DEFINIÇÃO DE ORDENS NAS CADEIAS LATERAIS DE PROTEÍNAS PARA CÁLCULOS COMBINATÓRIOS DE ESTRUTURAS TRIDIMENSIONAIS

Virginia Silva da Costa

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia de Sistemas e Computação.

Orientadores: Nelson Maculan Filho
Luiz Mariano Paes de Carvalho Filho

Rio de Janeiro
Fevereiro de 2013

DEFINIÇÃO DE ORDENS NAS CADEIAS LATERAIS DE PROTEÍNAS PARA CÁLCULOS COMBINATÓRIOS DE ESTRUTURAS TRIDIMENSIONAIS

Virginia Silva da Costa

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Examinada por:

Prof. Nelson Maculan Filho, D.Habil.

Prof. Luiz Mariano Paes de Carvalho Filho, Ph.D.

Prof. Carlile Campos Lavor, Ph.D.

Prof. Antonio Mucherino, Ph.D.

Prof. Carlos Antonio de Moura, Ph.D.

Prof[a]. Márcia Helena Costa Fampa, Ph.D.

RIO DE JANEIRO, RJ – BRASIL
FEVEREIRO DE 2013

Costa, Virginia Silva da

Definição de ordens nas cadeias laterais de proteínas para cálculos combinatórios de estruturas tridimensionais/Virginia Silva da Costa. – Rio de Janeiro: UFRJ/COPPE, 2013.

XII, 127 p.: il.; 29, 7cm.

Orientadores: Nelson Maculan Filho

Luiz Mariano Paes de Carvalho Filho

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2013.

Referências Bibliográficas: p. 110 – 115.

1. Molecular Distance Geometry Problem. 2. Branch-and-Prune. 3. Proteína. I. Maculan Filho, Nelson *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

Ao meu maior tesouro, minha querida filha Vitória da Costa Cunha.

Agradecimentos

A meus pais, Álvaro da Costa e Genilda Maria da Silva, que fizeram tudo o que podiam para que eu estudasse e me tornasse a cidadã que sou hoje.

A Carlos Roberto da Cunha, companheiro de anos, meu melhor amigo, que acompanhou todo o desenvolvimento deste e de muitos trabalhos, bem de perto, sempre ao meu lado, cuidando de mim.

A meus orientadores, Prof. Nelson Maculan Filho, que me recebeu de braços abertos como sua aluna na COPPE e a quem tenho a mais profunda admiração, e Prof. Luiz Mariano Paes de Carvalho Filho, que me acompanha desde a minha iniciação científica e presenciou diversas fases do meu desenvolvimento.

Ao Prof. Carlile Campos Lavor, que tem me acompanhado, juntamente com os meus orientadores, na elaboração desta tese.

Ao Prof. Antonio Mucherino, que me recebeu na Universidade de Rennes I durante o período sanduíche de meu doutorado.

À Maria de Fátima Cruz Marques, à Profª. Ana Flávia Uzeda e à Profª. Laura Patuzzi, pelos momentos agradáveis que passamos na UFRJ.

Às secretárias e funcionários do Programa de Engenharia de Sistemas e Computação.

A todos, sou muito grata.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

DEFINIÇÃO DE ORDENS NAS CADEIAS LATERAIS DE PROTEÍNAS PARA CÁLCULOS COMBINATÓRIOS DE ESTRUTURAS TRIDIMENSIONAIS

Virginia Silva da Costa

Fevereiro/2013

Orientadores: Nelson Maculan Filho
Luiz Mariano Paes de Carvalho Filho

Programa: Engenharia de Sistemas e Computação

As proteínas são as mais versáteis macromoléculas presentes em sistemas vivos e desempenham funções essenciais em todos os processos biológicos. O conhecimento sobre as funções desempenhadas por uma proteína está relacionado à determinação de sua estrutura molecular. Através da ressonância magnética nuclear (RMN) é possível se obter um conjunto de distâncias entre pares de átomos de uma molécula, que pode ser utilizado na obtenção de sua estrutura tridimensional, através da resolução de um problema conhecido como problema geométrico da distância molecular (MDGP, de Molecular Distance Geometry Problem). Em geral, o MDGP é resolvido por métodos de otimização global que fazem uma busca no espaço contínuo $\mathbb{R}^3$. A fim de se formular uma abordagem discreta para o MDGP, que considere distâncias intervalares obtidas a partir de experimentos de RMN, e de descrever um algoritmo eficiente para resolvê-la, em [1, 2], o Interval Discretizable MDGP (iDMDGP) e o algoritmo branch-and-prune intervalar (iBP) foram propostos. Em tal abordagem, algumas condições devem ser satisfeitas para a obtenção de um domínio de busca combinatório. Para satisfazer estas condições, em [3], uma ordenação para os átomos da cadeia principal de proteínas foi apresentada. Neste contexto, tendo em vista que as cadeias laterais de proteínas são responsáveis por distinguir um aminoácido de outro, conferindo-lhes as suas propriedades químicas, no presente trabalho, o conceito de ordem é estendido para as cadeias laterais e são propostas algumas técnicas computacionais para sua implementação.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

DEFINITION OF PROTEIN SIDE CHAINS ORDERS FOR COMBINATORIAL CALCULUS OF THREE-DIMENSIONAL STRUCTURE

Virginia Silva da Costa

February/2013

Advisors: Nelson Maculan Filho
Luiz Mariano Paes de Carvalho Filho

Department: Systems Engineering and Computer Science

Proteins are important molecules and they are widely studied in biology. Due to their three-dimensional conformations give clues about their function, an optimal methodology for the identification of such conformations has been researched for many years. Nuclear Magnetic Resonance (NMR) Experiments are able to estimate distances between some pairs of atoms which form the protein, and the problem of identifying the possible conformations that satisfy the available distance constraints is known in the scientific literature as the Molecular Distance Geometry Problem (MDGP). In order to formulate a discretized approach for the MDGP, which considers interval data from NMR experiments, and to describe an efficient algorithm to solve it, in [1, 2], the Interval Discretizable MDGP (iDMDGP) and the Interval Branch & Prune (iBP) algorithm were proposed. In such approach, some conditions must be satisfied for obtaining a combinatorial and discretized search domain, represented by a search tree. To satisfy these conditions, in [3], a special hand-craft atomic order was shown for protein backbone atoms. In this context, considering that protein side chains are responsible for distinguishing an amino acid from the other, due to they give them their chemical properties, in this work, the concept of order is extended for the side chains and techniques for its implementation are given.

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

As proteínas são as mais versáteis macromoléculas presentes em sistemas vivos e desempenham funções essenciais em todos os processos biológicos. O conhecimento sobre as funções desempenhadas pelas proteínas está relacionado à obtenção de mapas tridimensionais mais precisos.

A introdução da *ressonância magnética nuclear* (RMN) [4] como técnica para determinação de estruturas de proteínas [5] tornou possível a obtenção de estruturas com um grau mais elevado de precisão em um ambiente muito mais próximo do encontrado em um organismo vivo – mais próximo do que o ambiente requerido por um único cristal na cristalografia de uma proteína [6]. Embora as medições de proteínas via RMN sejam utilizadas com o objetivo de determinar suas estruturas tridimensionais, esta técnica não produz uma imagem, uma molécula claramente definida. O que se tem é um conjunto de informações estruturais que permite, através de modelos matemáticos e cálculos extensivos, determinar a geometria da molécula em questão. Dentro desse conjunto, a principal informação medida consiste de uma rede de *distâncias* entre átomos, em sua maioria de hidrogênio, espacialmente próximos. Desta forma, para se determinar a estrutura tridimensional de uma proteína, é preciso resolver um problema que se inicia a partir de distâncias conhecidas entre pares de átomos. A este problema, dá-se o nome de *problema geométrico da distância molecular* (*MDGP*, isto é, *molecular distance geometry problem*) [7], que será visto no Capítulo 3 deste texto.

São muitos os métodos que se propõem a resolver o MDGP. Alguns são baseados em buscas no espaço contínuo $\mathbb{R}^3$. Neste caso, pode-se citar como exemplos os métodos *spatial branch-and-bound* (*sBB*) [8, 9], *variable neighborhood search* (*VNS*) [10] e o *double VNS with smoothing* [11], o algoritmo heurístico multistart *SobolOpt* [12], o algoritmo *DC Optimization* [13, 14], o *alternating projections algorithm* (*APA*) [15], o algoritmo *geometric build-up* (*GB*) [16], o método iterativo *GNOMAD* [17], o algoritmo *monotonic basin hopping* (*MDH*) [18], o método por programação semidefinida [19] e o *stochasting proximity embedding* (*SPE*) [20]. Ainda é possível

citar algumas formulações e métodos discretos para o MDGP, tais como o *ABBIE* [21] e o *discretizable molecular distance geometry problem* (*DMDGP*) [22, 23]. Esse último método é uma reformulação combinatorial para o MDGP que, apesar de ser considerado um problema NP-difícil, segundo [22], pode ser tratado de forma a ser resolvido pela aplicação do algoritmo *branch-and-prune* (BP) [22, 23].

No presente trabalho, essa reformulação será vista com especial atenção, principalmente no que se refere a técnicas computacionais capazes de tornar o DMDGP factível, utilizando informações conhecidas sobre a estrutura molecular da proteína a ser determinada, tais como: comprimento de ligações covalentes e ângulos de ligação.

Inicialmente, é vista uma descrição detalhada do BP clássico e uma breve comparação com o algoritmo *revised updated geometric build-up* (RUGB, uma variação do geometric build-up já citado acima), onde os dois algoritmos foram executados a fim de se determinar as coordenadas dos átomos das cadeias principais de um conjunto de proteínas. O objetivo dessa comparação é mostrar o potencial do BP diante de dificuldades como, por exemplo, sistemas mal condicionados presentes no RUGB. Em seguida, este trabalho se direciona a uma nova abordagem para o BP, iniciando uma investigação sobre as modificações feitas no DMDGP, visando à inclusão das cadeias laterais dos aminoácidos-padrão. Devido a algumas limitações existentes nos dados oriundos da RMN, foi proposto em [1, 2] uma nova versão do BP, o *branch-and-prune intervalar* (*i*BP, de *interval branch-and-prune*). Nessa versão, consideram-se não somente distâncias exatas, mas também intervalos, além do fato de não haver a dependência em relação a informação obtida via RMN (veja o capítulo 6 para maiores detalhes). No algoritmo BP, tanto na sua versão clássica quanto na intervalar, é de grande importância a construção de uma ordem entre os átomos a serem determinados que satisfaça às condições do DMDGP, possibilitando, assim, a aplicação do BP. Primeiramente, em [1, 2], foi proposta uma ordem atômica somente para átomos das cadeias principais de proteínas. Dentro deste contexto, este trabalho expande o conceito de ordens para os átomos das cadeias laterais de proteínas, além de apresentar a implementação de um algoritmo capaz de gerar sequências de aminoácidos independentemente de informações obtida por RMN, utilizando dados conhecidos sobre as ligações covalentes e os ângulos formados por estas ligações.

Para tal, serão utilizados, nos próximos capítulos, dados oriundos de um repositório de estruturas tridimensionais de proteínas e aminoácidos, o *Protein Data Bank* (*PDB*) [24]. Este banco de dados armazena estruturas provenientes de experimentos de RMN, Raio-X ou desenvolvimento teórico realizados por pesquisadores no mundo todo. Estas informações estão disponíveis *on-line* podendo ser acessadas de forma gratuita por qualquer pessoa.

Sendo assim, esta tese pode ser organizada da seguinte forma:

O capítulo 2 aborda alguns conceitos básicos sobre aspectos químicos e estru-

turais das proteínas, fornecendo um conjunto de nomenclaturas e propriedades que serão utilizadas ao longo deste trabalho. O capítulo 3 apresenta o problema geométrico da distância molecular e um método de comparação de estruturas tridimensionais de proteínas. No capítulo 4 serão vistos o algoritmo *geometric build-up* e suas variações. O capítulo 5 mostra o DMDGP e oferece uma visão detalhada do algoritmo BP, em sua versão clássica. Também faz uma breve comparação entre o BP e o RUGB. O capítulo 6 apresenta o DMDGP intervalar e a versão intervalar do BP (iBP). Mostra, também, a descrição de uma ordem atômica que viabiliza a aplicação do iBP, além de descrever a implementação desta ordem.

Finalmente, no capítulo 7, tem-se a conclusão de todos os estudos aqui apresentados.

Capítulo 2

Visão Geral sobre Proteínas

Este capítulo apresenta os aspectos bioquímicos teóricos das proteínas, definindo-se nomenclaturas e propriedades muito citadas ao longo deste trabalho, assim como, em suas referências bibliográficas. Será feita, nas seções de 2.1 a 2.4, uma descrição das proteínas que se inicia em sua molécula formadora, o *aminoácido*, e se estende até a descrição de estruturas mais complexas.

2.1 Aminoácidos e a Estrutura Primária

As proteínas são polímeros lineares de aminoácidos e é a sequência de componentes distintos de aminoácidos que determina a estrutura tridimensional final da proteína. A esta sequência, chama-se *estrutura primária*. O conceito de proteínas como polímeros lineares de aminoácidos foi inicialmente proposto por Fischer e Hofmeister em 1902 [25]. Naquela época, a teoria prevalecente era de que as proteínas não teriam uma estrutura regular e seriam constituídas de associações livres de pequenas moléculas (coloides). Esta questão foi muito debatida por mais de 20 anos, até que a teoria do polímero linear alcançou aceitação geral no final de 1920 [25]. Em 1952, Fred Sanger fez a importante descoberta de que as proteínas podem ser distinguidas por suas sequências de aminoácidos [26]. Na verdade, ele descobriu que as proteínas do mesmo tipo têm sequências idênticas. O trabalho de Sanger ajudou a retirar as dúvidas remanescentes sobre a exatidão da teoria do polímero linear.

Os aminoácidos são pequenas moléculas que contêm um grupo amina (NH2), um grupo carboxílico (COOH) e um átomo de hidrogênio ligado a um carbono central (α) (Figura 2.1). Além disso, os aminoácidos também têm uma *cadeia lateral* (ou grupo R) ligado ao carbono α. É este grupo que distingue um aminoácido de outro e confere suas propriedades químicas específicas.

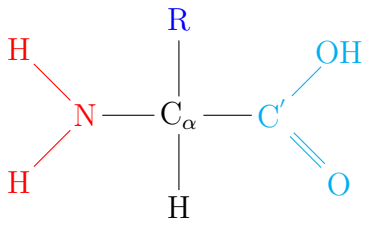


Figura 2.1: Estrutura geral de um *Aminoácido*. O grupo em vermelho é o *grupo amina*, o grupo em azul claro é o *grupo carboxílico* e o R em azul representa a *cadeia lateral*.

Um genoma pode conter instruções para produção de várias proteínas de diferentes tipos e, para tal, existem 20 aminoácidos que podem ser incorporados. A sequência resultante de uma proteína pode ser constituída por qualquer combinação desses 20 aminoácidos, em qualquer ordem. Embora essas pequenas moléculas já tivessem sido identificadas, no início do século XX, como os "*tijolos*"que constroem as proteínas, o número exato de aminoácidos só foi determinado em 1940 [25].

Os 20 aminoácidos-padrão [1] podem ser agrupados em classes com base nas propriedades químicas conferidas por suas cadeias laterais. As classes geralmente aceitas são: *apolares*, *polares*, *ácidos* e *básicos*. Dentro destas classes, subclassificações adicionais são possíveis, por exemplo, aromáticos ou alifáticos, grandes ou pequenos, e assim por diante [27]. A Figura 2.3 fornece a classificação dos 20 aminoácidos-padrão. Nessa figura, os aminoácidos estão representados como *zwitterions*. Um zwitterion é uma molécula que apresenta uma parte com uma carga positiva e outra parte com uma carga negativa. Nos aminoácidos, isto ocorre porque há a remoção (deprotonação) de um íon de hidrogênio (H^+) do grupo carboxílico (COOH) pelo grupo amina (NH_2), resultando em NH_3^+ e COO^-, como aparece na Figura 2.3. Este fato está relacionado à capacidade que os aminoácidos têm de atuar tanto como ácido quanto como base, à medida que há perda ou ganho de íons de hidrogênio. Substâncias aquosas que liberam unicamente íons de hidrogênio são consideradas ácidas e apresentam pH[2] menor que 7. Quando há perda de íons de hidrogênio, há o aumento do pH, dando ao aminoácido características de base[3]. A Figura 2.2 mostra as formas iônicas (catiônica, aniônica e zwitterion) de um aminoácido, desconsiderando qualquer ionização na cadeia lateral (grupo R).

[1] Aqui, *padrão* significa que estão presentes em todos os organismos já observados.

[2] O pH (potencial hidrogeniônico) é uma medida que indica se uma solução líquida é ácida (pH < 7, a 25 °C), neutra (pH $= 7$, a 25 °C), ou básica/alcalina (pH > 7, a 25°C).

[3] Base é qualquer substância que libera única e exclusivamente o ânion OH– (íons hidroxila)

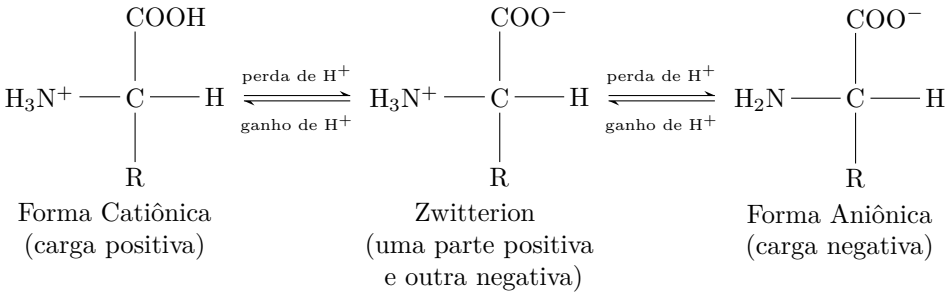


Figura 2.2: Formas iônicas de um aminoácido, mostradas sem considerar a cadeia lateral R. A forma catiônica apresenta baixo pH e, fazendo-se espécies catiônicas reagirem em um meio básico (que implica em perda de íons de hidrogênio H^+ e aumento do pH), tem-se zwitterions e, em seguida, dando-se continuidade ao processo de aplicação da base, tem-se formas aniônicas. Exemplos desse processo podem ser encontrados em [28].

2.2 Unidade Peptídica

Como já foi visto na seção 2.1, uma proteína é composta de aminoácidos, unidos linearmente, e pode ser descrita por uma sequência do tipo $(NH_2 - CHR_1 - CO) - (NH - CHR_2 - CO) - \cdots - (NH - CHR_n - COOH)$. Isto significa que os aminoácidos podem formar ligações uns com os outros através de uma reação entre seus grupos carboxílicos e amina. A ligação resultante é chamada *ligação peptídica*, e a dois ou mais aminoácidos ligados, dá-se o nome de *peptídeo* (Figura 2.4). Uma proteína é sintetizada pela formação de uma sucessão linear de ligações peptídicas entre muitos aminoácidos (obedecendo aos comandos do código genético) e pode assim ser tratada como um *polipeptídeo*. Uma vez que um aminoácido é incorporado a um peptídeo, ele passa a ter o nome de *resíduo* de aminoácido (ou somente *resíduo*, para simplificar).

O resíduo de aminoácido, no entanto, não é o mais recomendado para se entender a estereoquímica[4] e a conformação da cadeia de um polipeptídeo. Isso ocorre porque, segundo [29, 30], os grupos NH e CO interagem, uns com os outros, de tal forma que se produz uma ligação dupla de caráter parcial em relação à ligação $C - N$. As duas estruturas de ressonância que levam a esse caráter parcial de dupla ligação são mostradas na Figura 2.5 (retirada de [29, p. 281] por [30]). Como consequência, é formado um grupo planar, com uma rigidez considerável, que se inicia em um carbono α e se estende até o próximo carbono α, desta forma: $C_\alpha - C'O - NH - C_\alpha$. Este grupo é denominado *unidade peptídica*.

[4]Estereoquímica é um ramo da química que estuda as disposições espaciais de moléculas.

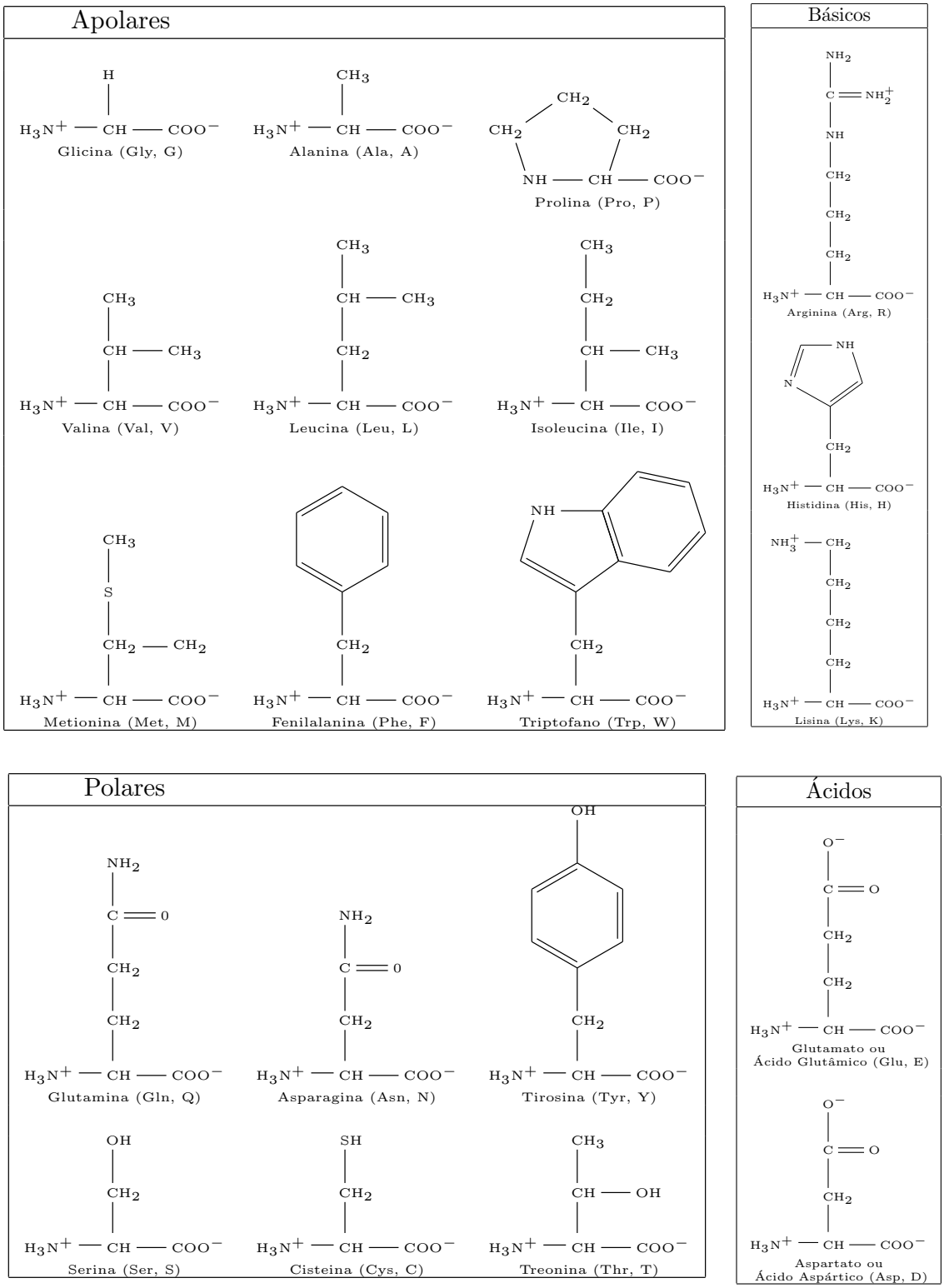


Figura 2.3: Classificação dos 20 aminoácidos-padrão. Os aminoácidos são apresentados na forma de zwitterions (ver seção 2.1 para maiores detalhes). Cada aminoácido é etiquetado com seu nome, seguido pelas abreviaturas de três letras e de uma letra.

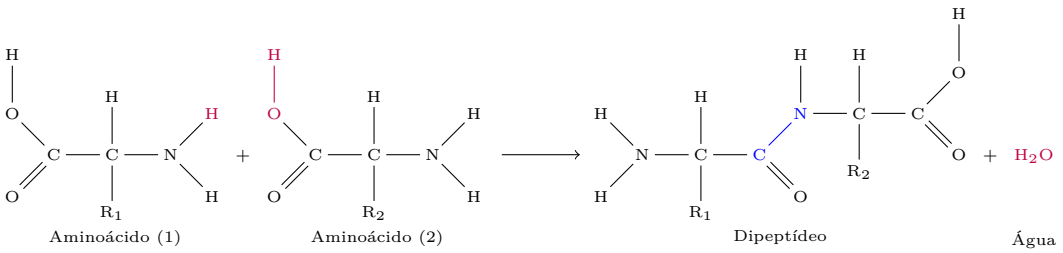


Figura 2.4: Formação da ligação peptídica por condensação (desidratação). No lado direito da equação, a ligação peptídica está representada em azul.

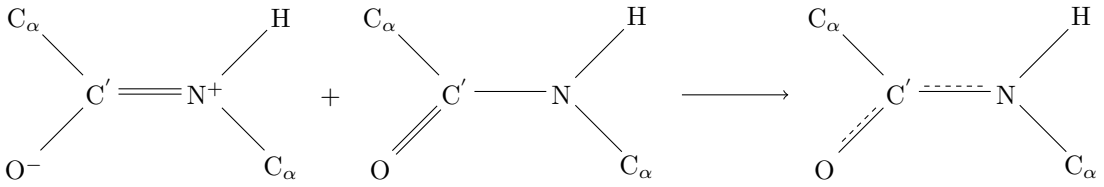
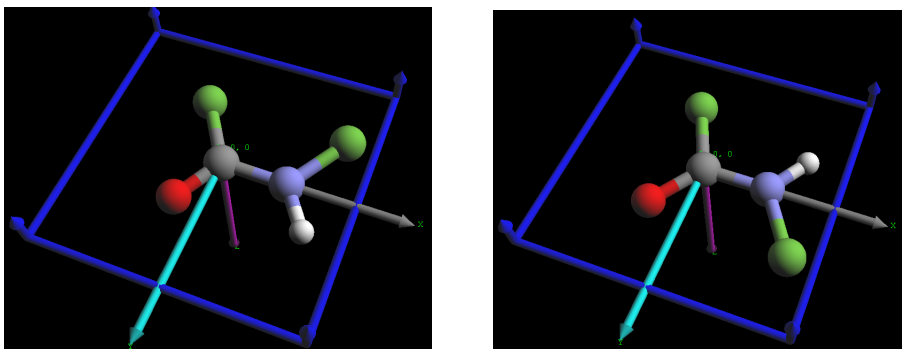


Figura 2.5: As duas estruturas de ressonância da unidade peptídica que produzem uma ligação dupla de caráter parcial na ligação peptídica $C' - N$.

A configuração planar da unidade peptídica é possível de duas formas, *cis* e *trans*. A forma trans é a que ocorre com mais frequência em polipeptídeos de cadeia aberta. Isso se deve, na forma cis, à ocorrência de uma grande repulsão entre os dois C_α, o que requer mais energia para sustentar a conformação. Na Figura 2.6a, tem-se uma unidade peptídica na forma cis e, na Figura 2.6b, uma unidade peptídica na forma trans.



(a) Cis (b) Trans

Figura 2.6: Configurações planares *cis* e *trans* de uma unidade peptídica. Nos desenhos, a bola verde representa o C_α, a cinza o C', a branca o H, a azul o N e a vermelha O.

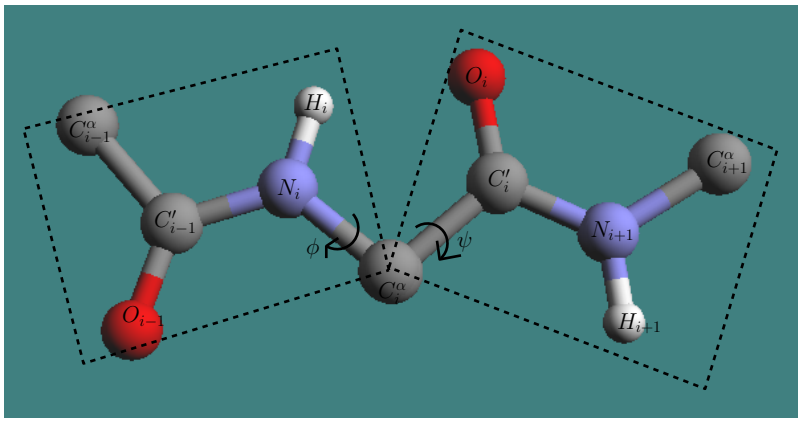


Figura 2.7: Ângulos Diedrais ϕ e ψ com valores iguais a 180°. Todos os grupos de peptídeos estão no mesmo plano.

2.3 Ângulos Diedrais e a Estrutura Secundária

De acordo com a seção 2.2, a cadeia principal de um polipeptídeo pode ser pensada, a priori, como uma série de planos consecutivos definidos pelos átomos da unidade peptídica, unidos por um ponto (C_α), como é visto nas Figuras 2.7 e 2.8. Este ponto pertence a dois eixos de rotação, $C_\alpha - C'$ e $C_\alpha - N$, e, assim sendo, seria possível, então, se obter um grande número de conformações de um polipeptídeo apenas fazendo-se rotações em torno desses eixos. Porém, para tal, algumas regras devem ser respeitadas.

Por convenção, os ângulos de rotação em torno dos eixos $C_\alpha - N$ e $C_\alpha - C'$ são rotulados, respectivamente, como ϕ e ψ. Também convencionou-se que tanto ϕ quanto ψ são definidos como 180° quando o polipeptídeo apresentar uma conformação completamente estendida e todos os grupos de peptídeos estiverem no mesmo plano, como mostra a Figura 2.7.

A princípio, ϕ e ψ podem assumir qualquer valor entre $-180°$ e $+180°$, mas muitos valores não podem ser utilizados devido ao efeito estérico[5] entre os átomos da cadeia principal e os polipeptídeos das cadeias laterais de aminoácidos. A conformação onde ϕ e ψ assumem o valor de 0° também não é permitida e é utilizada apenas como ponto de referência para a descrição dos ângulos de rotação (veja Figura 2.8).

Os valores permitidos para ϕ e ψ são expostos no *Gráfico de Ramachandran*, introduzido por G.N. Ramachandran [31] em 1963. Este gráfico apresenta os ângulos ϕ e ψ frequentemente observados em resíduos de aminoácidos pertencentes às

[5]Os efeitos estéricos advêm do fato de cada átomo dentro de uma molécula ocupar uma determinada quantidade de espaço. Se os átomos forem colocados muito próximos, há um custo em energia associado, devido à sobreposição das nuvens eletrônicas.

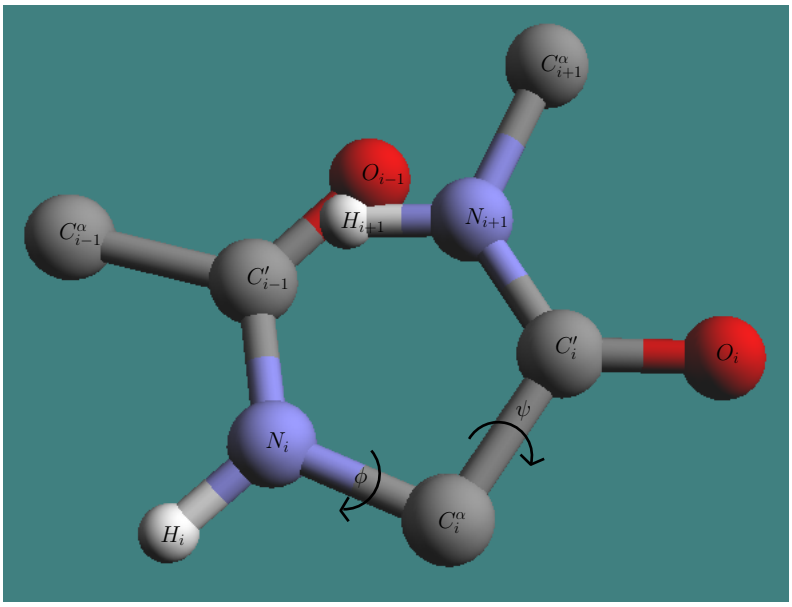


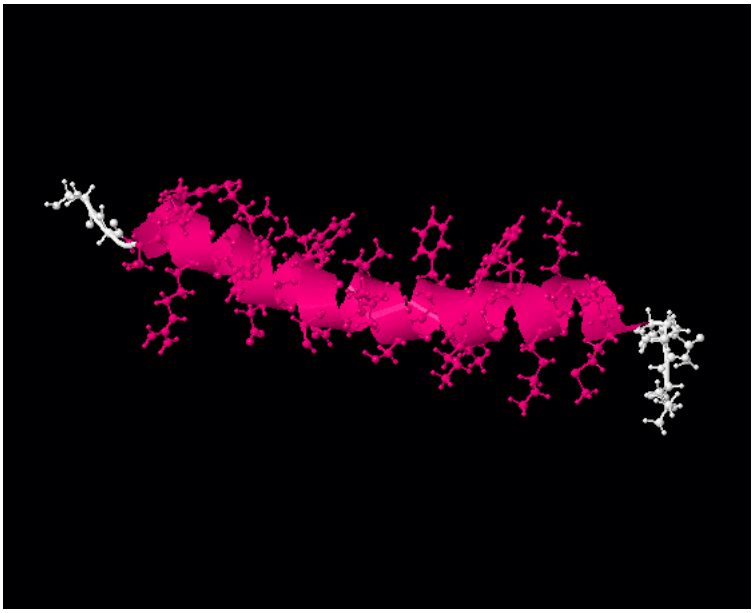
Figura 2.8: Ângulos Diedrais ϕ e ψ com valores iguais a 0°. Observe que o oxigênio do C' e o hidrogênio ligado ao nitrogênio da outra unidade peptídica se encontram.

estruturas α-hélice (esquerda e direita) e β-folha. A Figura 2.9b mostra um exemplo do gráfico de Ramachandran para a proteína 1R7C, juntamente com as regiões possíveis para ϕ e ψ. As regiões limitadas por linhas vermelhas são onde os pares (ϕ, ψ) são totalmente permitidos, pois não há influência do efeito estérico. As áreas limitadas por linhas amarelas são onde os pares (ϕ, ψ) são permitidos no limite do contato atômico (ou seja, chegando a menor distância em que um átomo pode estar do outro) ou quando há felxibilidade em relação ao ângulo de ligação. Os pontos pretos desta figura representam os pares de ângulos de cada resíduo de aminoácido da proteína 1R7C. Note que, como a proteína apresenta a maior parte de seus resíduos dentro de uma estrutura α-hélice (estrutura em vermelho na Figura 2.9a), a maioria dos pares de ângulos se concentra na região referente à esta estrutura (denominada na figura como Core R-Alpha), como era esperado de acordo com os estudos de G.N. Ramachandran.

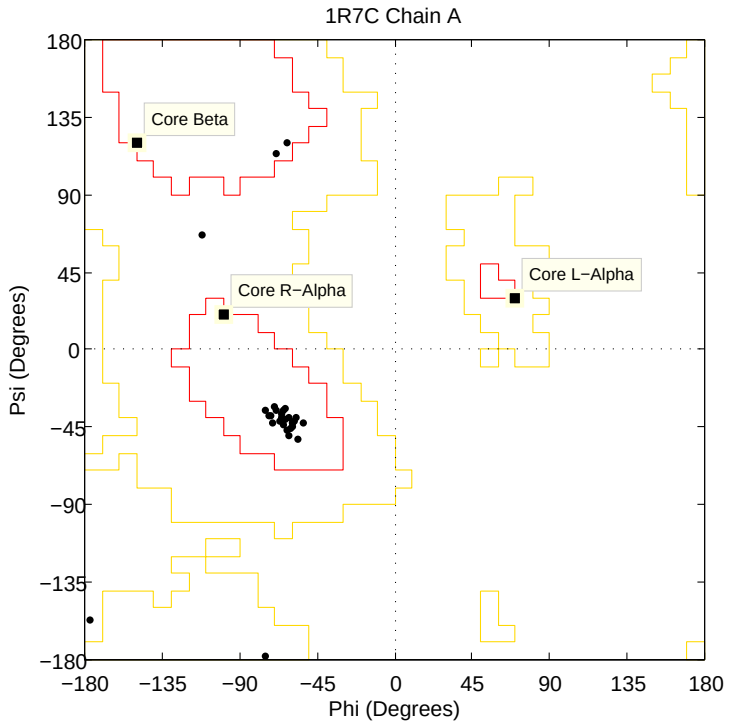
O conjunto de pares (ϕ, ψ) contém toda a informação necessária para se descrever a estrutura da cadeia principal do polipeptídeo [30]. Dentro deste contexto, tomando-se como exemplo três unidades peptídicas ligadas, desta forma

$$-\mathrm{C}^{\alpha}_1 - \mathrm{C}'_1\mathrm{O}_1 - \mathrm{N}_2\mathrm{H}_2 - \mathrm{C}^{\alpha}_2 - \mathrm{C}'_2\mathrm{O}_2 - \mathrm{N}_3\mathrm{H}_3 - \mathrm{C}^{\alpha}_3 - \mathrm{C}'_3\mathrm{O}_3 - \mathrm{N}_4\mathrm{H}_4 - \mathrm{C}^{\alpha}_4 -,$$

onde foi dado o mesmo número para todos os átomos do mesmo resíduo. Esta conformação poderá, então, ser especificada pelos pares de ângulos diedrais (ϕ_2, ψ_2) e (ϕ_3, ψ_3). Este procedimento mostra um método para se descrever a estrutura de uma



(a) Proteína 1R7C. Note que a maioria de seus resíduos de aminoácidos estão dispostos em uma estrutura α-hélice (em vermelho).



