrões estruturais (α-hélices, folhas-β e voltas-β);
- **Estrutura terciária**: descreve todos os aspectos da estrutura tridimensional enovelada de uma cadeia polipeptídica;
- **Estrutura quaternária**: corresponde ao arranjo espacial de uma proteína com duas ou mais cadeias polipeptídicas.

A figura 4 mostra os níveis estruturais de uma proteína.

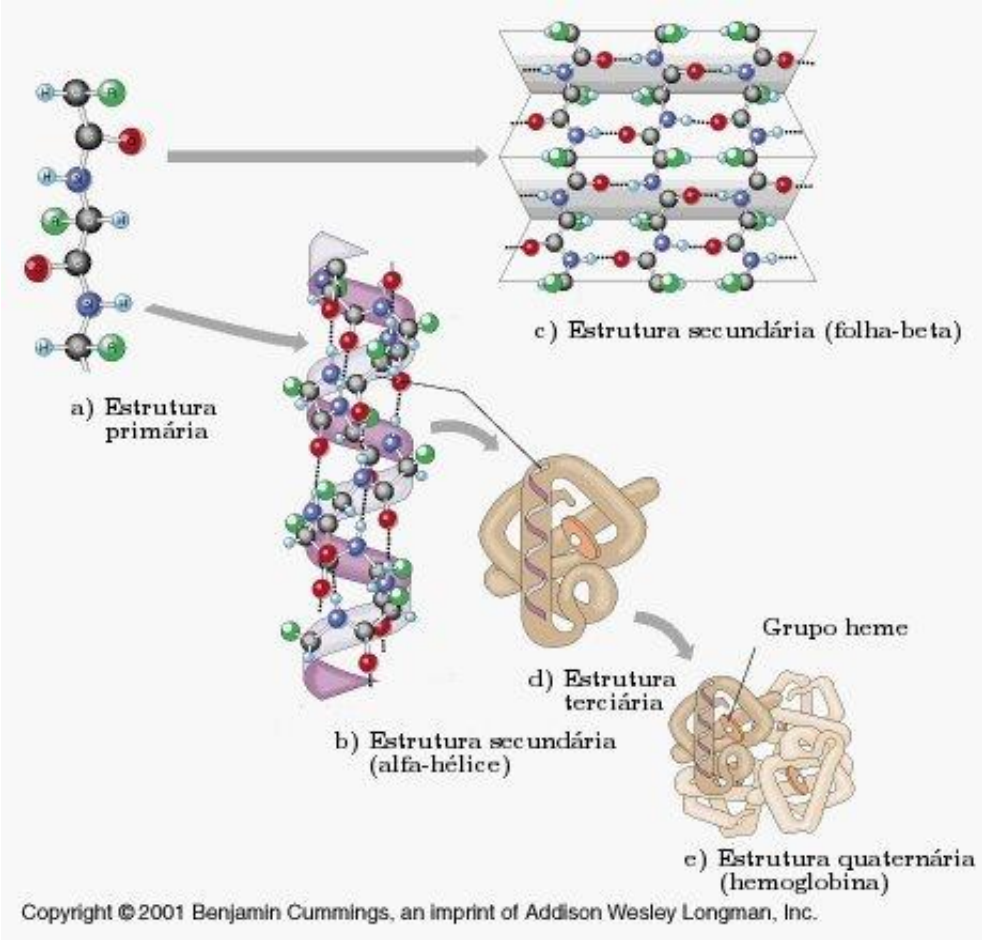


Figura 4 – Os níveis estruturais de uma proteína. Fonte: Marieb (2001).

2.3.1 Planaridade e rigidez da ligação peptídica

As ligações covalentes impõem importantes restrições às conformações de uma cadeia polipeptídica. Os carbonos alfa de resíduos adjacentes são separados por três ligações covalentes (C_α—C—N—C_α). Estudos de difração de raios X (PAULING; COREY; BRANSON, 1951) mostraram que a ligação peptídica é mais curta que a ligação C—N em uma amina e que os átomos associados a ela são coplanares (Figura 5). Devido ao

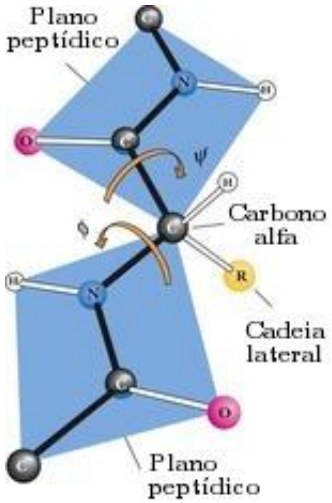


Figura 5 – Planaridade e rigidez da ligação peptídica. Fonte: deconhecida.

caráter parcial de dupla ligação, as ligações peptídicas não podem girar livremente, sendo permitido apenas rotações nas ligações N—C_α e C_α—C. O esqueleto peptídico pode, então, ser entendido como uma série de planos rígidos, com planos consecutivos compartilhando um ponto comum de rotação no carbono alfa. Portanto, as ligações peptídicas limitam o número de conformações possíveis de uma cadeia polipeptídica.

As conformações de uma cadeia polipeptídica podem ser definidas por três ângulos diedrais[3] ϕ, ψ e ω. Os ângulos ϕ e ψ, cuja rotação ocorre, respectivamente, sobre as ligações N—C_α e C_α—C podem assumir, à princípio, qualquer valor entre $-180°$ e $+180°$. Contudo, muitos desses valores não são permitidos devido ao impedimento estérico que ocorre entre os átomos do esqueleto polipeptídico e os átomos das cadeias laterais dos aminoácidos. O ângulo ω, cuja rotação ocorre sobre a ligação peptídica C—N, geralmente não é considerado, pois em 99% dos casos a ligação peptídica se encontra na configuração trans e, neste caso, o ângulo ω assume os valores $\pm 180°$.

O gráfico da figura 6 mostra os valores permitidos para os ângulos ϕ e ψ para o resíduo L-alanina. Nas áreas mais escuras desse gráfico, estão as conformações que não sofrem nenhum tipo de interferência estérica. Nas áreas um pouco mais clara, encontram-se as conformações que estão no limite extremo de contatos desfavoráveis. Nas outras áreas, estão as conformações cujos valores dos ângulos ϕ e ψ variam pouco e as conformações não

[3] Ângulo formado na intersecção de dois planos.

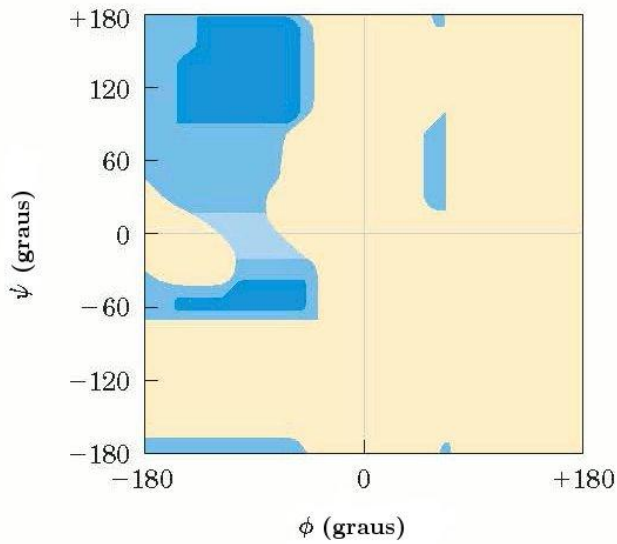


Figura 6 – Gráfico de Ramachandram para o resíduo L-alanina. Fonte: Lehninger, Nelson e Cox (2008).

permitidas. Este gráfico é chamado de gráfico de Ramachandram (RAMACHANDRAN; RAMAKRISHNAN; SASISEKHARAN, 1963).

2.3.2 Estrutura primária

A estrutura primária se refere a sequência de aminoácidos unidos por ligações peptídicas e dissulfetos. Proteínas diferentes possuem sequências de aminoácidos diferentes. Essa variação pode ocorrer em número e/ou combinação de aminoácidos.

A estrutura tridimensional de uma proteína está intimamente relacionada com sua sequência de aminoácidos. A relação entre estrutura primária e estrutura protéica pode ser melhor entendida através dos fatos abaixo, extraídos de Lehninger, Nelson e Cox (2008, p. 93).

> "A bactéria *Escherichia coli* produz mais de 3000 proteínas diferentes... Cada uma possui uma sequência de aminoácidos e uma estrutura tridimensional únicos que lhe confere uma função também única... Este fato sugere que a sequência de aminoácidos deve desempenhar um papel fundamental na determinação da estrutura tridimensional de uma proteína e, consequentemente, de sua função."

> "Milhares de doenças genéticas humana têm sua causa na produção de proteínas defeituosas. O defeito pode variar de uma simples mudança na sequência de aminoácidos (como na anemia falciforme) à deleção de uma grande porção da cadeia polipeptídica (como em muitos casos da

distrofia muscular de Duchenne). É evidente que, se a estrutura primária é alterada, a função da proteína também é alterada."

2.3.3 Estrutura secundária

O termo estrutura secundária se refere ao arranjo espacial dos átomos de qualquer segmento de uma cadeia polipeptídica, sem considerar a conformação das cadeias laterais e seu relacionamento com outros segmentos. Geralmente, uma estrutura secundária ocorre quando os valores dos ângulos ϕ e ψ permanecem aproximadamente os mesmos ao longo do segmento. Nas proteínas, existem poucos tipos de estruturas secundárias, sendo as mais comuns alfa-hélices e folhas-beta.

A alfa-hélice é uma estrutura helicoidal (PAULING; COREY; BRANSON, 1951). Nessa estrutura, o esqueleto polipeptídico se dispõe em forma de hélice estando as cadeias laterais voltadas para o lado de fora da hélice (Figura 7). A parte da hélice que

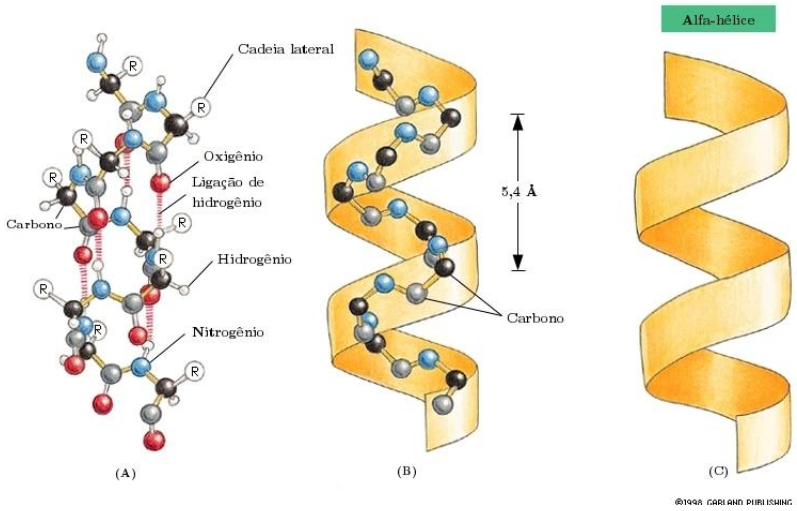


Figura 7 – A estrutura de uma alfa-hélice. Fonte: Alberts et al. (2007).

se repete tem aproximadamente 5,4 Å de comprimento e é formada por 3,6 resíduos. Em uma alfa-hélice, os ângulos ϕ e ψ medem, respectivamente, $-57°$ e $-47°$.

A estrutura da alfa-hélice é estabilizada por ligações de hidrogênio. Nessa estrutura, uma ligação de hidrogênio é formada entre o átomo de hidrogênio de um resíduo e o átomo de oxigênio do quarto resíduo adiante. Todos os resíduos, exceto aqueles das extremidades, formam ligações de hidrogênio. Cada volta da hélice é formada por três ou quatro dessas ligações, o que confere certa estabilidade a essa estrutura.

A conformação beta é uma estrutura estendida no qual o esqueleto polipeptídico se dispõem em zigue-zague (PAULING; COREY; BRANSON, 1951). Esses segmentos, quando dispostos lado a lado, formam uma estrutura semelhante a uma folha pregueada, chamada folha-beta (Figura 8). Nessa estrutura, as ligações de hidrogênio são formadas entre segmentos adjacentes. Geralmente, esses segmentos estão muito próximos na cadeia

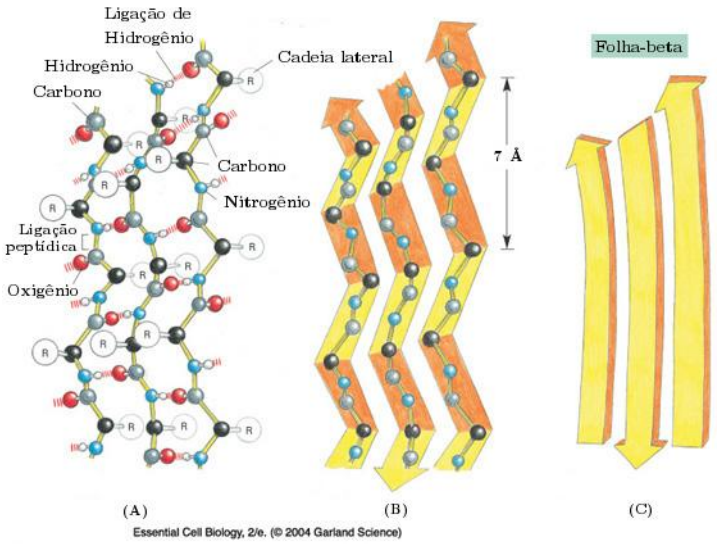


Figura 8 – A estrutura de uma folha-beta. Fonte: Alberts et al. (2007)

polipeptídica. Contudo, há casos onde estão distantes, podendo inclusive estarem em cadeias polipeptídicas diferentes. Nas folhas-beta, as cadeias laterais de resíduos adjacentes projetam-se em direções opostas, criando um padrão alternado. Os segmentos adjacentes podem ser paralelos (quando têm a mesma orientação) ou antiparalelos (quando têm orientações opostas). As estruturas paralela e antiparalela são bem semelhantes, porém na primeira a parte repetida é menor (6,5 Å contra 7 Å). Os padrões das ligações de hidrogênio também são diferentes. Na estrutura paralela, os ângulos diedrais ϕ e ψ medem $-119°$ e $+113°$, respectivamente. Já na antiparalela, os ângulos ϕ e ψ medem, respectivamente, $-139°$ e $+135°$. Contudo, estes valores podem variar, resultando em variações estruturais.