(b) Gráfico de Ramachandran da proteína 1R7C. Observe que a maioria dos resíduos de aminoácidos (representados por pontos pretos) apresentam ângulos ϕ e ψ pertencentes à região referente à estrutura α-hélice direita (denominada Core R-Alpha na figura).

Figura 2.9: Gráfico de Ramachandran exibido para a proteína 1R7C, utilizando a biblioteca de bioinformática do MATLAB.

cadeia polipeptídica que contém N resíduos. Neste caso, tem-se os pares $(\phi_i, \psi_i),\ i \in \{2 \dots N-1\}$ [6] descrevendo a estrutura da cadeia principal do polipeptídeo.

É importante ressaltar que toda a teoria vista até aqui sobre os ângulos diedrais leva em consideração a estrutura planar da unidade peptídica proposta por Linus Pauling e Robert Corey. Porém, segundo [30, p.294], na prática, isso não ocorre e se faz necessário apresentar mais parâmetros, além dos pares (ϕ, ψ), para se especificar uma conformação. Sendo assim, tem-se um ângulo diedral extra chamado ω, que representa uma rotação em torno da ligação peptídica $\mathrm{C}' - \mathrm{N}$ e pode ser definido como o ângulo entre os planos $\mathrm{C}_\alpha - \mathrm{C}' - \mathrm{N}$ e $\mathrm{C}' - \mathrm{N} - \mathrm{C}_\alpha$. Mais especificamente, ω_i é a rotação em torno da ligação $\mathrm{C}'_{\mathrm{i}} - \mathrm{N}_{\mathrm{i+1}}$, que liga o i-ésimo e o $(i+1)$-ésimo resíduo. O ângulo $\omega \in [0°, 360°]$ é medido no sentido horário, sendo que para $\omega = 0°$, tem-se uma unidade peptídica trans e, para $\omega = 180°$, tem-se uma unidade peptídica cis.

Agora, é possível se determinar a estrutura da cadeia principal de um peptídeo através dos parâmetros $(\phi_i, \psi_i, \omega_i)$. Porém, a atribuição de valores nulos para estes parâmetros deve respeitar a três condições listadas abaixo:

1. $\phi_i = 0°$ quando os átomos $\mathrm{C}'_{\mathrm{i-1}}$ e C'_{i} forem trans em relação a ligação $\mathrm{N_i} - \mathrm{C}^{\alpha}_{\mathrm{i}}$ (ver Figura 2.10a),

2. $\psi_i = 0°$ quando os átomos $\mathrm{N_i}$ e $\mathrm{N_{i+1}}$ forem trans em relação a ligação $\mathrm{C}^{\alpha}_{\mathrm{i}} - \mathrm{C}'_{\mathrm{i}}$ (ver Figura 2.10b),

3. $\omega_i = 0°$ quando os átomos $\mathrm{C}^{\alpha}_{\mathrm{i}}$ e $\mathrm{C}^{\alpha}_{\mathrm{i+1}}$ forem trans em relação a ligação $\mathrm{C}'_{\mathrm{i}} - \mathrm{N_{i+1}}$ (ver Figura 2.10c).

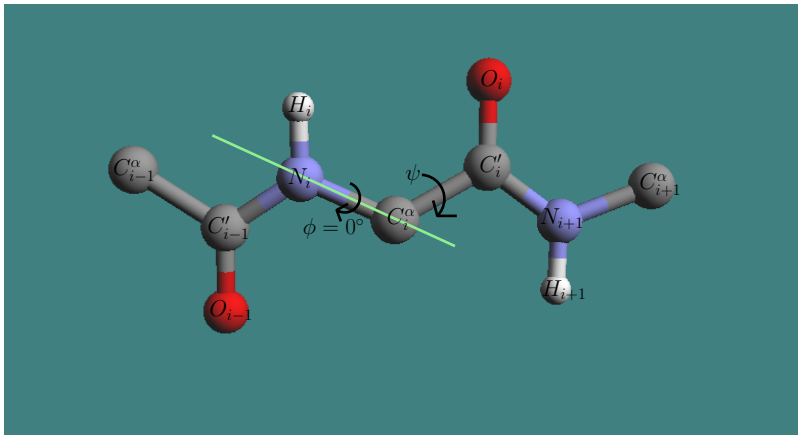
É importante lembrar que todos os ângulos são medidos no sentido horário em relação à direção de progressão da cadeia.

Os ângulos diedrais são responsáveis pela construção da *estrutura secundária* de uma proteína. A estrutura secundária é o formato assumido pelas cadeias peptídicas que constituem essa proteína. À medida que se atribuem valores a estes ângulos, dentro das condições atômicas já vistas, tem-se vários formatos de cadeias, sendo que os principais conhecidos são o *α-hélice* e o *β-folha*.

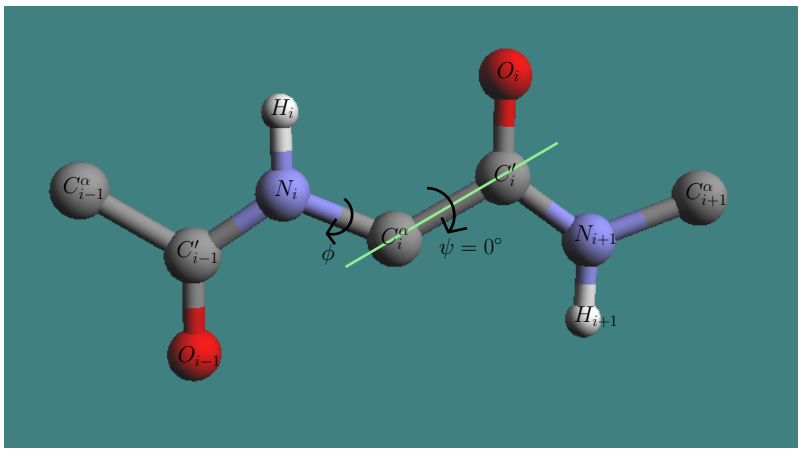
Segundo [32], o arranjo mais simples que uma cadeia peptídica pode assumir com suas ligações peptídicas rígidas[7] (e outras ligações não rígidas) é uma estrutura helicoidal que Linus Pauling e Robert Corey denominaram α-hélice. Neste arranjo,

[6] Assumindo-se que o primeiro e o último resíduos não apresentam, respectivamente, os ângulos ϕ_1 e ψ_N.

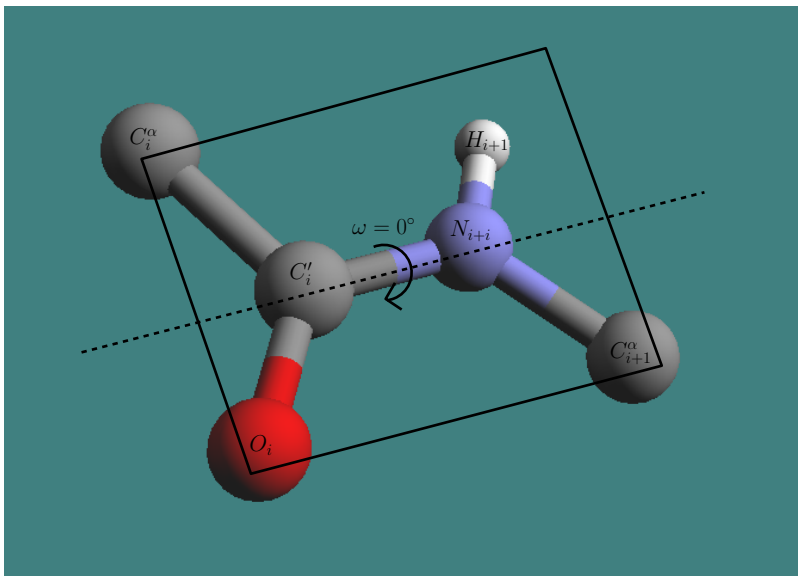
[7] O termo ligação peptídica rígida se refere à ligação peptídica presente no grupo planar denominado unidade peptídica e descrito na seção 2.2.



(a) $\phi_i = 0°$ quando os átomos C'_{i-1} e C'_i forem trans em relação a ligação $N_i - C^{\alpha}_i$.



(b) $\psi_i = 0°$ quando os átomos N_i e N_{i+1} forem trans em relação a ligação $C^{\alpha}_i - C'_i$.



(c) $\omega_i = 0°$ quando os átomos C^{α}_i e C^{α}_{i+1} forem trans em relação a ligação $C'_i - N_{i+1}$.

Figura 2.10: Condições para que os ângulos ϕ e ψ sejam nulos.

a cadeia polipeptídica é "enrolada"em torno de um eixo imaginário longitudinal que passa pelo meio da hélice e os grupos R dos resíduos são projetados para fora dessa estrutura helicoidal. A estrutura é estabilizada por *ligações de hidrogênio*[8] entre os átomos de hidrogênio, ligados aos átomos de nitrogênios eletronegativos da ligação peptídica, e os átomos de oxigênio eletronegativos dos grupos carboxílicos. Um exemplo deste arranjo pode ser visualizado nas Figuras 2.11 e 2.12.

Pauling e Corey previram um segundo tipo de arranjo repetitivo, mais extenso, de cadeias polipeptídicas, a conformação β. Nesta, as cadeias principais dos polipeptídeo apresentam forma de "ziguezague", diferente da α-hélice que é helicoidal. As cadeias polipeptídicas em ziguezague podem ser dispostas lado a lado para formar uma estrutura pregueada. Nesse arranjo, chamado de β-folha, ligações de hidrogênio são formadas entre os segmentos adjacentes da cadeia polipeptídica. Estes segmentos podem pertencer a mesma cadeia ou a cadeias diferentes. A Figura 2.13 mostra dois tipos de arranjo entre os segmentos adjacentes da estrutura β-folha: o *paralelo* e o *anti-paralelo*.

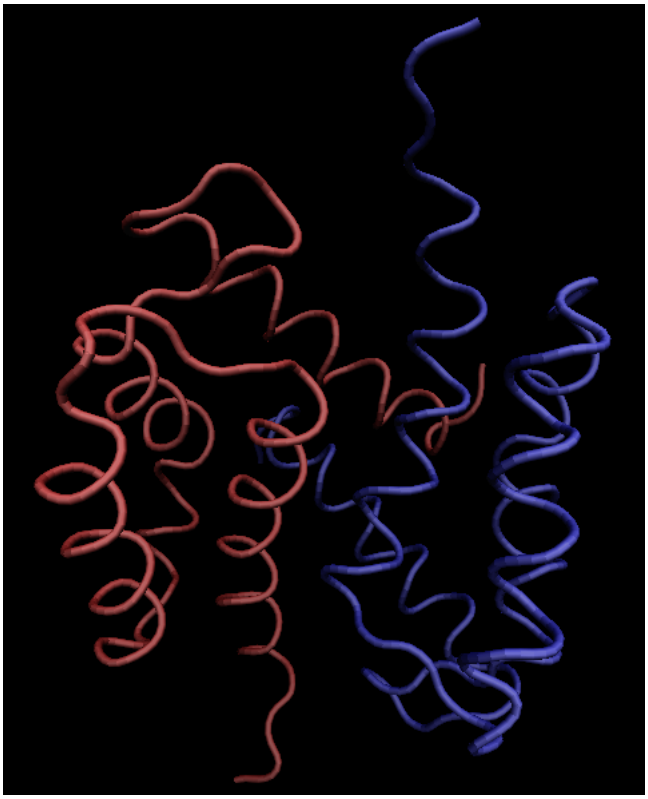
2.4 Estruturas Terciária e Quaternária

Além das estruturas primária e secundária, vistas nas seções 2.1 e 2.3, há mais dois níveis estruturais que uma proteína pode ter: as estruturas *terciária* e *quaternária*.

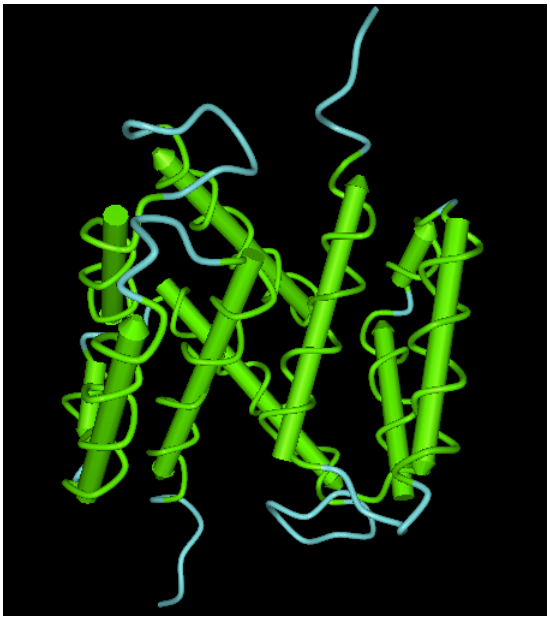
A estrutura terciária de uma proteína é definida como a sua estrutura tridimensional. Há muitas maneiras de se combinar as estruturas em hélice, folha, *loop* ou *coil*[9] para se produzir uma conformação completa. Estas combinações se realizam através de interações entre as cadeias laterais dos resíduos de aminoácido. Assim, na estrutura terciária, o grupo R tem um papel mais ativo na constituição final da proteína, diferente da estrutura secundária onde era a cadeia principal que dava a forma do arranjo. A maioria das proteínas tem, cada uma, uma estrutura terciária peculiar. Os elementos da estrutura secundária sempre se dobrarão (apresentarão formas contorcidas) do mesmo modo para produzir a mesma estrutura terciária. Essa consistência é vital para a função desempenhada pela proteína.

[8] Alguns autores denominam estas ligações como *pontes de hidrogênio*.

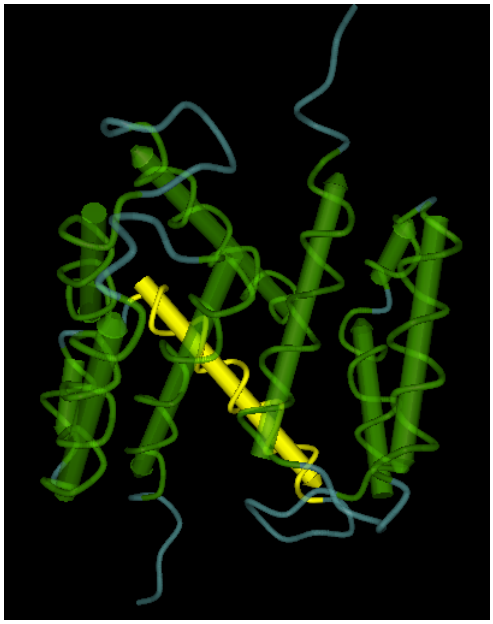
[9] α-hélice e β-folha representam a maioria das estruturas secundárias observadas em proteínas. No entanto, estas estruturas regulares são intercaladas com regiões de estruturas irregulares conhecidas como *loop* e *coil*. Neste trabalho, estas estruturas não serão abordadas.



(a) Representação da Proteína PDB ID 2LOP através de suas cadeias de aminoácidos. Esta proteína é constituída por duas cadeias que são formadas por estruturas α-hélices.



(b) Proteína PDB ID 2LOP representada através de suas estruturas secundárias (em verde, em torno de eixos cilíndricos), todas α-hélice.



(c) Seleção da sequência de aminoácidos de 1 a 20 (em amarelo), dada pelos aminoácidos SELETAMETLINVFHAHSG (ver abreviaturas na Figura 2.3, cada letra é um aminoácido). Esta sequência forma uma estrutura α-hélice.

Figura 2.11: Exemplo de estruturas α-hélices

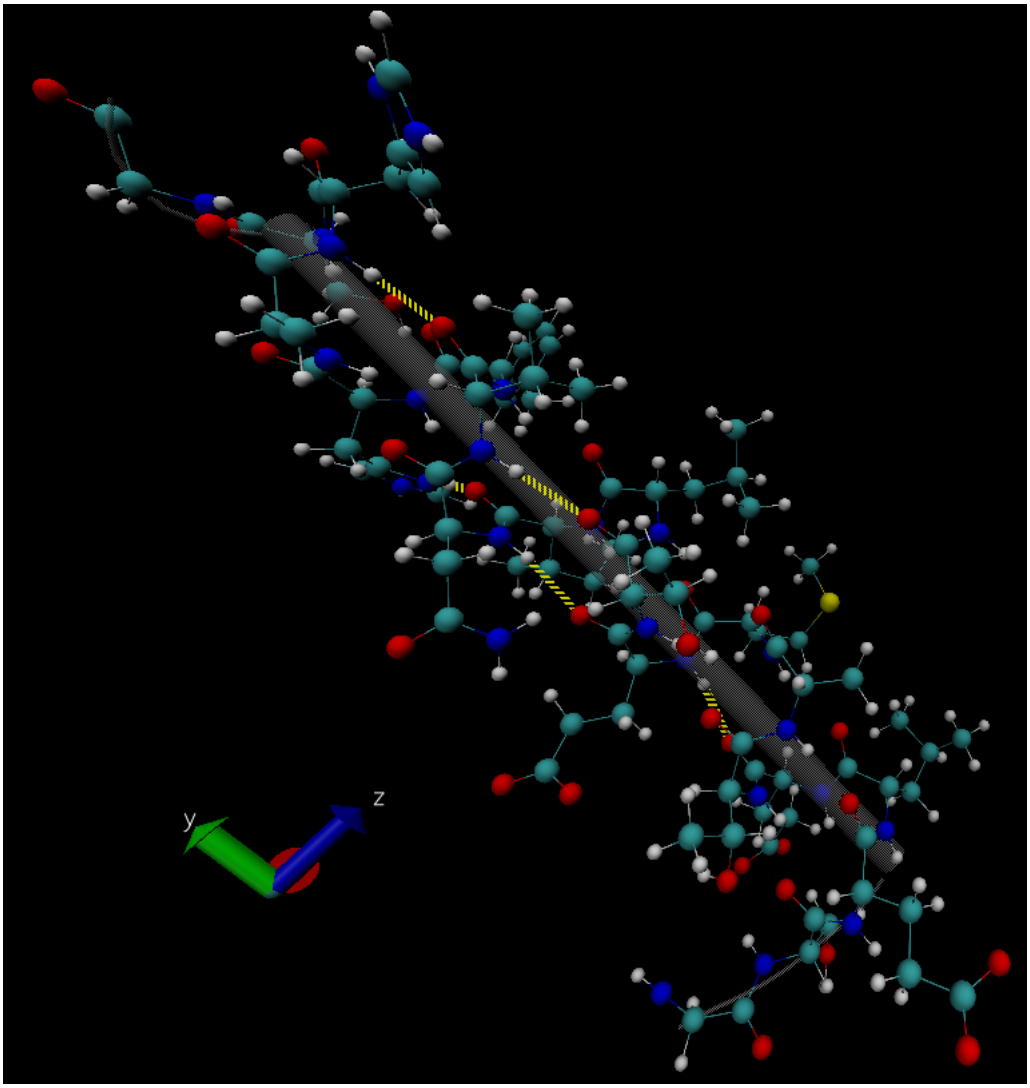
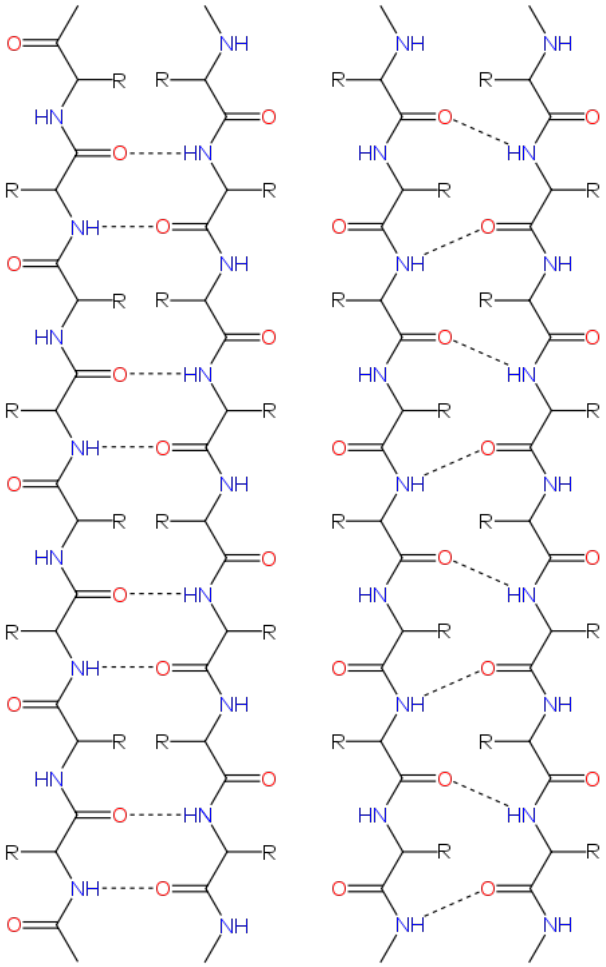
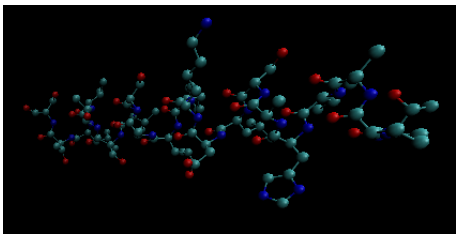


Figura 2.12: Estrutura α-hélice selecionada na Figura 2.11c. Nesta figura, observa-se as ligações de hidrogênio (linhas pontilhadas em amarelo). Os átomos verdes são carbonos, os vermelhos são oxigênios, os brancos são hidrogênios, os azuis são nitrogênios e os amarelos são átomos de enxofre.

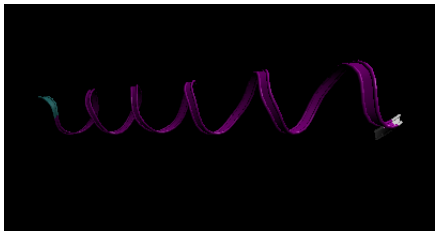


(a) β-folha anti-paralela. (b) β-folha paralela.

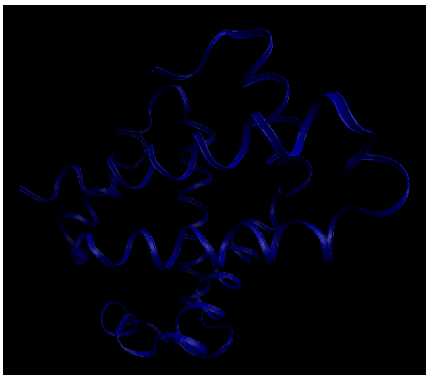
Figura 2.13: Estrutura β-folha. As pontes de hidrogênio estão representadas por linhas pontilhadas.



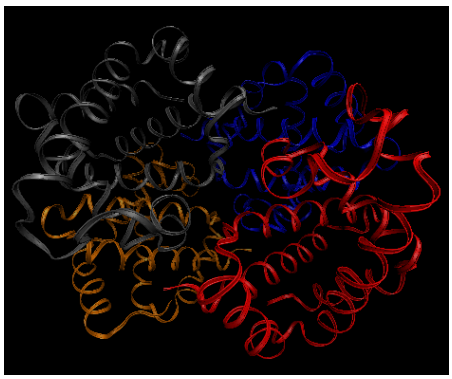
(a) Sequência de aminoácidos de 118 a 138, isto é, TPAVHASLDKFLASVSTVLTS: estrutura primária.



(b) A sequência da Figura 2.14a forma uma estrutura α-hélice: estrutura secundária.



(c) Várias estruturas α-hélice formam uma das cadeias da proteína: estrutura terciária.

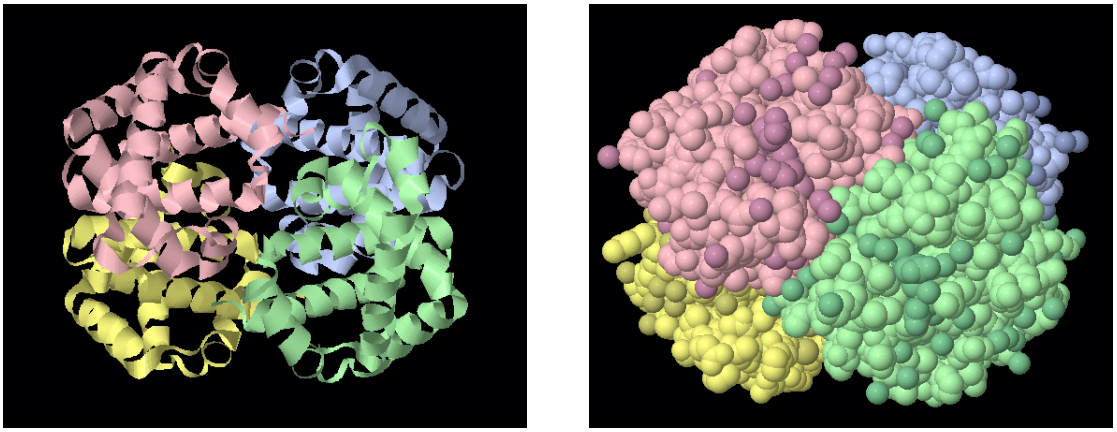


(d) A deoxihemoglobina é formada por quatro cadeias polipeptídicas: estrutura quaternária.

Figura 2.14: Os quatro níveis hierárquicos da estrutura molecular da deoxihemoglobina.

A estrutura terciária descreve a organização de uma única cadeia polipeptídica. Porém, muitas proteínas não apresentam uma única cadeia. Pelo contrário, elas existem como uma associação não covalente de duas ou mais cadeias polipeptídicas independentes. Essas proteínas são denominadas *multisubunidade* ou *multiméricas* (do inglês, *multisubunit* ou *multimeric*) e pode-se dizer que têm uma estrutura quaternária. As subunidades podem ser idênticas, resultando em uma proteína *homomérica* (do inglês, *homomeric*), ou diferentes, resultando em uma proteína heteromérica (do inglês, *heteromeric*). A Figura 2.14 mostra os quatro níveis hierárquicos da estrutura molecular da deoxihemoglobina[10] (do inglês, *deoxyhemoglobin*). A deoxihemoglobina é uma proteína multimérica e heteromérica composta por quatro subunidades, todas helicoidais. A figura começa com um exemplo de estrutura primária no canto superior esquerdo e prossegue até a estrutura quaternária. Tem-se ainda, na Figura 2.15, imagens da estrutura quaternária da deoxihemoglobina produzidas em MATLAB a partir do arquivo PDB ID 2HHB, onde foi utilizada a biblioteca de bioinformática.

[10]Hemoglobina que não está transportando oxigênio.



(a) Representação em fita (ribbon). (b) Modelo *space-filling*.

Figura 2.15: Estrutura quaternária da deoxihemoglobina (PDB ID 2HHB). Análise por raios-X de difração da desoxihemoglobina (hemoglobina sem moléculas de oxigênio ligadas), mostrando como as quatro subunidades polipeptídicas aparecem juntas. As imagens foram produzidas por MATLAB.

Do próximo capítulo em diante, serão tratados os aspectos matemáticos e computacionais presentes na determinação das estruturas moleculares das proteínas, utilizando-se todos os conceitos discutidos neste capítulo.

Capítulo 3

Problema Geométrico da Distância Molecular

3.1 Formulação

Iniciando-se os aspectos matemáticos presentes na determinação da estrutura tridimensional de uma proteína, apresenta-se, neste capítulo, o *problema geométrico da distância molecular* (que para simplificar, será chamado de MDGP, do inglês *molecular distance geometry problem*).

Sabe-se que, como foi mencionado no capítulo 1, com experimentos feitos via RMN são obtidas informações acerca da molécula estudada. Dentro destas informações, encontra-se um conjunto de distâncias entre átomos, em sua maioria hidrogênio, espacialmente próximos (distâncias, geralmente, em torno de 5 a 6Å). Com essas distâncias é possível representar a molécula como um grafo não direcionado $G = (V, E, d)$, onde V, E e d são definidos abaixo:

1. Sejam A o conjunto de todos os n átomos da molécula a ser determinada e a função bijetora $v : A \to \{1, 2, \ldots, n\}$, define-se V como o conjunto dos pares (a, i), onde $a \in A$ e $i \in \{1, 2, \ldots, n\}$. Para tornar a notação mais simples, i será utilizado no lugar de (a, i). Assim, em outras palavras, V é o conjunto dos vértices i do grafo G onde cada i está associado a um átomo.

2. E é definido como o conjunto das arestas $\{i, j\}$ de G $(i, j \in V)$ e $|E|$ é o número de elementos de E. Uma aresta $\{i, j\}$ existe em E se e somente se existe uma distância entre os átomos associados a i e j. É importante notar que o fato de se ter uma distância entre dois átomos não implica em uma ligação química.

3. d é uma função $E \to \mathbb{R}_+$, onde $d(\{i, j\}) = d_{ij} = d_{ji}$ são distâncias euclidianas entre os átomos associados aos vértices i e j $(\{i, j\} \in E)$.

Para se ter uma escrita mais simples, a partir daqui, a expressão *átomo i* deve ser entendida como um vértice i associado a um átomo da molécula.

Considerando-se o grafo G, deseja-se determinar a função $x : V \to \mathbb{R}^3$, que atribui coordenadas cartesianas aos átomos, tal que:

$$||x_i - x_j|| = d_{ij}, \ \forall \{i, j\} \in E, \tag{3.1}$$

onde $|| \cdot ||$ é a norma euclidiana e $x_i = x(i), \ i \in V$. Em outras palavras, deseja-se encontrar a função x responsável por retornar as coordenadas dos átomos com distâncias definidas. Note que a equação (3.1) representa um sistema de equações não-lineares, composto por $|E|$ (quantidade de arestas de G) equações.

Neste contexto, algumas observações podem ser feitas sobre a construção do conjunto X das posições x_i ($i \in V$) dos átomos de uma proteína. Desta forma, seja o conjunto $\bar{X} = \{\bar{x} \mid \bar{x} \text{ satisfaz (3.1)}\}$. Este conjunto pode ser não enumerável, pois para uma solução qualquer $\bar{x} \in \bar{X}$ e uma transformação ortogonal qualquer T em $\mathbb{R}^3$, tem-se que $T\bar{x} \in \bar{X}$, pois as transformações ortogonais preservam a norma – lembrando-se que existe uma quantidade não enumerável de transformações em $\mathbb{R}^3$. É preciso, então, construir um conjunto X enumerável. Assim, seja $\sim$ a relação de equivalência em $\bar{X}$, onde $\bar{x} \sim \bar{y}$ se e somente se existe uma transformação ortogonal T tal que $\bar{y} = T\bar{x}$ ($\bar{x}, \bar{y} \in \bar{X}$). Seja $[\bar{y}] = \{\bar{x} \in \bar{X} \mid \bar{x} \sim \bar{y}\}$ a classe de equivalência de um elemento $\bar{y} \in \bar{X}$, é possível reunir todas as classes de equivalência de $\bar{X}$ em um conjunto quociente $\bar{\bar{X}} = \bar{X}/\sim \ = \ \{[\bar{x}] \mid \bar{x} \in \bar{X}\}$. Agora, a partir de $\bar{\bar{X}}$, é necessário um critério de escolha para se determinar um único representante de cada classe de equivalência $[\bar{x}]$ ($\bar{x} \in \bar{X}$). X então será constituído por estes representantes. Este critério pode ser obtido fixando-se a posição de 4 vértices em V de forma a satisfazer a equação (3.1) e obtendo-se as demais posições a partir dos vértices já determinados [34]. Essa prática poderá ser percebida nos algoritmos discutidos a seguir.

Numericamente, resolver a equação não-linear (3.1) diretamente é uma tarefa difícil. Alguns subsistemas de (3.1), porém, são, às vezes, considerados e resolvidos como parte de um método de solução mais complexo [16, 33]. Não são raras as vezes que o sistema (3.1) é reescrito como uma função de penalidade a ser minimizada, assim:

$$\min_{x} \sum_{\{i,j\} \in E} (||x_i - x_j||^2 - d_{ij}^2)^2. \tag{3.2}$$

Na equação acima, obtêm-se os quadrados dos lados esquerdo e direito de (3.1) e depois o quadrado da diferença, de modo que a função objetivo resultante não tenha a operação raiz quadrada (onde se pode ter problemas numéricos para argumentos próximos de zero, caso o método de resolução utilize a derivada). É importante destacar que, segundo [34], (3.2) é um problema de otimização não convexo em x, e

se enquadra na categoria de otimização global (OG).

Formalizando, então, o problema associado a equação (3.1) é o seguinte:

Definição 3.1 (Problema Geométrico da Distância Molecular (MDGP))**.** *Dado o grafo, não direcionado, $G = (V, E, d)$ (descrito acima) com $d : E \to \mathbb{R}_+$, deve-se encontrar $x : V \to \mathbb{R}^3$ tal que a equação*

$$||x_i - x_j|| = d_{ij}, \ \forall \{i, j\} \in E,$$

seja satisfeita.

A modelagem da estrutura de proteínas através de problema geométrico de distância foi introduzida de forma pioneira por Crippen e Havel [7], que desenvolveram o algoritmo EMBED. Neste algoritmo uma classe particular de MDGP foi considerada: quando todas as distâncias exatas a todos os pares de átomos estão disponíveis. Assim, resolve-se o problema decompondo a matriz formada pelas distâncias, através de *decomposição em valores singulares* (*SVD* de *singular value decomposition*) vista no teorema A.1.

No caso do EMBED, será utilizada a forma reduzida do teorema A.1 vista no teorema A.2.

O teorema A.2 também pode ser aplicado quando a matriz A apresenta posto $r < n$. Neste caso, U_1 poderia ser escrita como uma matriz de dimensão $m \times r$ e Σ_1 como uma matriz diagonal de dimensão $r \times r$ e elementos estritamente positivos (para maiores detalhes, ver [36]).

Desta forma, dadas todas as distâncias d_{ij} exatas entre todos os pares de átomos, sejam a matriz simétrica $d = [d_{ij}]$ e $x_1, x_2, \ldots, x_n$ um conjunto de coordenadas consistentes, isto é, que satisfazem

$$||x_i - x_j|| = d_{ij}, \quad i, j = 1, \ldots, n, \tag{3.3}$$

posicionando-se o primeiro ponto na origem do sistema, isto é, fazendo-se

$$x_1 = (0, 0, 0)^T,$$

(3.3) pode ser reescrita de forma equivalente como

$$||x_i - x_1||^2 = ||x_i - (0, 0, 0)||^2 = ||x_i||^2 = d_{i1}^2, \tag{3.4}$$

$$||x_i - x_j||^2 = d_{ij}^2, \quad i, j = 2, \ldots, n. \tag{3.5}$$

Fazendo-se

$$x_i = \begin{bmatrix} u_i \\ v_i \\ w_i \end{bmatrix} = (u_i, v_i, w_i)^T$$

e desenvolvendo-se a equação (3.5), tem-se

$$\begin{aligned} ||x_i - x_j||^2 &= d_{ij}^2 \\ ||(u_i - u_j, v_i - v_j, w_i - w_j)^T||^2 &= d_{ij}^2 \\ (u_i - u_j)^2 + (v_i - v_j)^2 + (w_i - w_j)^2 &= d_{ij}^2 \\ (u_i^2 + v_i^2 + w_i^2) + (u_j^2 + v_j^2 + w_j^2) - 2(u_i u_j + v_i v_j + w_i w_j) &= d_{ij}^2, \end{aligned}$$

isto é,

$$||x_i||^2 + ||x_j||^2 - 2x_i^T x_j = d_{ij}^2. \tag{3.6}$$

Como x_1 está posicionado na origem, por (3.4), pode-se rescrever (3.6) da seguinte forma

$$d_{i1}^2 + d_{j1}^2 - 2x_i^T x_j = d_{ij}^2,$$

ou

$$d_{i1}^2 - d_{ij}^2 + d_{j1}^2 = 2x_i^T x_j, \quad i, j = 2, \ldots, n.$$

Seja $D = [D_{ij}]$, onde $D_{ij} = (d_{i1}^2 - d_{ij}^2 + d_{j1}^2)/2$, é possível definir

$$D = XX^T,$$

onde

$$X = \begin{bmatrix} x_2^T \\ x_3^T \\ \vdots \\ x_n^T \end{bmatrix} \text{ e } X^T = [x_2 \ x_3 \ \ldots \ x_n].$$

Note que a matriz X tem dimensão $(n-1) \times 3$ e posto $r(X) \leq \min\{n-1, 3\}$ e a matriz D é positiva semi-definida pois, seja $y \in \mathbb{R}^3$, tem-se

$$y^T D y = y^T X X^T y = (X^T y)^T (X^T y) = ||X^T y||^2 \geq 0.$$

Logo,

$$r(D) = r(X) \leq 3.$$

Neste contexto, pelo teorema A.1, pode-se decompor a matriz X como segue:

$$X = \bar{U}\bar{\Sigma}\bar{V}^T,$$

onde $\bar{U}$ é uma matriz ortogonal $(n-1)\times(n-1)$, $\bar{\Sigma}$ é uma matriz diagonal $(n-1)\times 3$ e $\bar{V}$ é uma matriz ortogonal 3×3. Sendo assim, utilizando-se o teorema A.2, tem-se

$$\begin{aligned} U &= \bar{U}(:, 1:3) = [u_1\ u_2\ \ldots\ u_3] \in \mathbb{R}^{(n-1)\times 3} \\ \Sigma_X &= \bar{\Sigma}(1:3, 1:3) = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_3) \in \mathbb{R}^{3\times 3} \\ V &= \bar{V} \in \mathbb{R}^{3\times 3}, \end{aligned}$$

e, assim,

$$\begin{aligned} X &= U\Sigma_X V^T \\ XX^T &= U\Sigma_X V^T X^T \\ XX^T &= U\Sigma_X V^T V \Sigma_X^T U^T \\ XX^T &= U\Sigma_X^2 U^T, \end{aligned}$$

isto é,

$$D = U\Sigma_D U^T,$$

onde $\Sigma_D = \Sigma_X^2$ é a matriz diagonal formada pelos valores singulares de D. Conclui-se, então, que uma solução para $D = XX^T$ é obtida fazendo-se

$$X = U\Sigma_D^{1/2}.$$

Dong e Wu, em [33], propuseram um algoritmo que segue uma linha de raciocínio parecida com o EMBED, cuja solução pode ser obtida em tempo linear (ver seção 4).

Este capítulo, desde seu início, leva à conclusão de que o MDGP envolve uma busca no espaço contínuo $\mathbb{R}^3$, o que é verdade. Porém, isto dependerá da estrutura do problema. No caso em questão, são distâncias moleculares, o que remete à procura de propriedades atômicas que facilitem essa busca, tornando possível a discretização desse espaço em um conjunto de pontos, abordagem utilizada em [23] onde é descrito o algoritmo branch-and-prune. Outra abordagem é se considerar subsistemas de (3.1) e resolvê-los, como é feito em [33], quando se tem todas as distâncias entre todos os átomos, e em [16], quando se tem algumas distâncias.

Sendo assim, na seção 3.2, será visto um método usado para se fazer comparações entre estruturas tridimensionais de proteínas e utilizado, também, em [37], como será visto na seção 4.4: A *RMSD* (*root mean square deviation*).

3.2 Método para comparação de estruturas

Ao se resolver o MDGP, pode-se obter mais de uma solução. Isso ocorre, pois, o MDGP é resolvido em função das distâncias disponíveis entre os átomos de uma

molécula que podem apresentar mais de um arranjo possível. Porém, existe ainda a possibilidade de uma única solução se apresentar em diferentes posições no sistema cartesiano, o que pode levar à ilusão de haver mais de uma conformação, quando, na verdade, o que houve foram rotações e translações de uma mesma estrutura. Para evitar este equívoco, existe um método utilizado com frequência, conhecido como *RMSD* (*root mean square deviation*).

Desta forma, sejam $\bar{X}$ e $\bar{Y}$ duas matrizes $n \times 3$, formadas pelas coordenadas $x_i = (x_{i1}, x_{i2}, x_{i3})^T$ e $y_i = (y_{i1}, y_{i2}, y_{i3})^T$, com $i = 1, 2, \ldots, n$, dos n átomos que compõem as moléculas A e B respectivamente. Sejam os *centros geométricos* destas estruturas definidos por

$$\begin{aligned} x_c(j) &= \frac{1}{n}\sum_{i=1}^{n} \bar{X}(i,j),\ j = 1,2,3 \\ y_c(j) &= \frac{1}{n}\sum_{i=1}^{n} \bar{Y}(i,j),\ j = 1,2,3 \end{aligned}$$

e a Norma de Frobenius da matriz G dada por

$$||G||_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} |g_{ij}|^2},$$

onde G é uma matriz $m \times n$, e, além disso, sabendo-se que

$$||G||_F = \sqrt{\mathrm{Tr}(GG^T)} = \sqrt{\mathrm{Tr}(G^TG)}, \tag{3.7}$$

onde $\mathrm{Tr}(GG^T)$ representa o traço da matriz GG^T e G^T é a transposta de G (ver [36, p. 23]).

Fazendo-se com que as estruturas representadas por $\bar{X}$ e $\bar{Y}$ sejam posicionadas de tal forma que seus centros geométricos coincidam com a origem, isto é, fazendo-se

$$\begin{aligned} X &= \bar{X} - \zeta_n x_c^T, \\ Y &= \bar{Y} - \zeta_n y_c^T, \end{aligned}$$

onde $\zeta_n = (1, \ldots, 1)^T \in \mathbb{R}^n$. Tem-se que a RMSD é dada por

$$\mathrm{RMSD}(X,Y) = \min_{Q} ||X - YQ||_F/\sqrt{n}, \tag{3.8}$$

onde Q é uma matriz 3×3 ortogonal. É importante notar que

$$||X - YQ||_F^2 = \mathrm{Tr}\left[(X - YQ)^T(X - YQ)\right],$$

considerando-se a equação (3.7). Continuando-se o desenvolvimento, tem-se

$$\begin{aligned} ||X - YQ||_F^2 &= \text{Tr}\left[(X - YQ)^T(X - YQ)\right] \\ &= \text{Tr}\left[(X^T - Q^TY^T)(X - YQ)\right] \\ &= \text{Tr}\left[X^TX - X^TYQ - Q^TY^TX + Q^TY^TYQ\right] \\ &= \text{Tr}\left[X^TX - X^TYQ - (X^TYQ)^T + Q^TY^TYQ\right] \\ &= \text{Tr}\left[X^TX\right] - \text{Tr}\left[X^TYQ\right] - \text{Tr}\left[(X^TYQ)^T\right] + \text{Tr}\left[Q^TY^TYQ\right]. \end{aligned}$$

Das propriedades do traço de uma matriz, pode-se dizer que

$$\begin{aligned} \text{Tr}\left[X^TYQ\right] &= \text{Tr}\left[(X^TYQ)^T\right] \\ \text{Tr}\left[Q^TY^TYQ\right] &= \text{Tr}\left[Y^TYQQ^T\right] = \text{Tr}\left[Y^TY\right] \end{aligned}$$

e que

$$||X - YQ||_F^2 = \text{Tr}(X^TX) + \text{Tr}(Y^TY) - 2\text{Tr}(Q^TY^TX). \tag{3.9}$$

Assim, da equação (3.9), conclui-se que minimizar $||X - YQ||_F$ equivale a maximizar $\text{Tr}(Q^TY^TX)$. Definindo-se $C = Y^TX$ e seja $C = U\Sigma V^T$ a decomposição em valores singulares de C (ver teorema A.1), pode-se escrever que

$$\text{Tr}(Q^TY^TX) = \text{Tr}(Q^TC) = \text{Tr}(Q^TU\Sigma V^T) = \text{Tr}(V^TQ^TU\Sigma).$$

Note que V^TQ^TU é uma matriz ortogonal. Desta forma, os vetores que compõem suas linhas e colunas são ortonormais. A matriz Σ, por sua vez, é diagonal com todos os valores positivos. Sejam γ_{ii} os elementos da diagonal de V^TQ^TU e σ_{ii} os elementos da diagonal de Σ, tem-se

$$\text{Tr}(V^TQ^TU\Sigma) = \sum_{i=1}^{3} \gamma_{ii}\sigma_{ii}.$$

Por ser V^TQ^TU ortogonal, $|\gamma_{ii}| \leqslant 1$. Sendo $\sigma_{ii} \geqslant 0$ pelo teorema A.1, pode-se escrever

$$\text{Tr}(V^TQ^TU\Sigma) = \sum_{i=1}^{3} \gamma_{ii}\sigma_{ii} \leqslant \sum_{i=1}^{3} \sigma_{ii} = \text{Tr}(\Sigma),$$

que representa um limite superior para $\text{Tr}(Q^TY^TX)$. Sendo assim, fazendo-se

$$Q = UV^T,$$

maximiza-se $\text{Tr}(Q^TY^TX)$ e, consequentemente, minimiza-se $||X - YQ||_F$, como se queria de início.

Neste trabalho, a RMSD será utilizada para calcular o erro existente entre as

conformações obtidas pelos algoritmos aqui analisados e as conformações originais obtidas através dos arquivos do PDB. Também será utilizada nos algoritmos da seção 4.4.

O próximo capítulo mostrará uma abordagem para o MDGP que se baseia na resolução de subsistemas, com o objetivo de se determinar a posição de um átomo através da interseção de três esferas. O algoritmo RUGB, descrito no capítulo 4, será comparado com o BP clássico no capítulo 5, a fim de se avaliar o desempenho do BP na determinação das coordenadas de átomos das cadeias principais de proteínas. Posteriormente, o BP será também utilizado na obtenção de estruturas tridimensionais das cadeias laterais dos aminoácidos-padrão.

Capítulo 4

Geometric Build-up

Este capítulo mostra o algoritmo geometric build-up e suas variações, propostos em [16, 33, 37, 39]. Na seção 4.1, será vista uma abordagem para o caso de se ter todas as distâncias exatas disponíveis. Na seção 4.2, o algoritmo é reescrito para o caso onde se tem somente algumas distâncias exatas. A seção 4.3 expõe um método de escolha de uma base inicial para a determinação dos primeiros átomos (aqui chamada de *base métrica inicial*).

Finalmente, na seção 4.4, serão vistas as variações do algoritmo geometric build-up: o *updated geometric build-up* (UGB) e o *revised updated geometric build-up* (RUGB). Este último será utilizado mais adiante pela seção 5.4.

4.1 Todas as distâncias exatas disponíveis

Dong e Wu [33] propuseram um algoritmo capaz de resolver o MDGP exato em tempo linear, se todas as distâncias entre todos os pares de átomos estiverem disponíveis. Baseia-se na relação geométrica simples entre distâncias e coordenadas. Para se entender este algoritmo em sua essência, considere o seguinte exemplo.

Exemplo 4.1. *Sejam 3 pontos* $x_1,\ x_2,\ x_3 \in \mathbb{R}^2$ *e um conjunto de distâncias entre átomos que forma a matriz*

$$\begin{bmatrix} 0 & d_{12} & d_{13} \\ d_{12} & 0 & d_{23} \\ d_{13} & d_{23} & 0 \end{bmatrix},$$

onde $d_{ij} = \|x_i - x_j\|,\ (i,j) \in \{1,2,3\}$, *e* $d_{ij} = d_{ji}$.

Com essas informações, é possível se determinar as coordenadas dos três pontos, colocando-se o primeiro ponto, x_1, *na origem, isto é,* $x_1 = (0\ 0)^T$. *Assim, um eixo será determinado pelos pontos* x_1 *e* x_2. *O ponto* x_2 *será colocado a uma distância* d_{12} *de* x_1 *no sentido positivo, isto é, de* x_1 *para* x_2. *O terceiro ponto pode ser encontrado como mostra a figura 4.1. Neste caso, pode-se escolher a posição positiva em relação*

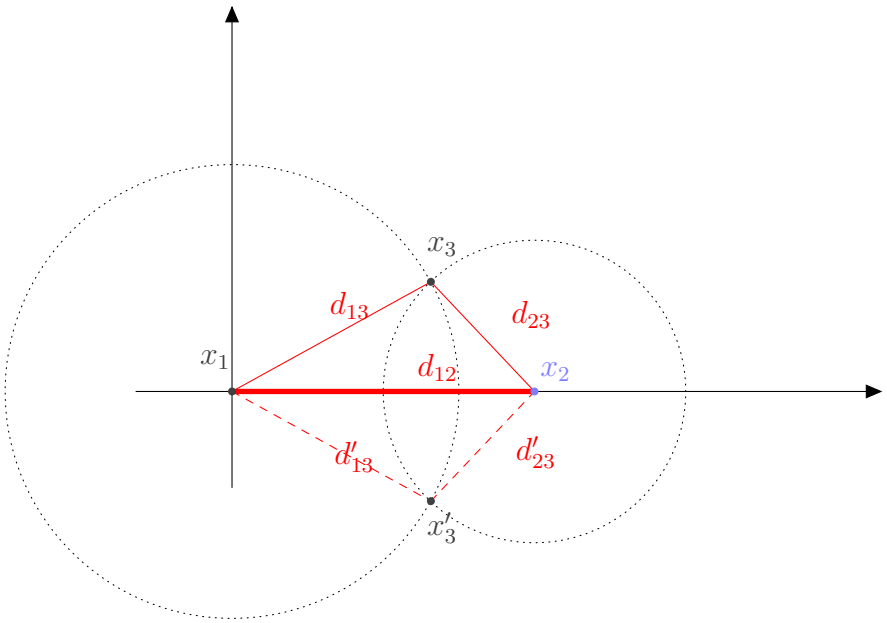


Figura 4.1: Encontrando as coordenadas de 3 pontos em 2 dimensões.

ao sistema definido por x_1 e x_2. Estes 3 pontos são únicos a menos de translações e rotações e, como eles não são colineares, estas coordenadas podem ser usadas para se encontrar um quarto ponto x_4[1], utilizando-se as distâncias em relação a este ponto.

No exemplo 4.1, (x_1, x_2, x_3) formam uma *base métrica* e são denominados *independentes*, pois não são colineares. Sendo assim, segue a definição formal de base métrica e pontos independentes. Além disso, serão definidos *hiperplano* e *pontos vizinhos*.

Definição 4.1 (Base Métrica)**.** *Um conjunto de pontos B (com coordenadas conhecidas) num espaço vetorial é uma* ***base métrica*** *de um conjunto S dado, quando as coordenadas de cada ponto de S são unicamente determinadas pelas distâncias entre este ponto e os pontos de B.*

Definição 4.2 (Hiperplano)**.** *Sejam $a_1, a_2, \ldots, a_n$ escalares com pelo menos um não-nulos e a constante real c. Então, o conjunto S dos vetores*

$$s = \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{bmatrix} \in \mathbb{R}^n,$$

[1]Neste caso, x_4 não é determinado de forma única como será visto mais adiante

tais que

$$a^T s = c$$

é um subconjunto de $\mathbb{R}^n$ *chamado* ***hiperplano****, onde* $a = (a_1, a_2, \ldots, a_n)^T$.

Definição 4.3 (Pontos Independentes)**.** *Um conjunto de* $k+1$ *pontos em* $\mathbb{R}^k$ *é chamado* ***independente*** *se os pontos deste conjunto não pertencerem a um hiperplano de dimensão* $k-1$ *(ver definição 4.2).*

Por exemplo, um conjunto de $k+1$ pontos em $\mathbb{R}^k$ é independente se os pontos deste conjunto não forem colineares, caso $k = 2$, ou coplanares, caso $k = 3$.

Definição 4.4 (Pontos Vizinhos)**.** *Um ponto* x_i *é chamado* ***vizinho*** *de um ponto* x_j *se existe* d_{ij}*, isto é, uma distância entre* x_i *e* x_j*. No grafo* $G = (V, E, d)$*, os vértices representados por estes pontos são adjacentes.*

O exemplo bidimensional é uma forma de se ter sensibilidade para o caso tridimensional, que é o que se pretende abordar. Segundo Robert T. Davis, em [37], dado um conjunto de distâncias entre quatro pontos não coplanares em $\mathbb{R}^3$, então as coordenadas destes pontos podem ser unicamente determinadas a menos de *movimentos rígidos*, que é uma combinação de uma translação, uma rotação e, possivelmente, uma reflexão. Isso se deve ao fato de as distâncias entre quatro pontos não coplanares definirem um tetraedro. Essa afirmação torna possível, então, a formação de uma base métrica tridimensional e, quando associada ao lema 4.1 e ao teorema 4.1, serve de base para o algoritmo *geometric build-up* (GB).

Lema 4.1. *Sejam 4 átomos de coordenadas* $x_i \in \mathbb{R}^3$, $i = 1, \ldots, 4$*, determinadas, e seja* $d_{ij} \in \mathbb{R}$ *as distâncias entre os átomos* i *e os átomos* j, $j = 1, \ldots, 4$*. Se esses átomos forem não coplanares, então a matriz* A *dada por*

$$A = \begin{bmatrix} (x_2 - x_1)^T \\ (x_3 - x_1)^T \\ (x_4 - x_1)^T \end{bmatrix}$$

apresenta linhas e colunas linearmente independentes.

Demonstração. Tendo todas as distâncias entre os 4 átomos disponíveis, isto é,

$$||x_i - x_j|| = d_{ij},$$

tem-se, pela definição de norma, $x_i - x_j \neq 0$. Primeiramente, será mostrado que as linhas $(x_i - x_j)^T$ são linearmente independentes. A demonstração será feita por absurdo. Suponha que as linha da matriz A são linearmente dependentes, isto é,

existem $\alpha, \beta, \gamma \in \mathbb{R}$ tais que

$$\alpha(x_2 - x_1)^T + \beta(x_3 - x_1)^T + \gamma(x_4 - x_1)^T = 0^T,$$

ou

$$\alpha(x_2 - x_1) + \beta(x_3 - x_1) + \gamma(x_4 - x_1) = 0,$$

onde pelo menos um dos escalares α, β, γ é diferente de zero. Reescrevendo a equação, tem-se

$$\begin{aligned} \alpha x_2 + \beta x_3 + \gamma x_4 - (\alpha + \beta + \gamma)x_1 &= 0 \\ \frac{\alpha}{\alpha + \beta + \gamma} x_2 + \frac{\beta}{\alpha + \beta + \gamma} x_3 + \frac{\gamma}{\alpha + \beta + \gamma} x_4 &= x_1, \end{aligned}$$

isto é, x_1 seria uma combinação linear dos vetores x_2, x_3, x_4 e os 4 átomos estariam no mesmo plano, o que contradiz a hipótese. Como, pelo teorema do posto, o posto linha é igual ao posto coluna, o resultado também vale para as colunas. Logo, as linhas e colunas de A são linearmente independentes. □

Teorema 4.1 (Base Métrica em $\mathbb{R}^3$)**.** *Se as coordenadas de 4 átomos não coplanares x_i, $i = 1, 2, 3, 4$, e as distâncias d_{ij}, $i = 1, 2, 3, 4$, ao átomo x_j são dadas, então, as coordenadas do quinto átomo x_j podem ser unicamente determinadas. Em outras palavras, qualquer conjunto de 4 pontos independentes em $\mathbb{R}^3$ forma uma base métrica para $\mathbb{R}^3$.*

Demonstração. Sejam $x_i = (u_i,\ v_i,\ w_i)^T$, $i = 1, 2, 3, 4$, as coordenadas dos 4 primeiros átomos e $x_j = (u_j,\ v_j,\ w_j)^T$ as coordenadas do quinto átomo. Tem-se então um conjunto de equações $\|x_i - x_j\| = d_{ij}$, $i = 1, 2, 3, 4$. Elevando-se as equações ao quadrado e expandindo-as, tem-se

$$\begin{aligned} ||x_i - x_j||^2 &= d_{ij}^2 \\ ||(u_i - u_j, v_i - v_j, w_i - w_j)^T||^2 &= d_{ij}^2 \\ (u_i - u_j)^2 + (v_i - v_j)^2 + (w_i - w_j)^2 &= d_{ij}^2 \\ (u_i^2 + v_i^2 + w_i^2) + (u_j^2 + v_j^2 + w_j^2) - 2(u_i u_j + v_i v_j + w_i w_j) &= d_{ij}^2, \end{aligned}$$

isto é,

$$||x_i||^2 + ||x_j||^2 - 2x_i^T x_j = d_{ij}^2,\ i = 1, 2, 3, 4.$$

Subtraindo-se a primeira equação das demais, pode-se reduzir o número de equações de quatro para três, como segue

$$-2(x_i - x_1)^T x_j = (d_{ij}^2 - d_{ij}^2) - (\|x_i\|^2 - \|x_1\|^2),\ i = 2, 3, 4.$$

Definindo-se a matriz A e o vetor b por:

$$A = -2\begin{bmatrix}(x_2 - x_1)^T\\(x_3 - x_1)^T\\(x_4 - x_1)^T\end{bmatrix} \text{ e } b = \begin{bmatrix}(d_{2j}^2 - d_{1j}^2) - (\|x_2\|^2 - \|x_1\|^2)\\(d_{3j}^2 - d_{1j}^2) - (\|x_3\|^2 - \|x_1\|^2)\\(d_{4j}^2 - d_{1j}^2) - (\|x_4\|^2 - \|x_1\|^2)\end{bmatrix}$$

tem-se

$$Ax_j = b.$$

Como x_i, $i = 1, 2, 3, 4$, não são coplanares, a matriz A possui linhas e colunas linearmente independentes (lema 4.1) e por isso determinante diferente de zero. Assim, o sistema acima pode ser resolvido para se obter uma única solução x_j. Esta demonstração pode ser encontrada em [33, 37, 38]. □

É importante notar a questão da instabilidade do sistema do teorema 4.1. Quando os pontos da base métrica formam tetraedros com alturas muito pequenas (isto é, são quase coplanares), computacionalmente, a matriz A pode apresentar linhas ou colunas linearmente dependentes, resultando em soluções ruins para esse sistema.

O algoritmo 4.1 se baseia no teorema 4.1. Neste algoritmo é possível observar que, se houver n átomos em uma molécula, resolve-se o sistema proposto no teorema 4.1, no máximo, $n - 4$ vezes.

Tem-se, então, um algoritmo para determinar a estrutura de uma molécula que pode ser resolvido em tempo de execução linear quando todas as distâncias exatas entre todos os pares de átomos são dadas. Mais ainda, quando todas as distâncias estão disponíveis uma única base métrica pode ser utilizada para se determinar todos os átomos.

Ainda há a dúvida quanto ao que se fazer quando se tem apenas algumas distâncias (*distâncias esparsas*). Para se esclarecer esta dúvida, passa-se ao próximo título, a seção 4.2.

Algoritmo 4.1 GB - para MDGP com todas as distâncias exatas

1: Encontre as coordenadas de 4 átomos que não estejam no mesmo plano e que possuam distâncias entre si.
2: **for all** (Átomo não posicionado) **do**
3: Determine suas coordenadas utilizando suas distâncias em relação a base métrica encontrada em 1.
4: **end for**
5: Todos os átomos foram determinados.

4.2 Distâncias exatas e esparsas

Embora que, na subseção 4.1, tenha sido descrito um algoritmo capaz de resolver o MDGP em um tempo linear de execução, sabe-se que não se tem todas as distâncias a todos os átomos, conforme o algoritmo pede. Levando-se em consideração que, para se determinar cada átomo, utilizam-se somente quatro distâncias de todas aquelas que se tem disponíveis[2], e que, para os átomos da base métrica, são necessárias seis distâncias, pode-se reescrever o algoritmo de tal forma que seja possível a sua execução a partir de apenas algumas distâncias. Para tal, é importante observar os teoremas 4.2 e 4.3 enunciados a seguir.

Teorema 4.2 (Condição Necessária [38])**.** *Uma condição necessária para a determinação única das coordenadas de um grupo de pontos $x_1, x_2, \ldots, x_n \in \mathbb{R}^k$, dado um conjunto de distâncias entre estes pontos, é que cada ponto deve ter no mínimo $k+1$ distâncias a $k+1$ pontos, assumindo que os $k+1$ pontos não pertençam a um hiperplano de dimensão k.*

O teorema 4.2, enunciado em [38], se baseia no fato de que, em $\mathbb{R}^k$, um ponto pode ser definido unicamente somente se ele tem $k+1$ distâncias de $k+1$ pontos independentes. Se ele tiver somente k distâncias de k pontos, o ponto a ser determinado terá duas posições (ocorre uma reflexão em relação ao hiperplano definido pelos demais k pontos).

Teorema 4.3 (Condição Suficiente [38])**.** *Uma condição suficiente para a determinação única das coordenadas de um grupo de pontos $x_1, x_2, \ldots, x_n \in \mathbb{R}^k$, dado um conjunto de distâncias entre estes pontos é que em todos os passos do algoritmo geometric build-up exista um ponto indeterminado com $k+1$ distâncias de $k+1$ pontos independentes e determinados.*

Note que, se a condição dada pelo teorema 4.3 é satisfeita em todos os passos do algoritmo 4.2, significa que este é capaz de determinar as coordenadas de todos os pontos de forma única.

[2]Quatro distâncias entre o átomo a ser determinado e os átomos da base métrica

Algoritmo 4.2 GB - para MDGP com distâncias esparsas exatas

1: Encontre as coordenadas de 4 átomos que não estejam no mesmo plano e que possuam distâncias entre si;
2: **repeat**
3: Para um átomo não posicionado, encontre quatro átomos posicionados para servir de base métrica;
4: Determine as coordenadas deste novo átomo utilizando as distâncias entre ele e os átomos da base métrica;
5: Adicione este átomo ao conjunto de átomos posicionados;
6: **until** (Todos os átomos determinados)
7: **if** (Nenhum átomo for determinado durante toda a estrutura de repetição) **then**
8: Não foi encontrada uma solução;
9: **end if**
10: Todos os átomos foram determinados.

Note que a principal diferença entre os algoritmos 4.1 e 4.2 é que, no segundo, há frequentemente mudança de base, enquanto que no primeiro a base métrica permanece até o final do algoritmo.

É importante observar que tanto no algoritmo 4.1 quanto no algoritmo 4.2 a escolha da base não é trivial, pois, além de se ter uma quantidade grande de pontos que podem ser escolhidos para formar este conjunto de quatro pontos, ainda há a possibilidade de haver mais de um conjunto (caso se considere as rotações, são infinitos).

Agora, o que está faltando é uma forma para se calcular as coordenadas dos quatro átomos da base métrica inicial. É o que propõe a subseção a seguir.

4.3 Coordenadas dos átomos da base métrica inicial

O processo de construção geométrica da estrutura molecular de uma proteína, segundo os algoritmos 4.1 e 4.2, se inicia através da determinação das coordenadas dos quatro átomos da base métrica inicial. Sendo assim, fazendo-se $x_i = (u_i, v_i, w_i)^T$, $i \in \{1, 2, 3, 4\}$, como foi feito no exemplo 4.1, posiciona-se o primeiro átomo, x_1, na origem, isto é,

$$x_1 = (0, 0, 0)^T.$$

Para o segundo átomo, tem-se

$$x_2 = (d_{12}, 0, 0)^T,$$

assim como no exemplo. E para o terceiro átomo, pode-se escrever

$$\begin{aligned} u_3^2 + v_3^2 &= d_{13}^2 \\ (u_3 - u_2)^2 + v_3^2 &= d_{23}^2 \\ w_3 &= 0. \end{aligned}$$

Resolvendo-se o sistema, tem-se as coordenadas de x_3, assim

$$\begin{aligned} u_3 &= (d_{13}^2 - d_{23}^2)/(2u_2) + u_2/2 \\ v_3 &= \pm\sqrt{d_{13}^2 - u_3^2} \\ w_3 &= 0, \end{aligned}$$

onde é possível escolher v_3 positivo sem comprometer a estrutura da molécula.

O quarto átomo pode ser encontrado de forma parecida, assim

$$\begin{aligned} u_4^2 + v_4^2 + w_4^2 &= d_{14}^2 \\ (u_4 - u_2)^2 + v_4^2 + w_4^2 &= d_{24}^2 \\ (u_4 - u_3)^2 + (v_4 - v_3) + w_4^2 &= d_{34}^2. \end{aligned}$$

Encontrando-se $x_4 = (u_4, v_4, w_4)^T$, tem-se

$$\begin{aligned} u_4 &= (d_{14}^2 - d_{24}^2)/(2u_2) + u_2/2 \\ v_4 &= (d_{24}^2 - d_{34}^2 - (u_4 - u_2)^2 + (u_4 - u_3)^2)/(2v_3) + v_3/2 \\ w_4 &= \pm\sqrt{d_{14}^2 - u_4^2 - v_4^2}. \end{aligned}$$

Como no cálculo das coordenadas de x_3, w_4 também pode ser considerado positivo.

4.4 Variações do algoritmo Geometric Build-up

Nesta seção serão vistas duas modificações do algoritmo geometric build-up. A primeira trata a questão da estabilidade numérica e originou o algoritmo *updated geometric build-up*, visto na seção 4.4.1. A segunda modificação trata a questão do desempenho computacional e originou o algoritmo *revised updated geometric build-up*, visto na seção 4.4.2. Somente este último será considerado, neste trabalho, para efeito de testes e comparações.

4.4.1 Updated Geometric Build-up

Neste capítulo, foi visto o algoritmo geometric buid-up (GB) para resolução do MDGP. Segundo [37–39], o maior problema no GB é a instabilidade numérica quando a proteína apresenta um grande número de átomos. Esta instabilidade numérica, que se deve à dificuldade de se resolver os sistemas lineares quase singulares que aparecem quando os vetores da base métrica são quase linearmente dependentes, introduz erros nas coordenadas calculadas e, devido a natureza iterativa do algoritmo, estes erros podem se acumular. Por isso, Di Wu e Zhijun Wu, em [39], propuseram, a partir do GB, um novo algoritmo capaz de reduzir esses erros, o *updated geometric buid-up* (UGB). A ideia principal do UGB é recalcular as coordenadas dos quatro átomos da base métrica a cada iteração, utilizando-se, para tal, as distâncias originais d_{ij}.

Assim, seja F o conjunto de todos os átomos determinados pelo UGB (isto é, cujas coordenadas já foram encontradas). No momento de se calcular as coordenadas de mais um átomo a, escolhe-se uma base métrica cujos elementos $b_1, b_2, b_3, b_4 \in F$ possuam todas as distâncias entre si. Desta forma, a cada passo do UGB, é possível obter dois conjuntos de coordenadas para os átomos da base métrica: o conjunto $\bar{X}$, formado pelas coordenadas de b_1, b_2, b_3, b_4 que foram calculadas pela resolução do sistema linear proposto pelo teorema 4.1 a partir de outras bases métricas, e o conjunto $\bar{Y}$, formado pelas coordenadas de b_1, b_2, b_3, b_4 recalculadas a partir das suas distâncias, obtidas conforme a seção 4.3. Estes dois conjuntos, aqui representados por duas matrizes 4×3, podem ser visualizados como dois tetraedros. O objetivo da rotina de atualização das coordenadas da base métrica, então, é, através de movimentos de rotações e translações (este último se deve ao fato de $\bar{X}$ estar na estrutura da molécula e $\bar{Y}$ ter um de seus vértices na origem), retirar o tetraedro $\bar{X}$ da estrutura da proteína e substituí-lo por um tetraedro Y, mais preciso. O problema, agora, é sobrepor o tetraedro $\bar{Y}$ em $\bar{X}$ de tal forma que a diferença das coordenadas de seus vértices seja mínima. Logo, trata-se de uma comparação de estruturas, onde pode ser aplicada a RMSD, vista na seção 3.2.

Então, considere

$$\bar{X} = \begin{bmatrix} \bar{x}_1^T \\ \bar{x}_2^T \\ \bar{x}_3^T \\ \bar{x}_4^T \end{bmatrix} = \begin{bmatrix} \bar{x}_{11} & \bar{x}_{12} & \bar{x}_{13} \\ \bar{x}_{21} & \bar{x}_{22} & \bar{x}_{23} \\ \bar{x}_{31} & \bar{x}_{32} & \bar{x}_{33} \\ \bar{x}_{41} & \bar{x}_{42} & \bar{x}_{43} \end{bmatrix} \text{ e } \bar{Y} = \begin{bmatrix} \bar{y}_1^T \\ \bar{y}_2^T \\ \bar{y}_3^T \\ \bar{y}_4^T \end{bmatrix} = \begin{bmatrix} \bar{y}_{11} & \bar{y}_{12} & \bar{y}_{13} \\ \bar{y}_{21} & \bar{y}_{22} & \bar{y}_{23} \\ \bar{y}_{31} & \bar{y}_{32} & \bar{y}_{33} \\ \bar{y}_{41} & \bar{y}_{42} & \bar{y}_{43} \end{bmatrix},$$

lembrando-se que $\bar{x}_i$ são as coordenadas dos b_i, $i = 1 \dots 4$, calculadas a partir do teorema 4.1, e y_i são as coordenada de b_i, $i = 1 \dots 4$, recalculadas a partir das

distâncias conforme a seção 4.3, isto é,

$$\bar{y}_1 = \begin{bmatrix} \bar{y}_{11} \\ \bar{y}_{12} \\ \bar{y}_{13} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad \bar{y}_2 = \begin{bmatrix} \bar{y}_{21} \\ \bar{y}_{22} \\ \bar{y}_{23} \end{bmatrix} = \begin{bmatrix} d_{12} \\ 0 \\ 0 \end{bmatrix}, \quad \bar{y}_3 = \begin{bmatrix} \bar{y}_{31} \\ \bar{y}_{32} \\ \bar{y}_{33} \end{bmatrix} = \begin{bmatrix} (d_{13}^2 - d_{23}^2)/(2d_{12}) + d_{12}/2 \\ \sqrt{d_{13}^2 - \bar{y}_{31}^2} \\ 0 \end{bmatrix},$$

$$\bar{y}_4 = \begin{bmatrix} \bar{y}_{41} \\ \bar{y}_{42} \\ \bar{y}_{43} \end{bmatrix} = \begin{bmatrix} (d_{14}^2 - d_{24}^2)/(2d_{12}) + d_{12}/2 \\ (d_{24}^2 - d_{34}^2 - (\bar{y}_{41} - d_{12})^2 + (\bar{y}_{41} - \bar{y}_{31})^2)/(2\bar{y}_{32}) + \bar{y}_{32}/2 \\ \sqrt{d_{14}^2 - \bar{y}_{41}^2 - \bar{y}_{42}^2} \end{bmatrix},$$

onde d_{ij} é a distância entre os átomos b_i e b_j da base a ser recalculada.

Em primeiro lugar, ajustam-se as estruturas $\bar{X}$ e $\bar{Y}$ de modo que apresentem o mesmo centro geométrico, posicionando-se ambos na origem. Expressando-se este procedimento matematicamente, pode-se dizer que se deve apurar

$$\begin{aligned} \bar{x}_c(j) &= \frac{1}{4}\sum_{i=1}^{4} \bar{X}(i,j), \ j = 1,2,3 \\ \bar{y}_c(j) &= \frac{1}{4}\sum_{i=1}^{4} \bar{Y}(i,j), \ j = 1,2,3 \end{aligned}$$

e, logo depois, atualizar

$$\begin{aligned} X &= \bar{X} - \zeta_4 \bar{x}_c^T, \\ Y &= \bar{Y} - \zeta_4 \bar{y}_c^T. \end{aligned}$$

Tem-se que a rotina de atualização das coordenadas da base métrica deve resolver o problema

$$\text{RMSD}(X,Y) = \min_{Q} ||X - YQ||_F/\sqrt{4},$$

obtendo-se a matriz ortogonal Q, para $n = 4$, conforme visto na seção 3.2. Esta matriz será utilizada para rotacionar o tetraedro Y, fazendo-se

$$\bar{Y} = YQ.$$

Agora, para que o tetraedro $\bar{Y}$ seja posicionado dentro da estrutura da proteína, deve-se fazer uma translação de $\bar{x}_c$ em cada um de seus vértices, isto é,

$$Y = \bar{Y} + \zeta_4 \bar{x}_c^T.$$

O exemplo 4.2 e a figura 4.2, a seguir, mostram como acontece a atualização da base métrica.