Outra estrutura secundária comum são as voltas. As voltas servem para conectar as várias estruturas secundárias de uma proteína. Por exemplo, os segmentos adjacentes em folhas-beta antiparalelas são conectados por voltas-beta. Essa estrutura é formada por quatro resíduos, com o oxigênio do primeiro formando uma ligação de hidrogênio com o hidrogênio do quarto resíduo. O ângulo formado por esses resíduos mede 180°. As voltas-beta são geralmente encontradas na superfície das proteínas, onde os dois resíduos centrais formam ligações de hidrogênio com a água.

2.3.4 Estrutura terciária e quaternária

A disposição espacial dos átomos de uma proteína é normalmente referenciada como estrutura terciária. De acordo com a estrutura terciária, as proteínas podem ser classificadas em proteínas fibrosas ou proteínas globulares. As proteínas fibrosas possuem geralmente um único tipo de estrutura secundária e desempenham, principalmente, funções estruturais. Já as proteínas globulares possuem vários tipos de estruturas secundárias e desempenham diversas funções.

Algumas proteínas são formadas por duas ou mais cadeias polipeptídicas idênticas ou não. O complexo tridimensional formado por essas cadeias polipeptídicas recebe o nome de estrutura quaternária.

2.4 Estabilidade estrutural

As conformações nativas são apenas parcialmente estáveis. Em condições fisiológicas, a energia livre que separa os estados desnaturado e funcional varia muito pouco (LEHNINGER; NELSON; COX, 2008). O estado desnaturado é caracterizado por uma alta entropia conformacional[4]. Essa entropia e as ligações de hidrogênio, formadas entre os vários grupos da cadeia polipeptídica e a água, tendem a estabilizar este estado. Entretanto, este efeito é contrariado pelas ligações dissulfeto e as interações fracas (ligações de hidrôgenio, interações iônicas e hidrofóbicas).

As ligações dissulfetos são muito mais fortes que as interações fracas. Contudo, as interações fracas funcionam como força estabilizadora da estrutura nativa, devido a sua grande quantidade nas proteínas. De forma grosseira, podemos dizer que a conformação de mais baixa energia livre é aquela com o maior número de interações fracas, onde predominam, geralmente, as interações hidrofóbicas.

Em solução aquosa, o aumento da entropia é a força termodinâmina orientadora da associação de grupos hidrofóbicos. A formação de ligações de hidrogênio nas proteínas é orientada por esse efeito. Dessa forma, grupos polares podem formar ligações de hidrogênio com a água.

As interações hidrofóbicas são muito importantes para a estabilidade conformacional. O interior de uma proteína é, geralmente, um núcleo denso formado por cadeias laterais de aminoácidos hidrofóbicos. Outro fato importante é que grupos polares ou carregados, no interior da proteína, formam ligações de hidrogênio ou interagem iônicamente com outros grupos. A variação de energia livre, resultado da formação de ligações de hidrogênio por diversos desses grupos e a solução, pode ser maior que a diferença de energia livre entre os estados desnaturado e enovelado. Nestes grupos, as ligações de hidrogênio se formam de forma cooperativa (DILL; FIEBIG; CHAN, 1993) dando origem as estruturas secundárias. Portanto, as ligações de hidrogênio têm um papel fundamental no processo de enovelamento.

2.5 Enovelamento

O enovelamento é o processo pelo qual uma proteína adquire sua conformação nativa. De acordo com a hipótese termodinâmica de Anfinsen (1973), o enovelamento

[4] Uma medida dos graus de liberdade conformacional de uma proteína.

não é um processo biológico, mas um processo puramente físico que depende apenas da sequência de aminoácidos da proteína e do solvente ao seu redor.

Em células vivas, as proteínas são sintetizadas muito rapidamente. Por exemplo, em células de *E. coli*, uma proteína biologicamente ativa contendo 100 aminoácidos, pode ser sintetizada em aproximadamente 5 segundos (LEHNINGER; NELSON; COX, 2008, p. 142). Em 1968, Levinthal mostrou que o estado nativo de uma proteína jamais poderia ser alcançado se o enovelamento fosse um processo aleatório. Como, então, as proteínas conseguem enovelar tão rapidamente? Hoje, sabemos que as proteínas utilizam rotas para alcançar o estado nativo. Contudo, as rotas utilizadas pelas proteínas durante o enovelamento são bastante complicadas, não sendo bem compreendidas ainda.

Existem evidências (CRIPPEN, 1978; BALDWIN; ROSE, 1999) de que o enovelamento é um processo hierárquico. Segundo esta hipótese, estruturas secundárias como alfa-hélices e folhas-beta se formariam primeiro. A formação dessas estruturas ocorreria devido a interações locais na cadeia polipeptídica. As estruturas secundárias formadas, por sua vez, interagiriam para formar estruturas estáveis maiores (estruturas supersecundárias). O processo continuaria até que a proteína estivesse completamente enovelada. Em outro modelo, chamado colapso hidrofóbico, as estruturas secundárias e terciária se formariam simultaneamente, dando origem a uma estrutura compacta chamada **molten globule**. Acredita-se que o processo de enovelamento deva possuir características de ambos os modelos.

2.6 Determinação da estrutura tridimensional de proteínas

Atualmente, existem dois métodos experimentais que são usados para determinar a estrutura tridimensional de proteínas: cristalografia por difração de raios X e espectroscopia por ressonância magnética nuclear (RMN). As estruturas determinadas por ambas as técnicas são geralmente bem semelhantes e juntas, elas são responsáveis pelo crescimento da disponibilidade de informação sobre diversas macromoléculas biológicas.

2.6.1 Cristalografia por difração de raios X

A determinação da estrutura tridimensional de uma proteína, por este técnica, pode ser resumida da seguinte forma: um cristal da proteína, cuja estrutura deve ser determinada, é colocado entre uma fonte de raios X e um filme fotográfico. Ao atingir os átomos do cristal, o feixe de raios X é difratado e uma matriz de pontos de intesidades diferentes é produzida no filme fotográfico. Um mapa de densidade eletrônica é, então, construído a partir dessa matriz de pontos. Com o auxílio de um computador, um modelo da estrutura é construído baseado no mapa de densidade eletrônica. A cristalografia por difração de raios X possui, entretanto, algumas limitações. Por exemplo, o ambiente físico do cristal é totalmente diferente do ambiente da proteína em solução. Além disso, este é

um método trabalhoso e que fornece pouca informação sobre o movimento molecular da proteína.

2.6.2 Espectroscopia por ressonância magnética nuclear

A ressonância magnética nuclear (RMN) é uma manifestação do momento ângular do spin nuclear, uma propriedade da mecânica quântica do núcleo atômico. Apenas certos átomos, incluindo ^{1}H, ^{13}C, ^{15}N, ^{19}F e ^{31}P possuem o tipo de spin nuclear que dá origem ao sinal RMN. Quando, por exemplo, um campo magnético estático e forte é aplicado a uma solução contendo uma proteína, surge uma interação que por sua vez tende a alinhar os dipolos magnéticos provenientes do spin presente nos núcleos. Este alinhamento, pode ser em duas direções: paralela (baixa energia) ou antiparalela (alta energia). Um pulso eletromagnético de curta duração ($\sim 10\mu s$) e de frequência adequada é aplicado perpendicularmente aos núcleos alinhados. Os núcleos, então, absorvem certa quantidade de enegia que os fazem atingir estados energéticos mais altos. O espectro resultante desta absorção é utilizado para obter informações sobre os núcleos e o ambiente químico a seu redor. Esse procedimento é realizado vários vezes e os dados obtidos são utilizados para gerar um espectro RMN unidimensional. Contudo, devido a grande quantidade de átomos de ^{1}H nas proteínas, mesmo nas pequenas, o espectro RMN unidimensional torna-se complexo demais para ser analisado. Desta forma, a análise estrutural de proteínas só é possível graças as técnicas de RMN bi e tridimensional.

Diferente da cristalografia por difração de raios X, a RMN utiliza proteínas em solução. Por isso, esta técnica pode esclarecer o lado dinâmico da estrutura protéica, incluíndo mudanças conformacionais, o enovelamento e interações com outras moléculas. Entretanto, a RMN é uma técnica cara.

3 Conceitos básicos sobre otimização

3.1 Problemas de otimização

Os problemas de otimização ocorrem em várias áreas. Nos problemas de otimização, estamos interessados em soluções que sejam ótimas ou "quase" ótimas em relação a algum objetivo. Mais especificamente, estamos interessados em soluções que maximizem ou minimizem uma função de avaliação definida sobre o critério de avaliação selecionado. A função de avaliação, também chamada de função objetivo, atribui um valor a cada solução possível e mede a qualidade de soluções diferentes. Geralmente, não podemos escolher entre todas as soluções disponíveis, pois existem restrições que as limitam. De forma geral, os problemas de otimização possuem as seguintes características:

- Estão disponíveis diferentes soluções;
- Restrições limitam o número de soluções disponíveis;
- Cada solução pode ter um efeito diferente sobre o critério de avaliação e;
- Uma função de avaliação, definida sobre as soluções, descreve o efeito da escolha de uma solução.

Resolver um problema de otimização não é uma tarefa de uma única etapa. Geralmente, o processo de solução consiste de várias etapas que são executadas uma após a outra. De forma geral, podemos dizer que as etapas utilizadas no processo de solução são: reconhecer e definir o problema, construir e resolver o modelo e avaliar as soluções.

Ao modelar um problema de otimização, as soluções são, geralmente, representadas por vetores

$$\boldsymbol{x} = (x_1, \ldots, x_n)$$

de n variáveis de decisão. As variáveis de decisão podem ser contínuas ($\boldsymbol{x} \in \mathbb{R}^n$) ou discretas ($\boldsymbol{x} \in \mathbb{Z}^n$), consequentemente, os problemas de otimização podem ser contínuos ou combinatórios. De acordo com Papadimitriou e Steiglitz (1998), as soluções para problemas de otimização combinatória são geralmente números inteiros, permutações, conjuntos ou grafos.

Uma instância de um problema de otimização é um par (X, f) onde X é o conjunto de soluções viáveis e $f : X \to \mathbb{R}$ é uma função de avaliação que atribui um valor real a cada elemento $\boldsymbol{x} \in X$. Uma solução é viável se satifaz todas as restrições. Em um problema de otimização, o objetivo é encontrar uma solução $\boldsymbol{x}^* \in X$ tal que

$$f(\boldsymbol{x}^*) \leq f(\boldsymbol{x}), \ \forall \boldsymbol{x} \in X \tag{3.1}$$

ou

$$f(\boldsymbol{x}^*) \geq f(\boldsymbol{x}),\ \forall \boldsymbol{x} \in X \tag{3.2}$$

onde $\boldsymbol{x}^*$ é chamado de ótimo global para a instância do problema. As inequações (3.1) e (3.2) definem, respectivamente, um problema de minimização e de maximização. Portanto, um problema de otimização pode ser definido como um conjunto I de instâncias de um problema de otimização. De forma geral, os problemas de minimização são formulados da seguinte forma:

$$\begin{array}{lll} \text{Minimizar} & f(\boldsymbol{x}) & \\ \text{sujeito a} & g_i(\boldsymbol{x}) \geq 0, & i = 1, \ldots, m \\ & h_j(\boldsymbol{x}) = 0, & j = 1, \ldots, p \\ & \boldsymbol{x} \in W_1 \times \ldots \times W_n,\ W_k \in \{\mathbb{B}, \mathbb{Z}, \mathbb{R}\}, & k = 1, \ldots, n \end{array}$$

onde $f(\boldsymbol{x})$ é a função objetivo, $g_i(\boldsymbol{x})$ e $h_j(\boldsymbol{x})$ são as restrições sobre $\boldsymbol{x}$ e $\mathbb{B}$ é o conjunto de números binários.

3.2 Vizinhança

O conceito de vizinhança é muito importante para os problemas de otimização, pois determina as soluções que são próximas. Uma vizinhança é um mapeamento

$$N : X \to 2^X$$

onde X é conjunto de soluções viáveis do problema e 2^X representa todos os subconjuntos possíveis de X. Uma vizinhança é, portanto, um mapeamento que atribui a cada solução $\boldsymbol{x} \in X$ um conjunto de soluções $Y \subset X$, sendo Y representado por $N(x)$.