Exemplo 4.2. *Supondo-se que, para determinar o átomo* $\boldsymbol{a}$*, foi utilizada a seguinte*

base

$$\bar{X} = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 1 \\ 1 & 2 & 2 \\ 2 & 1 & 2 \end{bmatrix},$$

onde todas as distâncias são conhecidas e iguais a $\sqrt{2}$*. Agora, calculam-se as coordenadas necessárias para se formar a matriz* $\bar{Y}$*, conforme a subseção 4.3. Assim, tem-se*

$$\bar{Y} = \begin{bmatrix} 0 & 0 & 0 \\ 1,4142 & 0 & 0 \\ 0,7071 & 1,2247 & 0 \\ 0,7071 & 0,4082 & 1,1547 \end{bmatrix}.$$

Apurando-se os centros geométricos para $\bar{X}$ *e* $\bar{Y}$*, tem-se*

$$\begin{aligned} \bar{x}_c(j) &= \frac{1}{4}\sum_{i=1}^{4} \bar{X}(i,j) = (1,5 \quad 1,5 \quad 1,5)^T, \ j = 1,2,3 \\ \bar{y}_c(j) &= \frac{1}{4}\sum_{i=1}^{4} \bar{Y}(i,j) = (0,7071 \quad 0,4082 \quad 0,2887)^T, \ j = 1,2,3. \end{aligned}$$

Na figura 4.2a, é possível visualizar os tetraedros, $\bar{X}$ *em vermelho e* $\bar{Y}$ *em azul, e seus centros geométricos e, na figura 4.2b, pode-se ver a translação dos tetraedros, quando seus centros geométricos coincidem com a origem (ponto em verde). Esta translação foi feita da seguinte forma:*

$$X = \bar{X} - \zeta_4 \bar{x}_c^T = \begin{bmatrix} -0,5 & -0,5 & -0,5 \\ 0,5 & 0,5 & -0,5 \\ -0,5 & 0,5 & 0,5 \\ 0,5 & -0,5 & 0,5 \end{bmatrix} \quad e$$

$$Y = \bar{Y} - \zeta_4 \bar{y}_c^T = \begin{bmatrix} -0,7071 & -0,4082 & -0,2887 \\ 0,7071 & -0,4082 & -0,2887 \\ 0 & -0,8165 & -0,2887 \\ 0 & 0 & 0,8660 \end{bmatrix}.$$

Ainda na figura 4.2b, estas novas matrizes X *e* Y *estão representadas em vermelho e azul, respectivamente, e em traço cheio. Os tetraedros tracejados são as posições antigas,* $\bar{X}$ *e* $\bar{Y}$*. Tendo-se os dois tetraedros com centros geométricos na origem,*

resolve-se, então, a RMSD (seção 3.2) e obtem-se a matriz ortogonal

$$Q \approx \begin{bmatrix} 0.7071 & 0.7071 & -0.0000 \\ -0.4082 & 0.4082 & 0.8165 \\ 0.5774 & -0.5774 & 0.5774 \end{bmatrix},$$

que multiplicada pela direita por Y resulta em

$$\bar{Y} = YQ = \begin{bmatrix} -0.5 & -0.5 & -0.5 \\ 0.5 & 0.5 & -0.5 \\ -0.5 & 0.5 & 0.5 \\ 0.5 & -0.5 & 0.5 \end{bmatrix}.$$

A figura 4.2c mostra a rotação do tetraedro Y. Note que o tetraedro X foi sobreposto por YQ. Fazendo-se a translação do tetraedro $\bar{Y}$, utilizando-se o centro geométrico $\bar{x}_c$, tem-se

$$Y = \bar{Y} + \zeta_4 \bar{x}_c^T = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 1 \\ 1 & 2 & 2 \\ 2 & 1 & 2 \end{bmatrix},$$

visto na figura 4.2d, onde Y aparece na posição de $\bar{X}$. É importante observar que houve uma sobreposição de $\bar{X}$ por Y e que, se $\bar{X}$ estivesse com as arestas diferentes de $\sqrt{2}$, seria substituído por Y, que teria as arestas no tamanho correto, conforme as distâncias d_{ij} originais. É desta forma que o algoritmo UGB atualiza as coordenadas dos átomos da base métrica a cada passo.

O algoritmo 4.3 mostra o UGB. Nota-se, neste algoritmo, um importante fato: tem que se encontrar uma base métrica a cada iteração. Lembrando-se que, para a determinação de um átomo a, deve-se encontrar quatro átomos determinados e vizinhos a a, com todas as distâncias entre si, isto é, utilizando-se a linguagem da teoria de grafos, deve-se encontrar um 4-clique[3]. No algoritmo que será visto a seguir, trata-se este problema organizando-se os dados de forma diferente e, além disso, utilizando-se triângulos no lugar de tetraedros, isto é, buscando a cada iteração um 3-clique, onde se busca somente três distâncias e não seis como no caso de 4-cliques.

[3]Um *clique* em um grafo G é um subgrafo completo. Em outras palavras, é uma subgrafo onde cada vértice é adjacente aos demais vértices. Um k-clique é um clique de k vértices.

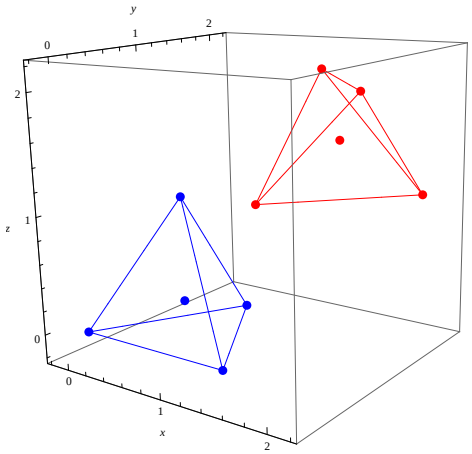
Algoritmo 4.3 UGB - para MDGP com distâncias esparsas exatas

1: Encontre quatro átomos que não estejam no mesmo plano e que tenham todas as distâncias entre si;
2: Determine suas coordenadas;
3: Seja L o conjunto de átomos sem coordenadas determinadas;
4: **while** $L \neq \emptyset$ **do**
5: **for** $a \in L$ **do**
6: Encontre quatro átomos determinados que sirva como base métrica para a e que tenham todas as distâncias entre si;
7: **if** os quatro átomos foram encontrados **then**
8: Encontre as coordenadas de a;
9: Remova a de L;
10: Atualize as coordenadas dos quatro átomos da base;
11: Coloque os átomos novamente na estrutura original;
12: **end if**
13: **end for**
14: Se nenhum átomo for determinado, pare;
15: **end while**
16: Todos os átomos foram determinados!

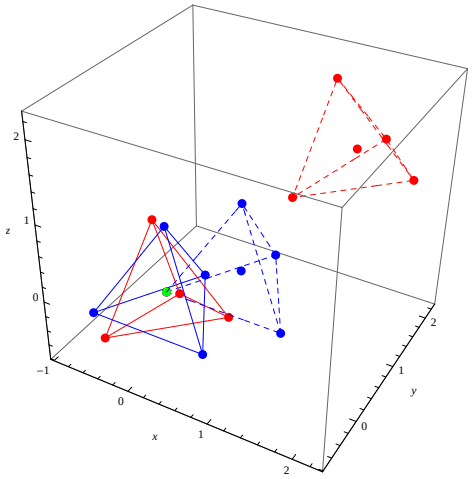
Note que, para a execução do algoritmo 4.3, consideram-se como satisfeitas as condições expressas nos teoremas 4.2 e 4.3.

4.4.2 Revised Updated Geometric Build-up

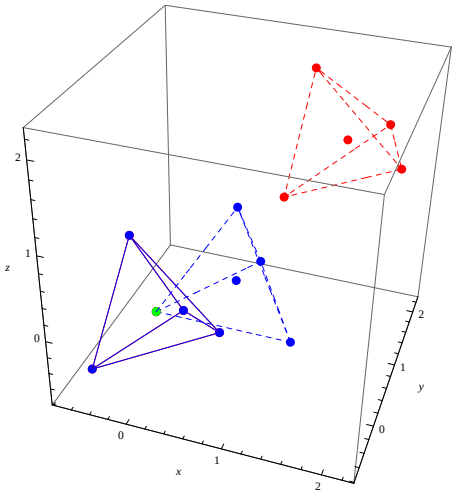
Embora o UGB, visto na subseção 4.4.1, consiga controlar a propagação de erros, segundo [37–39], este algoritmo requer uma busca por quatro átomos que tenham todas as distâncias entre si a cada iteração. Em [37, 38], foi utilizada, então, uma nova abordagem para o algoritmo geometric build-up, onde uma nova estrutura de dados foi empregada e cada átomo passou a ser determinado rigidamente, isto é, necessitando apenas de três distâncias a três átomos determinados. O algoritmo que utiliza esta nova abordagem é conhecido como *revised updated geometric build-up* (RUGB).



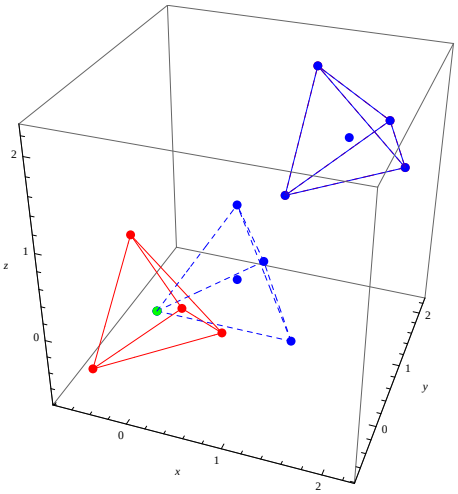
(a) Tetraedros $\bar{X}$ e $\bar{Y}$ do exemplo 4.2 nas posições iniciais.



(b) Tetraedros X e Y com centros geométricos na origem. Tetraedros pontilhados representam as posições iniciais $\bar{X}$ e $\bar{Y}$.



(c) Rotação do tetraedro Y aplicando-se Q pela direita.



(d) O tetraedro Y toma o lugar do tetraedro $\bar{X}$ na posição inicial (dentro da estrutura da molécula).

Figura 4.2: Atualização da base métrica no UGB.

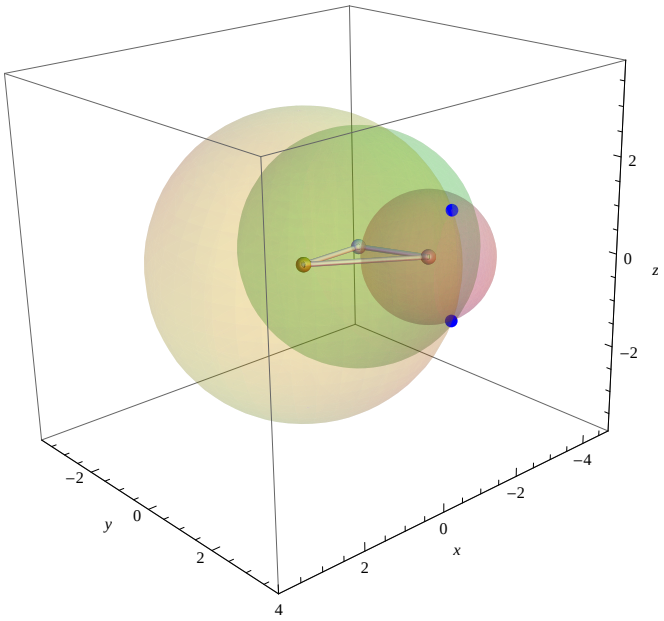


Figura 4.3: Determinação rígida do átomo a, representado em azul, com duas posições na figura.

Na figura 4.3, é possível visualizar a determinação rígida de um átomo. O algoritmo RUGB é baseado no UGB, mas com uma relaxação na busca da base métrica e uma estrutura de dados mais adaptada para facilitar esta busca. A relaxação consiste em considerar apenas três átomos para base métrica com distâncias entre si disponíveis e um átomo adicional, não coplanar aos três átomos da base e vizinho ao átomo a ser determinado, que servirá como referência para se escolher uma única solução da determinação rígida. A atualização baseada na RMSD, presente no algoritmo UGB, também é feita no RUGB, porém, no lugar de translações e rotações de tetraedros, tem-se rotações e translações de triângulos. Na figura 4.4 e no exemplo 4.3, pode-se ver esta atualização.

Exemplo 4.3. *Seja*

$$\bar{X} = \begin{bmatrix} \bar{x}_1^T \\ \bar{x}_2^T \\ \bar{x}_3^T \end{bmatrix} = \begin{bmatrix} -0.3396 & 1.2233 & -2.3965 \\ 0.8787 & 0.6059 & -1.8755 \\ 2.0688 & 1.3737 & -2.4223 \end{bmatrix}$$

a matriz formada pelas coordenadas dos átomos b_1, b_2, b_3 da base métrica. Sejam

$$\bar{Y} = \begin{bmatrix} \bar{y}_1^T \\ \bar{y}_2^T \\ \bar{y}_3^T \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 1.4618 & 0 & 0 \\ 1.9344 & 1.4427 & 0 \end{bmatrix}$$

a matriz formada pelas coordenadas recalculadas da base e

$$\begin{aligned}\bar{a}_1 &= (1.2106 \quad 2.4275 \quad 0.0683)^T \\ \bar{a}_2 &= (1.2106 \quad 2.4275 \quad -0.0683)^T\end{aligned}$$

as coordenadas do átomo $\boldsymbol{a}$ *determinadas rigidamente, conforme a subseção 4.3, a partir das coordenadas de* $\bar{Y}$*. Apurando-se os centros geométricos de* $\bar{X}$ *e* $\bar{Y}$*, tem-se*

$$\begin{aligned}\bar{x}_c(j) &= \frac{1}{3}\sum_{i=1}^{3}\bar{X}(i,j) = (0.8693 \quad 1.0676 \quad -2.2315)^T,\ j = 1,2,3 \\ \bar{y}_c(j) &= \frac{1}{3}\sum_{i=1}^{3}\bar{Y}(i,j) = (1.1321 \quad 0.4809 \quad 0)^T,\ j = 1,2,3.\end{aligned}$$

Assim, é possível fazer translações dos triângulos $\bar{X}$ *e* $\bar{Y}$ *desta forma*

$$X = \bar{X} - \zeta_3\bar{x}_c^T = \begin{bmatrix} -1.2089 & 0.1557 & -0.1651 \\ 0.0094 & -0.4617 & 0.3559 \\ 1.1994 & 0.3061 & -0.1909 \end{bmatrix} \quad e$$

$$Y = \bar{Y} - \zeta_3\bar{y}_c^T = \begin{bmatrix} -1.1321 & -0.4809 & 0 \\ 0.3297 & -0.4809 & 0 \\ 0.8023 & 0.9618 & 0 \end{bmatrix}.$$

Na figura 4.4a, é possível visualizar os dois triângulos, $\bar{X}$ *em vermelho e* $\bar{Y}$ *em azul, e seus centros geométricos e, na figura 4.4b, pode-se ver a translação dos triângulos, quando seus centros geométricos coincidem com a origem (ponto em verde). No RUGB, há também a translação de* $\bar{a}_1$ *e* $\bar{a}_2$*,*

$$\begin{aligned}a_1 &= \bar{a}_1 - \bar{y}_c = (0.0785 \quad 1.9467 \quad 0.0683)^T \\ a_2 &= \bar{a}_2 - \bar{y}_c = (0.0785 \quad 1.9467 \quad -0.0683)^T.\end{aligned}$$

Resolvendo-se, a RMSD (seção 3.2), obtem-se a matriz ortogonal

$$Q = \begin{bmatrix} 0.8334 & -0.4224 & 0.3564 \\ 0.5519 & 0.6706 & -0.4958 \\ -0.0296 & 0.6099 & 0.7919 \end{bmatrix},$$

que aplicada pela direita em Y *resulta em*

$$\bar{Y} = YQ = \begin{bmatrix} -1.2089 & 0.1557 & -0.1651 \\ 0.0094 & -0.4617 & 0.3559 \\ 1.1994 & 0.3061 & -0.1909 \end{bmatrix}$$

e, em a_1 e a_2, resulta em

$$\begin{aligned} \bar{a}_1 &= a_1^T Q = (1.1377 \quad 1.3139 \quad -0.8830)^T \\ \bar{a}_2 &= a_2^T Q = (1.1417 \quad 1.2305 \quad -0.9912)^T. \end{aligned}$$

A figura 4.4c mostra a nova posição do triângulo $\bar{=}YQ$. Note que o triângulo X foi sobreposto por $\bar{Y} = YQ$. Fazendo-se a translação de $\bar{Y}$, utilizando-se o centro geométrico $\bar{x}_c$, tem-se

$$Y = \bar{Y} + \zeta_3 \bar{x}_c^T = \begin{bmatrix} -0.3396 & 1.2233 & -2.3965 \\ 0.8787 & 0.6059 & -1.8755 \\ 2.0688 & 1.3737 & -2.4223 \end{bmatrix},$$

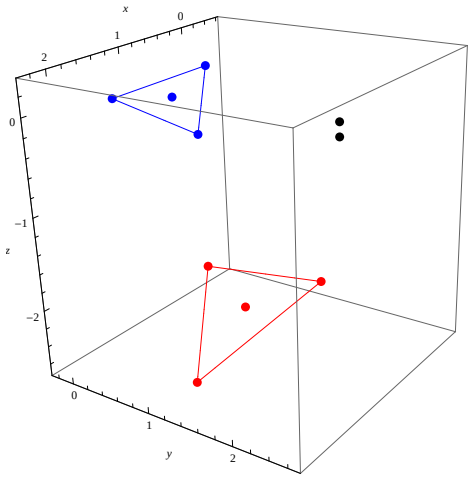
e, finalmente, para a_1 e a_2, tem-se

$$\begin{aligned} a_1 &= \bar{a}_1 + \bar{x}_c = (2.0070 \quad 2.3815 \quad -3.1144)^T \\ a_2 &= \bar{a}_2 + \bar{x}_c = (2.0110 \quad 2.2981 \quad -3.2227)^T. \end{aligned}$$

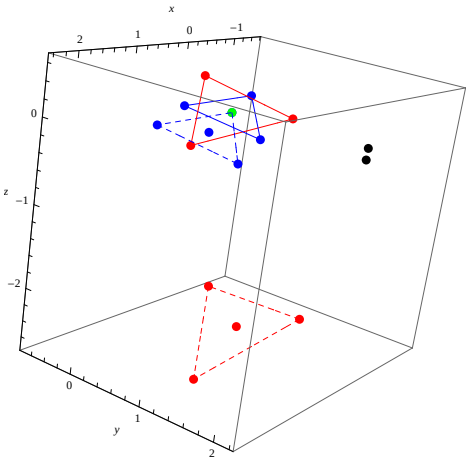
O final deste cálculo pode ser visto na figura 4.4d, onde Y se sobrepõe a $\bar{X}$. As duas posições possíveis para o átomo $\boldsymbol{a}$ também aparecem dentro da estrutura da molécula (dois pontos pretos). Basta, agora, tomar um outro átomo com distância a $\boldsymbol{a}$ disponível para servir de referência para a escolha de uma única solução.

Uma segunda modificação aplicada ao RUGB é a criação de uma estrutura de dados mais apropriada à busca da base, onde o acesso aos átomos vizinhos de cada átomo foi facilitado, utilizando-se o conceito de *grau* de um átomo, que é o número de vizinhos de cada átomo. Chamando-se o maior grau dentre todos os átomos de uma molécula de d_{max}, pode-se gerar uma matriz $n \times d_{max}$ onde cada linha i armazena os vizinhos j ($j = 1 \ldots d_{max}$) do átomo i. Assim, ao se buscar os quatro átomos, os três da base e o adicional, faz-se uma busca em, no máximo, d_{max} posições, evitando-se uma procura por todos os n átomos.

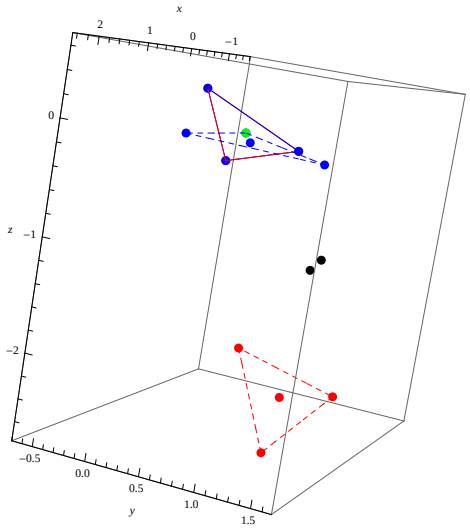
No UGB, considerando-se somente a busca exaustiva da base métrica por todos os n átomos, pode custar até $\mathcal{O}(n^4)$ passos, resultando em um tempo de execução total de $\mathcal{O}(n^6)$, acrescentando-se as duas estruturas de repetição presentes no algoritmo. Já no RUGB, devido a matriz $n \times d_{max}$, pode custar em torno de $\mathcal{O}(d_{max}^3)$ para se encontrar a base e d_{max} passos para se encontrar o quarto átomo, o que resulta em um tempo de execução total de $\mathcal{O}(n^2 d_{max}^3)$. Tem-se neste fato um ganho computacional, pois $d_{max} < n$.



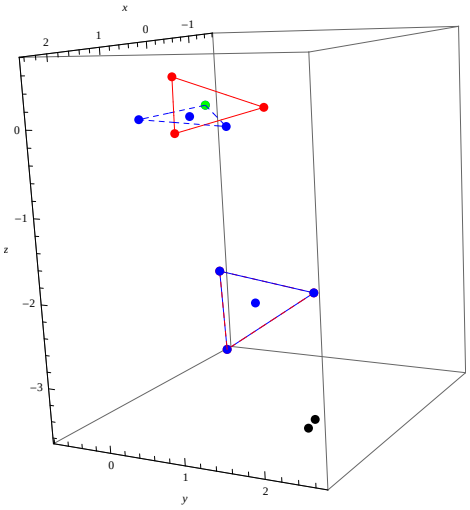
(a) Triângulos $\bar{X}$ e $\bar{Y}$ do exemplo 4.3 nas posições iniciais.



(b) Triângulos X e Y com centros geométricos na origem. Triângulos pontilhados representam as posições iniciais, $\bar{X}$ e $\bar{Y}$.



(c) Rotação do triângulo Y aplicando-se Q pela direita.



(d) O triângulo $\bar{Y} = YQ$ toma o lugar do triângulo $\bar{X}$ na posição inicial (dentro da estrutura da molécula). Note que há duas posições para o átomo a ser determinado, representadas por dois pontos pretos em todas as figuras.

Figura 4.4: Atualização da base métrica no RUGB.

Algoritmo 4.4 RUGB - para MDGP com distâncias esparsas exatas

1: Encontre quatro átomos que não estejam no mesmo plano e que tenham todas as distâncias entre si;
2: Determine suas coordenadas;
3: Seja L o conjunto de átomos sem coordenadas determinadas;
4: **while** $L \neq \emptyset$ **do**
5: **for** $a \in L$ **do**
6: Encontre três átomos b_1, b_2, b_3 vizinhos a a e que tenham todas as distâncias entre si;
7: Encontre um átomo adicional b_4 vizinho a a;
8: Verifique se b_1, b_2, b_3, b_4 não estão no mesmo plano;
9: **if** Nenhum átomo satisfaz as características acima **then**
10: Encerra-se o algoritmo e nenhum átomo pode ser determinado;
11: **end if**
12: **if** os quatro átomos satisfazem as características acima **then**
13: Atualize as coordenadas dos átomos b_1, b_2, b_3 e a, isto determinará duas posições para a;
14: Coloque os átomos novamente na estrutura original;
15: Encontre a posição de a que minimiza o erro entre $d(b_4, a)$ e a distância real;
16: Remova a de L.
17: **end if**
18: **end for**
19: **end while**

Embora o algoritmo 4.4, assim como o GB e o UGB, tenha um método bem definido para a seleção dos quatro primeiros átomos da base métrica, não há garantias de que a escolha de uma base arbitrária para a inicialização irá resultar na determinação completa da estrutura de uma proteína. Nesse caso, pode-se começar novamente, selecionando-se um conjunto diferente de átomos para a inicialização.

O teorema 4.4 analisa o limite superior de complexidade computacional, não importando se é a determinação da estrutura de uma proteína ou de um grafo, utilizando o algoritmo 4.4.

Teorema 4.4 (visto em [37])**.** *Dado um conjunto de distâncias esparsas de uma proteína, então custa, no máximo,* $\mathcal{O}(n^3 d_{max}^6)$ *para se determinar se a estrutura da proteína pode ser resolvida usando o algoritmo 4.4.*

Demonstração. Na estrutura de uma proteína, há no máximo $\mathcal{O}(nd_{max}^3)$ grupos de quatro átomos que não são coplanares e têm todas as distâncias entre si. Qualquer

um desses grupos pode ser considerado como base inicial. Porém, considerando o pior caso, pode acontecer de todas essas possíveis bases falharem até a última ou todas falharem. Para cada tentativa, tem-se um tempo computacional de $\mathcal{O}(n^2 d_{max}^3)$. Assim, para se saber se é possível resolver uma estrutura, custa em torno de

$$\mathcal{O}(nd_{max}^3)\mathcal{O}(n^2 d_{max}^3) = \mathcal{O}(n^3 d_{max}^6).$$

□

Daqui por diante, será visto um outro tipo de abordagem para resolução do MDGP. Todos os algoritmos discutidos neste capítulo serão considerados na seção 5.4.

Capítulo 5

Abordagem discreta para o Problema Geométrico da Distância Molecular (DMDGP)

Este capítulo se dedica a uma abordagem discreta para o MDGP, conhecida como DMDGP (*discretizable molecular distance geometry problem*). Segundo [40], sob algumas condições, geralmente satisfeitas em moléculas de proteínas, pode-se obter uma reformulação combinatorial do MDGP de tal forma que o espaço de busca a ser considerado passa de contínuo a discreto. Além disso, trata de um algoritmo capaz de resolver o DMDGP de forma eficiente, o *branch-and-prune* (BP), proposto em [1, 2, 22, 23, 40]. Assim, a seção 5.1 apresenta a formulação do DMDGP. As seções 5.2 e 5.3 dão detalhes sobre o algoritmo BP. Finalmente, a seção 5.4 mostra uma comparação entre o algoritmo BP e o RUGB, este último visto no capítulo anterior.

5.1 Formulação

Esta seção mostrará uma formulação discreta para o MDGP, cuja definição pode ser vista a seguir:

Definição 5.1 (Discretizable Molecular Distance Geometry Problem – DMDGP)**.** *Dado um grafo não direcionado $G = (V, E, d)$, tal que existe uma ordem $v_1, v_2, \ldots, v_n$ em V que satisfaça os seguintes requisitos:*

1. *E contém todos os cliques*[1] *num grupo de quatro vértices consecutivos, isto é,*

$$\forall k \in \{4, \ldots, n\} \text{ e } \forall i, j \in \{k-3, k-2, k-1, k\}, \text{ com } i \neq j, \text{ então } \{i, j\} \in E \text{ ;}$$

[1] *Cliques* de um grafo não direcionado é um subconjunto de vértices tais que cada dois vértices são conectados por uma aresta.

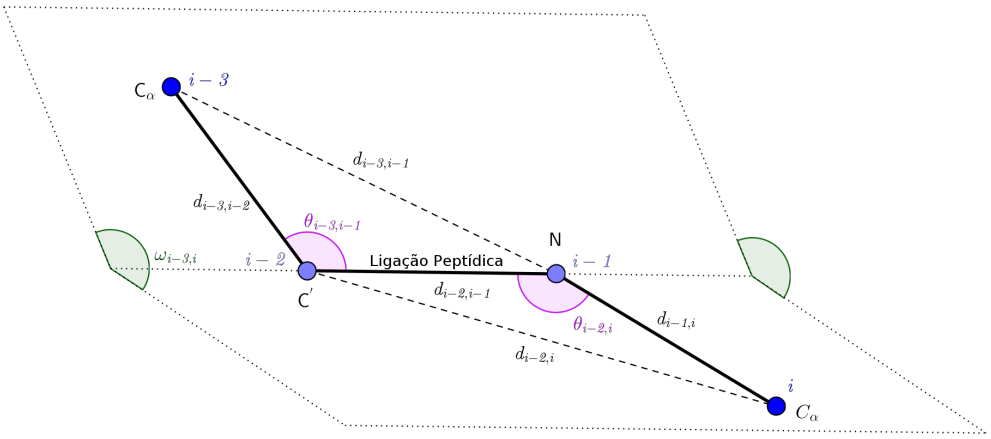


Figura 5.1: Ângulos e distâncias de ligação e ângulo de torção na unidade peptídica.

2. *Vale a desigualdade triangular estrita, isto é,*

$$\forall i \in \{2, \ldots, n-1\},\ d_{i-1,1+1} < d_{i-1,i} + d_{i,i+1},$$

o problema em questão é encontrar $x : V \to \mathbb{R}^3$ *tal que* $||x_i - x_j|| = d_{ij}$*, para cada* $\{i, j\} \in E$*.*

Neste contexto, as distâncias d_{ij} são chamadas *distâncias de ligação* e os ângulos $\theta_{i-2,i}$ entre os átomos dos vértices v_{i-2}, v_{i-1}, v_i são chamados *ângulos de ligação*. Uma visão mais detalhada, com definições e propriedades, das distâncias de ligação e dos ângulos de ligação será dada na seção 5.3.

Além das distâncias de ligação d_{ij}, $i = 2, \ldots, n$, e ângulos de ligação $\theta_{i-2,i}$, $i = 3, \ldots, n$, há também o *ângulo de torção* $\omega_{i-3,1}$, $i = 4, \ldots, n$, entre as duas normais aos planos definidos pelos átomos $i-3, i-2, i-1$ e $i-2, i-1, i$ (veja figura 5.1), já mencionado no capítulo 2 e detalhado matematicamente no seção 5.3.

Sabendo-se *a priori*[2] esses ângulos e essas distâncias, é possível fixar posições para os três primeiros átomos e, em seguida, calcula-se a posição do quarto átomo através do ângulo de torção $\omega_{1,4}$ e depois, através de $\omega_{2,5}$, calcula-se a posição do quinto átomo e assim por diante.

A intuição geométrica por trás desta formulação discreta do MDGP é que o i-ésimo átomo está na interseção de três esferas centradas nos átomos $i-3$, $i-2$ e $i-1$, com raios iguais a $d_{i-3,i}$, $d_{i-2,i}$ e $d_{i-1,i}$, respectivamente. Da definição 5.1 e do fato de que dois átomos não podem assumir a mesma posição no espaço, tem-se que a interseção das três esferas define no máximo dois pontos (veja figuras 5.2 e 5.3), o que permite montar uma árvore com 2^{n-3} conformações possíveis (embora que algumas destas conformações poderão ser descartadas, como será visto mais tarde).

[2] A seção 5.3 mostrará que estes ângulos podem ser calculados.

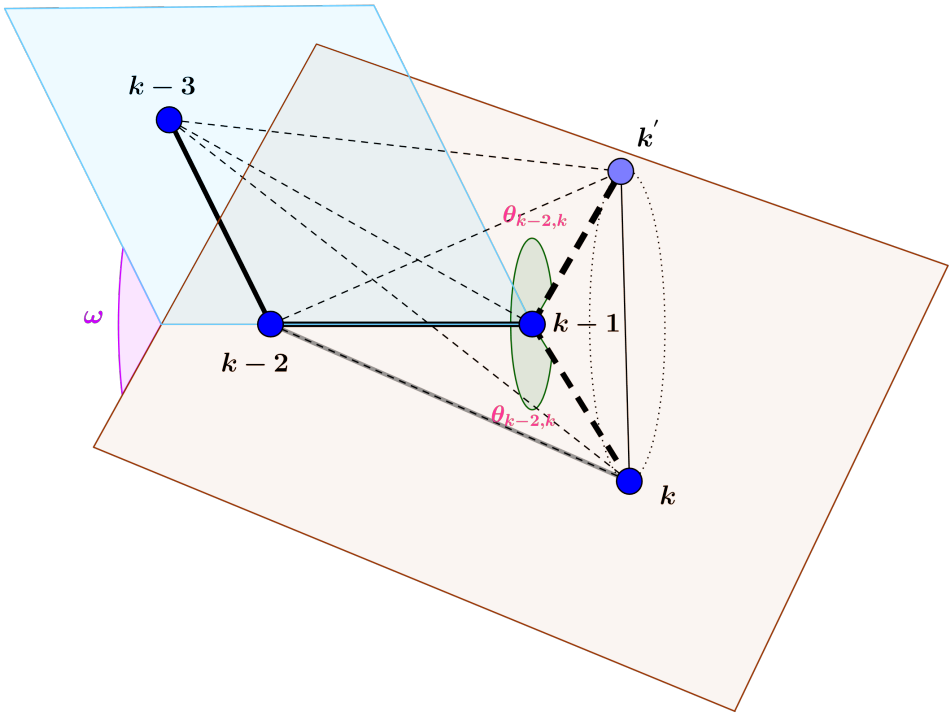


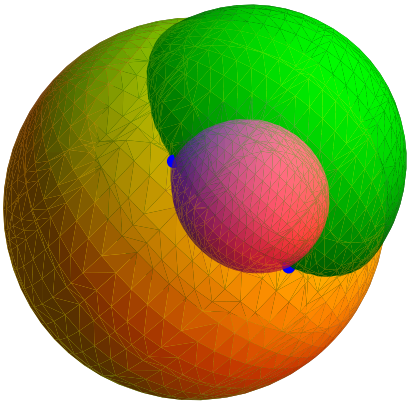
Figura 5.2: Discretização do MDGP. O quarto átomo só poderia estar nas duas posições representadas por k e k' para que seja viável com respeito a distância $d_{k-3,k}$. Note que o círculo pontilhado desenhado acima é dado pelo ângulo de torção $\omega_{k-3,k}$.

Assim, dados, então, as distâncias de ligação $d_{12}, \ldots, d_{n-1,n}$, os ângulos de ligação $\theta_{13}, \ldots, \theta_{n-2,n}$ e os ângulos de torção $\omega_{14}, \ldots, \omega_{n-3,n}$ de uma molécula com n átomos, as coordenadas cartesianas de cada átomo podem ser obtidas por

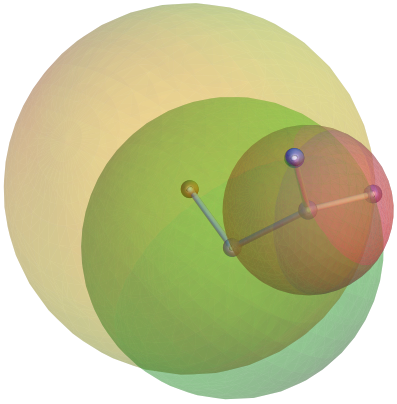
$$\begin{bmatrix} x_{i1} \\ x_{i2} \\ x_{i3} \\ 1 \end{bmatrix} = B_1 B_2 B_3 \ldots B_i \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \ i \in \{4, \ldots n\}, \tag{5.1}$$

onde B_i, $i = 4, \ldots, n$, são matrizes definidas por

$$B_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$



(a) A interseção entre as três esferas resulta em dois pontos possíveis (pontos azuis).



(b) Vista das distâncias de ligação. Note que os três primeiros átomos estão posicionados nos centros das esferas. Há duas possibilidades para o quarto átomo.

Figura 5.3: Encontro das três esferas da formulação discreta.

$$B_2 = \begin{bmatrix} -1 & 0 & 0 & -d_{12} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, B_3 = \begin{bmatrix} -\cos(\theta_{13}) & -\operatorname{sen}(\theta_{13}) & 0 & -d_{23}\cos(\theta_{13}) \\ \operatorname{sen}(\theta_{13}) & -\cos(\theta_{13}) & 0 & d_{23}\operatorname{sen}(\theta_{13}) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ e} \tag{5.2}$$

$$B_i = \begin{bmatrix} -\cos(\theta_{i-2,i}) & -\operatorname{sen}(\theta_{i-2,i}) & 0 & -d_{i-1,i}\cos(\theta_{i-2,i}) \\ \operatorname{sen}(\theta_{i-2,i})\cos(\omega_{i-3,i}) & -\cos(\theta_{i-2,i})\cos(\omega_{i-3,i}) & -\operatorname{sen}(\omega_{i-3,i}) & d_{i-1,i}\operatorname{sen}(\theta_{i-2,i})\cos(\omega_{i-3,i}) \\ \operatorname{sen}(\theta_{i-2,i})\operatorname{sen}(\omega_{i-3,i}) & -\cos(\theta_{i-2,i})\operatorname{sen}(\omega_{i-3,i}) & \cos(\omega_{i-3,i}) & d_{i-1,i}\operatorname{sen}(\theta_{i-2,i})\operatorname{sen}(\omega_{i-3,i}) \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{5.3}$$

fixando-se $x_1 = (0,0,0)^T$. O uso da quarta coordenada igual a 1 na equação 5.1 será abordado na seção 5.3. Também de acordo com esta seção, x_1 pode ser fixado como $x_1 = (0,0,0,1)^T$.

Desta forma, as coordenadas cartesianas de todos os átomos da cadeia principal da molécula são completamente determinadas através do $\cos(\theta_{i-2,i})$, $\operatorname{sen}(\theta_{i-2,i})$, $\cos(\omega_{i-3,i})$ e do $\operatorname{sen}(\omega_{i-3,i})$, onde este último admite dois valores possíveis, isto é,

$$\operatorname{sen}(\omega_{i-3,i}) = \pm\sqrt{1-\cos^2(\omega_{i-3,i})},$$

o que possibilita as duas posições, vistas nas figuras 5.2 e 5.3. Esta questão do sinal do ângulo de torção será vista na proposição 5.1.

É importante, antes de se encerrar esta seção, deixar registrados três resultados cujas demonstrações podem ser encontradas em [40]. Estes resultados serão utilizados no algoritmo branch-and-prune constantemente. São eles: lemas 5.1 e 5.2 e o teorema 5.1, enunciados a seguir.

Lema 5.1. *Seja a matriz $\mathcal{P}_i$ tal que*

$$\mathcal{P}_i = B_4 \dots B_i,$$

para $i \in \{4, \dots n\}$, onde seus elementos são dados por

$$\mathcal{P}_i = \begin{bmatrix} \rho_{11}^i & \rho_{12}^i & \rho_{13}^i & \rho_{14}^i \\ \rho_{21}^i & \rho_{22}^i & \rho_{23}^i & \rho_{24}^i \\ \rho_{31}^i & \rho_{32}^i & \rho_{33}^i & \rho_{34}^i \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Sejam as matrizes $B_i^{'}$ obtidas ao se inverter os sinais de $\operatorname{sen}(\omega_{i-3,i})$, para $i \in \{4, \dots n\}$. Definindo-se

$$\mathcal{P}_i^{'} = B_4^{'} \dots B_i^{'},$$

tem-se que

$$\mathcal{P}_i' = \begin{bmatrix} \rho_{11}^i & \rho_{12}^i & -\rho_{13}^i & \rho_{14}^i \\ \rho_{21}^i & \rho_{22}^i & -\rho_{23}^i & \rho_{24}^i \\ -\rho_{31}^i & -\rho_{32}^i & \rho_{33}^i & -\rho_{34}^i \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

para $i \in \{4, \dots n\}$.

Lema 5.2. *Seja* $x_1, \dots, x_n \in \mathbb{R}^3$ *as coordenadas cartesianas definidas pelos ângulos de torção* $\omega_{14}, \dots, \omega_{n-3,n}$. *Se os sinais dos* $\operatorname{sen}(\omega_{i-3,i})$ *forem invertidos em toda a* B_i, *para* $i \in \{4, \dots n\}$, *então, as novas coordenadas cartesianas* $x_1^{'}, \dots, x_n^{'} \in \mathbb{R}^3$ *são dadas por*

$$\begin{bmatrix} x_{i1}^{'} \\ x_{i2}^{'} \\ x_{i3}^{'} \end{bmatrix} = \begin{bmatrix} x_{i1} \\ x_{i2} \\ -x_{i3} \end{bmatrix}$$

para $i \in \{1, \dots n\}$.

Teorema 5.1. *Seja* $x : V \to \mathbb{R}^3$ *uma função que resolva uma instância*[3] *do DMDGP, definida pelos seus ângulos de torção* $\omega_{14}, \dots, \omega_{n-3,n}$. *Se os sinais dos* $\operatorname{sen}(\omega_{i-3,i})$ *forem invertidos em toda a* B_i, *para* $i \in \{4, \dots n\}$, *obtém-se uma nova função* $x' : V \to \mathbb{R}^3$ *que resolve a mesma instância.*

Os lemas 5.1 e 5.2 e o teorema 5.1 facilitam o cálculo das coordenadas dos átomos, pois evitam operações para se obter as duas posições possíveis, vistas nas figuras 5.2 e 5.3. Por eles, basta se calcular i para se ter i', ou vice-versa.

A próxima seção descreverá o algoritmo branch-and-prune, que é uma implementação da formulação discreta do MDGP. Lembrando-se que questões como cálculo dos ângulos, obtenção das matrizes B_i e coordenadas homogêneas serão abordadas no seção 5.3.

5.2 Branch-and-Prune

Nesta seção, será apresentado um algoritmo, conhecido como *branch-and-prune* (BP), em sua formulação clássica [22, 23, 40], que é utilizado no cálculo de soluções para o DMDGP. A cada passo do BP, encontram-se duas coordenadas possíveis para o i-ésimo átomo, x_i e $x_i^{'}$. No entanto, se uma dessas posições (ou as duas) forem inviáveis, isto é, não atendam ao *critério de poda* ou *pruning test*, a solução

[3]Uma *instância*, até então, é um problema obtido a partir de uma molécula real, geralmente um arquivo PDB, onde se tem todos os pontos já determinados. Desse arquivo, se calculam as distâncias entre os átomos de toda a molécula ou de uma parte dela (da cadeia principal, por exemplo). Esses dados servem como entrada para os algoritmos aqui estudados. Na seção 6.3, o conceito de instância será aprofundado.

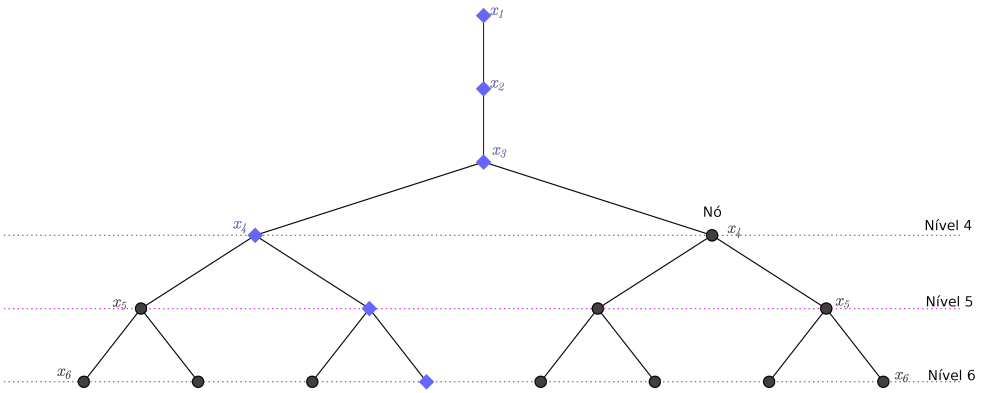


Figura 5.4: Exemplo de árvore de busca para uma molécula de 6 átomos ($n = 6$). Os losangos em azul representam uma solução completa.

em questão é descartada. O *pruning test* utilizado neste trabalho pelo BP clássico é o DDF (*direct distance feasibility*)[41], definido como segue.

Definição 5.2 (Direct Distance Feasibility – DDF)**.** *Sejam x_i a posição do átomo i e x_j a posição do átomo j vizinho a i, para que posição x_i seja considerada viável, a condição*

$$|||x_i - x_j|| - d_{ij}| < \varepsilon, \ \forall j < i,$$

onde $\{j, i\} \in E$ e $\varepsilon > 0$, deve ser satisfeita. Caso contrário, a posição x_i e as posições calculadas a partir de x_i são descartadas. Esta condição representa o pruning test DDF (direct distance feasibility).

À medida que os átomos de uma proteína são determinados, gera-se uma *árvore de busca*, cujos ramos que representam soluções inviáveis são descartados. Cada posição ocupada por um átomo i nesta árvore, chama-se *nó*, e ao conjunto de posições possíveis ocupadas por este átomo chama-se *nível i*. Desta forma, seja T uma árvore de busca, inicia-se os três primeiros nós de T sem nenhuma ramificação, pois estes átomos são fixados em posições viáveis, x_1, x_2 e x_3, como será visto na seção 5.3. O quarto átomo também pode ser fixado, de acordo com o teorema 5.1. Na figura 5.4, é possível observar um exemplo de uma árvore onde nenhuma ramificação foi descartada.

De início, em cada nó da árvore de busca são armazenadas as seguintes informações:

- a posição $x_i \in \mathbb{R}^3$ do i-ésimo átomo;
- a matriz $\mathcal{P}_i$;
- um ponteiro para o nó pai $P(i)$;
- ponteiros para os nós filhos $L(i)$ e $R(i)$.

Assim, sejam $y = (0, 0, 0, 1)^T, \varepsilon > 0$ e v um nó do nível $i - 1$, tem-se o branch-and-prune descrito pelos algoritmos 5.1 e 5.2.

Seja $|E|$ o número de arestas do grafo G, cada solução encontrada pelo BP pode ser avaliada quanto a sua precisão utilizando-se o *LDE*, isto é, *largest distance error*, definido por

$$LDE = \frac{1}{|E|} \sum_{\{i,j\} \in E} \frac{|||x_i - x_j|| - d_{ij}|}{d_{ij}}, \tag{5.4}$$

onde a solução encontrada pode ser comparada com a estrutura original (geralmente obtida do PDB). Quanto menor for este erro, mais próxima da estrutura original é a solução encontrada. Para tal, também pode ser utilizada a RMSD, vista no capítulo 3.

Algoritmo 5.1 Algoritmo Branch-and-Prune detalhado

```
1: BranchAndPrune(T,v,i)
2: if (i < n) then
3:    {Cálculo das posições possíveis para o átomo i.}
4:    Calcule B_i e B_i' e leia P_{i-1} de P(v);
5:    P_i ← P_{i-1}B_i e P_i' ← P_{i-1}'B_i';
6:    x_i ← P_i y e x_i' ← P_i' y;
7:    {Pruning Tests}
8:    if (x_i é viável) then
9:       Crie um nó z em T;
10:      Armazene P_i, x_i em z;
11:      P(z) ← v e L(v) ← z;
12:      BranchAndPrune(T,z,i + 1);
13:   else
14:      L(v) ← PRUNED;
15:   end if
16:   if (x_i' é viável) then
17:      Crie um nó z' em T;
18:      Armazene P_i', x_i' em z';
19:      P(z') ← v e R(v) ← z';
20:      BranchAndPrune(T,z',i + 1);
21:   else
22:      R(v) ← PRUNED;
23:   end if
24: else
25:   Solução encontrada!
26: end if
```

É possível ainda escrever o algoritmo 5.1 de forma mais simples, como mostra o algoritmo 5.2.

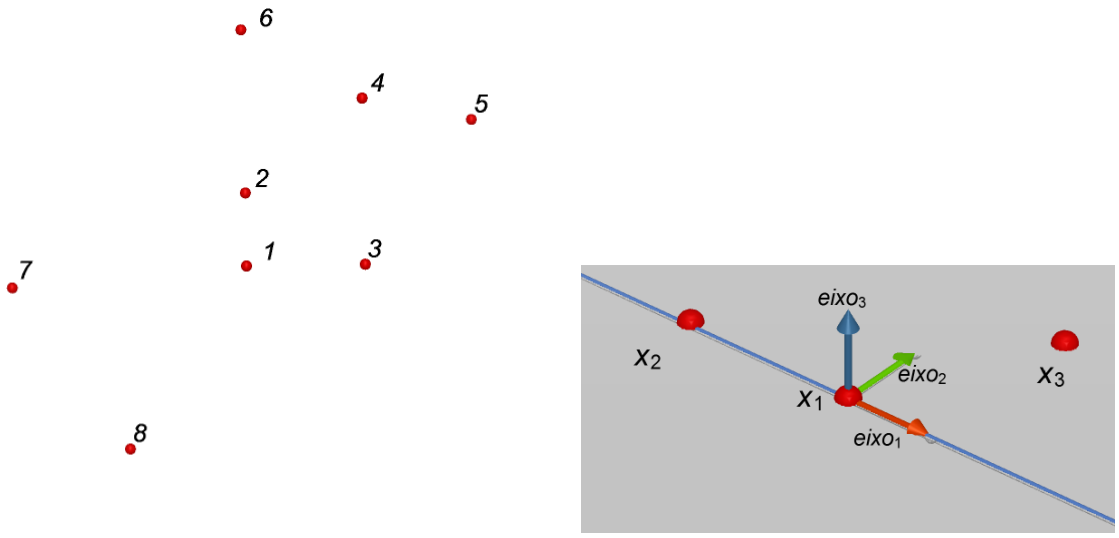
Algoritmo 5.2 Algoritmo Branch-and-Prune (resumido)

1: BranchAndPrune(i, x_{i-1}, T)
2: **if** ($i < n$) **then**
3: Calcule B_i e $B_i^{'}$;
4: Calcule $\prod_{j=1}^{i} B_j$ e $\prod_{j=1}^{i} B_j^{'}$, $1 \leq j \leq i$;
5: $x_i \leftarrow (\prod_{j=1}^{i} B_j)e_4$ e $x_i^{'} \leftarrow (\prod_{j=1}^{i} B_j^{'})e_4$;
6: {Pruning Tests}
7: **if** (x_i é viável) **then**
8: Insira x_i em um novo nó a esquerda em T;
9: BranchAndPrune($i+1$, x_i, T);
10: **else**
11: Ramo relativo a x_i é descartado em T;
12: **end if**
13: **if** ($x_i^{'}$ é viável) **then**
14: Insira x_i em um novo nó a direita em T;
15: BranchAndPrune($i+1$, $x_{i-1}^{'}$, T);
16: **else**
17: Ramo relativo a $x_i^{'}$ é descartado em T;
18: **end if**
19: **end if**

Nesta seção foi vista uma breve explicação sobre o BP, porém há alguns aspectos que voltarão a ser discutidos neste trabalho. Na próxima seção, volta-se a abordar o DMDGP e o BP de forma mais detalhada, apontando-se suas características matemáticas mais importantes.

5.3 Descrição Matemática do Algoritmo Branch-and-Prune

Nesta seção serão apresentadas três formulações complementares para a representação espacial de átomos via algoritmo branch-and-prune. Na subseção 5.3.1, faz-se o cálculo de grandezas relevantes a partir das coordenadas cartesianas de um conjunto de pontos em $\mathbb{R}^3$. Na subseção 5.3.2, usando as informações sobre as distâncias de ligação, dos ângulos de ligação e de torção, discute-se a fundamentação matemática de um algoritmo para os cálculos das coordenadas cartesianas. As informações sobre proteínas, provenientes de dados de Ressonância Magnética Nuclear, são fornecidas através das distâncias entre seus átomos. Assim, na subseção 5.3.3, serão calculadas algumas grandezas relativas aos ângulos entre os átomos.



(a) Pontos ordenados.

(b) Eixos ortogonais e três pontos iniciais.

Figura 5.5: Estrutura *Vitoria8*.

5.3.1 Pontos em Coordenadas Cartesianas

Dados n pontos distintos em $\mathbb{R}^3$, faz-se uma enumeração de 1 a n, ver Figura 5.5a. A seguir, eixos de referência ortogonais são determinados atendendo às seguintes condições:

1. o ponto 1 fica na origem,
2. o eixo relativo à primeira ordenada é definido pelos pontos 1 e 2 e tem sentido de 2 para 1,
3. os pontos 1, 2 e 3 estão no plano definido pelos eixos relativos às primeiras e segundas ordenadas,
4. o eixo relativo à segunda ordenada tem o sentido que garanta valor positivo ou nulo para a segunda ordenada do ponto 3.

A orientação positiva é definida pela regra da mão direita. Os pontos serão representados por x_i com $i = 1 \ldots n$ e uma representação dos pontos x_1, x_2 e x_3, com eixos ortogonais que atendam às condições acima, como mostra a Figura 5.5b. Cada uma das três ordenadas de cada ponto será representada por x_{ij}, com $i = 1 \ldots n$ e $j = 1 \ldots 3$, ver Figura 5.6. Outra hipótese necessária à formulação do problema é que três pontos consecutivos não são colineares.

No restante desta subseção, serão introduzidos objetos que podem ser bem definidos a partir desses pontos, dessa enumeração e desses eixos.

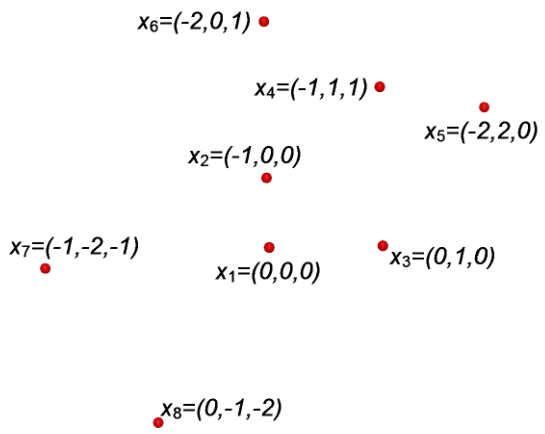


Figura 5.6: Pontos com coordenadas cartesianas em $\mathbb{R}^3$.

Distâncias de Ligação

A distância euclidiana entre dois pontos sucessivos é denominada *distância de ligação*, $d_{i-1,i}$ com $i = 2 \dots n$. A fórmula da distância de ligação é dada por:

$$d_{i-1,i} = \left(\sum_{j=1}^{3} (x_{ij} - x_{(i-1)j})^2 \right)^{\frac{1}{2}} = \|x_i - x_{i-1}\|_2. \tag{5.5}$$

Nessa parte, definem-se distâncias apenas entre pontos sucessivos; mais adiante, serão utilizadas distâncias entre todos os n pontos, mas não serão denominadas distâncias de ligação.

Vetores de Ligação

A partir dos n pontos, tem-se $(n-1)$ *vetores de ligação*, $v_{i-1,i}$ com $i = 2 \dots n$, onde o sentido desses vetores será de $i-1$ para i. Suas ordenadas são definidas pela diferença entre dois pontos consecutivos, ver Figura 5.7:

$$v_{i-1,i} = x_i - x_{i-1}. \tag{5.6}$$

Ou seja, $d_{i-1,i} = \|v_{i-1,i}\|_2$.

Ângulos de Ligação

Seja o ângulo determinado por três pontos sucessivos, x_{i-2}, x_{i-1} e x_i, com $i = 3 \dots n$, onde as semi-retas que definem o ângulo tem origem no ponto x_{i-1}. Dessa forma, obtêm-se $(n-2)$ *ângulos de ligação*, $\theta_{i-2,i}$ com $i = 3 \dots n$. Utilizando-se um radiano como a unidade de medida de ângulos, esses ângulos estarão no intervalo fechado

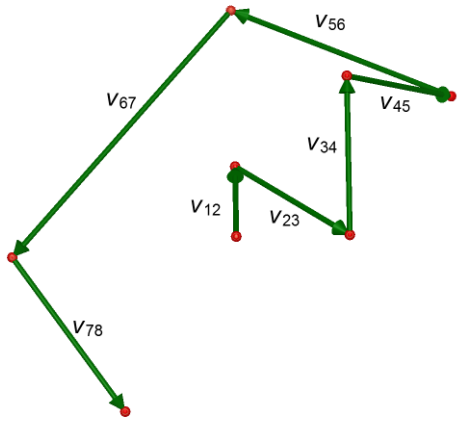


Figura 5.7: Vetores definidos por dois pontos sucessivos, $v_{i-1,i}$.

$[0, \pi]$; se a unidade for um grau, então os ângulos estarão entre $[0°, 180°]$. A fórmula a seguir mostra o cálculo dos ângulos de ligação:

$$\theta_{i-2,i} = \pi - \arccos\left[\left[\frac{v_{i-2,i-1}}{\|v_{i-2,i-1}\|_2}\right]^T \left[\frac{v_{i-1,i}}{\|v_{i-1,i}\|_2}\right]\right], \tag{5.7}$$

onde T representa o operador de transposição matricial e $v^T u$ é o produto interno euclidiano entre vetores v e u, com ordenadas reais (ver Figura 5.8). Caso seja usado grau como unidade de ângulo, então π deve ser substituído por 180° na fórmula (5.7).

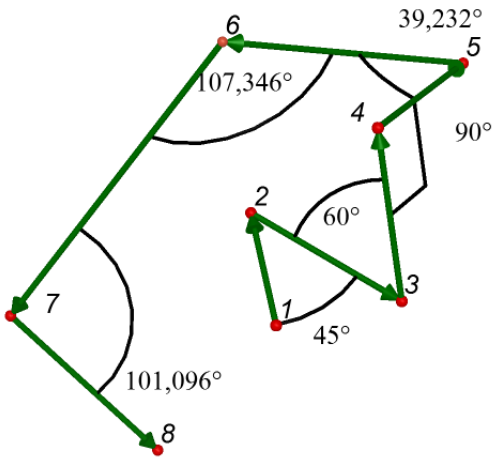


Figura 5.8: Ângulos definidos por três pontos sucessivos, $\theta_{i-2,i}$ e medidos em graus.

Note que, através do produto interno entre os vetores $v_{i-2,i-1}/||v_{i-2,i-1}||_2$ e

$v_{i-1,i}/||v_{i-1,i}||_2$, obtém-se o ângulo λ, dado por

$$\lambda = \arccos\left[\left[\frac{v_{i-2,i-1}}{\|v_{i-2,i-1}\|_2}\right]^T \left[\frac{v_{i-1,i}}{\|v_{i-1,i}\|_2}\right]\right],$$

e que $\lambda + \theta_{i-2,i} = \pi$ ou, se a unidade escolhida for um grau, $\lambda + \theta_{i-2,i} = 180°$, como mostra a Figura 5.9. Sendo assim, tem-se $\theta_{i-2,i} = \pi - \lambda$, como mostra a Equação (5.7).

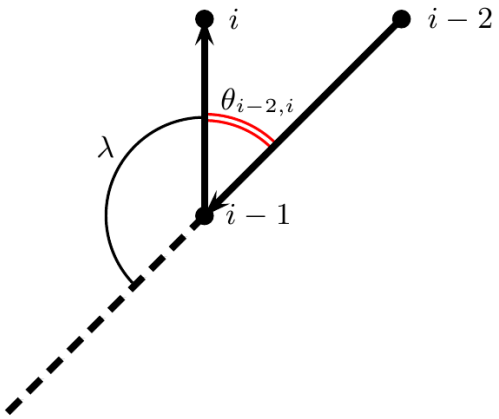


Figura 5.9: Ângulos definidos por três pontos sucessivos, $\theta_{i-2,i}$, e seu complementar λ, que é o ângulo entre os vetores de ligação em questão, $v_{i-2,i-1}$ e $v_{i-1,i}$.

Vetores Normais

Os vetores unitários ortogonais aos planos definidos por três pontos sucessivos e que atendam à orientação dos eixos ortogonais escolhidos[4] são chamados *vetores normais*, $\eta_{i-2,i}$ com $i = 3 \ldots n$. Para o cálculo, usa-se a fórmula

$$\eta_{i-2,i} = \frac{v_{i-2,i-1} \times v_{i-1,i}}{\|v_{i-2,i-1} \times v_{i-1,i}\|_2}, \tag{5.8}$$

onde $\times$ representa o produto vetorial entre dois vetores, ver Figura 5.10. Observa-se que a hipótese de não colinearidade de três pontos sucessivos garante a existência de todas as normais.

Ângulos de Torção

Ângulos de torção são ângulos orientados entre dois vetores normais sucessivos. Utilizando-se os vetores normais definidos anteriormente, haverá $(n-3)$ ângulos

[4]Neste caso, os eixos ortogonais escolhidos atendem aos itens de 1 a 4 da página 57, onde a orientação positiva para o produto vetorial é dada pela regra da mão direita.

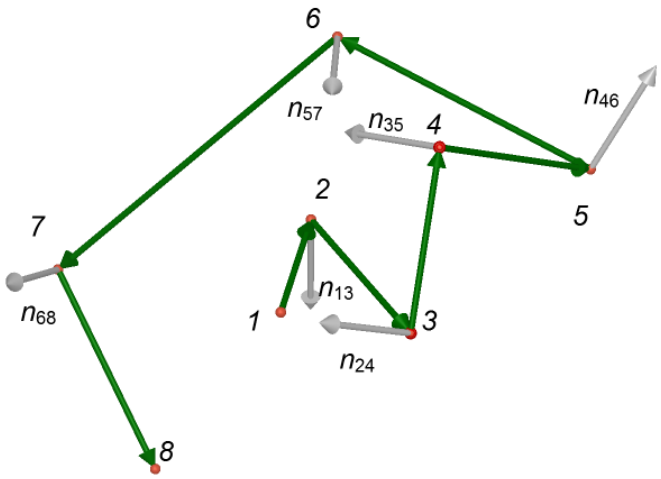


Figura 5.10: Normais, $\eta_{i-2,i}$, aos planos definidos por três pontos sucessivos e utilizadas para o cálculo dos ângulos de torção.

de torção, $\omega_{i-3,i}$ com $i = 4 \ldots n$. Como os ângulos de torção são ângulos orientados, seus valores se encontram no intervalo $-\pi < \omega_{i-3,i} \leq \pi$, ou $-180° < \omega_{i-3,i} \leq 180°$ (ver Figura 5.11), e são calculados pela fórmula:

$$\omega_{i-3,i} = \text{sign}\left((\eta_{i-3,i-1})^T v_{i-1,i}\right) \arccos\left((\eta_{i-3,i-1})^T \eta_{i-2,i}\right), \tag{5.9}$$

onde sign é uma função real com resultados inteiros, definida por

$$\text{sign}(x) = \begin{cases} -1, & \text{se } x < 0, \\ 1, & \text{se } x \geq 0. \end{cases}$$

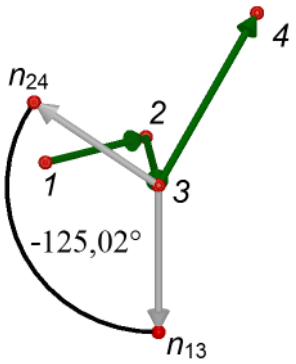
em eixo

Toda a rotação em $\mathbb{R}^3$ pode ser definida por um ângulo e por um eixo de rotação. No caso do ângulo de torção, que é o ângulo entre duas normais $\eta_{i-3,i-1}$ e $\eta_{i-2,i}$, o eixo de rotação é o vetor $v_{i-2,i-1}$, pois sendo

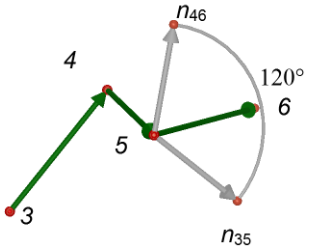
$$\eta_{i-3,i-1} = \frac{v_{i-3,i-2} \times v_{i-2,i-1}}{||v_{i-3,i-2} \times v_{i-2,i-1}||} \quad \text{e} \quad \eta_{i-2,i} = \frac{v_{i-2,i-1} \times v_{i-1,i}}{||v_{i-2,i-1} \times v_{i-1,i}||},$$

temos que $v_{i-2,i-1}$ é perpendicular ao plano gerado por estes vetores unitários.

Para se determinar o ângulo de torção entre os vetores unitários $\eta_{i-3,i-1}$ e $\eta_{i-2,i}$, a Equação (5.9) se utiliza de dois fatores: o segundo fator, $\arccos\left((\eta_{i-3,i-1})^T \eta_{i-2,i}\right)$, é a forma usual para se determinar ângulos entre vetores

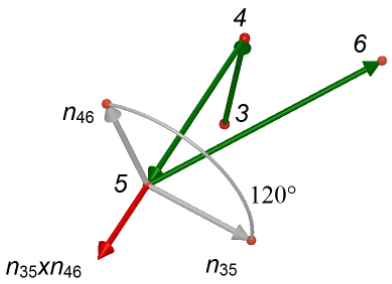


(a) Detalhe do ângulo de torção $\omega_{1,4}$.

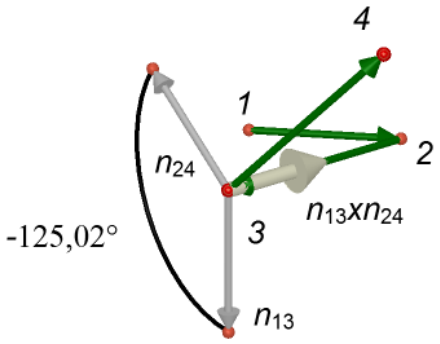


(b) Detalhe do ângulo de torção $\omega_{3,6}$.

Figura 5.11: Ângulos de torção.



(a) Ângulo positivo pois $n_{35} \times n_{46}$ e v_{45} têm mesmo sentido.



(b) Ângulo negativo pois $n_{13} \times n_{24}$ e v_{23} têm sentido oposto.

Figura 5.12: Ângulos de torção definidos pela regra da mão direita entre vetores normais e vetor definido por dois pontos.

unitários. Já o primeiro fator está relacionado com o sentido de rotação em torno do eixo dado por $v_{i-2,i-1}$. Assim, respeitando a regra da mão direita, o ângulo de torção será positivo quando o vetor $v_{i-2,i-1}$ e o produto vetorial das normais envolvidas, $\eta_{i-3,i-1} \times \eta_{i-2,i}$, tiverem o mesmo sentido (ver Figura 5.12a) e negativo quando os sentidos forem contrários (ver Figura 5.12b). Daí a necessidade do primeiro fator, que pode ser justificado pela proposição a seguir.

Proposição 5.1 (Cálculo do Sinal do Ângulo de Torção)**.** *O sinal do ângulo de torção entre as normais* $\eta_{i-3,i-1}$ *e* $\eta_{i-2,i}$ *é dada pela fórmula* $\operatorname{sign}\left((\eta_{i-3,i-1})^T v_{i-1,i}\right)$.

Demonstração. O objetivo é estudar o sinal do produto interno

$$(v_{i-2,i-1})^T(\eta_{i-3,i-1} \times \eta_{i-2,i}). \tag{5.10}$$

Desenvolvendo-se o produto vetorial das normais:

$$\eta_{i-3,i-1} \times \eta_{i-2,i} = (v_{i-3,i-2} \times v_{i-2,i-1}) \times (v_{i-2,i-1} \times v_{i-1,i}).$$

Utilizando-se a fórmula de Grassmann (ver proposição C.3) para um produto vetorial triplo[5],tem-se que:

$$\begin{aligned}(v_{i-3,i-2} \times v_{i-2,i-1}) \times (v_{i-2,i-1} \times v_{i-1,i}) = \\ = \left(v_{i-2,i-1}(v_{i-1,i})^T - v_{i-1,i}(v_{i-2,i-1})^T\right)(v_{i-3,i-2} \times v_{i-2,i-1}) = \\ = v_{i-2,i-1}(v_{i-1,i})^T(v_{i-3,i-2} \times v_{i-2,i-1}).\end{aligned}$$

Fazendo-se o produto interno em (5.10)

$$\begin{aligned}(v_{i-2,i-1})^T v_{i-2,i-1}(v_{i-1,i})^T(v_{i-3,i-2} \times v_{i-2,i-1}) = \\ = \|v_{i-2,i-1})\|_2^2(v_{i-1,i})^T(v_{i-3,i-2} \times v_{i-2,i-1}) = \\ = \|v_{i-2,i-1})\|_2^2(v_{i-1,i})^T\eta_{i-3,i-1}.\end{aligned}$$

Logo, o sinal do ângulo de torção será definido pelo sinal de $(v_{i-1,i})^T\eta_{i-3,i-1} = (\eta_{i-3,i-1})^T v_{i-1,i}$. □

Note que, na proposição 5.1, inicia-se o estudo do sinal do ângulo de torção através do produto interno expresso em (5.10). Tem-se este fato, pois, como $v_{i-2,i-1}$ é perpendicular ao plano gerado por $\eta_{i-3,i-1}$ e $\eta_{i-2,i}$, pode-se concluir que $v_{i-2,i-1}$ é paralelo ao vetor $\eta_{i-3,i-1} \times \eta_{i-2,i}$, o que significa que o ângulo entre estes dois vetores só poderá ser:

- π, que implica em

  $$(v_{i-2,i-1})^T(\eta_{i-3,i-1} \times \eta_{i-2,i}) = \|v_{i-2,i-1}\|\|\eta_{i-3,i-1} \times \eta_{i-2,i}\| \cos(\pi) < 0$$

  e sentidos opostos, ou

- 0, que implica em

  $$(v_{i-2,i-1})^T(\eta_{i-3,i-1} \times \eta_{i-2,i}) = \|v_{i-2,i-1}\|\|\eta_{i-3,i-1} \times \eta_{i-2,i}\| \cos(0) > 0$$

  e mesmo sentido.

[5]Sejam os vetores u, v e r, então a fórmula de Grassmann para um produto vetorial triplo diz que $u \times (v \times r) = (vr^T - rv^T)u$, onde as multiplicações são multiplicações matriciais usuais.

5.3.2 Distâncias e Ângulos

Nesta seção, usando as informações sobre as distâncias de ligação, ângulos de ligação e de torção, discute-se a fundamentação matemática de um algoritmo para os cálculos das coordenadas cartesianas de cada ponto. Para tal, enuncia-se o seguinte problema:

Problema 5.1. *Dados n pontos diferentes e ordenados em $\mathbb{R}^3$, suas distâncias de ligação, seus ângulos de ligação e seus ângulos de torção, calcular as coordenadas cartesianas do ponto x_i tal que os eixos de referência ortogonais sejam determinados atendendo às seguintes condições:*

1. *o ponto 1 fica na origem,*
2. *o eixo relativo à primeira ordenada é definido pelos pontos 1 e 2 e tem sentido de 2 para 1,*
3. *os pontos 1, 2 e 3 estão no plano definido pelos eixos relativos às primeiras e segundas ordenadas,*
4. *o eixo relativo à segunda ordenada tem o sentido que garanta valor positivo ou nulo para a segunda ordenada do ponto 3.*

A orientação positiva é definida pela regra da mão direita.

No estudo deste problema, é necessário o uso de translações em $\mathbb{R}^3$. Porém, uma translação em $\mathbb{R}^3$ não pode ser representada por uma matriz de ordem 3, ou seja, não é uma transformação linear de $\mathbb{R}^3$ em $\mathbb{R}^3$. No entanto, ao se utilizar o conceito de *coordenadas homegêneas*, [42, p. 14] (ver definição 5.3), uma translação em $\mathbb{R}^3$ pode ser representada por uma transformação linear de $\mathbb{R}^4$ em $\mathbb{R}^4$.

Definição 5.3 (Coordenadas Homogêneas)**.** *Dados $x, y, z, w \in \mathbb{R}$ com $w \neq 0$ então (x, y, z, w) é a representação em* ***coordenadas homogêneas*** *do ponto $\left(\frac{x}{w}, \frac{y}{w}, \frac{z}{w}\right) \in \mathbb{R}^3$ em coordenadas cartesianas.*

Assim, uma translação de um vetor qualquer de $\mathbb{R}^3$ pelo vetor (x_{01}, x_{02}, x_{03}) pode ser representada pela matriz

$$T = \begin{bmatrix} 1 & 0 & 0 & x_{01} \\ 0 & 1 & 0 & x_{02} \\ 0 & 0 & 1 & x_{03} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{5.11}$$

com o resultado sendo dado em coordenadas homogêneas, isto é, $T(x, y, z, 1) = (x + x_{01}, y + x_{02}, z + x_{03}, 1)$. A matriz representada em (5.11) é um exemplo de *matriz homogênea*, cuja definição pode ser vista logo abaixo.

Definição 5.4. *Uma matriz associada a uma transformação linear de vetores representados em coordenadas homogêneas será denominada* ***matriz homogênea****.*

Voltando ao problema 5.1, têm-se, em $\mathbb{R}^3$, distâncias de ligação, ângulos de ligação e de torção para se determinar as coordenadas cartesianas de n pontos. Para se alterar esses dados, recorre-se às definições 5.3 e 5.4, associando-se a cada dado uma matriz homogênea, da seguinte forma:

- Para se alterar a distância entre os pontos $i-1$ e i, dada por $d_{i-1,i} = \|v_{i-1,i}\|$, utiliza-se a matriz

$$T_i = \begin{bmatrix} 1 & 0 & 0 & d_{i-1,i} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{5.12}$$

  que é uma matriz homogênea de translação em $\mathbb{R}^3$ pelo vetor $(d_{i-1,i}, 0, 0)$ (ver apêndice B.1).