3.3 Soluções ótimas

Em um problema de minimização, o ótimo global pode ser definido como a solução $\boldsymbol{x}^* \in X$ tal que

$$f(\boldsymbol{x}^*) \leq f(\boldsymbol{x}),\ \forall \boldsymbol{x} \in X.$$

Portanto, a definição de ótimo global não depende da definição de vizinhança.

Dada uma instância (X, f) de um problema de minimização e uma estrutura de vizinhança N, uma solução viável $\boldsymbol{x'} \in X$ é chamada de ótimo local em relação à vizinhança N se

$$f(\boldsymbol{x'}) \leq f(\boldsymbol{x}),\ \forall \boldsymbol{x} \in N(\boldsymbol{x'}).$$

Portanto, não existe ótimo local se uma estrutura de vizinhança não for definida.

3.4 Classes de complexidade

A teoria da complexidade computacional (COOK, 1971; GAREY; JOHNSON, 1979) nos permite classificar os problemas de otimização de acordo com sua dificuldade. A dificuldade de um problema está relacionada a quantidade de recursos computacionais necessários para resolvê-lo. De modo geral, o esforço computacional necessário para resolver um problema de otimização é determinado por sua complexidade de tempo e espaço. A complexidade de tempo se refere a quantidade de passos necessários para resolver o problema. Já a complexidade de espaço se refere a quantidade de espaço, geralmente memória de computador, necessária para resolver o mesmo problema. A complexidade de tempo e espaço dependem do tamanho da instância do problema.

Uma classe de complexidade é um conjunto de problemas computacionais que apresentam o mesmo comportamento assintótico, ou seja, a quantidade de recursos computacionais necessários para resolver um determinado problema desse conjunto é a mesma. Limites podem ser atribuídos à complexidade computacional de uma classe de complexidade. Geralmente, esses limites dependem do tamanho da instância do problema, sendo este muito menor que o tamanho do espaço de busca.

3.4.1 A classe P

A classe de complexidade P (acrônimo em inglês para tempo polinomial determinístico) é definida como o conjunto de problemas que podem ser resolvidos, por um algoritmo, no pior caso, em tempo polinomial. O tempo necessário para resolver um problema em P é limitado assintoticamente ($n > n_0$) por uma função $O(n^k)$, onde n é o tamanho da entrada e n_0 e k são constantes. Para todos os problemas em P, exite um algoritmo que pode resolver qualquer instância de um problema no tempo $O(n^k)$. Portanto, todos os problemas em P podem ser resolvidos, no pior caso, eficientemente.

3.4.2 A classe NP

A classe de complexidade NP (acrônimo em inglês para tempo polinomial não-determinístico) é definida como o conjunto de problemas onde uma solução para um problema pode ser verificada em tempo polinomial. Portanto, todos os problemas em NP podem ter suas soluções efetivamente verificadas.

3.4.3 Problemas tratáveis e intratáveis

Ao usarmos um algoritmo para resolver um problema de otimização, geralmente, estamos interessados no tempo de execução desse algoritmo. De forma geral, podemos distinguir entre tempo de execução polinomial e tempo de execução exponencial. Os problemas que podem ser resolvidos em tempo polinomial, por algum algoritmo, são ditos

tratáveis. Por exemplo, encontrar o menor inteiro positivo em uma lista desordenada de tamanho n é tratável, pois existem algoritmos que resolvem este problema em $O(n)$. Geralmente, os problemas tratáveis são fáceis de resolver. Por outro lado, os problemas que não podem ser resolvidos em tempo polinomial, por algum algoritmo, são intratáveis. Para estes problemas, existe um limite inferior para o tempo de execução, dado por $\Omega(k^n)$, onde n é o tamanho da entrada e k é uma constante maior que 1. A tabela 1 mostra a taxa de crescimento de algumas funções comuns.

Tabela 1 – Taxa de crescimento de algumas funções.

Função	Taxa de crescimento
Constante	$O(1)$
Logarítmica	$O(\log n)$
Linear	$O(n)$
Quase linear	$O(n \log n)$
Quadrática	$O(n^2)$
Polinomial	$O(n^k)$, $k > 1$
Exponencial	$O(k^n)$
Fatorial	$O(n!)$
Super exponencial	$O(n^n)$

3.4.4 Problema NP-difícil e NP-completo

Todos os problemas que estão em P são tratáveis e assim podem ser facilmente resolvidos. Se considerarmos que P $\neq$ NP, então existem problemas que estão em NP, mas não em P. Estes problemas são difíceis, pois não se conhece algoritmo que possa resolvê-los em tempo polinomial. Um problema é NP-difícil se existir um algoritmo que seja redutível a um algoritmo de tempo polinomial e que seja capaz de resolver qualquer problema em NP. Portanto, os problemas NP-difíceis são pelo menos tão difíceis quanto qualquer outro problema em NP, embora possam ser mais difíceis. Contudo, os problemas NP-difíceis não estão necessariamente em NP.

Em 1971, Cook introduziu um conjunto de problemas chamado NP-completo e o definiu como um subconjunto de NP. Portanto, um problema é NP-completo se está em NP e é NP-difícil. Nenhum outro problema em NP é mais difícil, por um fator polinomial, que qualquer problema NP-completo, sendo estes os problemas mais difíceis em NP.

3.5 Métodos de otimização

Os problemas de otimização podem ser resolvidos por métodos de otimização diferentes. O objetivo de um método de otimização é encontrar, com pouco esforço computacional, uma solução ótima ou “quase” ótima para um problema de otimização.

Nesse contexto, a palavra esforço se refere ao tempo e espaço (memória de computador) que o método consome.

Podemos classificar os métodos de otimização em: exatos e não-exatos ou heurísticos. Nos métodos exatos, existe a garantia de que a solução encontrada seja a ótima (ótimo global). Por outro lado, nos métodos não-exatos ou heurísticos não há garantia de que o ótimo global seja encontrado. Geralmente, os métodos exatos são utilizados quando o esforço computacional cresce polinomialmente com o tamanho da entrada. Como vimos, tais problemas pertencem à classe P. Porém, quando os problemas de otimização são NP-difíceis, os métodos exatos não são uma boa opção, pois mesmo para entradas de tamanho médio, o esforço cresce de forma exponencial e o problema se torna intratável. Nestes casos, uma opção seria usar os métodos heurísticos. Os métodos heurísticos exploram as características do problema, sendo, portanto, específicos para cada problema. Além disso, apresentam bom desempenho para muitos problemas NP-completo.

3.5.1 Métodos heurísticos

O conceito de método heurístico surgiu no início da década de 1940 com Pólya (1945), tornando-se um conceito comum em meados da década de 1960.

Rothlauf (2011) divide os métodos heurísticos em: heurísticas, algoritmos de aproximação e heurísticas modernas. As heurísticas, por sua vez, podem ser divididas em: heurísticas de construção e heurísticas de refinamento. As heurísticas de construção constroem uma solução, executando iterativamente passos de construção. As heurísticas de refinamento partem de uma solução pronta e pesquisam o espaço de busca modificando a solução corrente.

Os algoritmos de aproximação são métodos que retornam uma solução aproximada. Além disso, garantem uma solução de boa qualidade. A principal diferença em relação as heurísticas é a existência de limites sobre a qualidade da solução. Se pudermos atribuir limites a qualidade da solução retornada por uma heurística, então temos um algoritmo de aproximação.

Pólya (1945), Romanycia e Pelletier (1985) definem heurística moderna como uma heurística de propósito geral, ou seja, ela não depende do problema. Na Pesquisa Operacional (PO), estes métodos são chamados de metaheurísticas (GLOVER, 1986). Uma característica das metaheurísticas é que a estratégia busca pode ser utilizada, com sucesso, a uma grande variedade de problemas. Tais estratégias geralmente possuem uma fase de intensificação e uma fase de diversificação. Durante a fase de intensificação, áreas promissoras do espaço de busca são pesquisadas e durante a fase de diversificação novas áreas do espaço de busca são exploradas.

3.6 *Greedy Randomized Adaptive Search Procedure*

O *Greedy Randomized Adaptive Search Procedure (GRASP)* (RESENDE; RIBEIRO, 2002) é uma metaheurística de partidas múltiplas na qual cada iteração consiste de duas fases: construção e busca local. Na fase de construção, uma solução viável é construída. Na fase de busca local, a vizinhança da solução construída é investigada até que um ótimo local seja encontrado. Ao final das iterações, a melhor solução encontrada é retornada. A metaheurística GRASP é mostrada a seguir.

Algoritmo 1 A metaheurística GRASP.

```
1: procedure GRASP(it_max, α)
2:     x* ← ∅                                   ▷ Melhor solução encontrada
3:     f* ← +∞
4:     for i = 1,...,it_max do
5:         x ← Construcao(α);
6:         x' ← BuscaLocal(x)
7:         if f(x') < f* then
8:             x* ← x'
9:             f* ← f(x*)
10:        end if
11:    end for
12:    return x*
13: end procedure
```

Na fase de construção , o conjunto de elementos candidatos E é formado por todos os elementos viáveis $e \in E$ que podem ser incorporados à solução em construção. A cada iteração, o próximo elemento a ser incorporado à solução é determinado pela avaliação de todos os elementos de E de acordo com uma função gulosa $g(e)$. Esta função gulosa representa, geralmente, um incremento na função objetivo devido a incorporação deste elemento à solução. A avaliação dos elementos por esta função conduz a criação de uma lista restrita de candidatos (LRC) formada pelos melhores elementos, isto é, aqueles cuja incorporação na solução parcial resultam no menor custo incremental (aspecto guloso).

A LRC é formada por todos os elementos viáveis e que podem ser inseridos na solução parcial em construção sem perda de viabilidade e cuja qualidade é superior a um valor limite dado por

$$g(e) \in [g_{min}, g_{min} + \alpha(g_{max} - g_{min})],$$

onde g_{min} e g_{max} correspondem, respectivamente, ao menor e ao maior custo incremental. A LRC pode ser limitada pelo número de elementos (baseado em cardinalidade) ou pela qualidade dos elementos (baseado em valor). Além disso, ela está associada a um parâmetro $\alpha \in [0, 1]$. Se $\alpha = 0$, então a construção é totalmente gulosa. Se $\alpha = 1$, então a contrução é totalmente aleatória. O elemento e a ser incorporado à solução parcial é aleatoriamente selecionado da LRC (aspecto aleatório). Uma vez selecionado, esse elemento é incorporado à solução parcial, a LRC é atualizada e os custos reavaliados (aspecto adaptativo). A seguir, mostramos o procedimento de construção da metaheurístca GRASP.

Algoritmo 2 O procedimento de construção.

```
1: procedure CONSTRUCAO(α)
2:     x ← ∅
3:     Avalie o custo incremental g(e) dos elementos e ∈ E
4:     while E ≠ ∅ do
5:         g_min ← min{g(e) | e ∈ E}
6:         g_max ← max{g(e) | e ∈ E}
7:         LRC ← {e ∈ E | g(e) ≤ g_min + α(g_max − g_min)}
8:         Selecione, aleatoriamente, e ∈ LRC
9:         x ← x ∪ {e}
10:        Atualize a lista de candidatos E
11:        Reavalie o custo incremental g(e) dos elementos e ∈ E
12:    end while
13:    return x
14: end procedure
```

As soluções construídas na fase de construção não são necessariamente ótimas, mesmo em relação a vizinhanças simples. A fase de busca local geralmente melhora a solução construída. De forma geral, um procedimento de busca local funciona de forma iterativa substituindo a solução corrente por uma solução melhor na vizinhança da solução corrente. A seguir é apresentado o pseudo-código de um procedimento de busca local básico.

Algoritmo 3 Um procedimento de busca local básico.

```
1: procedure BUSCALOCAL(x)
2:     while x não for um ótimo local do
3:         Encontre x′ ∈ N(x) com f(x′) < f(x)
4:         x ← x′
5:     end while
6:     return x
7: end procedure
```

3.7 *Variable Neighborhood Descent*

O *Variable Neighborhood Descent (VND)* (HANSEN; MLADENOVIĆ, 2003) é um procedimento de busca local que explora a idéia de mudança sistemática de vizinhança para escapar de ótimos locais. O VND é mostrado a sequir.

Algoritmo 4 O procedimento de busca local VND.

```
1: procedure VND(x)
2:     Selecione as estruturas de vizinhança N_k(x), k = 1, ..., k_max
3:     k ← 1
4:     while k ≤ k_max do
5:         Encontre o melhor vizinho x' ∈ N_k(x)
6:         if f(x') < f(x) then
7:             x ← x'
8:             k ← 1
9:         else
10:            k ← k + 1
11:        end if
12:    end while
13:    return x
14: end procedure
```

Na $k-$ésima iteração, a vizinhança N_k da solução corrente será investigada até que um mínimo local seja encontrado. Se a solução encontrada for melhor que a solução atual, então essa solução passa a ser a melhor solução encontrada até o momento e sua vizinhança N_k será investigada na iteração seguinte. Caso contrário, a vizinhança N_{k+1} da solução corrente será investigada. Isso se repete até que as k_{max} estruturas de vizinhanças da solução corrente tenham sido investigadas, sendo retornada a melhor solução encontrada.