- Para se alterar o ângulo de ligação, utiliza-se a matriz

$$R_{\pi-\theta_{i-2,i},\eta_{i-2,i}} = \begin{bmatrix} \cos(\pi - \theta_{i-2,i}) & -\operatorname{sen}(\pi - \theta_{i-2,i}) & 0 & 0 \\ \operatorname{sen}(\pi - \theta_{i-2,i}) & \cos(\pi - \theta_{i-2,i}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$= \begin{bmatrix} -\cos(\theta_{i-2,i}) & -\operatorname{sen}(\theta_{i-2,i}) & 0 & 0 \\ \operatorname{sen}(\theta_{i-2,i}) & -\cos(\theta_{i-2,i}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{5.13}$$

  que é uma matriz homogênea que representa uma rotação em $\mathbb{R}^3$ de um ângulo $\pi - \theta_{i-2,i}$ em torno do eixo definido por $\eta_{i-2,i}$ (ver apêndice B.2).

- Para se alterar o ângulo de torção, utiliza-se a matriz

$$R_{\omega_{i-3,i},v_{i-2,i-1}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\omega_{i-3,i}) & -\operatorname{sen}(\omega_{i-3,i}) & 0 \\ 0 & \operatorname{sen}(\omega_{i-3,i}) & \cos(\omega_{i-3,i}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{5.14}$$

que é uma matriz homogênea que representa uma rotação em $\mathbb{R}^3$ de um ângulo $\omega_{i-3,i}$ em torno do eixo definido por $v_{i-2,i-1}$ (ver apêndice B.3).

A partir das matrizes (5.12), (5.13) e (5.14), pode-se definir a matriz homogênea

$$B_i = R_{\omega_{i-3,i},v_{i-2,i-1}}\ R_{\pi-\theta_{i-2,i},\eta_{i-2,i}}\ T_i \tag{5.15}$$

e, desenvolvendo-se as multiplicações matriciais, tem-se as matrizes

$$B_i = \begin{bmatrix} -\cos(\theta_{i-2,i}) & -\operatorname{sen}(\theta_{i-2,i}) & 0 & -d_{i-1,i}\cos(\theta_{i-2,i}) \\ \operatorname{sen}(\theta_{i-2,i})\cos(\omega_{i-3,i}) & -\cos(\theta_{i-2,i})\cos(\omega_{i-3,i}) & -\operatorname{sen}(\omega_{i-3,i}) & d_{i-1,i}\operatorname{sen}(\theta_{i-2,i})\cos(\omega_{i-3,i}) \\ \operatorname{sen}(\theta_{i-2,i})\operatorname{sen}(\omega_{i-3,i}) & -\cos(\theta_{i-2,i})\operatorname{sen}(\omega_{i-3,i}) & \cos(\omega_{i-3,i}) & d_{i-1,i}\operatorname{sen}(\theta_{i-2,i})\operatorname{sen}(\omega_{i-3,i}) \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{5.16}$$

utilizadas pelo algoritmo branch-and-prune, [40], na determinação das coordenadas de um ponto no $\mathbb{R}^3$.

Essas matrizes homogêneas não estão representadas na base canônica, as bases utilizadas são discutidas adiante e estão ligadas aos vetores normais e de ligação associados a pontos próximos ao ponto i.

O algoritmo para o cálculo das coordenadas homogêneas de um ponto x_i a partir das distâncias de ligação, ângulos de ligação e de torção, para todo $1 \leq j \leq i$, é dado pela seguinte fórmula[6]:

$$x_i = \Big(\prod_{j=1}^{i} B_j\Big)e_4, \tag{5.17}$$

onde e_4 é o vetor canônico $(0,0,0,1)$ e as matrizes B_j, para $j = 1,2,3$, são definidas por: $B_1 = I_4$ (identidade de ordem 4),

$$B_2 = \begin{bmatrix} -1 & 0 & 0 & -d_{1,2} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{5.18}$$

e

$$B_3 = \begin{bmatrix} -\cos(\theta_{1,3}) & -\operatorname{sen}(\theta_{1,3}) & 0 & -d_{2,3}\cos(\theta_{1,3}) \\ \operatorname{sen}(\theta_{1,3}) & -\cos(\theta_{1,3}) & 0 & d_{2,3}\operatorname{sen}(\theta_{1,3}) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{5.19}$$

As matrizes B_j, para $j = 1,2,3$ podem ser representadas seguindo o formato da matriz em (5.16), desde que sejam assumidas as seguintes convenções:

[6]Estamos usando a mesma notação, x_i, tanto para representar as coordenadas cartesianas quanto para as homogêneas do ponto, uma vez que há uma bijeção entre elas quando se considera a última ordenada homogênea sempre igual a 1.

1. em relação ao ponto x_1 (a origem dos eixos): $d_{0,1} = 0$, $\theta_{-1,1} = \pi$, $\omega_{-2,1} = 0$,

2. em relação ao ponto x_2 : $d_{1,2}$ é dada, $\theta_{0,2} = 0$, $\omega_{-1,2} = \pi$,

3. em relação ao ponto x_3 : $d_{2,3}$ é dada, $\theta_{1,3}$ é dado, $\omega_{0,3} = 0$.

Uma interpretação possível do algoritmo (5.17) é considerar cada multiplicação por uma matriz B_j, $1 \leq j \leq i$, como o cálculo das ordenadas do ponto x_i tomando-se como origem dos eixos ortogonais o ponto x_{j-1}. Os eixos ortogonais em relação aos quais as ordenadas estão sendo expressas são definidos:

1. em relação à primeira ordenada: pelo vetor de ligação $v_{j-2,j-1}$,

2. em relação à segunda ordenada: pelo produto vetorial $\eta_{j-3,j-1} \times v_{j-2,j-1}$,

3. em relação à terceira ordenada: pela normal $\eta_{j-3,j-1}$.

Como $\eta_{j-3,j-1} = v_{j-3,j-2} \times v_{j-2,j-1}$, os eixos são ortogonais e atendem à regra da mão direita.

Analisando o início do algoritmo para o cálculo das coordenadas cartesianas do ponto x_i, ou seja, $B_i\ e_4$. Tem-se

$$B_i\ e_4 = R_{\omega_{i-3,i},v_{i-2,i-1}}\ R_{\pi-\theta_{i-2,i},\eta_{i-2,i}}\ T_i\ e_4,$$

descrevendo-se cada operação:

$$T_i\ e_4 = \begin{bmatrix} d_{i-1,i} \\ 0 \\ 0 \\ 1 \end{bmatrix}, \tag{5.20}$$

$$R_{\pi-\theta_{i-2,i},\eta_{i-2,i}}\ T_i\ e_4 = \begin{bmatrix} -d_{i-1,i}\cos(\theta_{i-2,i}) \\ d_{i-1,i}\,\mathrm{sen}\,(\theta_{i-2,i}) \\ 0 \\ 1 \end{bmatrix}, \tag{5.21}$$

$$R_{\omega_{i-3,i},v_{i-2,i-1}}\ R_{\pi-\theta_{i-2,i},\eta_{i-2,i}}\ T_i\ e_4 = \begin{bmatrix} -d_{i-1,i}\cos(\theta_{i-2,i}) \\ d_{i-1,i}\,\mathrm{sen}\,(\theta_{i-2,i})\cos(\omega_{i-3,i}) \\ d_{i-1,i}\,\mathrm{sen}\,(\theta_{i-2,i})\,\mathrm{sen}\,(\omega_{i-3,i}) \\ 1 \end{bmatrix}. \tag{5.22}$$

A equação (5.20) define o tamanho da translação a ser feita, e deve-se lembrar que $\|v_{i-1,i}\|_2 = d_{i-1,i}$. A primeira ordenada do vetor em (5.21) pode ser interpretada como a norma da projeção ortogonal de $v_{i-1,i}$ em $v_{i-2,i-1}$, uma vez que $\pi - \theta_{i-2,i}$ é ângulo entre $v_{i-1,i}$ e $v_{i-2,i-1}$. A segunda ordenada desse vetor seria a norma da

projeção ortogonal de $v_{i-1,i}$ no espaço ortogonal a $v_{i-2,i-1}$. Ver Figura 5.13, para exemplo com pontos x_6, x_7 e x_8. A seguir demonstram-se essas afirmações.

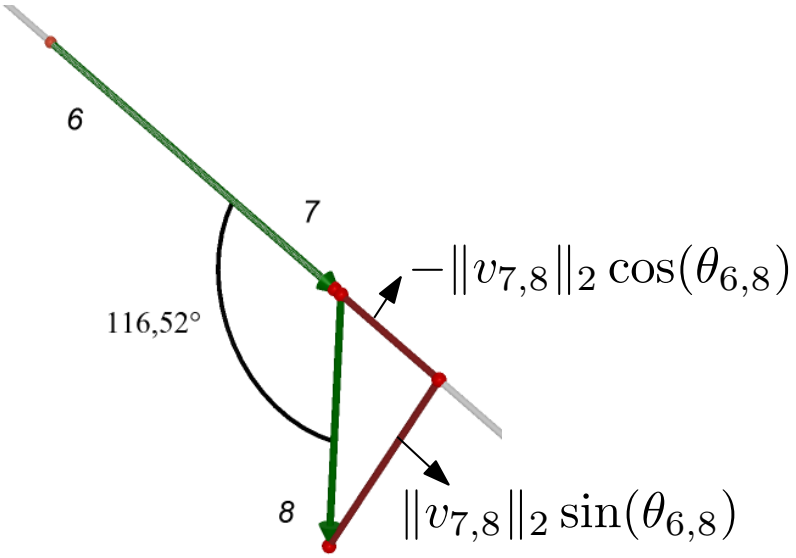


Figura 5.13: Projeções de $v_{7,8}$ em relação a $v_{6,7}$.

Proposição 5.2. *A primeira ordenada do vetor em* (5.21) *é calculada utilizando-se a norma da projeção ortogonal de* $v_{i-1,i}$ *em* $v_{i-2,i-1}$. *E como consequência, o subespaço sobre o qual ela está sendo projetada é a reta suporte do vetor* $v_{i-2,i-1}$.

Demonstração. A operador de projeção ortogonal sobre o espaço definido por $v_{i-2,i-1}$ é dado por

$$\frac{v_{i-2,i-1}}{\|v_{i-2,i-1}\|}\frac{v_{i-2,i-1}^T}{\|v_{i-2,i-1}\|}; \tag{5.23}$$

Sendo assim, a projeção ortogonal de $v_{i-1,i}$ em $v_{i-2,i-1}$ é dada por

$$\begin{aligned}\frac{v_{i-2,i-1}}{\|v_{i-2,i-1}\|}\frac{v_{i-2,i-1}^T}{\|v_{i-2,i-1}\|}v_{i-1,i} = \\ = \frac{v_{i-2,i-1}}{\|v_{i-2,i-1}\|}\frac{v_{i-2,i-1}^T}{\|v_{i-2,i-1}\|}v_{i-1,i}\frac{\|v_{i-1,i}\|}{\|v_{i-1,i}\|} = \\ = \frac{v_{i-2,i-1}}{\|v_{i-2,i-1}\|}\cos(\pi-\theta_{i-2,i})\|v_{i-1,i}\| = \\ = -d_{i-1,i}\cos(\theta_{i-2,i})\frac{v_{i-2,i-1}}{\|v_{i-2,i-1}\|}.\end{aligned} \tag{5.24}$$

O sinal negativo está relacionado ao uso do ângulo de ligação $\theta_{i-2,i}$, uma vez que o ângulo entre os vetores $v_{i-2,i-1}$ e $v_{i-1,i}$ vale $\pi-\theta_{i-2,i}$. E, também, pode-se interpretar que o vetor da base relacionado à primeira ordenada, ou seja, o primeiro vetor da base ortonormal utilizada é o $\frac{v_{i-2,i-1}}{\|v_{i-2,i-1}\|}$. □

Proposição 5.3. *A segunda ordenada do vetor em* (5.21) *é calculada utilizando-se a norma da projeção ortogonal de* $v_{i-1,i}$ *no espaço ortogonal a* $v_{i-2,i-1}$*, que nesse caso é um plano ortogonal a* $v_{i-2,i-1}$*.*

Demonstração. O projetor ortogonal no espaço ortogonal a $v_{i-2,i-1}$ é dado por

$$I - \frac{v_{i-2,i-1}}{\|v_{i-2,i-1}\|}\frac{v_{i-2,i-1}^T}{\|v_{i-2,i-1}\|}. \tag{5.25}$$

O objetivo é mostrar que a norma da projeção do vetor $v_{i-1,i}$ nesse espaço é igual à segunda ordenada do vetor em (5.21). Para diminuir a confusão visual causada pelo excesso de índices, considera-se que

$$v = v_{i-2,i-1}, \quad \overline{v} = \frac{v_{i-2,i-1}}{\|v_{i-2,i-1}\|}, \quad u = v_{i-1,i} \text{ e } \quad \overline{u} = \frac{v_{i-1,i}}{\|v_{i-1,i}\|}.$$

Com isso, o projetor ortogonal no espaço ortogonal a $v_{i-2,i-1}$ aplicado ao vetor $v_{i-1,i}$, será:

$$\begin{aligned}(I - \overline{v}\,\overline{v}^T)u \quad &= \quad u\overline{v}^T\,\overline{v} - \overline{v}\overline{v}^T u \quad = \quad (u\overline{v}^T - \overline{v}u^T)\overline{v} \quad = \\ &(u\overline{v}^T - \overline{v}u^T)\overline{v}\frac{\|u\|}{\|u\|} \quad = \quad \overline{v} \times \left[\overline{u} \times \overline{v}\right] \|u\|, \end{aligned} \tag{5.26}$$

devido à igualdade $(uv^T - vu^T)v = v \times (u \times v)$. Esse vetor é ortogonal ao vetor $v = v_{i-2,i-1}$ e sua norma pode ser calculada usando-se a fórmula $\|u \times v\|^2 = \|u\|^2\|v\|^2 - (v^T u)^2$:

$$\|\overline{v} \times \left[\overline{u} \times \overline{v}\right] \|u\| \,\| = \sqrt{\frac{\|v\|^2}{\|v\|^2}\left\|\overline{u} \times \overline{v}\right\|^2}\|u\| \tag{5.27}$$

uma vez que $\overline{v}^T \left[\overline{u} \times \overline{v}\right] = 0$, e assim a norma do vetor calculado em (5.26) será, retomando a notação com índices,:

$$\sqrt{\text{sen}^{\,2}(\theta_{i-2,i})}\|v_{i-1,i}\| = d_{i-1,i}\,\text{sen}\,(\theta_{i-2,i}), \tag{5.28}$$

já que, pela definição de ângulo de ligação, o seno sempre será positivo.

□

Proposição 5.4. *Os vetores* $n_{i-3,i-1}$*,* $n_{i-2,i}$ *e* $v_{i-1,i}\,\text{sen}\,(\theta_{i-2,i})$ *pertencem a um mesmo plano.*

Demonstração. Como estamos em $\mathbb{R}^3$, podemos assumir que $v_{i-1,i}\,\text{sen}\,(\theta_{i-2,i})$ está em plano ortogonal a $v_{i-2,i-1}$. Por sua vez

$$n_{i-3,i-1} = \frac{v_{i-3,i-2} \times v_{i-2,i-1}}{\|v_{i-3,i-2} \times v_{i-2,i-1}\|}$$

e

$$n_{i-2,i} = \frac{v_{i-2,i-1} \times v_{i-1,i}}{\|v_{i-2,i-1} \times v_{i-1,i}\|},$$

logo, os três vetores estão em um subespaço ortogonal a $v_{i-2,i-1}$ que é único, no caso tridimensional esse subespaço é um plano. □

Proposição 5.5. *Os vetores $n_{i-2,i}$ e $v_{i-1,i}\,\mathrm{sen}\,(\theta_{i-2,i})$ são ortogonais.*

Demonstração. Da proposição 5.3, tem-se que

$$v_{i-1,i}\,\mathrm{sen}\,(\theta_{i-2,i}) = \left[I - \frac{v_{i-2,i-1}}{\|v_{i-2,i-1}\|}\frac{v_{i-2,i-1}^T}{\|v_{i-2,i-1}\|}\right] v_{i-1,i} \tag{5.29}$$

e, por definição, que

$$n_{i-2,i} = \frac{v_{i-2,i-1} \times v_{i-1,i}}{\|v_{i-2,i-1} \times v_{i-1,i}\|}$$

Desenvolvendo (5.29), tem-se que

$$\left[v_{i-1,i} - \left[\frac{v_{i-2,i-1}^T}{\|v_{i-2,i-1}\|} v_{i-1,i}\right] \frac{v_{i-2,i-1}}{\|v_{i-2,i-1}\|}\right].$$

Ou seja, é uma combinação linear dos vetores $v_{i-2,i-1}$ e $v_{i-1,i}$ e, portanto, ortogonal a $n_{i-2,i}$. □

5.3.3 Distâncias entre Pontos

Esta seção se propõe a apresentar e justificar os cálculos para os ângulos de ligação e de torção quando são dadas todas as distâncias entre quatro pontos sucessivos. Será usada a notação $d_{i,j}$ para representar a distância entre os pontos i e j.

Em primeiro lugar, a lei dos cossenos será relembrada brevemente. Dado o triângulo de lados A, B e C medindo, respectivamente, a, b e c e cujos ângulos opostos aos lados A, B e C são, respectivamente, α, β e γ, então a lei dos cossenos afirma que:

$$\begin{aligned} a^2 + b^2 - c^2 &= 2ab\cos\gamma, \\ b^2 + c^2 - a^2 &= 2bc\cos\alpha, \\ a^2 + c^2 - b^2 &= 2ac\cos\beta. \end{aligned}$$

Proposição 5.6 (Cálculo de Ângulo de Ligação)**.** *Dados três pontos sucessivos, i, $i-1$ e $i-2$ cujas distâncias $d_{i-1,i}$, $d_{i-2,i-1}$ e $d_{i-2,i}$ são conhecidas, então o cosseno do ângulo de ligação $\theta_{i-2,i}$ é dado pela fórmula*

$$\cos(\theta_{i-2,i}) = \frac{d_{i-1,i}^2 + d_{i-2,i-1}^2 - d_{i-2,i}^2}{2d_{i-1,i}d_{i-2,i-1}}. \tag{5.30}$$

Esse ângulo é único.

Demonstração. É uma aplicação imediata da lei dos cossenos (ver Figura 5.14). A unicidade vem da definição de ângulo de ligação, pois nesse caso a função cosseno, entre $[0, \pi]$, é bijetiva. □

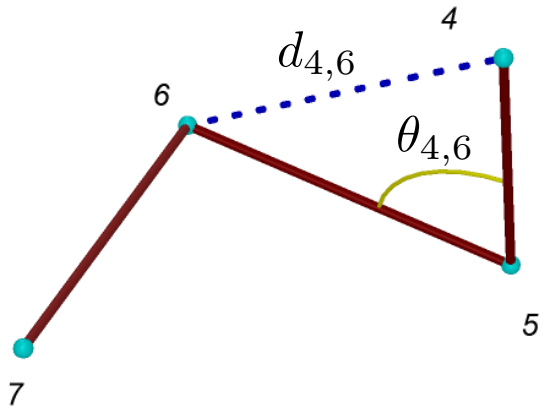


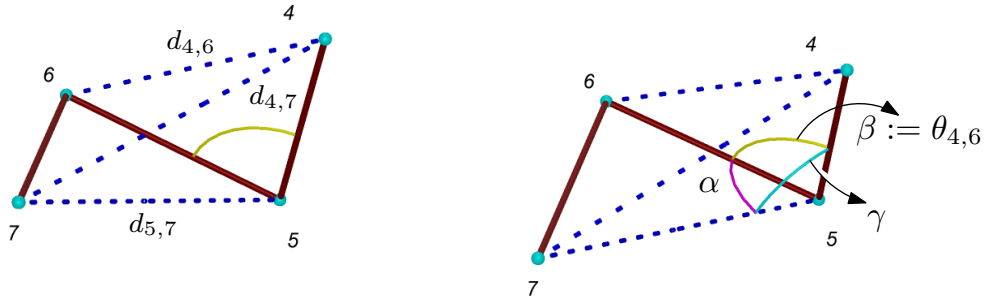
Figura 5.14: Quatro pontos sucessivos com um ângulo de ligação e distância auxiliar entre os pontos 4 e 6.

Para o cálculo do cosseno do ângulo de torção precisamos de alguns elementos acessórios. Na Figura 5.15a, mostramos uma caso particular das três distâncias auxiliares necessárias, $d_{4,6}$, $d_{5,7}$ e $d_{4,7}$; no caso geral serão $d_{i-3,i-1}$, $d_{i-2,i}$ e $d_{i-3,i}$. Observa-se que não se tratam de distâncias de ligação como definidas anteriormente, mas são fornecidas por algum meio; por exemplo, através de ressonância magnética nuclear. Dadas essas distâncias é possível calcular os cossenos dos ângulos α e γ opostos aos lados cujos comprimentos são, respectivamente, $d_{6,7}$ e $d_{4,7}$ (ver Figura 5.15b), no caso geral serão opostos aos lados $d_{i-1,i}$ e $d_{i-3,i}$:

$$\begin{aligned} \cos(\alpha) &= \frac{d_{i-2,i-1}^2 + d_{i-2,i}^2 - d_{i-1,i}^2}{2d_{i-2,i-1}d_{i-2,i}}, \\ \cos(\gamma) &= \frac{d_{i-3,i-2}^2 + d_{i-2,i}^2 - d_{i-3,i}^2}{2d_{i-3,i-2}d_{i-2,i}}. \end{aligned} \tag{5.31}$$

Assim como os ângulos de ligação, esses ângulos auxiliares não são ângulos orientados, logo os cossenos são suficientes para a unicidade dos ângulos α e γ.

Com essas construções acessórias, pode-se calcular o cosseno de ângulo de torção. Nesse caso, o conhecimento dessa medida não é suficiente para se conhecer o ângulo de torção: há duas possibilidades dependendo do sinal do seno.



(a) Distâncias adicionais.

(b) Ângulos adicionais.

Figura 5.15: Distâncias e ângulos acessórios para cálculo de cosseno de ângulo de torção a partir das distâncias dadas.

Observação 1. *Como visto na proposição 5.1, são necessárias as coordenadas do vetor de ligação $v_{i+2,i+3}$ para a definição do sinal do ângulo de torção, sinal que define de forma única o ângulo, como as coordenadas não estão disponíveis, não é possível esse cálculo.*

Proposição 5.7 (Cálculo do Cosseno de Ângulo de Torção)**.** *Sejam os pontos ordenados i, $i-1$, $i-2$, $i-3$ tais que sejam conhecidas todas as distâncias entre eles. Sejam o ângulo de ligação $\beta := \theta_{i-2,i}$, e os ângulos auxiliares α e γ. Seja o triângulo formado por lados cujos comprimentos são $d_{i-2,i}$, $d_{i-1,i}$, $d_{i-2,i-1}$ e seja α o ângulo oposto ao lado cujo comprimento é $d_{i-1,i}$. Seja o triângulo formado por lados cujos comprimentos são $d_{i-3,i-2}$, $d_{i-2,i}$, $d_{i-3,i}$ e seja γ o ângulo oposto ao lado cujo comprimento é $d_{i-3,i}$. Então o cosseno do ângulo de torção $\omega_{i-3,i}$ é dado por*

$$\cos\,\omega_{i-3,i} = \frac{\cos\,\gamma - \cos\,\alpha\cos\,\beta}{\sqrt{1-\cos^2\,\alpha}\sqrt{1-\cos^2\,\beta}}. \tag{5.32}$$

Esse valor não garante a unicidade do ângulo de torção, podendo, no entanto, haver dois valores possíveis definidos a partir do seno do ângulo de torção.

Demonstração. Da definição de ângulo de torção (ver seção 5.3.1), tem-se que

$$\cos\omega_{i-3,i} = (\eta_{i-3,i-1}, \eta_{i-2,i}),$$

onde $\eta_{i-3,i-1}$ é a normal ao plano definido pelos pontos $i-3$, $i-2$, $i-1$ e $\eta_{i-2,i}$ é a normal ao plano definido pelos pontos $i-2$, $i-1$, i. As representações dessas normais podem ser feitas utilizando quaisquer dos vetores linearmente independentes dos planos aos quais as normais são ortogonais e não apenas os vetores de ligação, como foi feito anteriormente. Na Figura 5.16, é possível ver uma representação

particular de vetores de ligação e auxiliares. Definindo-se os seguintes vetores:

$$u := v_{i-2,i-1}, \quad w := -v_{i-3,i-2}, \quad v := v_{i-2,i}.$$

A normal $\eta_{i-3,i-1}$ pode ser representada por

$$\eta_{i-3,i-1} = \frac{v_{i-3,i-2} \times v_{i-2,i-1}}{\|v_{i-3,i-2}\| \times v_{i-2,i-1}} = -\frac{v_{i-2,i-1} \times v_{i-3,i-2}}{\|v_{i-2,i-1} \times v_{i-3,i-2}\|} = \frac{u \times w}{\|u \times w\|}.$$

Como o vetor v pertence ao plano gerado pelos vetores $u := v_{i-2,i-1}$ e $v_{i-1,i}$, e como estamos supondo que três pontos sucessivos não jamais colineares, então a normal $\eta_{i-2,i}$ pode ser escrita como:

$$\eta_{i-2,i} = \frac{v_{i-2,i-1} \times v_{i-1,i}}{\|v_{i-2,i-1} \times v_{i-1,i}\|} = \frac{u \times v}{\|u \times v\|},$$

pois a regra da mão direita é preservada por essa ordem no produto vetorial. Com essas convenções temos que

$$(\eta_{i-3,i-1}, \eta_{i-2,i}) = \left[\frac{u \times w}{\|u \times w\|}, \frac{u \times v}{\|u \times v\|}\right] = \frac{1}{\|u \times w\|\|u \times v\|}(u \times w, u \times v) =$$
$$= \frac{1}{\|u \times w\|\|u \times v\|}\Big((u,u)(v,w) - (u,v)(u,w)\Big).$$

Desenvolvendo o numerador tem-se (ver corolário C.2 na pág. 120):

$$(u,u)(v,w) - (u,v)(u,w) = \|u\|^2\|v\|\|w\|\cos\,\gamma - \|u\|\|v\|\cos\,\alpha\|u\|\|w\|\cos\,\beta =$$
$$= \|u\|^2\|v\|\|w\|(\cos\,\gamma - \cos\,\alpha\cos\,\beta).$$

O denominador será:

$$\|u \times w\|\|u \times v\| = \|u\|\|w\|\,|\operatorname{sen}\ \beta|\|u\|\|v\|\,|\operatorname{sen}\ \alpha| = \|u\|^2\|v\|\|w\|\,|\operatorname{sen}\ \beta|\,|\operatorname{sen}\ \alpha| =$$
$$= \|u\|^2\|v\|\|w\|\sqrt{1-\cos^2\beta}\sqrt{1-\cos^2\alpha}.$$

Com isso tem-se que:

$$\cos\,\omega_{i-3,i} = (\eta_{i-3,i-1}, \eta_{i-2,i}) = \frac{\cos\,\gamma - \cos\,\alpha\cos\,\beta}{\sqrt{1-\cos^2\alpha}\sqrt{1-\cos^2\beta}}.$$

□

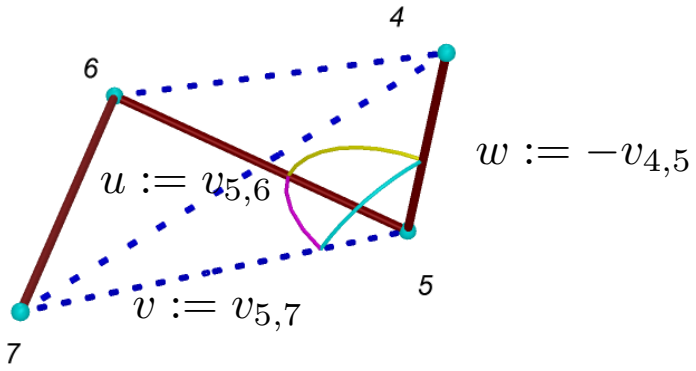


Figura 5.16: Quatro pontos sucessivos com vetores auxiliares.

5.4 Branch-and-Prune e o Revised Updated Geometric Build-up

No capítulo 4, foi visto o algoritmo GB e suas variações. Uma delas, o RUGB, nesta seção, será comparada com a versão clássica do algoritmo BP, com a finalidade de se avaliar o comportamento do BP diante de algumas dificuldades observadas no RUGB. Primeiramente, a subseção 5.4.1 fará alguns comentários sobre essas dificuldades. Em seguida, a subseção 5.4.2 apresentará os testes feitos com esses dois algoritmos.

5.4.1 Aspectos Computacionais do Revised Updated Geometric Build-up

Escolha da base métrica inicial

Iniciando-se uma análise do RUGB, que é descrito pelo algoritmo 4.4, nota-se que, na linha 1 deste algoritmo, escolhe-se um conjunto de quatro átomos, com todas as distâncias entre si e que não estejam no mesmo plano. A questão principal a ser pensada nessa linha é como se deve escolher estes átomos. Na página 46 deste trabalho, há a seguinte informação, que pode ser encontrada em [37]: não há garantias de que uma base arbitrária escolhida levará a determinação completa de uma conformação. Esta afirmação abre espaço para algumas propostas. A primeira delas seria o pensamento mais imediato: formar um conjunto de todas as combinações possíveis de quatro átomos, fazendo-se uma seleção daquelas combinações que formam tetraedros, isto é, cujos quatro átomos não sejam coplanares e tenham distâncias entre si, e escolher uma delas para se iniciar o algoritmo. Caso não se chegue a uma

solução para a estrutura da proteína, trocam-se os quatro átomos iniciais. Analisando esta proposta com cuidado, supondo um caso onde se tem todas as distâncias exatas, como no algoritmo 4.1 que representa o GB, onde também há a escolha dos quatro átomos logo de início, seja n o número total de átomos da proteína a ser determinada, seriam formados, então,

$$C_4^n = \frac{n!}{4!(n-4)!} = \frac{n(n-1)(n-2)(n-3)}{4!}$$

conjuntos para essa proposta, sem levar em consideração o fato de as combinações serem coplanares ou não. Para se determinar todos esses conjuntos seriam necessários $\mathcal{O}(n^4)$ passos, o que seria inviável, tendo em vista que o algoritmo 4.1, pode ser resolvido em $\mathcal{O}(n)$ passos. Isto significa que levaria mais tempo para se escolher os quatro átomos iniciais que para se determinar toda a estrutura. Voltando para o caso do RUGB e utilizando o mesmo raciocínio, é possível escrever um algoritmo para esta proposta. Sabendo-se, da subseção 4.4.2, que cada átomo apresenta um conjunto formado por seus vizinhos, cujo número de elementos chega no máximo a $d_{max} < n$, definindo-se $m(i)$ como o conjunto dos átomos vizinhos de i que sejam maiores que i, $m(i)_j$ como o vizinho j de i, $j = 1 \ldots \#m(i)$, e $\#m(i)$ como número de elementos de $m(i)$, tem-se o seguinte:

Algoritmo 5.3 Combinações de quatro átomos para inicialização do RUGB.

1: **for** $i = 1 \ldots n$ **do**
2: **if** $\#m(i) \geqslant 3$ **then**
3: **for** $j = 1 \ldots \#m(i)$ **do**
4: **for** $k = j + 1 \ldots \#m(i)$, com $j < \#m(i)$ **do**
5: **for** $p = k + 1 \ldots \#m(i)$, com $k < \#m(i)$ **do**
6: Forme o trio $\{m(i)_j, m(i)_k, m(i)_p\}$;
7: **if** $\{m(i)_j, m(i)_k, m(i)_p\}$ formar um triângulo **then**
8: **if** $\{m(i)_j, m(i)_k, m(i)_p\}$ não for coplanar a i **then**
9: $T(i) \leftarrow t$, onde $T(i)$ é o conjunto dos triângulos formados pelos vizinhos de i;
10: **end if**
11: **end if**
12: **end for**
13: **end for**
14: **end for**
15: **end if**
16: **end for**

O algoritmo 5.3 retorna todas as bases métricas iniciais possíveis para o RUGB.

Nota-se que, como $\#m(i)$ é no máximo d_{max}, este algoritmo pode ser executado em $\mathcal{O}(nd_{max}^3)$ passos, no pior dos casos, o que seria equivalente a executar o bloco *for* da linha 5 do algoritmo 4.4. Com esses dois exemplos, percebe-se que testar possíveis bases métricas iniciais pode não ser uma boa ideia em se tratando de se obter um método rápido de escolha.

Uma segunda proposta ainda pode ser feita: construir um conjunto de quatro átomos não coplanares, que tenham todas as distâncias entre si, isto é, um tetraedro, de tal forma que o número de átomos conectados aos seus vértices seja o maior possível. Esta proposta é sugerida em [38] para o algoritmo UGB, visto na seção 4.4.1. O método consiste em criar uma matriz de adjacência A, onde, para cada par de átomos ij,

$$a_{ij} = \begin{cases} 1, \text{ se } \exists d_{ij}, \\ 0, \text{ caso contrário.} \end{cases}$$

Logo após a criação de A, cada quatro linhas consecutivas desta matriz são somadas (ex.: linhas 1,2,3,4, linhas 2,3,4,5, linhas 3,4,5,6, etc.), obtendo-se uma nova matriz de ordem $n-3 \times n$. A partir dessa nova matriz, escolhe-se a linha que apresentar o maior número de quatros (4) e que represente um tetraedro válido para iniciar o algoritmo. Ter o maior número de quatros, neste caso, significa ter o maior número de átomos conectados ao tetraedro e, por isso, ter a maior quantidade de átomos determinados logo na primeira base métrica.

Assim, toda vez que a linha 6 do algoritmo 4.3 for executada, pode-se verificar se a base métrica inicial pode ser utilizada na determinação do átomo. Quanto maior o número de átomos conectados, maior a quantidade de iterações que não precisarão construir uma base métrica. No caso do RUGB, este método também pode ser utilizado. Para este trabalho, foi feito um programa em MATLAB que reproduz o algoritmo RUGB, conforme [37, 38]. Nesse programa foi utilizado o método acima descrito para se obter a primeira base métrica. Um segundo método de obtenção da base métrica inicial, *método dos triângulos de grau máximo* (algoritmo 5.5), pode ainda ser proposto. Consiste em utilizar a estrutura do RUGB para se determinar, em primeiro lugar, o átomo com o maior número de vizinhos e, a partir de sua lista de átomos vizinhos, obtêm-se os triângulos com maior grau (ver explicação na seção 4.4.2), isto é, os triângulos cujos graus dos átomos formadores são maiores. Desta forma, tem-se uma lista de bases métricas iniciais. Na tabela 5.1, é possível observar uma comparação, via RMSD, entre as estruturas obtidas pelo programa em MATLAB e as estruturas originais dos arquivos PDB. A quarta e a quinta coluna representam os valores das RMSDs obtidos por esse programa e os valores publicados em [38, p.7], respectivamente. A terceira coluna mostra o método utilizado para encontrar a primeira base métrica, da seguinte forma: zero representa o método

PDB ID	Nº de Átomos	*tol*	Base	RMSD(1)	RMSD(2)
2DX2	174	0,000	0	2,99E-11	2,31E-11
		0,000	1	1,16E-11	
1ID7	189	0,000	0	2,98E-12	8,62E-14
		0,165	0	9,61E-14	
1B5N	332	0,000	0	8,90E-12	1,93E-10
1FW5	332	0,000	0	1,70E-12	1,65E-12
1SOL	353	0,000	0	1,05E-12	7,33E-13
		0,000	1	7,03E-13	
1JAV	360	0,000	0	1,40E-12	2,78E-12
1MEQ	405	0,000	0	2,73E-10	2,43E-12
		0,050	0	8,83E-13	
1AMB	438	0,000	0	1,13E-12	7,11E-12
1R7C	532	0,000	0	4,55E-10	8,62E-10
1HLL	540	0,000	0	2,44E-09	2,83E-12
		0,050	1	1,62E-12	
1VII	596	0,000	0	2,44E-10	3,56E-10
1HIP	617	0,000	0	3,66E-04	4,80E-10
		0,050	1	1,22E-10	
1ULR	677	0,000	0	1,93E-06	3,84E-10
		0,015	0	1,26E-07	
1BOM	700	0,000	0	2,08E-11	1,36E-09
1AIK	729	0,000	0	7,34E-06	9,19E-09
		0,050	1	1,02E-10	
1CEU	854	0,000	0	6,07E-11	3,15E-10
1KVX	954	0,000	0	4,36E+00	7,21E-04
		0,020	0	2,66E-07	
1VMP	1166	0,000	0	2,39E-06	1,01E-06
		0,050	0	1,08E-08	
1HSM	1251	0,000	0	7,76E-09	5,88E-07
1HAA	1310	0,000	0	1,27E-07	4,49E-10
		0,000	1	1,31E-10	

Tabela 5.1: Comparação entre as estruturas obtidas e as originais via RMSD, variando-se a base métrica inicial e a tolerância (*tol*, que será explicada na subseção 5.4.1) para os elementos b_1, b_2, b_3 e b_4 das bases métricas. As variações foram feitas até se obter valores próximos ou menores aos valores da última colunas, obtidos em [37].

proposto em [37] para o UGB (usado aqui no RUGB e descrito pelo algoritmo 5.4). Valores maiores que zero indicam o método dos triângulos de grau máximo (descrito pelo algoritmo 5.5), onde o valor disponível na tabela indica qual a base está sendo utilizada (um é a primeira base, dois a segunda e assim por diante). Nota-se, pela tabela 5.1, que o método dos triângulos de grau máximo, em relação ao proposto em [38], pode ajudar um pouco, como no caso da molécula 1HAA que a RMSD saiu de 10^{-7} para 10^{-10}.