4 Predição da estrutura tridimensional de proteínas

O problema da predição da estrutura protéica (do inglês *Protein Structure Prediction problem, PSP*) pode ser entendido como: determinar a estrutura tridimensional de uma proteína a partir de sua sequência de aminoácidos. Esse problema é considerado, atualmente, um dos grandes desafios da Bioquímica e da Biologia Computacional.

As limitações dos métodos experimentais de determinação da estrutura protéica, têm feito com que pesquisadores de diversas áreas busquem desenvolver métodos computacionais que sejam capazes de resolver o *PSP*. O desenvolvimento de tais métodos traria avanços significativos para as ciências da vida, principalmente, para a medicina, pois várias doenças, como por exemplo, a doença de Alzheimer, estão relacionadas ao mau enovelamento de certas proteínas. Outro motivo, não menos importante, é a substituição dos atuais métodos experimentais, utilizados na descoberta de novas drogas, por métodos computacionais rápidos e eficientes (DILL et al., 2007).

Para que um método computacional para o *PSP* tenha sucesso, alguns requisitos são necessários: a função de energia deve ser adequada, a cadeia polipeptídica deve ser representada de forma apropriada e o método de busca conformacional deve ser eficiente. Felizmente, tem havido progresso significativo nesse sentido, devido principalmente aos experimentos do *Critical Assessment of Techniques for Structure Prediction (CASP)*[1]. Os métodos computacionais disponíveis, atualmente, para a predição da estrutura protéica podem ser agrupados em três categorias: modelagem por homologia, reconhecimento do enovelamento (*Threading*) e modelagem livre.

4.1 Modelagem por homologia

A modelagem por homologia ou modelagem comparativa é uma classe de métodos que se baseia no fato de que proteínas com sequências similares possuem estruturas também similares. Segundo Lesk e Chothia (1980), as estruturas tridimensionais de proteínas, de uma mesma família, são mais conservadas que suas sequências de aminoácidos. Dessa forma, a estrutura tridimensional de uma proteína poderia ser modelada utilizando as estruturas de proteínas da mesma família que já foram resolvidas experimentalmente.

A modelagem por homologia é composta basicamente por cinco etapas: 1) alinhamento da sequência-alvo com sequências de estruturas conhecidas; 2) construção de um modelo estrutural inicial; 3) determinação da conformação das cadeias laterais do núcleo e de voltas; 4) refinamento e 5) validação do modelo. A qualidade do modelo gerado

[1] Encontro que acontece de dois em dois anos e tem como objetivo auxiliar nos avanços das técnicas de predição da estrutura protéica. Sua última edição aconteceu em 2012.

depende do grau de similaridade entre a sequência-alvo e as sequências das estruturas conhecidas. Quando, por exemplo, a correspondência entre as sequências é superior a 40%, os modelos gerados são tão acurados quanto as estruturas determinadas experimentalmente (KOPP; SCHWEDE, 2004). Quando a correspondência está entre 30% e 40%, obter um alinhamento correto passa a ser uma tarefa difícil, pois são frequentes inserções e deleções. Quando, porém, a correspondência é inferior a 30%, a identificação de estruturas homólogas passa a ser um problema e o alinhamento torna-se muito mais difícil.

Atualmente, os melhores métodos para predição da estrutura protéica pertencem a esta categoria.

4.2 Reconhecimento do enovelamento (*Threading*)

Os métodos de reconhecimento do enovelamento se baseam no fato de que o número total de estruturas protéicas é muito menor que o número total de sequências de DNA. Os métodos desta categoria utilizam bancos de dados de estruturas, como o *Protein Data Bank (PDB)*, para pesquisar por estruturas que sirvam como modelo (*template*) para a proteína-alvo. Mais especificamente, tais métodos procuram, nessas estruturas, padrões de enovelamento ou *motifs* estruturais semelhantes que possam ser utilizados na proteína-alvo. O *Threading* é semelhante à modelagem comparativa no sentido de que ambos tentam construir um modelo estrutural usando como modelo estruturas determinadas experimentalmente.

O *Threading* é capaz de detectar alinhamentos entre o alvo e os modelos (*templates*) independentemente de qualquer relação evolutiva. Por exemplo, proteínas que possuem padrões de enovelamento semelhantes podem ser detectadas, mesmo que haja baixa similaridade entre as sequências. Quando a similaridade é baixa, a identificação precisa do alinhamento alvo-modelos torna-se um problema. Para a eficiência destes métodos é essencial que se utilize boas funções de pontuação para o alinhamento.

4.3 Modelagem livre

A modelagem livre agrupa todos os métodos computacionais que utilizam princípios físico-químicos para determinar a estrutura tridimensional de proteínas. O princípio norteador para o desenvolvimento de tais métodos é a hipótese termodinâmica de Anfinsen (1973).

Os métodos desta categoria podem ser classificados em: métodos por primeiros princípios que utilizam ou não informações de bancos de dados. Estes últimos são chamados também de métodos *ab initio*. De forma geral, os métodos *ab initio* utilizam algoritmos de busca conformacional e campos de forças para determinar a conformação de uma proteína. Contudo, tem havido pouco progresso destes métodos (ROY; ZHANG, 2012). Por outro

lado, os métodos que utilizam informações de bancos de dados têm tido maior sucesso, sendo essas informações utilizadas na montagem das estruturas (SIMONS et al., 1997; XU; ZHANG, 2012).

4.4 Modelos protéicos em grade

O problema da predição da estrutura protéica pode ser entendido como um problema de otimização cuja solução está em um espaço de busca que cresce exponencialmente de acordo com o tamanho da sequência de aminoácidos da proteína. Com o objetivo de diminuir a complexidade do problema, diversos modelos protéicos simplificados têm sido propostos.

Princípios gerais da estrutura protéica, estabilidade e cinética do enovelamento têm sido explorados através de simulação computacional utilizando modelos protéicos simplificados (DILL et al., 1995). Em tais modelos, as proteínas são modeladas em um nível de abstração rudimentar, assim poucos parâmetros e aproximações são utilizados. Contudo, esses modelos permitem a exploração completa do espaço conformacional[2], pelo menos para cadeias protéicas pequenas.

Existem vários modelos protéicos simplificados entre os quais estão os modelos em grade (DILL, 1985). Nos modelos em grade, cada aminoácido é representado como uma esfera e as ligações entre os aminoácidos são representadas por linhas. A grade serve simplesmente para dividir o espaço em células de mesmo formato. Uma posição da grade pode estar ou não ocupada por um aminoácido. Os ângulos de ligação assumem poucos valores discretos e dependem da estrutura da grade. Existem vários tipos de grades em duas e três dimensões. São exemplos de grades bidimensionais as grades: triangular, quadrangular e hexagonal; e de tridimensionais as grades: cúbica e cúbica centrada em face.

Segundo Dill et al. (1995), os modelos protéicos em grade possuem as seguintes vantagens e desvantagens:

- A resolução do problema original é perdida. Além disso, os detalhes da estrutura protéica e da energética não são representados de forma acurada;
- Simulações no nível atômico podem explorar apenas pequenas mudanças conformacionais enquanto que, simulações com modelos em grade podem explorar mudanças conformacionais maiores;
- O cálculo da energia, em modelos atomísticos, é mais complexo, pois os campos de força possuem termos para as ligações covalentes. Já nos modelos em grade, o cálculo da energia é mais simples, pois esses termos são omitidos;

[2] Conjunto de todas as conformações possíveis de uma molécula

- Simulações utilizando modelos atômicos requerem muitos parâmetros e aproximações, além disso, envolvem a enumeração incompleta do espaço conformacional;
- Os modelos em grade podem ser utilizados para testar hipóteses e aproximações em modelos analíticos.

4.4.1 O modelo Hidrofóbico-Polar

O modelo Hidrofóbico-Polar de Lau e Dill (1989) é um modelo protéico em grade onde uma proteína é modelada como uma sequência de aminoácidos hidrofóbicos (H) e polares (P). Uma conformação é representada por um caminho sobre a grade. Cada resíduo pode ocupar apenas uma posição livre na grade. Os resíduos adjacentes na sequência são chamados de vizinhos conectados e os resíduos adjacentes no espaço, mas não adjacentes na sequência, são chamados de vizinhos topológicos. Todo contato hidrofóbico (HH) entre vizinhos topológicos possui um valor de energia livre ϵ, geralmente negativo. Todos os outros contatos (PH e PP) entre vizinhos topológicos possui valor de energia igual a zero. A energia total da conformação é dada pela soma das energias livres dos contatos hidrofóbicos (HH). A figura 9 mostra um exemplo de uma conformação para a sequência `HPPHPPHPHPPHPHPHHH` em uma grade bidimensional quadrangular.

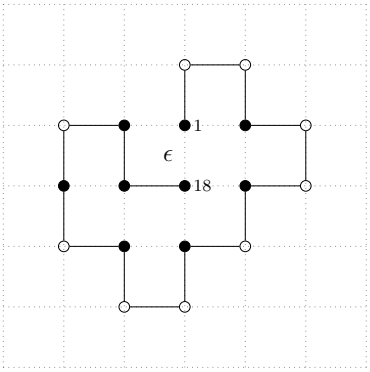


Figura 9 – Exemplo de uma conformação. As esferas pretas representam os resíduos hidrofóbicos (H) e as brancas representam os resíduos polares (P).

No exemplo da figura 9, os resíduos 17 e 18 são vizinhos conectados e os resíduos 1 e 18 são vizinhos topológicos cujo contato hidrofóbico possui energia ϵ. Se $\epsilon = -1$, então a energia total livre dessa conformação é igual a -9.

Apesar da simplicidade, o modelo HP apresenta algumas características importantes. As conformações desenoveladas possuem um pequeno número de contatos hidrofóbicos. À medida que o número de contatos hidrofóbicos aumenta, o número de conformações diminui e as mesmas passam a apresentar:

- Baixo valor de energia livre;

- Estrutura compacta;
- Núcleo formado por resíduos hidrofóbicos e;
- Estruturas secundárias.

4.4.2 Problema da predição da estrutura protéica no modelo HP

O problema da predição da estrutura protéica no modelo Hidrofóbico-Polar pode ser definido como: dada uma sequência de resíduos hidrofóbicos (H) e polares (P), encontrar uma conformação, dita nativa, com o maior número de contatos hidrofóbicos (HH) entre vizinhos topológicos, ou seja, com a menor energia livre. Esse problema pertence à classe de problemas NP-completo (CRESCENZI et al., 1998; BERGER; LEIGHTON, 1998).

4.4.3 Trabalhos relacionados

Nesta seção, mostramos algumas das diversas abordagens para o problema da predição da estrutura protéica no modelo Hidrofóbico-Polar.

Um dos primeiros trabalhos sobre simulação do enovelamento protéico utilizando o modelo HP foi o de Unger e Moult (1993b). Estes desenvolveram um algoritmo genético (AG) que pode ser entendido como uma extensão do método de Monte Carlo. Este AG utiliza um critério semelhante ao de Metrópolis (METROPOLIS et al., 1953) para selecionar as conformações de mais baixa energia.

Em 1995, Patton et al. desenvolveram um algoritmo genético semelhante ao de Unger e Moult (1993a). Ao contrário deste último, este AG utiliza um sistema de coordenadas internas relativas para representar os indivíduos, o que evita que um grande número de conformações inválidas seja gerada. Além disso, foi utilizada uma função de avaliação com penalidade, permitindo que regiões desconhecidas fossem exploradas. Para o desenvolvimento de algoritmos evolucionários (AE) e outros métodos heurísticos aplicados ao modelo HP, algumas recomendações específicas podem ser encontradas em Krasnogor et al. (1999).

O método *Hydrophobic Zipper (HZ)*, proposto por Fiebig e Dill (1993), é um método que se baseia na idéia de cooperatividade no enovelamento protéico (DILL; FIEBIG; CHAN, 1993). A cooperatividade descreve como um estado globalmente ótimo (estado nativo) pode ser encontrado sem uma busca exaustiva. Outro método que utiliza o conceito de cooperatividade é o método *Contact Interactions (CI)* de Toma e Toma (1996). Esse método também é uma extensão do método de Monte Carlo. A principal diferença em relação aos outros métodos, está no critério de aceitação de novas conformações. O método CI não se baseia na energia da molécula inteira (conformação), mas em fatores de resfriamento associados a cada resíduo, definindo regiões de baixa e alta mobilidade.

O método exato *Core-directed chain Growth (CG)* de Beutler e Dill (1996) é um algoritmo de crescimento de cadeia que utiliza uma função heurística para construir um núcleo hidrofóbico. A construção do núcleo hidrofóbico ocorre pela adição sistemática de segmentos, cujos tamanhos variam de acordo com um procedimento. O metódo CG foi testado com diversas instâncias do modelo HP em duas e três dimensões, mostrando-se superior aos métodos não exaustivos

O algoritmo evolucionário de Monte Carlo (EMC) de Liang e Wong (2001) é um método que utiliza o conceito de cooperatividade para acelerar as simulações. Este EMC utiliza populações de cadeias de Markov, onde cada cadeia possui uma temperatura diferente. As populações, cujas distribuições de Boltzmann são preservadas, são atualizadas através de operadores de mutação e cruzamento.