Algoritmo 5.4 Construção da base métrica inicial, algoritmo proposto em [38].

1: Construa a matriz A de adjacência;
2: {Quantidade de quatros por linha.}
3: $noffour \leftarrow 0$;
4: {Quantidade máxima de quatros encontrada em toda a matriz S.}
5: $noffourmax \leftarrow 0$;
6: **for** $i = 1 \ldots n - 3$ **do**
7: **for** $j = 1 \ldots n$ **do**
8: $S_{i,j} \leftarrow a_{i,j} + a_{i+1,j} + a_{i+2,j} + a_{i+3,j}$;
9: **if** $S_{i,j} = 4$ **then**
10: $noffour \leftarrow noffour + 1$;
11: **end if**
12: **end for**
13: **if** $noffourmax = noffour$ **then**
14: Acrescente i à lista de linhas com maior quantidade de quatros;
15: **else**
16: **if** $noffourmax < noffour$ **then**
17: $noffourmax \leftarrow noffour$;
18: Substitua todas as linhas armazenadas por i;
19: **end if**
20: **end if**
21: $noffour \leftarrow 0$;
22: **end for**
23: A base inicial é dada por $\{i, i+1, i+2, i+3\}$, onde i pertence à lista de linhas construída acima.

Algoritmo 5.5 Algoritmo dos Triângulos de Grau Máximo para construção da base métrica inicial.

1: Construa a estrutura do RUGB conforme a seção 4.4.2, selecionando as r linhas de maior grau;
2: {Grau de um triângulo.}
3: $g \leftarrow 0$;
4: {Grau máximo encontrado entre todos os triângulos.}
5: $g_{max} \leftarrow 0$;
6: **for** $i = 1 \ldots r$ **do**
7: **for** $j = 1 \ldots d_{max}$ **do**
8: **for** $k = j + 1 \ldots d_{max},\ j < d_{max}$ **do**
9: **for** $p = k + 1 \ldots d_{max},\ k < d_{max}$ **do**
10: Forme o trio $\{j, k, p\}$;
11: **if** $\{j, k, p\}$ formar um triângulo **then**
12: **if** i e $\{j, k, p\}$ não forem coplanares **then**
13: $g \leftarrow$ grau do triângulo $\{j, k, p\}$;
14: **if** $g_{max} = g$ **then**
15: Acrescente $\{j, k, p\}$ à lista de triângulos de maior grau;
16: **else**
17: **if** $g_{max} < g$ **then**
18: $g_{max} \leftarrow g$;
19: Substitua todos os triângulos já armazenados por $\{j, k, p\}$;
20: **end if**
21: **end if**
22: **end if**
23: **end if**
24: **end for**
25: **end for**
26: **end for**
27: **end for**
28: A base inicial é dada por $\{i, j, k, p\}$, onde i pertence ao conjunto de átomos de maior grau e $\{j, k, p\}$ pertence à lista de triângulos construída acima.

Com base nos algoritmos 5.4 e 5.5 é possível se fazer uma análise quanto ao número de passos gastos por cada um. No algoritmo 5.4, nas linhas 6 e 7, tem-se duas estruturas de repetição, o que, no pior dos casos, resulta em $\mathcal{O}(n^2)$ passos. Já no algoritmo 5.5, tem-se $\mathcal{O}(rd_{max}^3)$ passos, devido as linhas de 6 a 9, onde $r \ll n$.

Na tabela 5.1, existe ainda um outro parâmetro que alterado pode melhorar ou piorar os resultados para a RMSD. Este parâmetro será explicado no próximo tópico.

Determinação rígida da posição de um átomo

Acima, foram vistas algumas escolhas para a base métrica inicial e suas implicações na comparação, via RMSD, entre a estrutura obtida pelo RUGB e a estrutura original da molécula. Assim, foi apresentada parte da tabela 5.1. Porém, ficou faltando uma outra parte, que está relacionada à tolerância *tol*.

O processo de atualização descrito na seção 4.4.2, exemplo 4.3 e figura 4.4, pode ser considerado como a resolução de dois sistemas de equações. Assim, sejam

$$Y = \begin{bmatrix} y_1^T \\ y_2^T \\ y_3^T \end{bmatrix} = \begin{bmatrix} y_{11} & y_{12} & y_{13} \\ y_{21} & y_{22} & y_{23} \\ y_{31} & y_{32} & y_{33} \end{bmatrix}$$

as coordenadas de b_1, b_2, b_3 que serão recalculadas nesta seção, como foi feito na seção 4.4.2. Assim, da subseção 4.3, tem-se que

$$\begin{aligned} y_1 &= (0\ 0\ 0)^T \\ y_2 &= (d_{12}\ 0\ 0)^T. \end{aligned}$$

Desta forma, surge o primeiro sistema, que irá recalcular as coordenadas de y_3, como segue

$$||y_3 - y_1||^2 = d_{13}^2 \tag{5.33}$$
$$||y_3 - y_2||^2 = d_{23}^2. \tag{5.34}$$

Desenvolvendo-se as equações, obtêm-se

$$\begin{aligned} \|y_3\|^2 - 2y_3^T y_1 + \|y_1\|^2 = d_{13}^2, \\ \|y_3\|^2 - 2y_3^T y_2 + \|y_2\|^2 = d_{23}^2. \end{aligned}$$

Subtraindo-se as duas equações,tem-se

$$\begin{aligned} 2y_3^T y_2 - 2y_3^T y_1 + \|y_1\|^2 - \|y\|^2 &= d_{13}^2 - d_{23}^2 \\ 2d_{12}y_{31} - d_{12}^2 &= d_{13}^2 - d_{32}^2 \\ \frac{d_{13}^2}{2d_{12}} - \frac{d_{23}^2}{2d_{12}} + \frac{d_{12}}{2} &= y_{31}. \end{aligned}$$

Da equação (5.33),

$$\begin{aligned} ||y_3||^2 &= d_{13}^2 \\ y_{31}^2 + y_{32}^2 + y_{33}^2 &= d_{13}^2, \end{aligned}$$

e como y_1, y_2, y_3 estão no plano $x\hat{O}y$, tem-se $y_{33} = 0$ e

$$y_{32} = \pm\sqrt{d_{13}^2 - \left(\frac{d_{13}^2}{2d_{12}} - \frac{d_{23}^2}{2d_{12}} + \frac{d_{12}}{2}\right)^2},$$

Neste caso, segundo [38], escolhe-se a opção positiva para y_{32}. Este primeiro sistema, se refere à interseção entre duas esferas, $S(y_1, d_{13})$ e $S(y_2, d_{23})$[7], e o plano $x\hat{O}y$.

O segundo sistema se refere à interseção entre três esferas, $S(y_1, d_{1a})$, $S(y_2, d_{2a})$ e $S(y_3, d_{3a})$, considerando y_1, y_2, y_3, posicionados no plano $x\hat{O}y$. Essa interseção é descrita pelo seguinte sistema:

$$||y_a - y_1||^2 = d_{1a}^2, \tag{5.35}$$
$$||y_a - y_2||^2 = d_{2a}^2, \tag{5.36}$$
$$||y_a - y_3||^2 = d_{3a}^2, \tag{5.37}$$

onde $y_a = (y_{a1}\ y_{a2}\ \ y_{a3})^T$ é o ponto em que o átomo a (a ser determinado) está posicionado em relação ao conjunto Y de coordenadas e d_{ia} é a distância entre o átomo a e o átomo i, $i = 1 \ldots 3$. Esse sistema pode ser reescrito da seguinte forma:

$$y_{a1}^2 + y_{a2}^2 + y_{a3}^2 = d_{1a}^2, \tag{5.38}$$
$$(y_{a1} - d_{12})^2 + y_{a2}^2 + y_{a3}^2 = d_{2a}^2, \tag{5.39}$$
$$(y_{a1} - y_{31})^2 + (y_{a2} - y_{32})^2 + y_{a3}^2 = d_{3a}^2. \tag{5.40}$$

Resolvendo-se este sistema, obtêm-se

$$\begin{aligned} y_{a1} &= \frac{d_{12}}{2} + \frac{d_{1a}^2 - d_{2a}^2}{2d_{12}}, \\ y_{a2} &= \frac{d_{2a}^2 - d_{3a}^2 - (y_{a1} - d_{12})^2 + (y_{a1} - y_{31})^2}{2y_{32}} + \frac{y_{32}}{2}, \\ y_{3a} &= \pm\sqrt{d_{1a}^2 - y_{a1}^2 - y_{a2}^2}. \end{aligned}$$

Nota-se, no resultado do segundo sistema, que há duas soluções possíveis para y_a, pois y_{3a} pode assumir um valor positivo ou negativo. Para se escolher qual a melhor solução, utiliza-se o átomo b_4, não coplanar a b_1, b_2, b_3 e vizinho de a, logo depois que y_1, y_2, y_3 e as duas opções para y_a forem reposicionados na estrutura da molécula. A questão que ainda permanece é em que ocasião usa-se a tolerância *tol* mencionada na tabela 5.1. Imaginando-se os pontos y_1, y_2, y_3 colineares, pode ocorrer que o sistema, descrito pelas equações de (5.35) a (5.37), pode ter infinitas soluções

[7] $S(A, r)$ é esfera de centro em A e raio r.

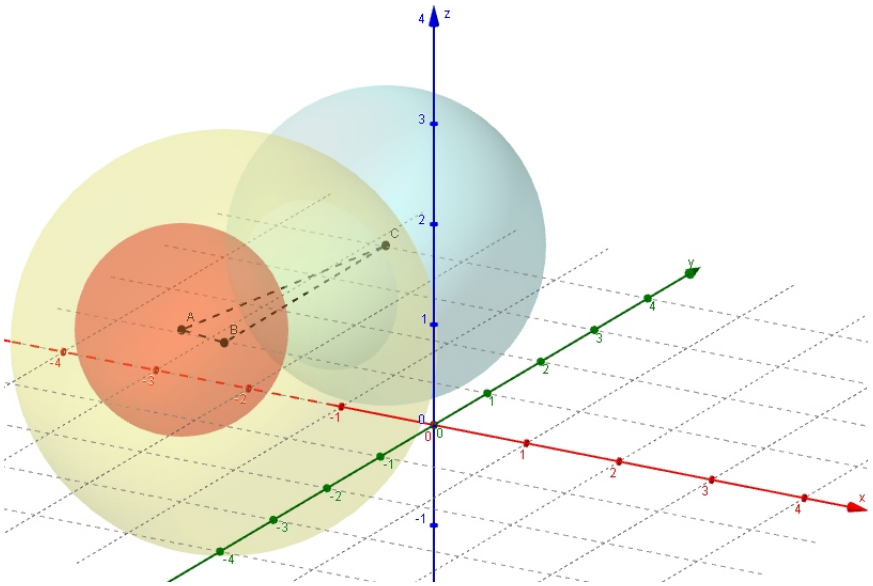


Figura 5.17: Interseção entre três esferas $S(A, r_a), S(B, r_b), S(C, r_c)$. Note que a esfera $S(A, r_a)$ está no interior de $S(B, r_b)$, isto é, $S(A, r_a) \cap S(B, r_b) = \emptyset$ e, consequentemente, $(S(A, r_a) \cap S(B, r_b)) \cap S(C, r_c) = \emptyset$ também. O triângulo ABC é um exemplo de triângulo "mal comportado".

ou nenhuma. Uma das formas de se evitar esse problema é construir triângulos "bem comportados" para servir de base para o átomo a. Bem comportado, neste caso, significa evitar lados muito pequenos, que façam os pontos y_1, y_2, y_3 serem considerados computacionalmente colineares (veja um exemplo na figura 5.17).

Uma outra questão importante, que também está relacionada com tol, é a determinação de y_a, ou de a, considerando-se que os dois valores de y_a já estão posicionados dentro da molécula. Usa-se, neste caso, a distância $d(a, b_4)$ do átomo a ao átomo b_4. O valor calculado que apresentar a distância $d(a, b_4)$ mais próxima de d_{ab_4} será o escolhido. Porém, existe um pequeno inconveniente: quando a distância entre o triângulo formado por b_1, b_2, b_3 e o átomo b_4 for muito pequena, estes quatro átomos podem ser considerados computacionalmente coplanares. Isto significa que, sejam a_1 e a_2 as duas posições possíveis para a, dentro da estrutura, se o triângulo b_1, b_2, b_3 e b_4 estiverem no mesmo plano, tem-se que $d(a_1, b_4) = d(a_2, b_4)$ e, com isso, não há critério de escolha para a posição de a.

Buscando-se amenizar estes inconvenientes, criou-se, neste trabalho, para a implementação em MATLAB do RUGB o parâmetro tol. Considerando-se que b_1, b_2, b_3 e b_4 assumem as posições x_{b1}, x_{b2}, x_{b3} e x_{b4} dentro da estrutura molecular, tem-se que a distância entre o plano definido pelo átomos b_i, b_j, b_k e o átomo b_p é dada por

$$h_i = \frac{\left|(x_{b_p} - x_{b_i})^T \left[(x_{b_j} - x_{b_i}) \times (x_{b_k} - x_{b_i})\right]\right|}{\left\|(x_{b_j} - x_{b_i}) \times (x_{b_k} - x_{b_i})\right\|}, \ i, j, k, p \in \{1, 2, 3, 4\}. \tag{5.41}$$

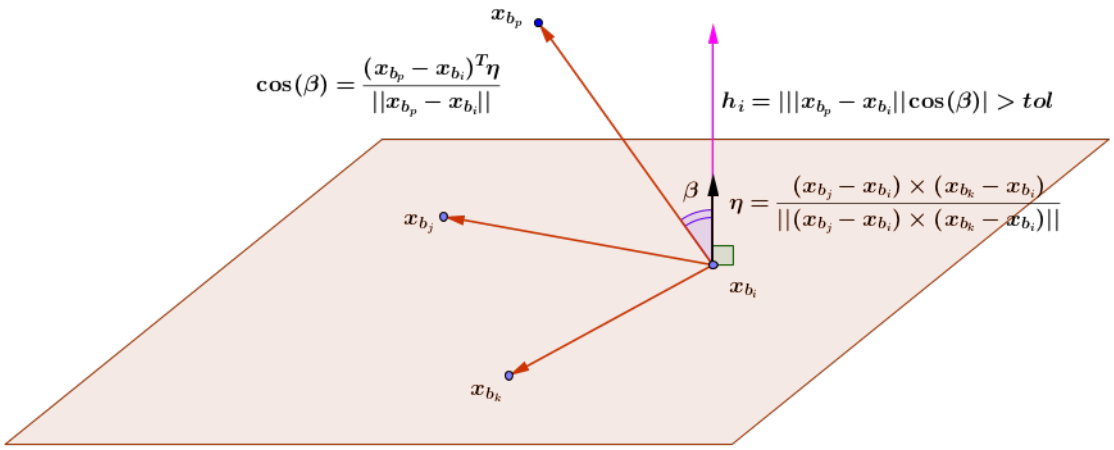


Figura 5.18: Explicação geométrica para Equação (5.41).

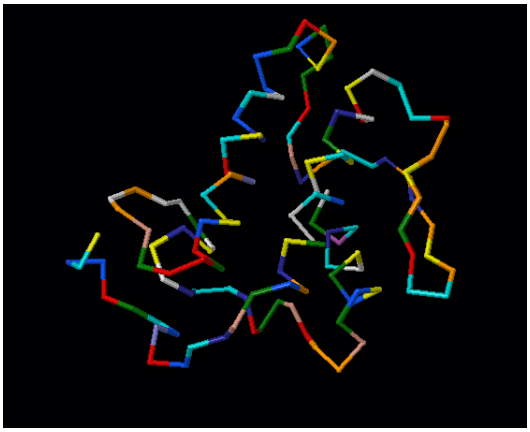
Pensando-se que os quatro átomos b_1, b_2, b_3 e b_4 formam um tetraedro, pode-se dizer que x_{b_i} é a posição de um de seus vértices e h_i é a altura em relação ao plano definido pelos pontos $x_{b_i}, x_{b_j}, x_{b_k}$ e o ponto x_{b_p}. Assim, para que os átomos b_1, b_2, b_3 e b_4 sejam considerados aptos a formarem uma base métrica para a, deve-se ter $h_i > tol,\ i = 1 \ldots 4$.

Na tabela 5.1, é possível ver um exemplo do uso de *tol* para se obter um resultado melhor para a RMSD no PDB ID 1KVX. Nota-se, neste caso, que, quando se aumenta a tolerância para 0,02, a implementação do RUGB é capaz de encontrar uma RMSD melhor. Na figura 5.19 é possível visualizar as diferenças desses resultados.

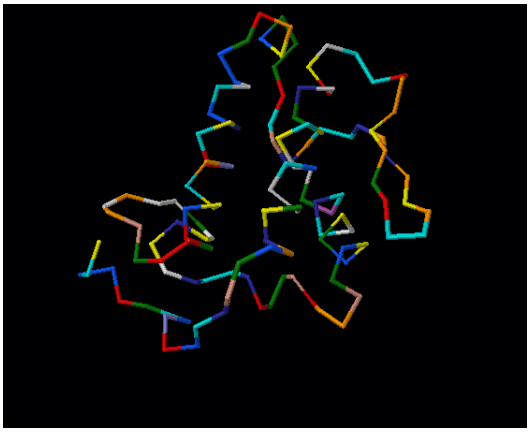
5.4.2 Comparação via RMSD

O objetivo desta seção é fazer uma comparação entre o algoritmo BP e o RUGB. Para tal, serão utilizadas as mesmas moléculas cujos PDB IDs se encontram na tabela 5.1, com exceção daquelas que apresentam mais de uma cadeia principal.

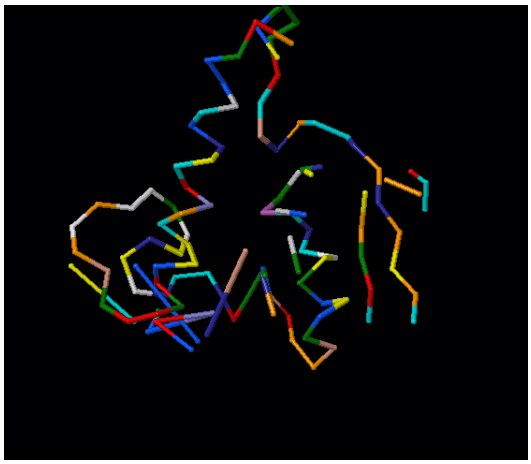
O método de comparação utilizado consiste em, a partir dos arquivos PDBs, construir instâncias que contenham somente os átomos das cadeias principais das moléculas. Foram selecionados pares de átomos capazes de gerar distâncias de até 5Å, onde a ordem atômica considerada para o conjunto V é a mesma do arquivo PDB. Através dos dois programas, serão determinadas posições para esses átomos. Então, utilizando-se a RMSD, descrita na seção 3.2, as posições calculadas pelos programas serão comparadas com as posições originais contidas nos PDBs. Os dois programas foram feitos em MATLAB (versão R2010b) e foram executados em um computador não dedicado, Intel Core 2 Quad Q6600, com 4GB de memória RAM, cujo sistema operacional é o Linux Ubuntu 12.04.



(a) Visualização do arquivo PDB ID 1KVX.



(b) Visualização do resultado da implementação do RUGB em MATLAB para $tol = 0.020$ (RMSD = 2,66E-07).



(c) Visualização do resultado da implementação do RUGB em MATLAB para $tol = 0.000$ (RMSD = 4,36E+00). É possível perceber a falta de vários trechos da molécula.

Figura 5.19: Resultados gráficos da implementação do RUGB em MATLAB para o *backbone* de PDB ID 1KVX.

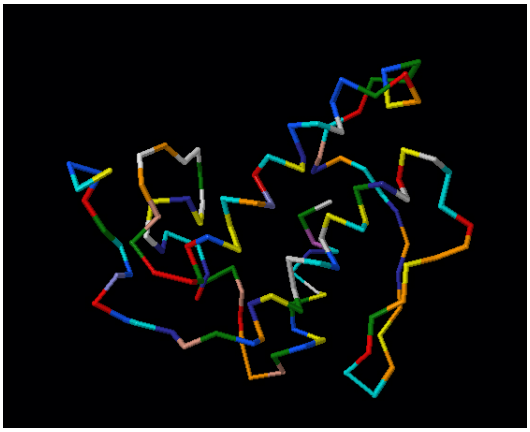
PDB ID	Nº de Átomos/ Cadeia Principal	RMSD (RUGB)	RMSD) (BP)
2DX2	174/33	9.59E-12	1,29E-10
1ID7	189/36	1.22E-12	3,01E-14
1B5N	332/69	2.08E-11	6,89E-11
1FW5	332/60	2.44E-12	3,31E-11
1SOL	353/60	1.47E-13	3,36E-12
1JAV	360/57	8.63E-13	2,06E-12
1MEQ	405/69	2.33E-12	1,52E-12
1AMB	438/84	3.52E-13	5,47E-12
1R7C	532/93	2.84E-09	3,61E-11
1HLL	540/96	9.85E-10	1.01E-10
1VII	596/108	7.01E-13	1,43E-12
1HIP	617/255	4.38E-05	6,63E-11
1ULR	677/261	5.70E-07	3,70E-12
1CEU	854/153	1,41E-11	1,36E-12
1KVX	954/369	**9.43e+00**	7,24E-11
1VMP	1166/213	3,71E-11	1,58E-10
1HSM	1251/237	**6.82E+00**	1,11E-10

Tabela 5.2: Comparação de estruturas via RMSD, entre o algoritmo BP e o RUGB. Nestes testes, determinam-se, somente, as cadeias principais. As instâncias geradas apresentam distâncias de até 5Å.

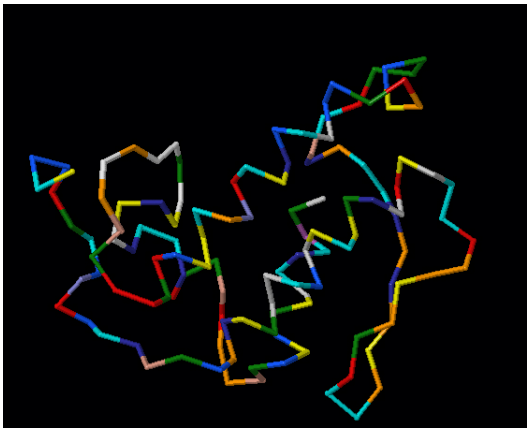
Na tabela 5.2, os resultados dos testes feitos com os dois algoritmos podem ser vistos. É importante lembrar que o RUGB, nestes testes, foi executado utilizando-se base 0 e *tol* 0.0 (ver mais detalhes sobre esses parâmetros nas subseções 5.4.1 e 5.4.1).

Ainda, na tabela 5.2 é possível se verificar que o BP obteve resultados similares ao RUGB, ao se comparar a estrutura das soluções obtidas com as estruturas originais do PDB. Nessa tabela, nota-se duas exceções, as proteínas 1KVX e 1HSM, onde o BP se destacou quanto à determinação de suas cadeias principais. Essas proteínas são casos onde se tem bases métricas formadas por pontos computacionalmente colineares, resultando, assim, na mesma situação discutida na subseção 5.4.1. O BP faz uso de operadores de rotação e translação para calcular a posição de um átomo de uma molécula de proteína, o que ajuda a evitar esse problema. Além disso, é de grande relevância se destacar a presença de uma ordem para o conjunto V de vértices de G (veja notação na seção 5.1), capaz de permitir que cada átomo, no momento do cálculo de sua posição, utilize como base as três posições anteriores. Considerou-se, nos testes acima, a mesma ordem utilizada pelos arquivos PDB para o conjunto V. O próximo capítulo se dedicará a uma nova abordagem para o MDGP, que abrange não só distâncias exatas, mas também intervalos de distâncias, o DMDGP intervalar (iDMDGP), onde se discutirá a construção de uma ordem atômica capaz de permitir a determinação da estrutura molecular de uma proteína através de uma nova versão do BP, o BP intervalar (iBP).

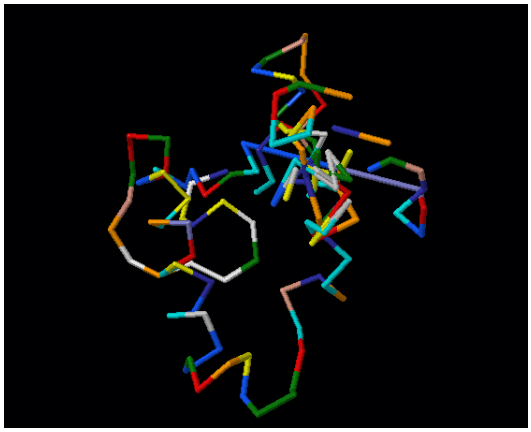
Antes de se iniciar uma nova abordagem para o MDGP, já é possível se destacar



(a) Visualização da cadeia principal da molécula do arquivo PDB ID 1KVX.



(b) Visualização do resultado da implementação do BP em MATLAB, exposto na linha 17 da tabela 5.2.



(c) Visualização do resultado da implementação em MATLAB do algoritmo RUGB.

Figura 5.20: Resultados gráficos da implementação do RUGB e do BP em MATLAB para o *backbone* do PDB ID 1KVX. Os resultados podem ser consultados numericamente na tabela 5.2.

algumas contribuições deste trabalho, tais como as implementações em MATLAB tanto do BP clássico como do RUGB, sendo esta última construída somente a partir das referências bibliográficas, pois, diferente do BP clássico que se encontra disponível em linguagem C, não foram encontradas implementações para o RUGB. Também é possível destacar a elaboração de uma estratégia que trata das alturas dos tetraedros formados pelas coordenadas dos átomos da base métrica a cada iteração do RUGB, observando-se, assim, a redução do erro na determinação da posição espacial dos átomos da cadeia principal de proteínas que apresentam sistemas mal condicionados. Além disso, foi feita a comparação entre os dois algoritmos, verificando-se casos onde o BP apresenta vantagens em relação ao RUGB, devido ao uso de rotações e translação para se determinar a estrutura tridimensional de uma proteína.

Capítulo 6

Abordagem discreta para o Problema Geométrico da Distância Molecular Intervalar (iDMDGP)

Nas seções 5.1, 5.2 e 5.3, foram vistas uma formulação discreta para o MDGP e a descrição do algoritmo branch-and-prune, onde foram consideradas distâncias exatas entre alguns pares de átomos. Para que o BP possa ser aplicado, as condições do DMDGP devem ser satisfeitas. A primeira condição exige que existam todas as distâncias exatas entre quatro átomos consecutivos, supondo-se que exista uma ordem entre eles. A segunda condição exige que, dada essa ordem, três átomos consecutivos não sejam colineares. Obedecer à primeira condição utilizando dados provenientes de RMN é uma tarefa difícil, pois pode não haver distâncias suficientes para tal, além do fato destes dados estarem sujeitos a erros de medição que fazem com que as distâncias fornecidas não sejam exatas, ou seja, o melhor que se obtém é um intervalo do tipo $[d_{ij}^L, d_{ij}^U]$. Outro elemento complicador para aplicação do BP, na determinação de estrutura de proteínas usando dados reais, é o fato de experimentos de RMN retornarem, em sua maioria, distâncias entre átomos de hidrogênio.

Assim, foi proposta uma nova abordagem para o MDGP em [1, 2] denominada *Interval Discretizable Molecular Distance Geometry Problem* (iDMDGP), onde não se utilizam somente distâncias obtidas via RMN, mas, também, distâncias calculadas a partir de informações acerca da molécula estudada. Desta forma, seja $G = (V, E, d)$ um grafo não direcionado, definido como na seção 3.1. Supondo-se que existe uma ordem em V[1], é possível considerar as seguintes afirmações:

[1]Caso não exista uma ordem em V, se faz necessária uma busca por uma base a ser utilizada no cálculo da posição espacial de cada átomo. Utilizando-se uma ordem atômica, essa busca não é necessária pelo fato de se considerar como base átomos anteriores (maiores detalhes na seção 6.2).

1. Distâncias entre os átomos $\{i, j\}$ separados por duas ligações covalentes são exatas e podem ser calculadas a partir dos comprimentos dessas ligações covalentes e do ângulo de ligação θ_{ij}. Segundo [43, 44], ângulos e distâncias de ligações covalentes podem ser considerados fixos em moléculas de proteínas. Define-se como $E' \subset E$ o conjunto de arestas $\{i, j\}$ que apresentam estas características, juntamente com as arestas que representam comprimento de ligações covalentes.

2. Distâncias entre átomos $\{i, j\}$ separados por três ligações covalentes não são exatas. Neste caso, calcula-se os extremos do intervalo $[d_{ij}^L, d_{ij}^U]$ e D distâncias pertencentes a este intervalo. O conjunto dos pares $\{i, j\}$ que obedecem a estas características é definido como $E'' \subset E$.

3. Distâncias obtidas via RMN são dadas por intervalos e serão utilizadas somente para teste de viabilidade de solução. Lembrando-se que a RMN não fornece todas as distâncias necessárias para satisfazer a primeira condição do DMDGP, pois, em geral, se obtêm distâncias entre 4 e 5Å, além do fato de se ter, em sua maioria, distâncias entre átomos de hidrogênio. O conjunto das arestas que apresentam estas características é definido como $F \subset E$. Assim, $E = E' \cup E'' \cup F$, onde $E' \cap E'' \cap F = \emptyset$.

Observando-se as afirmações 1 e 2, é possível descrever um método que utiliza somente distâncias entre pares de átomos $\{i, j\} \in E' \cup E''$, para a discretização do espaço de busca de soluções, e $\{i, j\} \in F$, para a verificação de soluções inviáveis. Assim, dada uma ordem em V, no iDMDGP encontra-se a posição de um átomo i considerando-se a interseção entre entre duas esferas S_{i-1} e S_{i-2}, e uma concha esférica S_{i-3}^h. Aqui, $S_{i-1} = S(x_{i-1}, d_{i-1,i})$, isto é, esfera de centro em x_{i-1} e raio $d_{i-1,i}$, enquanto que $S_{i-2} = S(x_{i-2}, d_{i-2,i})$ é a esfera de centro em x_{i-2} e raio $d_{i-2,i}$. Usando-se a mesma representação, tem-se a concha esférica $S_{i-3}^h = S(x_{i-3}, [d_{i-3,i}^L, d_{i-3,i}^U])$, onde seu centro encontra-se no ponto x_{i-3} e seu raio pertence ao intervalo $[d_{i-3,i}^L, d_{i-3,i}^U]$. Note que S_{i-3}^h também pode ser uma esfera, caso a distância $d_{i-3,i}$ seja exata. A Figura 6.1 mostra a interseção $S_{i-1} \cap S_{i-2} \cap S_{i-3}^h$. No DMDGP, para cada átomo i se tinha duas posições x_i e x_i'. No iDMDGP, além da possibilidade de se ter duas soluções para a posição de cada átomo, caso $d_{k,i}$ seja exata para todo $k \in \{i-3, i-2, i-1\}$, também se tem $x_i \in [x_i^L, x_i^U]$ e $x_i' \in [x_i'^L, x_i'^U]$, caso $d_{k,i} \in [d_{k,i}^L, d_{k,i}^U]$ para $k = i-3$. A seguir, encontra-se a definição do iDMDGP.

Definição 6.1 (Interval Discretizable Molecular Distance Geometry Problem – iDMDGP)**.** *Dado um grafo não direcionado $G = (V, E, d)$, tal que existe uma ordem $v_1, v_2, \ldots, v_n$ em V que satisfaça os seguintes requisitos:*

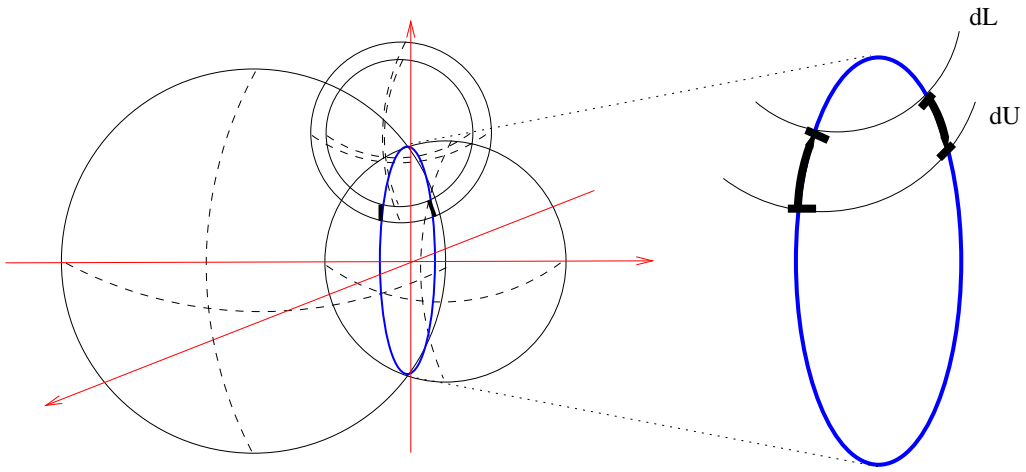


Figura 6.1: Interseção entre duas esferas S_{i-1} e S_{i-2} e concha esférica S^h_{i-3}. Note que os lugares geométricos de x_i e x'_i são intervalos. Figura retirada de [2].

1. *E contém todos os cliques*[2] *num grupo de quatro vértices consecutivos, isto é,*

$$\forall k \in \{4, \ldots, n\} \text{ e } \forall i, j \in \{k-3, k-2, k-1, k\}, \text{ com } i \neq j, \text{ então } \{i, j\} \in E \text{ ;}$$

2. *Vale a desigualdade triangular estrita, isto é,*

$$\forall i \in \{2, \ldots, n-1\}, \; d_{i-1,1+1} < d_{i-1,i} + d_{i,i+1};$$

3. $\forall k \in \{4, \ldots, n\}$ *e* $\forall i \in \{k-1, k-2\}, \{i, k\} \in E'$*;*

4. $\forall k \in \{4, \ldots, n\}, \; \{k-3, k\} \in E' \cup E''$*;*

o problema em questão é encontrar $x : V \to \mathbb{R}^3$ *tal que* $||x_i - x_j|| = d_{ij}$*, para cada* $\{i, j\} \in E$*.*

Tendo em vista as mudanças acima expostas, é preciso, então, modificar o algoritmo BP para encontrar as soluções desejadas. Além disso, se faz necessária a descrição de uma ordem para V que satisfaça os requisitos da definição 6.1. A próxima seção irá discutir algumas modificações no BP que irão resultar em um novo algoritmo, o *branch-and-prune intervalar* (iBP, de *interval* branch-and-prune [1, 2]). Já a seção 6.2 será dedicada à descrição de uma ordem para V que viabilize o uso do iBP.

[2] *Cliques* de um grafo não direcionado é um subconjunto de vértices tais que cada dois vértices são conectados por uma aresta.

6.1 Branch-and-Prune Intervalar

Nas seções 5.2 e 5.3, foi visto o algoritmo branch-and-prune para obtenção de soluções para o DMDGP. Porém, de acordo com o que foi visto no início deste capítulo, a aplicação do BP em dados reais oriundos de experimentos de RMN pode não ser possível, havendo assim a necessidade de definir uma classe de problemas onde se considera informações sobre as moléculas de proteínas, tais como o comprimento de ligações covalentes e os ângulos de ligação. Essa abordagem foi denominada iDMDGP e nela é possível haver, além de um conjunto de distâncias exatas, um conjunto de intervalos de distâncias. Desta forma, para atender aos novos requisitos do iDMDGP, o algoritmo *branch-and-prune intervalar (iBP)* foi proposto em [1, 2]. O iBP é uma extensão do BP, onde, para cada átomo i, se faz a verificação da distância $d_{i-1,i}$ da seguinte forma:

1. Se $d_{i-3,i}$ for exata, há a aplicação do BP clássico, podendo retornar duas soluções viáveis x_i e x_i'.

2. Se $d_{i-3,i} \in [d^L_{i-3,i}, d^U_{i-3,i}]$, então são selecionadas D distâncias pertencentes a esse instervalo para as quais são calculadas, da mesma forma apresentada nas seções 5.2 e 5.3, duas posições possíveis para o átomo a ser determinado.