Alguns exemplos de heurísticas são: o algoritmo memético de Krasnogor et al. (2002) e os algoritmos Busca Tabu (BT) de Lesh, Mitzenmacher e Whitesides (2003) e de Blazewicz et al. (2004). Dentre as várias heurísticas, podemos destacar o método *Ant Colony Optimization (ACO)* de Shmygelska e Hoos (2005). Apelidado de ACO-HPPFP-3, este algoritmo consiste em uma melhoria de seu algoritmo ACO anterior (SHMYGELSKA; HOOS, 2003) e sua extensão para o modelo HP tridimensional.

Thachuk, Shmygelska e Hoos (2007) apresentaram uma implementação do método *Replica Exchange Monte Carlo (REMC)* que utiliza estruturas de vizinhanças diferentes. As estruturas de vizinhança utilizadas são geradas a partir de dois conjunto de movimentos: *VSHD* e *pull moves* (LESH; MITZENMACHER; WHITESIDES, 2003). Este trabalho foi o primeiro a utilizar o *pull moves* em uma grade tridimensional cúbica.

5 A heurística proposta

Neste capítulo, apresentamos a heurística proposta. O problema da predição da estrutura protéica no modelo HP é abordado como um problema de otimização combinatória e uma formulação é proposta.

5.1 Representação da solução

A conformação de uma proteína no modelo HP pode ser representada de três formas diferentes: coordenadas cartesianas, coordenadas internas (UNGER; MOULT, 1993b; UNGER; MOULT, 1993a; PATTON et al., 1995) e matriz de distâncias. Neste trabalho, optamos pela representação que utiliza coordenadas cartesianas. A escolha desta representação se deve principalmente à facilidade de implementação do tipo de movimento utilizado na fase de busca local.

Sejam Σ^+ o conjunto de todas as cadeias em {H,P}, excluindo a cadeia vazia, e s uma cadeia em Σ^+ de comprimento m. Mais especificamente, podemos escrever s como $s_1 s_2 \ldots s_m$, onde s_i, $i = 1, \ldots, m$, representa o símbolo i. Devido ao contexto, utilizaremos as palavras sequência e resíduo como sinônimos de cadeia e símbolo, respectivamente[1]. Uma conformação para s será representada por um vetor

$$\boldsymbol{c} = (c_1, \ldots, c_m)$$

tal que c_i representa a posição do resíduo i no espaço inteiro n dimensional. Em uma grade qualquer, para que uma conformação seja válida, ela deve obedecer as seguintes restrições:

- A distância entre resíduos adjacentes na sequência deve ser a mesma e;
- Dois resíduos quaisquer não podem ocupar uma mesma posição na grade.

A figura 10 mostra um exemplo de uma conformação para a sequência HPPHPPHPPH em uma grade bidimensional quadrangular.

[1] Para um melhor entendimento sobre cadeias e linguagens, recomendamos o livro de Hopcroft et al. (2000).

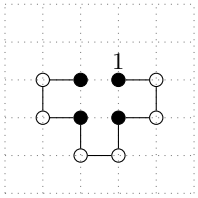


Figura 10 – Exemplo de uma conformação para a sequência `HPPHPPHPPH`. O primeiro resíduo está numerado com 1.

5.2 Função de energia

Krasnogor et al. (1999) observaram que a função de energia proposta por Lau e Dill (1989) não é capaz de diferenciar conformações com o mesmo número de contatos hidrofóbicos, pois apenas estes contribuem para a energia total da conformação. Um exemplo desta situação é mostrado na figura 11.

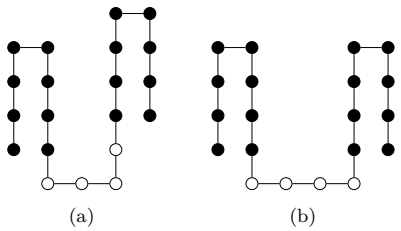


Figura 11 – Exemplo de conformações diferentes, mas que apresentam o mesmo valor energético. Fonte: Krasnogor et al. (1999)

As conformações a e b possuem o mesmo número de contatos hidrofóbicos e, portanto, possuem a mesma energia. Entretanto, a conformação a é mais compacta que a conformação b. Para resolver este problema, Krasnogor et al. (1999) propuseram uma função de energia que é dependente das distâncias entre os resíduos hidrofóbicos. Na literatura, existem algumas poucas funções que utilizam a mesma idéia. Neste trabalho, utilizaremos a função de energia de Berenboym e Avigal (2008), mostrada abaixo.

$$E(\boldsymbol{c}) = \sum_{s_i, s_j \in \{\mathtt{H}\}} \epsilon_{ij} \tag{5.1}$$

Nesta função, ϵ_{ij}, que é dado por

$$\epsilon_{ij} = \begin{cases} -1/d_{ij}^2 & \text{se } |i-j| \geq 2 \\ 0 & \text{caso contrário,} \end{cases}$$

representa a energia livre do contato hidrofóbico entre os resíduos i e j, onde d_{ij} é a distância euclidiana entre eles.

5.3 Formulação do problema

O problema da predição da estrutura protéica no modelo HP pode ser definido como: dada uma sequência $s \in \Sigma^+$ de m resíduos, encontrar uma conformação $\boldsymbol{c}$ para s com a menor energia livre. Esse problema pode ser formulado da seguinte forma:

$$\begin{aligned} \text{Minimizar} \quad & E(\boldsymbol{c}) \\ \text{sujeito a} \quad & d_{i,i+1} = 1, \qquad \forall i = 1, \ldots, m \qquad (5.2) \\ & c_i \neq c_j, \qquad \forall i \neq j = 1, \ldots, m \qquad (5.3) \\ & c_i \in \mathbb{Z}^n. \end{aligned}$$

Na formulação acima, $E(\boldsymbol{c})$ é a função de energia (5.1) e as sentenças (5.2) e (5.3) são as restrições sobre a conformação $\boldsymbol{c}$.

5.4 O procedimento principal

O procedimento principal da heurística é mostrado no algoritmo 5. A cada iteração desse procedimento, uma conformação viável é construída na fase de construção. A vizinhança da conformação construída é, então, investigada pelo procedimento VND (HANSEN; MLADENOVIĆ, 2003) até que um mínimo local seja encontrado. Se a energia da conformação encontrada (mínimo local) for menor que a energia da conformação corrente, então essa conformação passa a ser a melhor solução encontrada até o momento. Ao final das iterações a melhor conformação encontrada é retornada.

Algoritmo 5 O procedimento principal da heurística.

```
1: procedure HGRASP(it_max, s, α)
2:     c* ← ∅                                   ▷ Melhor solução encontrada
3:     E* ← +∞
4:     for i ← 1 to it_max do
5:         c ← Construcao(s, α)
6:         c' ← VND(c)
7:         if E(c') < E* then
8:             c* ← c'
9:             E* ← E(c*)
10:        end if
11:    end for
12:    return c*
13: end procedure
```

5.5 Fase de construção

Na fase de construção, uma conformação é construída de forma iterativa. A cada iteração do algoritmo de construção, um resíduo é colocado em uma posição livre da grade e a conformação parcial cresce. Ao final das iterações, a conformação construída é retornada. A seguir, mostramos como o algoritmo de construção (Algoritmo 6) funciona.

Algoritmo 6 Constrói iterativamente uma conformação.

```
1: procedure CONSTRUCAO(s, α)
2:     Posicione, em c, os dois primeiros resíduos
3:     for i ← 3 to m do
4:         L ← ∅
5:         c' ← (c_1, c_2 ..., c_i)
6:         Encontre A para o resíduo i − 1
7:         for j ← 1 to |A| do
8:             c'_i ← a_j
9:             if VALIDA(c') then
10:                Calcule E(c') usando s_1 s_2 ... s_i
11:                L ← L ∪ {a_j^(E)}
12:            end if
13:        end for
14:        Ordene L em ordem crescente de E
15:        LRC = {l ∈ L | E ≤ E_min + α(E_max − E_min)}
16:        Selecione, aleatoriamente, l ∈ LRC
17:        c_i ← l
18:    end for
19:    return c
20: end procedure
```

O algoritmo de construção começa com a criação de uma conformação $\boldsymbol{c}$. Em toda conformação construída, as posições dos dois primeiros resíduos são fixas. O primeiro resíduo é colocado na origem dos eixos e o segundo sobre a primeira unidade do eixo x. Para determinar as posições dos resíduos restantes devemos fazer: para cada resíduo i, $3 \leq i \leq m$, devemos determinar as posições adjacentes (horizontal e vertical) ao resíduo $i-1$. Seja A o conjunto formado pelas posições adjacentes ao resíduo $i-1$ tal que a_j é o j-ésimo elemento desse conjunto. O resíduo i é, então, colocado na posição adjacente a_j e a conformação $\boldsymbol{c'}$ resultante, que possui i resíduos, é avaliada (Algoritmo 7), isto é, se ela não viola as restrições (5.2) e (5.3). Note que, uma conformação $\boldsymbol{c'}$ diferente é construída

Algoritmo 7 Verifica se uma conformação é válida.

```
1: procedure VALIDA(c)
2:     for i ← 1 to m − 1 do
3:         if d_{i,i+1} ≠ 1 then
4:             return 0
5:         end if
6:         for j ← i + 2 to m do
7:             if c_i = c_j then
8:                 return 0
9:             end if
10:        end for
11:    end for
12:    return 1
13: end procedure
```

toda vez que o resíduo i é colocado em uma posição a_j. Seja E a energia da conformação $\boldsymbol{c'}$. Se $\boldsymbol{c'}$ for uma conformação válida, então sua energia é calculada usando a sequência $s_1 s_2 \ldots s_i$ e a dupla (a_j, E), que denotaremos por $a_j^{(E)}$, é inserida em uma lista L. Em

seguida, a lista L é ordenada em ordem crescente de E. Os elementos de L (elementos viáveis) são utilizados na construção da lista restrita de candidatos (LRC), conforme a expressão

$$LRC = \{l \in L \,|\, E \leq E_{min} + \alpha(E_{max} - E_{min})\},$$

cujo tamanho é definido pelo parâmetro $\alpha \in [0, 1]$. Nessa expressão, E_{min} e E_{max} correspondem, respectivamente, a menor e a maior energia das conformações $\boldsymbol{c'}$. Por fim, um elemento da LRC é escolhido aleatoriamente e o resíduo i colocado nessa posição.

Durante a construção da solução pode ocorrer o caso onde todas as posições adjacentes estão ocupadas. Neste caso, qualquer posição a_j torna a conformação $\boldsymbol{c'}$ inválida e, consequentemente, a LRC será vazia. Quando isto ocorre o procedimento principal (Algoritmo 5) passa para a próxima iteração. Um exemplo desta situação é mostrado na figura 12.

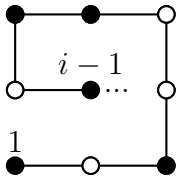


Figura 12 – Exemplo de solução parcial onde todas as posições adjacentes (horizontal e vertical) ao resíduo $i-1$ estão ocupadas, impedindo o crescimento da solução.

5.6 Fase de busca local

Na fase de busca local, utilizamos o procedimento VND de Hansen e Mladenović (2003). A vantagem deste método em relação aos métodos tradicionais de busca local é que ele utiliza várias estruturas de vizinhanças ao invés de uma única. Dessa forma, ele pode pesquisar por soluções mais distantes da solução atual, escapando de ótimos locais. Abaixo, vemos o pseudo-código desse procedimento.

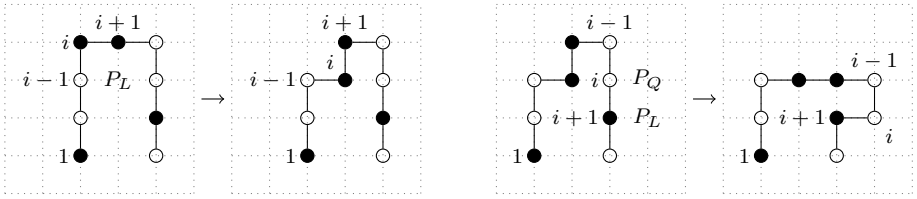
Algoritmo 8 O procedimento VND.

```
1: procedure VND(c, k_max)
2:     k ← 1
3:     while k ≤ k_max do
4:         c' ← BuscaLocal(c)
5:         if E(c') < E(c) then
6:             c ← c'
7:             k ← 1
8:         else
9:             k ← k + 1
10:        end if
11:    end while
12:    return c
13: end procedure
```

5.6.1 *Pull moves*

O *Pull moves* (LESH; MITZENMACHER; WHITESIDES, 2003) é um conjunto relativamente novo de movimentos, que vem sendo bastante utilizando em algoritmos para o problema da predição da estrutura protéica no modelo HP. Este conjunto possui algumas propriedades importantes como: completude, localidade e reversibilidade. Destas, a que mais nos interessa é a propriedade de completude, que diz que qualquer conformação válida pode ser obtida de uma outra através de uma sequência de movimentos. A seguir, mostramos, em uma grade bidimensional, como um movimento ocorre.

Considere um resíduo i no tempo t na posição $(x_i(t), y_i(t))$ e uma posição livre P_L adjacente a $(x_{i+1}(t), y_{i+1}(t))$ e diagonalmente adjacente a $(x_i(t), y_i(t))$ (Figura 13). As



(a) (b)

Figura 13 – Exemplo movimentos para resíduos não finais.

posições $(x_i(t), y_i(t))$, $(x_{i+1}(t), y_{i+1}(t))$ e P_L formam os três vértices de um quadrado. Seja P_Q o quarto vértice desse quadrado. Para que um movimento ocorra, a posição P_Q deve estar livre ou ser igual a $(x_{i-1}(t), y_{i-1}(t))$. Quando a posição P_Q é igual a $(x_{i-1}(t), y_{i-1}(t))$, o movimento é dito completo e o resíduo i é movido para a posição P_L (Figura 13 *a*). Quando a posição P_Q é livre, o resíduo i é movido para P_L e o resíduo $i-1$ é movido para P_Q (Figura 13 *b*). Neste último caso, enquanto uma conformação válida não for encontrada, é necessário que se faça

$$(x_j(t+1),\, y_j(t+1)) = (x_{j+2}(t),\, y_{j+2}(t))$$

do resíduo $j = i - 2$ até o resíduo 1. No exemplo da figura 13, os resíduos são movidos no sentido do último resíduo. O movimento pode ocorrer também no sentido inverso, isto é, no sentido do resíduo 1. Neste caso, a posição livre P_L é adjacente a $(x_{i-1}(t), y_{i-1}(t))$ e diagonalmente adjacente a $(x_i(t), y_i(t))$.

Vejamos o movimento para os resíduos finais. Considere duas posições P_L livres, com uma adjacente ao resíduo $m - 1$ e a outra diagonalmente adjacente a m (Figura 14). Para que um movimento ocorra, o resíduo m deve ser movido para qualquer uma das

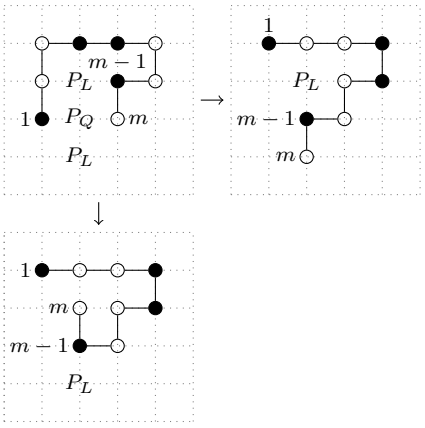


Figura 14 – Exemplo de movimentos para resíduos finais.

posições P_L livres e o resíduo $m - 1$ para a posição P_Q. Neste caso, também é necessário que se faça

$$(x_j(t+1),\, y_j(t+1)) = (x_{j+2}(t),\, y_{j+2}(t))$$

do resíduo $j = m - 2$ até o resíduo 1. Este último movimento também pode ser aplicado ao resíduo 1, mudando-se apenas o sentido do movimento.

Recentemente, Gyorffy, Zavodszky e Szilagyi (2012) mostraram que, em grade bidimensional quadrangular, o *Pull moves* não é completamente reversível, pois os movimentos de resíduos finais, que formam um gancho nas extremidades da conformação, são irreversíveis.

5.6.2 Determinando as posições P_L e P_Q

Nesta seção, mostramos como determinar as posições P_L e P_Q, para um resíduo i qualquer, nas duas grades utilizadas (bidimensional quadrangular e tridimensional cúbica).

Considere o quadrado $ABCD$ da figura 15, cujo lado mede duas unidades. Suponha que um resíduo i, $1 < i < m$, esteja no centro desse quadrado e que o resíduo $i + 1$ esteja no ponto médio M_{BC}. Neste caso, os vértices B e C correspondem às posições P_L e os pontos médios M_{AB} e M_{CD} correspondem às posições P_Q. Suponha, agora, que o resíduo $i + 1$ esteja em M_{AB}. Neste caso, A e B correspondem às posições P_L e os pontos médios M_{DA} e M_{BC}, às posições P_Q. Vejamos, agora, as posições P_L e P_Q quando

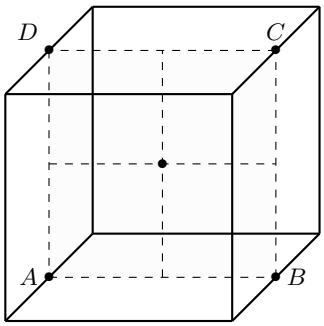


Figura 15 – As posições P_L e P_Q para um resíduo i em uma grade bidimensional quadrangular.

$i = 1$ ou $i = m$. Suponha que o resíduo m esteja no centro do quadrado $ABCD$ e que o resíduo $m - 1$ esteja em M_{DA}. Neste caso, os vértices A, B, C e D correspondem às posições P_L e os pontos médios M_{AB} e M_{CD}, às posições P_Q. Note que, quando i é um resíduo final sempre teremos quatro posições P_L e duas posições P_Q. Portanto, qualquer que seja o resíduo i posicionado no centro do quadrado $ABCD$, teremos sempre os vértices correspondendo as posições P_L e os pontos médios dos lados correspondendo as posições P_Q.

Para determinar as posições P_L e P_Q para um resíduo i qualquer, em uma grade tridimensional cúbica, usamos o mesmo raciocínio. Neste caso, o resíduo i deve estar no centro formado pela intersecção de três quadrados como mostra a figura 16.

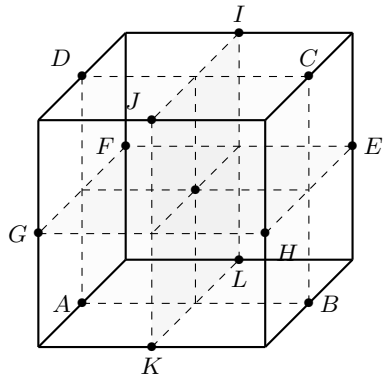


Figura 16 – As posições P_L e P_Q para um resíduo i em uma grade tridimensional cúbica.

5.6.3 Estrutura de vizinhança

A estrutura de vizinhança utilizada no procedimento VND (HANSEN; MLADENOVIĆ, 2003) é definida como: dada uma conformação $\boldsymbol{c}$, a vizinhança N dessa conformação é o conjunto

$$\{\boldsymbol{c'} \in C \mid \boldsymbol{c'} \text{ é obtido de } \boldsymbol{c}, \text{ movendo-se um resíduo } i \text{ para uma posição livre } P_L\}$$

onde C é o conjunto de conformações válidas. O algoritmo 9 mostra como esse conjunto é determinado.

Algoritmo 9 Encontra o melhor vizinho $\boldsymbol{c'} \in N(\boldsymbol{c})$.

```
1: procedure BuscaLocal(c)
2:     dir ← 1
3:     for k ← 1 to 2m do
4:         Selecione 1 ≤ i ≤ m, aleatoriamente
5:         if i = m then
6:             dir ← 1
7:             Determine P_L e P_Q usando c_{i−1}
8:         else if i = 1 then
9:             dir ← −1
10:            Determine P_L e P_Q usando c_{i+1}
11:        else
12:            Selecione dir ∈ {−1, 1}, aleatoriamente
13:            Determine P_L e P_Q usando c_{i+dir}
14:        end if
15:        Selecione 1 ≤ j ≤ r/2, aleatoriamente
16:        P_L ← v_{2j−1}
17:        if P_L estiver livre then
18:            c' ← c
19:            P_Q ← v_{2j}
20:            Move(c', P_L, P_Q, i, dir)
21:            if E(c') < E(c) then
22:                c ← c'
23:            end if
24:        end if
25:    end for
26:    return c
27: end procedure
```

Um resíduo i, $1 \leq i \leq m$, é aleatoriamente selecionado. Na subseção 5.6.1, vimos que um resíduo i pode ser movido no sentido do resíduo m ou no sentido do resíduo 1 e que, o movimento dos resíduos finais é sempre no sentido do resíduo selecionado. Seja $dir \in \{1, -1\}$ o sentido do movimento, tal que 1 representa o movimento no sentido do resíduo m e -1 representa o movimento no sentido do resíduo 1. Se $i = m$, então $dir = 1$ e as posições P_L e P_Q serão determinadas usando o resíduo $i - 1$ (ver subseção 5.6.2). Seja $\boldsymbol{v}$ um vetor de comprimento r, tal que $v_{2j-1} \in V$ e $v_{2j} \in M$, $\forall j = 1, \ldots, r$, onde V e M são, respectivamente, o conjunto de vértices e o conjunto de pontos médios dos lados de um quadrado de centro em c_i. As posições P_L e P_Q são, então, armazenadas em $\boldsymbol{v}$. Se $i = 1$, então $dir = -1$ e as posições P_L e P_Q serão determinadas usando o resíduo $i + 1$. Para todos os outros resíduos, isto é, $1 < i < m$, as posições P_L e P_Q serão determinadas usando o resíduo $i + dir$, sendo dir escolhido aleatoriamente. Por fim, uma posição P_L e uma posição P_Q são aletoriamente selecionadas em $\boldsymbol{v}$, e caso P_Q esteja livre, o resíduo i é movido (Algoritmo 10) para a posição P_L no sentido dir. O tamanho do conjunto N é igual a $2m$.

Algoritmo 10 Move o resíduo i para a posição livre P_L no sentido dir.

```
1: procedure MOVE(c, P_L, P_Q, i, dir)
2:     if P_Q estiver livre then
3:         t_1 ← c_i                          ▷ t é um vetor de três posições
4:         t_2 ← c_{i−dir}
5:         c_i ← P_L
6:         c_{i−dir} ← P_Q
7:         j ← i − 2dir
8:         while not VALIDA(c) do
9:             t_3 ← c_j
10:            c_j ← t_1
11:            j ← j − dir
12:            t_1 ← t_2
13:            t_2 ← t_3
14:        end while
15:    end if
16: end procedure
```

Neste trabalho, utilizamos apenas a posição P_L diagonalmente adjacente ao resíduo final. Dessa forma, evita-se os movimentos que formam ganchos nas extremidades da conformação.

5.7 Experimentos computacionais

Para testar o algoritmo proposto, utilizamos várias instâncias de referência (Tabelas 2 e 3) disponíveis na literatura. Os testes foram realizados em um computador com processador Intel Core $i5$ 3.10 GHz, 8GB de memória RAM e sistema operacional Ubuntu 10.04 LST Lucid Lynx. Os algoritmos foram implementados na linguagem Java.

Para cada uma das instâncias das tabelas 2 e 3, o algoritmo foi executado cinco vezes e o melhor valor encontrado foi utilizado como resultado. Repetimos esse procedimento para cada valor do parâmetro α no intervalo $(0, 1]$. O tempo de execução de cada valor encontrado corresponde à média aritmética das cinco execuções. Para as instâncias da tabela 2, utilizamos os seguintes parâmetros: número máximo de iterações (it_{max}) e número de estruturas de vizinhança (k_{max}) iguais a 100 para as instâncias 1, 2, 3 e 4; e $it_{max} = 10000$ e $k_{max} = 100$ para as outras instâncias. Para as instâncias da tabela 3, utilizamos os seguintes parâmetros: $it_{max} = 6000$ e $k_{max} = 100$.

Tabela 2 – Instâncias de referência para a modelo HP em grade bidimensional quadrangular. E - Melhor valor de energia conhecido.

Instância	Tamanho	E	Sequência
1	20	-9	HPHPPHHPHPPHPHHPPHPH
2	24	-9	HHPPHPPHPPHPPHPPHPPHPPHH
3	25	-8	PPHPPHHPPPPHHPPPPHHPPPPHH
4	36	-14	PPPHHPPHHPPPPPPHHHHHHHPPHHPPPPHHPPHPP
5	48	-23	PPHPPHHPPHHPPPPPPHHHHHHHHHHPPPPPPHHPPHHPPHPPHHHHH
6	50	-21	HHPHPHPHPHHHHPHPPPHPPPHPPPPHPPPHPPPHPHHHHPHPHPHPHH
7	60	-36	PPHHHPHHHHHHHHPPPHHHHHHHHHHPHPPPHHHHHHHHHHHHPPPPHHHHHHPHHPHP
8	64	-42	HHHHHHHHHHHHPHPHPPHHPPHHPPHPPHHPPHHPPHPPHHPPHPPHHPPHPHPHHHHHHHHHHHH
9	85	-53	HHHHPPPPHHHHHHHHHHHHPPPPPPHHHHHHHHHHHHPPPHHHHHHHHHHHPPPHHHHHHHHHHHPPPHPPHHPPHHPPHPH
10	100	-50	PPPHHPPHHHHPPHHHPHHPHPHHHHPPPPPPPHHHHHHPPHHHHHHPPPPPPPPPHPHHPHHHHHHHHHHPPHHHPHHPHP PHPHHHPPPPPPHHH
11	100	-48	PPPPPPHPHHPPPPPHHHPHHHHHPHPPPPHHPPHPHHHHHPHHHHHHHHHHPHHPHHHHHHPPPPPPPPPPPHHHHHHPP HPHHHPPPPPPHPHH

Fonte: Shmygelska e Hoos (2005).

Tabela 3 – Instâncias de referência para o modelo HP em grade tridimensional cúbica. E - Melhor valor de energia conhecido.

Instância	Tamanho	E	Sequência
1	48	−32	HPHHPPHHHHPHHHPPHHPPHPHHHPHPHHPPHHPPPHPPPPPPPPHH
2	48	−34	HHHHPHHPHHHHHPPHPPHHPPHPPPPPPHPPHPPPHPPHHPPHHHPH
3	48	−34	PHPHHPHHHHHHPPHPHPPHPHHPHPHPPPHPPHHPPHHPPHPHPPHP
4	48	−33	PHPHHPPHPHHHPPHHPHHPPPHHHHHPPHPHHPHPHPPPPHPPHPHP
5	48	−32	PPHPPPHPHHHHPPHHHHPHHPHHHPPHPHPHPPHPPPPPPHHPHHPH
6	48	−32	HHHPPPHHPHPHHPHHPHHPHPPPPPPPHPHPPHPPPHPPHHHHHHPH
7	48	−32	PHPPPPHPHHHPHPHHHHPHHPHHPPPHPHPPPHHHPPHHPPHHPPPH
8	48	−31	PHHPHHHPHHHHPPHHHPPPPPPHPHHPPHHPHPPPHHPHPHPHHPPP
9	48	−34	PHPHPPPPHPHPHPPHPHHHHHHPPHHHPHPPHPHHPPHPHHHPPPPH
10	48	−33	PHHPPPPPPHHPPPHHHPHPPHPHHPPHPPHPPHHPPHHHHHHHPPHH

Fonte: Shmygelska e Hoos (2005).

6 Resultados e discussões

Neste capítulo, apresentamos os resultados dos experimentos computacionais para os dois modelos HP utilizados.

6.1 Modelo HP em grade bidimensional quadrangular

Os resultados dos experimentos computacionais para as instâncias da tabela 2 são mostrados na tabela 4. O algoritmo HG foi capaz de encontrar o ótimo global para as intâncias 1, 2, 3, 4 e 6. Para as instâncias 5, 7 e 9, os valores encontrados ficaram muito próximos do ótimo global conhecido (Figura 17). O tempo médio necessário para encontrar o ótimo para as instâncias 1, 2, 3 e 4 ficou abaixo de 1 segundo, mostrando, portanto, ser bastante eficiente para instâncias pequenas.

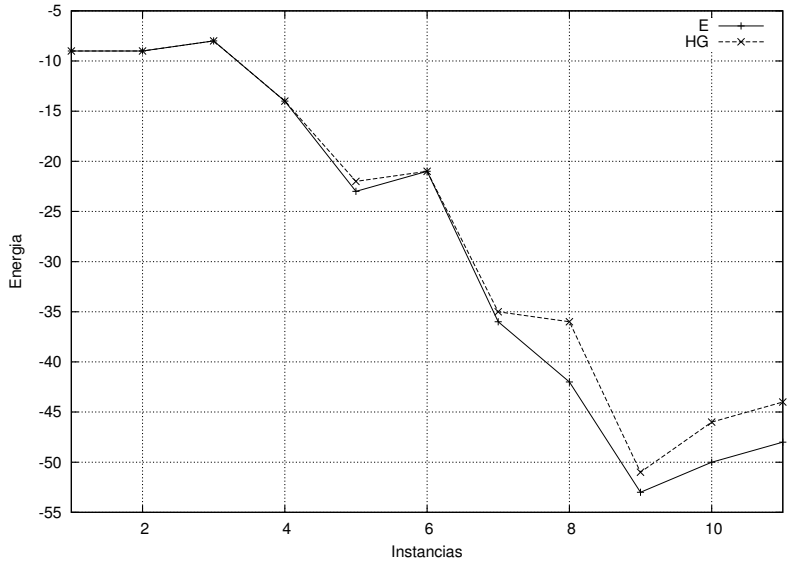


Figura 17 – Comparação entre os melhores valores conhecidos e os melhores valores encontrados pelo algoritmo HG para o modelo HP em grade bidimensional quadrangular.

A tabela 5 mostra um comparativo entre os resultados encontrados pelo algoritmo HG e os resultados de outros trabalhos para este modelo.

Tabela 4 – Valores e os respectivos tempos, em segundos, encontrados pelo algoritmo HG para cada valor do parâmetro α.

Instância	Tamanho	α									
		0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1,0
1	20	**−9** (0,30)	**−9** (0,28)	**−9** (0,28)	−8 (0,28)	**−9** (0,28)	**−9** (0,28)	**−9** (0,28)	**−9** (0,29)	**−9** (0,28)	**−9** (0,31)
2	24	−8 (0,32)	**−9** (0,33)	−8 (0,33)	−8 (0,34)	**−9** (0,35)	**−9** (0,34)	−8 (0,34)	**−9** (0,35)	**−9** (0,35)	**−9** (0,38)
3	25	−7 (0,43)	−7 (0,42)	−7 (0,42)	−7 (0,41)	**−8** (0,42)	−7 (0,42)	**−8** (0,42)	−7 (0,41)	−7 (0,43)	**−8** (0,47)
4	36	−13 (0,74)	−12 (0,73)	−13 (0,74)	**−14** (0,76)	−13 (0,78)	−13 (0,76)	−13 (0,78)	−12 (0,81)	−12 (0,79)	−12 (1,02)
5	48	−21 (135,21)	−20 (136,98)	−22 (136,50)	−22 (136,74)	−21 (137,86)	−21 (139,02)	−20 (137,82)	−21 (133,27)	−22 (126,08)	−20 (199,47)
6	50	−18 (133,17)	−19 (132,98)	−19 (130,03)	−19 (128,60)	**−21** (134,60)	−18 (140,43)	**−21** (141,15)	−20 (140,99)	−18 (136,60)	−17 (204,17)
7	60	−35 (356,92)	−35 (357,22)	−35 (360,13)	−35 (357,59)	−35 (355,22)	−35 (355,91)	−35 (329,95)	−35 (271,15)	−34 (190,56)	−35 (350,16)
8	64	−36 (235,38)	−34 (238,42)	−35 (252,61)	−34 (257,09)	−34 (263,64)	−34 (269,92)	−34 (270,53)	−34 (249,25)	−34 (197,44)	−34 (363,24)
9	85	−50 (601,21)	−51 (603,72)	−51 (597,47)	−50 (591,82)	−49 (574,83)	−50 (577,28)	−51 (525,01)	−51 (433,44)	−48 (257,02)	−49 (570,98)
10	100	−44 (396,56)	−42 (400,30)	−44 (411,64)	−44 (416,50)	−46 (433,11)	−43 (440,07)	−42 (429,12)	−43 (375,01)	−40 (283,68)	−41 (617,34)
11	100	−44 (402,42)	−43 (416,48)	−43 (414,94)	−41 (422,05)	−42 (412,73)	−43 (417,58)	−41 (401,53)	−41 (368,44)	−42 (259,27)	−40 (630,80)

Tabela 5 – Comparativo entre os melhores valores encontrados pelo algoritmo HG e os valores encontrados por outros algoritmos para o modelo HP em grade bidimensional quadrangular. E - Melhor valor conhecido. AG - Algoritmo genético de Unger e Moult (1993b). EMC - Algoritmo evolucionário de Monte Carlo de Liang e Wong (2001). CF - Algoritmo de otimização por colônia de formigas de Shmygelska e Hoos (2005).

Instância	Tamanho	E	AG	EMC	CF	HG
1	20	**−9**	**−9**	**−9**	**−9**	**−9**
2	24	**−9**	**−9**	**−9**	**−9**	**−9**
3	25	**−8**	**−8**	**−8**	**−8**	**−8**
4	36	**−14**	**−14**	**−14**	**−14**	**−14**
5	48	**−23**	−22	**−23**	**−23**	−22
6	50	**−21**	**−21**	**−21**	**−21**	**−21**
7	60	**−36**	−34	−35	**−36**	−35
8	64	**−42**	−37	−39	**−42**	−36
9	85	**−53**			**−53**	−51
10	100	**−50**			−49	−46
11	100	**−48**			−47	−44

6.2 Modelo HP em grade tridimensional cúbica

Os resultados dos experimentos computacionais para as instâncias da tabela 3 são mostrados na tabela 6. O algoritmo HG foi capaz de encontrar o ótimo global apenas para a intância 5, ficando os outros valores muito próximos do ótimo global conhecido (Figura 18). O tempo médio de execução para cada instância foi de aproximadamente 5 minutos.

Tabela 6 – Valores e os respectivos tempos, em segundos, encontrados pelo algoritmo HG para cada valor do parâmetro α.

Instância	Tamanho	α									
		0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1,0
1	48	−31 (396,69)	−31 (400,87)	−30 (390,62)	−30 (389,84)	−30 (390,65)	−30 (393,28)	−30 (395,19)	−30 (397,53)	−30 (396,99)	−30 (416,00)
2	48	−31 (345,58)	−31 (345,15)	−32 (346,89)	−31 (349,25)	−32 (352,94)	−32 (356,96)	−31 (363,47)	−31 (371,76)	−30 (385,64)	−31 (420,22)
3	48	−32 (326,41)	−31 (327,86)	−32 (328,27)	−32 (328,88)	−32 (330,59)	−31 (332,66)	−33 v337,60)	−32 (343,44)	−32 (350,75)	−30 (378,23)
4	48	−30 (342,62)	−30 (342,71)	−30 (342,35)	−30 (343,37)	−31 (344,62)	−30 (344,67)	−31 (347,70)	−30 (352,61)	−30 (356,62)	−30 (375,70)
5	48	−30 (362,48)	−31 (362,86)	−30 (363,81)	−30 (364,78)	−31 (368,10)	**−32** (370,03)	−30 (376,49)	−31 (381,89)	−30 (389,11)	−31 (409,28)
6	48	−29 (359,82)	−30 (359,25)	−30 (360,90)	−30 (362,58)	−30 (365,79)	−30 (367,65)	−29 (372,93)	−29 (382,21)	−30 (392,68)	−29 (427,16)
7	48	−30 (331,73)	−29 (331,74)	−30 (331,22)	−29 (333,18)	−30 (333,86)	−29 (337,18)	−29 (340,41)	−30 (346,92)	−29 (355,30)	−28 (379,31)
8	48	−30 (351,35)	−28 (350,46)	−29 (351,26)	−29 (353,29)	−29 (354,31)	−29 (358,96)	−29 (364,79)	−30 (370,79)	−29 (381,95)	−29 (407,43)
9	48	−31 (334,63)	−33 (334,14)	−32 (335,67)	−31 (337,44)	−31 (338,19v	−31 (340,85)	−31 (344,14)	−31 (348,50)	−32 (355,22)	−31 (368,95)
10	48	−30 (321,64)	−31 (321,96)	−31 (324,33)	−31 (327,07)	−31 (325,03)	−31 (327,31)	−32 (334,10)	−32 (343,16)	−31 (352,52)	−30 (381,56)

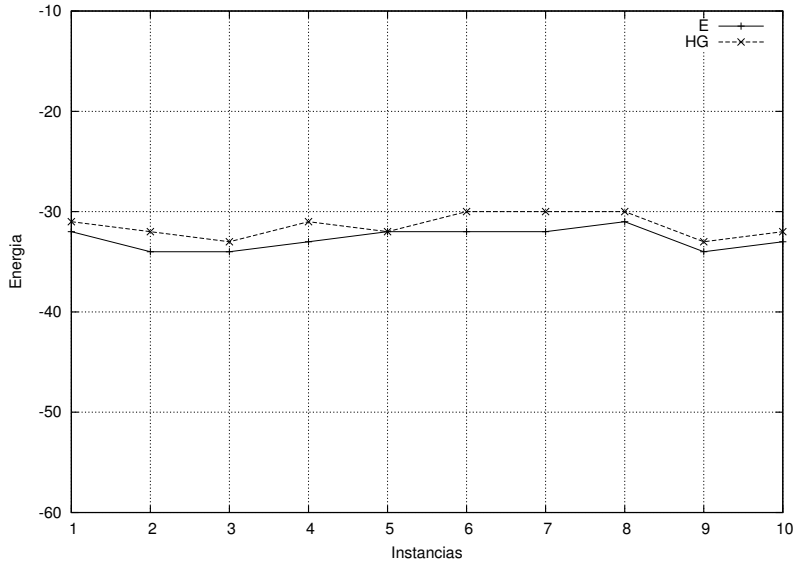


Figura 18 – Comparação entre os melhores valores conhecidos e os melhores valores encontrados pelo algoritmo HG para o modelo HP em grade tridimensional cúbica.

A tabela 7 mostra um comparativo entre os resultados encontrados pelo algoritmo HG e os resultados de outros trabalhos para este modelo.

Tabela 7 – Comparativo entre os melhores valores encontrados pelo algoritmo HG e os valores encontrados por outros algoritmos para o modelo HP tridimensional em grade cúbica. E - Melhor valor conhecido. MC - Algoritmo de Monte Carlo. HZ - Método *Hydrophobic Zipper* de Fiebig e Dill (1993). CG - Método exato *Core-directed chain Growth* de Beutler e Dill (1996).

Instância	Tamanho	E	MC	HZ	CG	HG
1	48	**−32**	−30	−31	**−32**	−31
2	48	**−34**	−30	−32	**−34**	−32
3	48	**−34**	−31	−31	**−34**	−33
4	48	**−33**	−30	−30	**−33**	−31
5	48	**−32**	−30	−30	**−32**	**−32**
6	48	**−32**	−30	−29	**−32**	−30
7	48	**−32**	−31	−29	**−32**	−30
8	48	**−31**	**−31**	−29	**−31**	−30
9	48	**−34**	−30	−31		−33
10	48	**−33**	−30	**−33**	**−33**	−32

7 Conclusões

Neste trabalho, apresentamos uma heurística GRASP híbrida para o problema da predição da estrutura de proteínas utilizando o modelo Hidrofóbico-Polar. Para as instâncias do modelo HP em grade bidimensional quadrangular, o método proposto apresentou desempenho muito semelhante ao do algoritmo genético de Unger e Moult (1993b) e ao do algoritmo evolucionário de Monte Carlo de Liang e Wong (2001). Contudo, apresentou desempenho inferior ao do algoritmo de otimização por colônia de formigas de Shmygelska e Hoos (2005), principalmente em relação as instâncias maiores. Para as instâncias do modelo HP em grade tridimensional cúbica, o método proposto apresentou desempenho superior tanto ao do algoritmo de Monte Carlo quanto ao método *Hydrophobic Zipper* de Fiebig e Dill (1993). Entretanto, apresentou desempenho inferior ao do método exato *Core-directed chain Growth* de Beutler e Dill (1996).

Como trabalho futuro, podemos propor um estudo sobre a utilização de outras funções de energia e a utilização de movimentos irreversíveis na fase de busca local. Além disso, este trabalho pode servir como ponto de partida para o desenvolvimento de métodos semelhantes para outros modelos em grade ou para o problema da busca conformacional em modelos mais detalhados.

Referências Bibliográficas

ALBERTS, B. et al. **Molecular Biology of the Cell**. 5. ed. [S.l.]: Garland Science, 2007. Hardcover. ISBN 0815341059.

ANFINSEN, C. B. Principles that govern the folding of protein chains. **Science**, AAAS, v. 181, n. 96, p. 223–230, 1973. Disponível em: <http://www.ncbi.nlm.nih.gov/pubmed/4124164>.

BALDWIN, R. L.; ROSE, G. D. Is protein folding hierarchic? i. local structure and peptide folding. **Trends in Biochemical Sciences**, v. 24, n. 1, p. 26 – 33, 1999. ISSN 0968-0004. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0968000498013462>.

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**. New York, NY, USA: ACM, 2008. (GECCO '08), p. 1097–1098. ISBN 978-1-60558-130-9. Disponível em: <http://doi.acm.org/10.1145/1389095.1389296>.

BERGER, B.; LEIGHTON, T. Protein folding in the hydrophobic-hydrophilic (hp) model is np-complete. **Journal of Computational Biology**, v. 5, n. 1, p. 27–40, 1998.

BEUTLER, T. C.; DILL, K. A. A fast conformational search strategy for finding low energy structures of model proteins. **Protein Science**, v. 5, p. 2037–2043, 1996.

BLAZEWICZ, J. et al. A tabu search strategy for finding low energy structures of proteins in hp-model. **Computational Methods in Science and Technology**, v. 10, p. 7–19, 2004.

COOK, S. A. The complexity of theorem-proving procedures. In: **Proceedings of the third annual ACM symposium on Theory of computing**. New York, NY, USA: ACM, 1971. (STOC '71), p. 151–158. Disponível em: <http://doi.acm.org/10.1145/800157.805047>.

CRESCENZI, P. et al. On the complexity of protein folding. **Journal of Computational Biology**, v. 5, p. 597–603, 1998.

CRIPPEN, G. M. The tree structural organization of proteins. **Journal of Molecular Biology**, v. 126, n. 3, p. 315 – 332, 1978. ISSN 0022-2836. Disponível em: <http://www.sciencedirect.com/science/article/pii/0022283678900438>.

DILL, K. A. Theory for the folding and stability of globular proteins. **Biochemistry**, American Chemical Society, v. 24, n. 6, p. 1501–1509, mar 1985. Disponível em: <http://dx.doi.org/10.1021/bi00327a032>.

DILL, K. A. et al. Principles of protein folding - a perspective from simple exact models. **Protein Science**, v. 4, n. 4, p. 561–602, 1995. ISSN 1469-896X. Disponível em: <http://dx.doi.org/10.1002/pro.5560040401>.

DILL, K. A.; FIEBIG, K. M.; CHAN, H. S. Cooperativity in protein-folding kinetics. **Proceedings of the National Academy of Sciences**, v. 90, n. 5, p. 1942–1946, 1993. Disponível em: <http://www.pnas.org/content/90/5/1942.abstract>.

DILL, K. A. et al. The protein folding problem: when will it be solved? **Current Opinion in Structural Biology**, v. 17, n. 3, p. 342–346, 2007. ISSN 0959-440X. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0959440X07000772>.

FIEBIG, K. M.; DILL, K. A. Protein core assembly processes. **The Journal of Chemical Physics**, AIP, v. 98, n. 4, p. 3475–3487, 1993. Disponível em: <http://link.aip.org/link/?JCP/98/3475/1>.

GAREY, M. R.; JOHNSON, D. S. **Computers and Intractability: A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences)**. First edition. [S.l.]: W. H. Freeman, 1979. Paperback. ISBN 0716710455.

GLOVER, F. Future paths for integer programming and links to artificial intelligence. **Comput. Oper. Res.**, Elsevier Science Ltd., Oxford, UK, UK, v. 13, n. 5, p. 533–549, may 1986. ISSN 0305-0548. Disponível em: <http://dx.doi.org/10.1016/0305-0548(86)90048-1>.

GYORFFY, D.; ZAVODSZKY, P.; SZILAGYI, A. Pull moves for rectangular lattice polymer models are not fully reversible. **IEEE/ACM Transactions on Computational Biology and Bioinformatics**, IEEE Computer Society, Los Alamitos, CA, USA, v. 9, n. 6, p. 1847–1849, 2012. ISSN 1545-5963.

HANSEN, P.; MLADENOVIĆ, N. Variable neighborhood search. In: GLOVER, F.; KOCHENBERGER, G. (Ed.). **Handbook of Metaheuristics**. [S.l.]: Springer New York, 2003, (International Series in Operations Research & Management Science, v. 57). p. 145–184. ISBN 978-0-306-48056-0.

HOPCROFT, J. E. et al. **Introduction to Automata Theory, Languages and Computability**. 2nd. ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2000. ISBN 0201441241.

KOPP, J.; SCHWEDE, T. Automated protein structure homology modeling: a progress report. **Pharmacogenomics**, v. 5, n. 4, p. 405–416, 2004. ISSN 1462-2416. Disponível em: <http://dx.doi.org/10.1517/14622416.5.4.405>.

KRASNOGOR, N. et al. Multimeme algorithms for protein structure prediction. In: **Proceedings of the 7th International Conference on Parallel Problem Solving from Nature**. London, UK: Springer-Verlag, 2002. (PPSN VII), p. 769–778. ISBN 3-540-44139-5. Disponível em: <http://dl.acm.org/citation.cfm?id=645826.669429>.

KRASNOGOR, N. et al. Protein structure prediction with evolutionary algorithms. In: BANZHAF, D. E. G. H. J.; SMITH (Ed.). **International Genetic and Evolutionary Computation Conference (GECCO99)**. Morgan Kaufmann, 1999. p. 1569–1601. Disponível em: <http://www.cs.nott.ac.uk/~nxk/PAPERS/gecco99.pdf>.

LAU, K. F.; DILL, K. A lattice statistical mechanics models of the conformational and sequence spaces of proteins. **Macromolecules**, v. 22, p. 3986–3997, 1989.

LEHNINGER, A.; NELSON, D. L.; COX, M. M. **Lehninger Principles of Biochemistry**. Fifth edition. [S.l.]: W. H. Freeman, 2008. Hardcover.

LESH, N.; MITZENMACHER, M.; WHITESIDES, S. A complete and effective move set for simplified protein folding. In: **Proceedings of the seventh annual international conference on Research in computational molecular biology**. New York, NY, USA: ACM, 2003. (RECOMB '03), p. 188–195. ISBN 1-58113-635-8. Disponível em: <http://dx.doi.org/10.1145/640075.640099>.

LESK, A. M.; CHOTHIA, C. How different amino acid sequences determine similar protein structures: The structure and evolutionary dynamics of the globins. **Journal of Molecular Biology**, v. 136, n. 3, p. 225 – 270, 1980. ISSN 0022-2836. Disponível em: <http://www.sciencedirect.com/science/article/pii/0022283680903733>.

LEVINTHAL, C. Are there pathways for protein folding? **Extrait du Journal de Chimie Physique**, v. 65, n. 1, 1968.

LIANG, F.; WONG, W. H. Evolutionary monte carlo for protein folding simulations. **The Journal of Chemical Physics**, v. 115, n. 7, p. 3374, 2001. Disponível em: <http://link.aip.org/link/JCPSA6/v115/i7/p3374/s1Agg=doi>.

MARIEB, E. N. **Human anatomy & physiology**. 5th ed.. ed. San Francisco: Benjamin Cummings, 2001.

METROPOLIS, N. et al. Equation of state calculations by fast computing machines. **The Journal of Chemical Physics**, AIP, v. 21, n. 6, p. 1087–1092, 1953. ISSN 00219606. Disponível em: <http://dx.doi.org/10.1063/1.1699114>.

PAPADIMITRIOU, C. H.; STEIGLITZ, K. **Combinatorial Optimization: Algorithms and Complexity**. [S.l.]: Dover Publications, 1998. Paperback. ISBN 0486402584.

PATTON, A. L. et al. A standard ga approach to native protein conformation prediction. In: **Proceedings of the Sixth International Conference on Genetic Algorithms**. [S.l.]: Morgan Kaufmann, 1995. p. 574–581.

PAULING, L.; COREY, R. B.; BRANSON, H. R. The structure of proteins: Two hydrogen-bonded helical configurations of the polypeptide chain. **Proceedings of the National Academy of Sciences**, v. 37, n. 4, p. 205–211, 1951.

PóLYA, G. **How to Solve It**. First edition. [S.l.]: Princeton University Press, 1945.

RAMACHANDRAN, G. N.; RAMAKRISHNAN, C.; SASISEKHARAN, V. Stereochemistry of polypeptide chain configurations. **Journal of molecular biology**, v. 7, p. 95–99, jul 1963. ISSN 0022-2836. Disponível em: <http://view.ncbi.nlm.nih.gov/pubmed/13990617>.

RESENDE, M.; RIBEIRO, C. Greedy randomized adaptive search procedures. In: GLOVER, F.; KOCHENBERGER, G. (Ed.). **Handbook of Metaheuristics**. [S.l.]: Kluwer Academic Publishers, 2002. p. 219–249.

ROMANYCIA, M. H. J.; PELLETIER, F. J. What is a heuristic? **Computational Intelligence**, Blackwell Publishing Ltd, v. 1, n. 1, p. 47–58, 1985. ISSN 1467-8640. Disponível em: <http://dx.doi.org/10.1111/j.1467-8640.1985.tb00058.x>.

ROTHLAUF, F. **Design of Modern Heuristics: Principles and Application**. [S.l.]: Springer Berlin Heidelberg, 2011. (Natural computing series). ISBN 9783540729624.

ROY, A.; ZHANG, Y. Protein structure prediction. **eLS**, John Wiley & Sons, Ltd, 2012.

SHMYGELSKA, A.; HOOS, H. An improved ant colony optimisation algorithm for the 2D HP protein folding problem. In: XIANG, Y.; CHAIB-DRAA, B. (Ed.). **Advances in Artificial Intelligence**. [S.l.]: Springer Berlin Heidelberg, 2003, (Lecture Notes in Computer Science, v. 2671). p. 400–417. ISBN 978-3-540-40300-5.

SHMYGELSKA, A.; HOOS, H. H. An ant colony optimisation algorithm for the 2D and 3D hydrophobic polar protein folding problem. **BMC Bioinformatics**, v. 6, p. 30, 2005.

SIMONS, K. T. et al. Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and bayesian scoring functions. **Journal of Molecular Biology**, v. 268, n. 1, p. 209 – 225, 1997. ISSN 0022-2836. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0022283697909591>.

THACHUK, C.; SHMYGELSKA, A.; HOOS, H. A replica exchange monte carlo algorithm for protein folding in the hp model. **BMC Bioinformatics**, v. 8, n. 1, p. 342, 2007. ISSN 1471-2105. Disponível em: <http://www.biomedcentral.com/1471-2105/8/342>.

TOMA, L.; TOMA, S. Contact interactions method: A new algorithm for protein folding simulations. **Protein Science**, Cold Spring Harbor Laboratory Press, v. 5, n. 1, p. 147–153, 1996. ISSN 1469-896X. Disponível em: <http://dx.doi.org/10.1002/pro.5560050118>.

UNGER, R.; MOULT, J. A genetic algorithm for 3d protein folding simulations. In: FORREST, S. (Ed.). **Proc. of the Fifth Int. Conf. on Genetic Algorithms,**. [S.l.]: Morgan Kaufmann, San Mateo, CA., 1993. p. 581–588.

UNGER, R.; MOULT, J. Genetic algorithms for protein folding simulations. **J. Mol. Biol.**, v. 231, p. 75–81, 1993.

URBANC LUIS CRUZ, D. B. T. B.; STANLEY, H. E. Computer simulations of alzheimers amyloid beta-protein folding and assembly. **Current Alzheimer Research**, v. 3, p. 493–504, 2006.

XU, D.; ZHANG, Y. Ab initio protein structure assembly using continuous structure fragments and optimized knowledge-based force field. **Proteins: Structure, Function, and Bioinformatics**, Wiley Subscription Services, Inc., A Wiley Company, v. 80, n. 7, p. 1715–1735, 2012. ISSN 1097-0134. Disponível em: <http://dx.doi.org/10.1002/prot.24065>.