Note que, o BP clássico retornava uma árvore de busca que continha todas as soluções encontradas por este algoritmo. No iBP, como são calculadas duas posições possíveis para um conjunto de D distâncias quando se tem $d_{i-3,i} \in [d^L_{i-3,i}, d^U_{i-3,i}]$, será retornada uma nova árvore, não mais binária, onde se tem até $2D$ ramificações em cada nó de determinados níveis. Na Figura 6.2, por exemplo, o nó x_1^6 tem t_1 ramificações, onde $t_1 \leqslant 2D$, considerando a possilidade de descartes de soluções inviáveis. O iBP é expresso no algoritmo 6.1. Seu primeiro bloco condicional verifica se o átomo é uma cópia de outro átomo calculado anteriormente. Este assunto será abordado na seção 6.2.

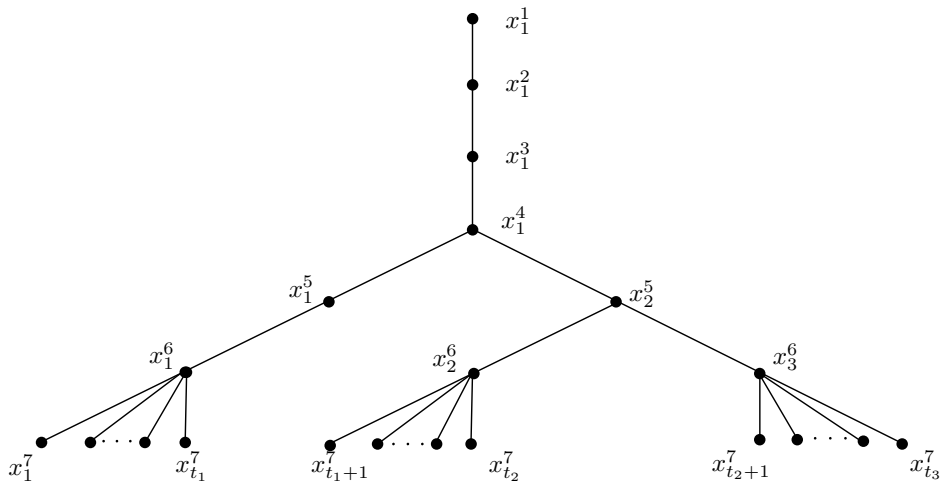


Figura 6.2: Árvore de busca no iBP. Note que $t_1 \leqslant 2D$, $t_2 - t_1 \leqslant 2D$ e $t_3 - t_2 \leqslant 2D$. Para se determinar a posição x^7, foram utilizadas as posições x^6, x^5 e x^4. O fato de o nível 7 ter mais de duas ramificações em cada nó significa que a distância $d_{i-3,i} = d_{4,7}$ é um intervalo.

Algoritmo 6.1 Algoritmo Branch-and-Prune Intervalar

1: $\underline{i\text{BP}}(j,r,d,D)$
2: **if** (r_j é uma cópia de outro átomo) **then**
3: {A cópia de átomos, tanto na cadeia principal quanto nas cadeias laterais de uma proteína, é um artifício que será explicado na seção 6.2.}
4: Copie as coordenadas do átomo original em $x_{r_j}^1$;
5: iBP($j+1$,r,d,D);
6: **else**
7: **if** (d_{r_{j-3},r_j} for uma distância exata) **then**
8: b = 2;
9: **else**
10: b = 2D;
11: **end if**
12: **for** $k \in \{1,\ldots,b\}$ **do**
13: Calcule a k-ésima posição $x_{r_j}^k$ para o r_j-ésimo átomo;
14: Verifique a viabilidade da posição $x_{r_j}^k$;
15: **if** ($x_{r_j}^k$ é viável) **then**
16: **if** ($j = |r|$) **then**
17: Foi encontrada uma solução;
18: **else**
19: iBP($j+1$,r,d,D);
20: **end if**
21: **end if**
22: **end for**
23: **end if**

É importante observar que, no algoritmo 6.1, se verifica se a distância d_{r_{j-3},r_j} é exata. Caso não seja, ocorre o cálculo de até $2D$ soluções para x_{r_j}. Lembrando-se

que o cálculo da posição do átomo continua da mesma forma expressa nas seções 5.2 e 5.3.

Para a aplicação do algoritmo 6.1, supõe-se a existência de uma ordem para o conjunto V tal que todos os requisitos da definição 6.1 sejam atendidos. A próxima seção deste capítulo apresentará uma ordem, proposta em [1, 2], capaz de atender a essas condições.

6.2 Descrição de uma ordem atômica para o Branch-and-Prune Intervalar

Até este capítulo, para a aplicação do algoritmo BP, utilizou-se a mesma ordem encontrada nos arquivos PDB de proteínas, para o conjunto V dos vértices de G. Porém, essa forma de se organizar os átomos antes de se determinar suas posições no espaço pode não satisfazer as condições impostas pelo iDMDGP (definição 6.1). Sendo assim, em [1, 2], uma ordem foi proposta para os átomos da cadeia principal. Antes de se abordar esse assunto nas subseções a seguir, se mostrará aqui a definição de *ordem com repetição* (ou, como na definição em inglês, *repetition order* ou *re-order*), encontrada em [1, 2].

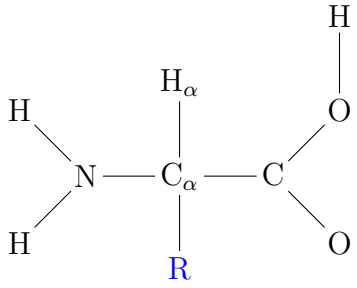
Definição 6.2 (Ordem com Repetição [1, 2])**.** *Uma* ***ordem com repetição*** *é uma sequência $r : \mathbb{N} \to V'$, tal que:*

- *Os vértices de índices $r_1,\ r_2,\ r_3$ têm todas as distâncias entre si.*
- *$\forall i \in \{4, \ldots, |r|\},\ \{r_{i-2}, r_i\}, \{r_{i-1}, r_i\} \in E'$, isto é, $\{r_{i-2}, r_i\}$ e $\{r_{i-1}, r_i\}$ são arestas de G que representam distâncias exatas.*
- *$\forall i \in \{4, \ldots, |r|\},$ ou $\{r_{i-3}, r_i\} \in E' \cup E''$ ou $r_{i-3} = r_i$, isto é, ocorre a repetição de um dos vértices de G.*

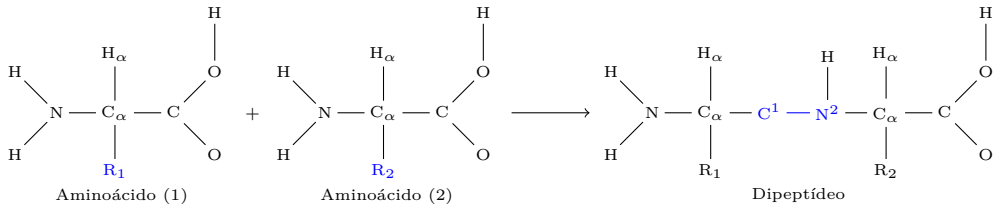
Nesta definição, $V' = V \cup \{0\}$ e $|r|$ é o número de elementos da sequência r.

Com a definição 6.2, é possível, então, construir uma nova estrutura de arestas baseada no grafo G. Nessa nova estrutura, alguns vértices podem ser repetidos a fim de se satisfazer às condições impostas pelo iDMDGP. As duas primeiras propriedades da definição 6.2 garantem que, no mínimo, dois antecessores do átomo a ser determinado (r_i), isto é r_{i-2} e r_{i-1}, apresentem distâncias exatas a r_i. Se $\{r_{i-3}, r_i\} \in E'$ ou $r_{i-3} = r_i$, então o átomo correspondente a r_i poderá ocupar duas posições[3], conforme as seções 5.2 e 5.3. Se $\{r_{i-3}, r_i\} \in E''$, então a posição de r_i pertence a intervalos, como mostra a Figura 6.1.

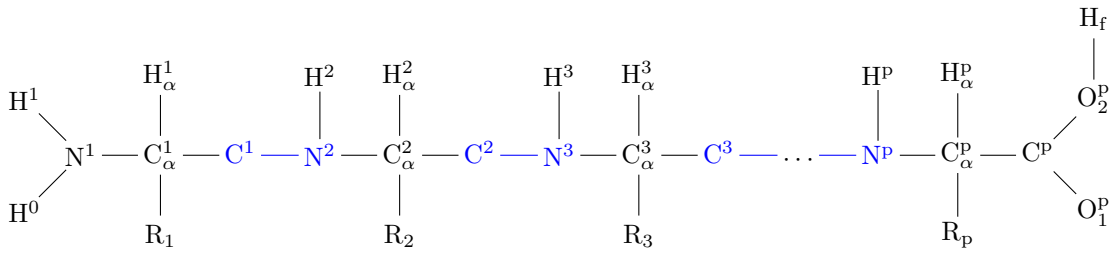
[3]Para simplificar a notação, denomina-se o *átomo correspondente a r_i* como *átomo r_i* ou somente r_i.



(a) Representação em Grafo de um aminoácido genérico. Parecida com a Figura 2.1, mas sem detalhes como ligações duplas entre átomos. Aqui, só é relevante saber onde se tem ligações covalentes.



(b) Junção de dois aminoácidos.



(c) Junção de p aminoácidos. Aqui, todos os átomos são numerados para facilitar a descrição da ordem.

Figura 6.3: Grafo de uma proteína.

Para a aplicação dessa ordem em uma proteína e construção de uma nova estrutura de arestas baseada no grafo G, a junção entre dois aminoácidos, proposta em [1, 2], será feita como na Figura 6.3b, onde o grupo carboxílico (COOH) é compactado no vértice C^1 e uma ligação N–H do grupo amina (NH2) é compactada no vértice N^2. Na Figura 6.3c, pode ser vista uma representação para uma cadeia de mais de 2 aminoácidos. É possível observar nessa compactação que um dos oxigênios do grupo carboxílico não aparece. Isso ocorre porque a posição desse átomo pode ser calculada a partir da posição dos outros átomos que podem ser vistos no grafo.

A ordem com repetição permite a obtenção de cadeias artificiais de proteínas, tanto principais como laterais, mais longas, devido a repetição de alguns átomos na sequência, mas capazes de serem determinadas eficientemente pelo algoritmo iBP. Nas próximas subseções, será visto como se dá a construção dessas cadeias.

6.2.1 Cadeia Principal de uma Proteína

A partir da definição 6.2 e utilizando-se o grafo expresso na Figura 6.3, em [1, 2], foi proposta uma ordem para o conjunto V para os átomos das cadeias principais de proteínas. Essa ordem, planejada de forma a atender às condições do iDMDGP, consiste no seguinte:

- Para o primeiro aminoácido, considera-se a sequência r^1_{PB}, assim

$$r^1_{PB} = \{N^1,\ H^1,\ H^0,\ C^1_\alpha,\ N^1,\ H^1_\alpha,\ C^1_\alpha,\ C^1\}.$$

- Para o segundo aminoácido, a sequência

$$r^2_{PB} = \{N^2,\ C^2_\alpha,\ H^2,\ N^2,\ C^2_\alpha,\ H^2_\alpha,\ C^2,\ C^2_\alpha\}.$$

- Para um aminoácido genérico, tem-se

$$r^i_{PB} = \{N^i,\ C^{i-1},\ C^i_\alpha,\ H^i,\ N^i,\ C^i_\alpha,\ H^i_\alpha,\ C^i,\ C^i_\alpha\},$$

  onde $i = 1, \ldots, p-1$ e $C^{i-1} = C^2$, quando $i = 1$.

- Finalmente, para o último aminoácido, tem-se

$$r^p_{PB} = \{N^p,\ C^{p-1},\ C^p_\alpha,\ H^p,\ N^p,\ C^p_\alpha,\ H^p_\alpha,\ C^p,\ C^p_\alpha,\ O^p_1,\ C^p,\ O^p_2\},$$

  onde p é o número de aminoácidos genéricos juntamente com o último aminoácido.

A Figura 6.4 apresenta um esquema dessas sequências. Note que, as cadeias laterais, representadas pelos grupos R, ainda não estão incluídas.

Seguindo as sequências r^1_{PB}, r^2_{PB}, r^i_{PB} e r^p_{PB}, cada átomo recebe um índice $k = 1, \ldots, |V_r|$, onde V_r é o conjunto obtido a partir dos vértices pertencentes a V, porém reordenados, considerando-se todas as repetições necessárias, e $|V_r|$ é o número de elementos de V_r.

A Tabela 6.1 mostra um exemplo de ordenação para uma cadeia principal de 4 aminoácidos. Tendo em vista essa tabela, é possível determinar a distância $d_{k,i}$ entre os vértices k e i, para $k = i-3, i-2, i-1$, de forma que o algoritmo iBP possa ser executado. Assim, as distância $d_{k,i}$ pode ser calculada como segue:

1. Se $\{k, i\} \in E'$, representar uma ligação covalente, então os vértices k e i apresentam distância $d_{k,i}$ conhecida.

1	N	20	H
2	H	21	N*
3	H	22	C_α*
4	C_α	23	H_α
5	N*	24	C
6	H_α	25	C_α*
7	C_α*	26	N
8	C	27	C*
9	N	28	C_α
10	C_α	29	H
11	H	30	N*
12	N*	31	C_α*
13	C_α*	32	H_α
14	H_α	33	C
15	C	34	C_α*
16	C_α*	35	O
17	N	36	C*
18	C*	37	O
19	C_α		

Tabela 6.1: Exemplo de ordenação para uma cadeia de quatro aminoácidos. O grafo G, neste caso, terá 37 vértices. O * indica a repetição de um átomo. Por exemplo, o átomo 5, que é um nitrogênio (N), é o átomo 1 repetido, seguindo o que é observado na Figura 6.4, onde o N dá origem a duas setas, a 1 e a 5.

2. Se existirem duas ligações covalentes entre os vértices k e i, então a distância $d_{k,i}$ será calculada a partir do ângulo de ligação (que é conhecido), utilizando-se a proposição 5.6.

3. Quando há 3 ligações covalentes entre k e i, calcula-se $d^L_{k,i}$ para o ângulo de torsão $\omega_{i-3,i} = 0$ e $d^U_{k,i}$ para $\omega_{i-3,i} = \pi$, utilizando-se a proposição 5.7 para se obter o intervalo $[d^L_{k,i}, d^U_{k,i}]$.

4. Distâncias entre átomos repetidos são consideradas exatas e iguais a zero.

Desta forma, tendo como exemplo os dados da Tabela 6.1, seja $i = 6$, tem-se, a partir da Figura 6.4, que a aresta $\{4, 6\}$, referente aos átomos C_α e H_α, representa uma ligação covalente, o que corresponde ao caso 1 dos itens acima mencionados. Já a aresta $\{5, 6\}$ apresenta duas ligações covalentes entre N* e H_α (N* – C_α – H_α), o que corresponde ao caso 2. Entre os vértices 3 e 6 (H e H_α), existem 3 ligações covalentes (H – N/N* – C_α – H_α), o que significa que $\{3, 6\} \in E''$ e que $[d^L_{k,i}, d^U_{k,i}]$ deve ser calculado de acordo com 3.

A próxima seção mostrará como se aplica o conceito de ordem nas cadeias laterais de proteínas.

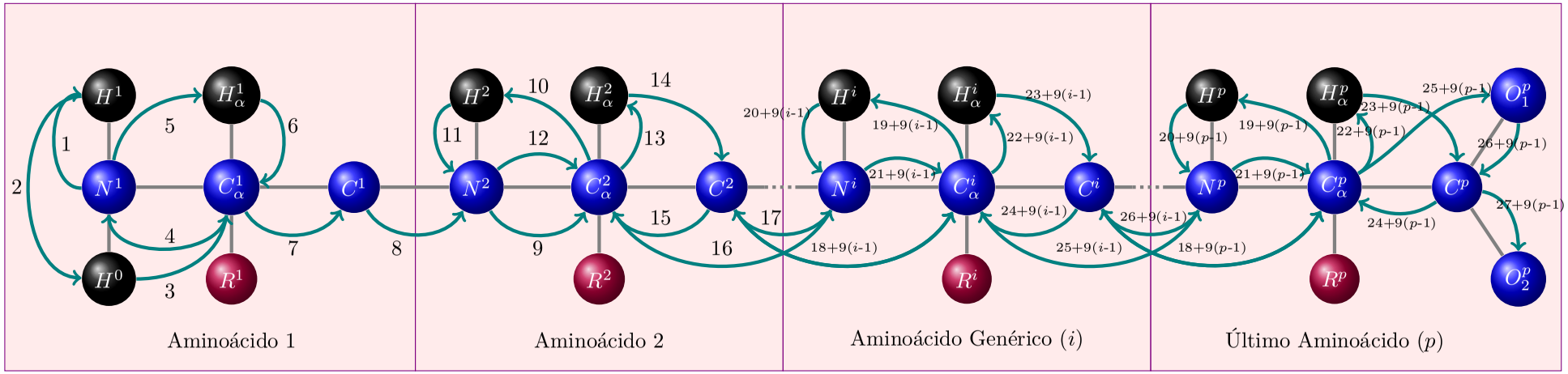


Figura 6.4: Ordem atômica para uma cadeia principal de $n = p+2$ aminoácidos. Onde p é a quantidade de aminoácidos genéricos juntamente com o último aminoácido e $i = 1, \ldots, p-1$.

6.2.2 Cadeias Laterais de uma Proteína

Nesta seção, até então, foram vistas a definição de ordem com repetição (definição 6.2) e sua aplicação na escolha de uma ordem para o posicionamento dos átomos das cadeias principais de proteínas no espaço euclidiano. Na Figura 6.4, que ilustra essa escolha, é possível notar a presença de grupos que não foram incluídos nas sequências r^k_{PB}, $k = 1, \ldots, p$: os grupos R. Esses grupos, vistos no capítulo 2, representam as *cadeias laterais* (*side chains*, em inglês) e são responsáveis por distinguir um aminoácido de outro, dando-lhes suas propriedades químicas. O objetivo desta subseção é incluir esses grupos dentro do conceito de ordem com repetição. Para tal, para cada aminoácido será elaborada uma sequência r^k_{SC}, onde as letras SC serão substituídas pelas abreviaturas de três letras vistas na Figura 2.3 e k se refere, agora, à posição do aminoácido dentro da proteína. Por exemplo, se uma proteína apresenta como segundo aminoácido a Glicina, a sequência que o representaria seria r^2_{GLY}. Cada sequência r^k_{SC} será formulada de forma que haja a possibilidade de conexão entre diferentes aminoácidos sem que as condições do iDMDGP deixem de serem atendidas. Assim, por exemplo, dado o grupo de aminoácidos GLY ASN GLU PHE ALA, pertencente a uma proteína, teria-se, então, o conjunto de sequências $\{r^1_{GLY},\ r^2_{ASN},\ r^3_{GLU},\ r^4_{PHE},\ r^5_{ALA}\}$ representando essa pequena parte da conformação. Cada aminoácido terá 3 sequências r^k_{SC}, correspondentes à primeira posição ($k = 1$), uma posição do meio ($1 < k < p$) e a última posição ($k = p$) dentro da proteína, onde p é o número total de aminoácidos. No apêndice D, encontram-se as ordenações r^k_{SC}, onde $k = 1, i, p$ e $1 < i < p$ e as Figuras 6.5, 6.6 e 6.7 ilustram as sequências r^k_{SC} para $k = 1$, de cada aminoácido.

A partir dessas ordens, elaboradas para as cadeias laterais de proteínas, é possível a construção de um programa capaz de gerar um conjunto de distâncias que atendam às condições do iDMDGP, utilizando-se distâncias e ângulos conhecidos a priori. A seção 6.3 irá propor uma implementação em linguagem C das sequências aqui apresentadas.

6.3 Geração de Instâncias

Na seção 6.2, foi discutida uma forma de se dispor os átomos de uma conformação a fim de se obedecer às condições impostas pelo iDMDGP. Para tal, foram apresentadas a definição de ordem com repetição (definição 6.2) e sua aplicação nos aminoácidos-padrão. Nesta seção, serão vistos os detalhes da implementação destes conceitos, que tem como objetivo principal a formação de um conjunto de distâncias para as ordenações vistas neste trabalho, fazendo com que a aplicação do algoritmo iBP seja viável. A este conjunto de distâncias, que é diferente daquele descrito na

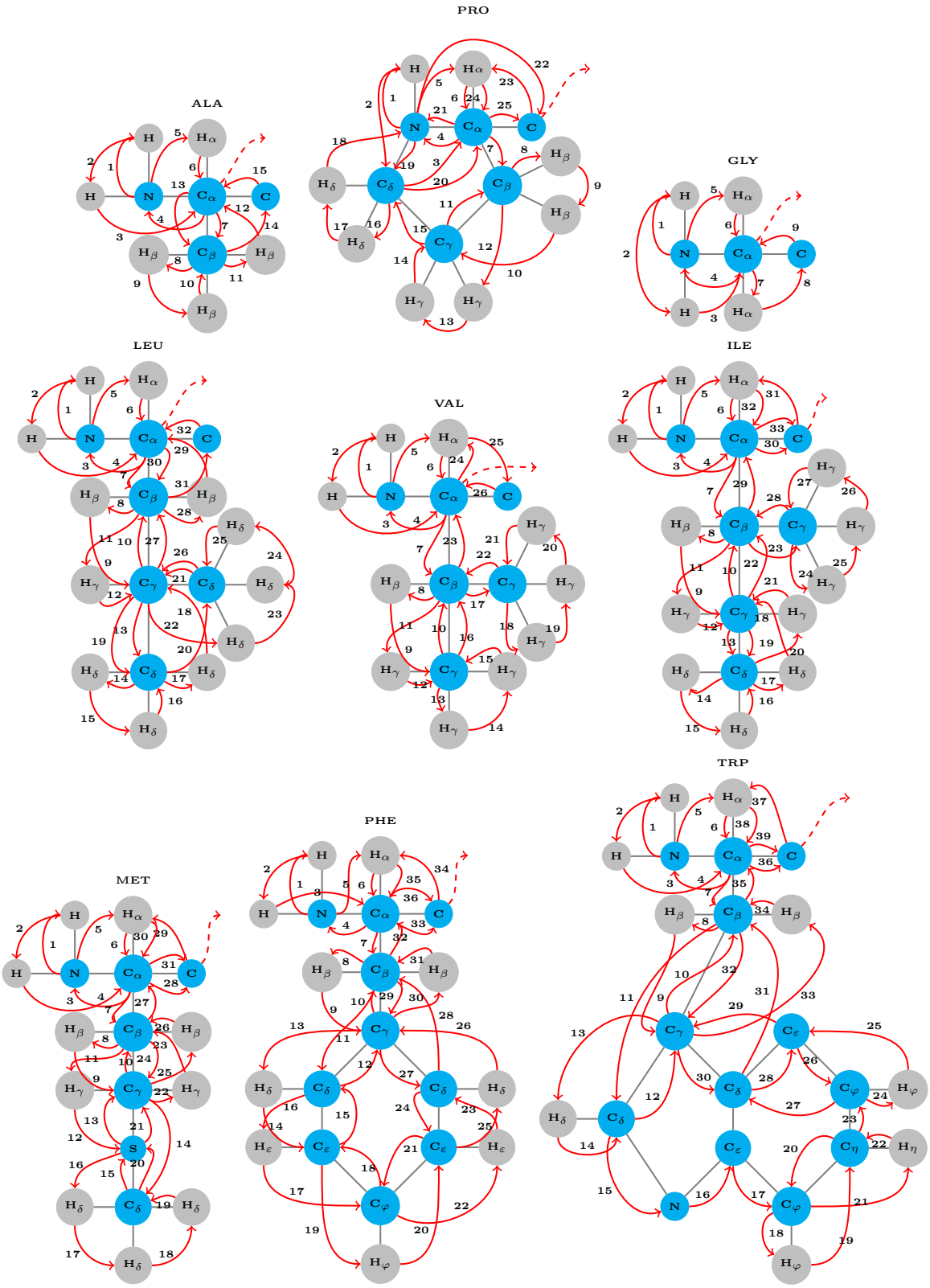


Figura 6.5: Ordenação para os átomos dos aminoácidos ALA, PRO, GLY, LEU, VAL, ILE, MET, PHE e TRP, que ocupam a primeira posição dentro de uma sequência.

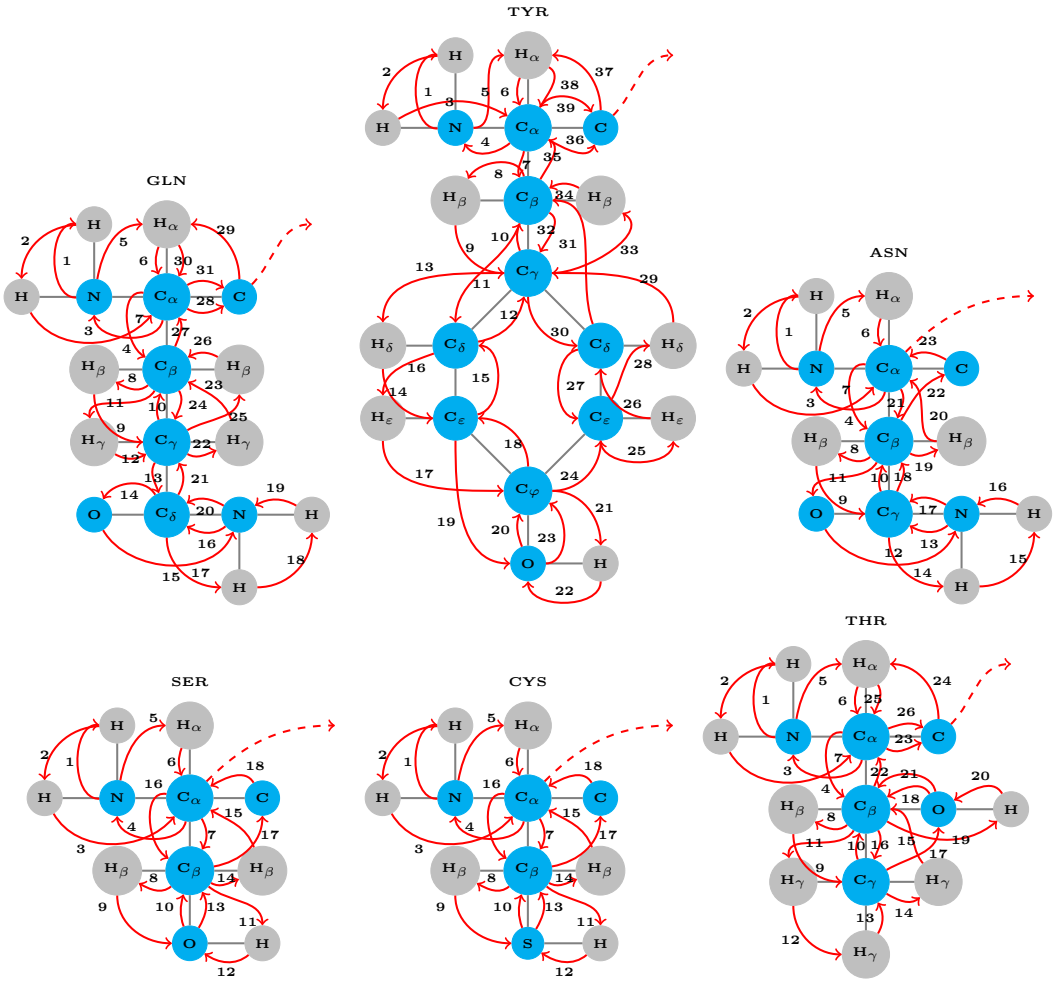


Figura 6.6: Ordenação para os átomos dos aminoácidos GLN, TYR, ASN, SER, CYS e THR, que ocupam a primeira posição dentro de uma sequência.

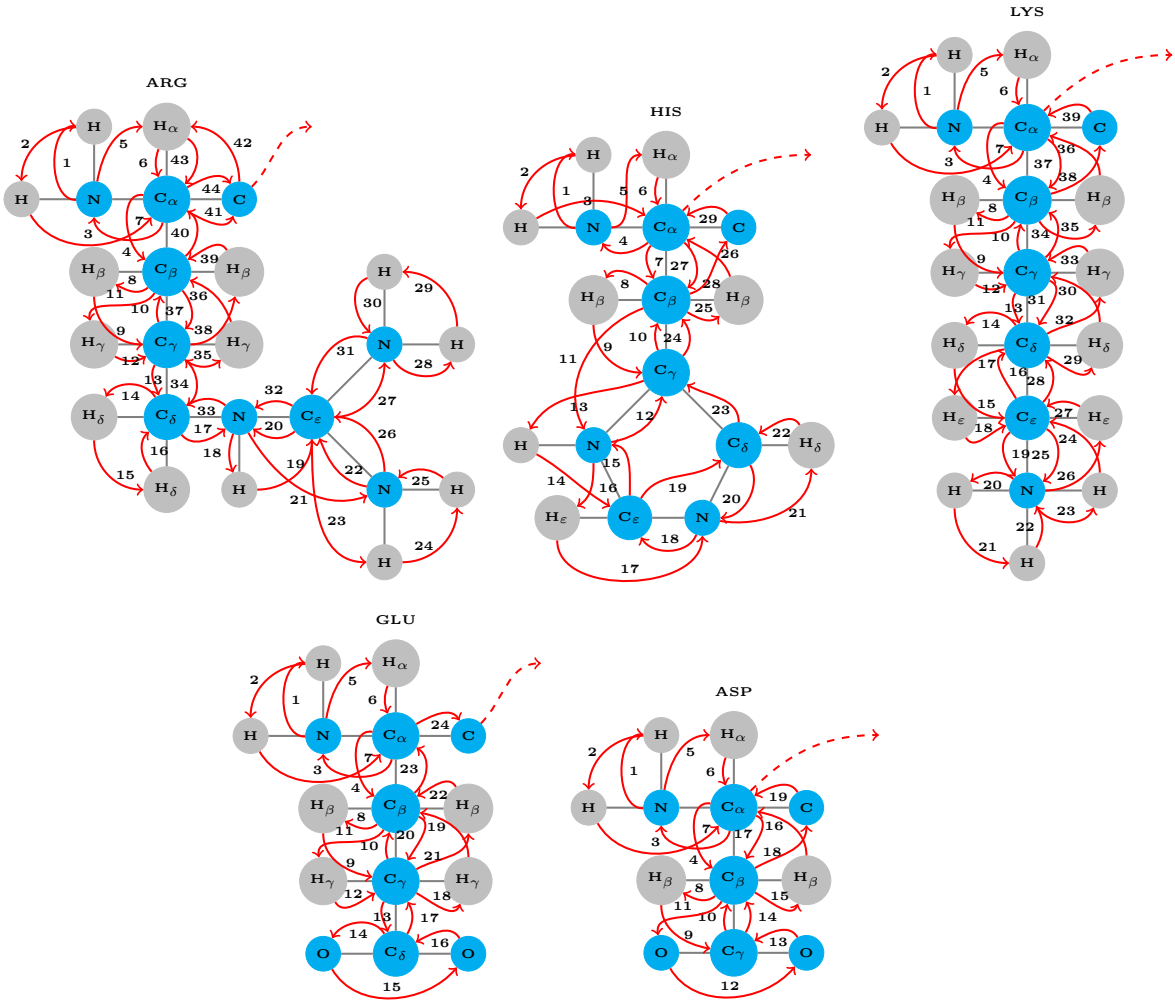


Figura 6.7: Ordenação para os átomos dos aminoácidos ARG, HIS, LYS, GLU e ASP, que ocupam a primeira posição dentro de uma sequência.

definição 6.1, denomina-se *instância* e se define conforme abaixo:

Definição 6.3. *O conjunto de distâncias de uma sequência de aminoácidos, cujos átomos foram reordenados de acordo com a definição 6.2, é denominado* **instância**.

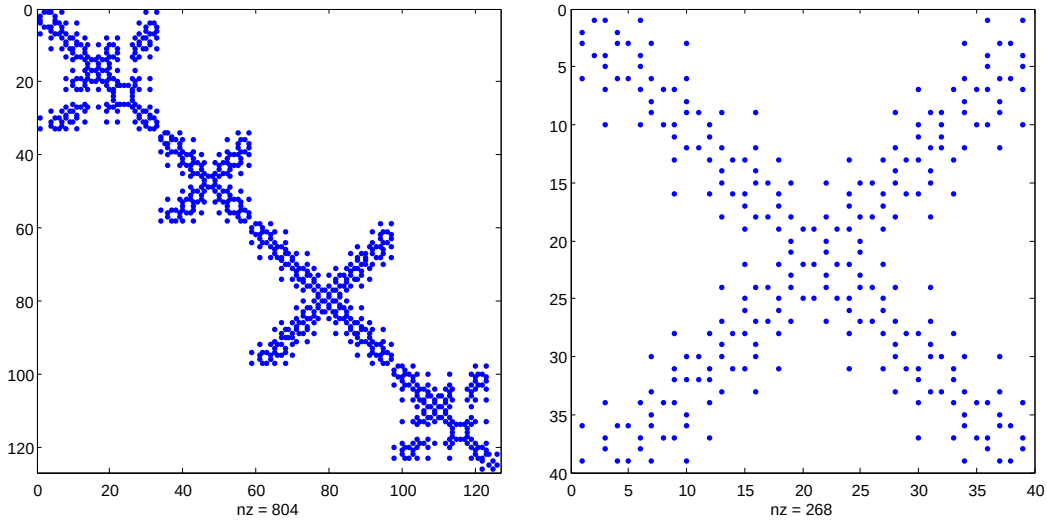
Assim, para que seja efetivada a construção deste conjunto, tendo em vista as definições 6.1, 6.2 e 6.3, considere o grafo $G_r = (V_r, E_r, d_r)$, obtido a partir de G, da seguinte forma:

- O conjunto V_r é formado pelos vértices pertencentes ao conjunto V, porém reordenados de acordo com a definição 6.2, considerando todas as repetições que se fizerem necessárias.
- O conjunto E_r é formado pelas arestas $\{i, j\}$, onde $i, j \in V_r$. Essas arestas podem estar relacionadas a distâncias exatas ou a intevalos. No primeiro caso, considera-se $\{i, j\} \in E'_r \subset E_r$ e, no segundo caso, $\{i, j\} \in E''_r \subset E_r$.
- d_r é uma função de $E_r \rightarrow \mathbb{R}_+$, onde $d_r(\{i, j\}) = \hat{d}_{ij} = \hat{d}_{ji}$ são distâncias euclidianas entre os átomos associados aos vértices $i, j \in V_r$.

Com base na construção do grafo G_r, tem-se, então, como objetivo encontrar as distâncias $\hat{d}_{ij}$. Sabe-se acerca dessas distâncias que elas podem estar relacionadas a arestas pertencentes a E'_r e, portanto, serem exatas. Podem, ainda, estar relacionadas a arestas de E''_r e não serem exatas, mas sim um intervalo real $[\hat{d}^L_{ij}, \hat{d}^U_{ij}]$. Sabe-se, também, do início deste capítulo, que a classificação dessas distâncias é determinada pelo número de ligações covalentes que existem entre dois átomos. Assim, é de extrema importância o mapeamento dessas ligações dentro de cada aminoácido, respeitando a ordenação estabelecida para V_r. Para tal, seja A_r a matriz de adjacência de G_r, será construída uma nova matriz $\mathcal{A}$. Essa matriz será formada pelos elementos de A_r que representam ligações covalentes. Desta forma, tem-se em $\mathcal{A}$ um mapeamento das ligações covalentes de toda a sequência de aminoácidos a ser determinada. Tendo em vista o que foi apresentado na seção 6.2, como as ordens estabelecidas para os aminoácidos são fixas, também é possível gerar matrizes menores, $\mathcal{A}_{SC}$, simétricas, formadas por elementos iguais a 0 e 1, que mapeiam suas ligações covalentes. Assim, tem-se, que

$$\mathcal{A} = \begin{bmatrix} \mathcal{A}^1_{SC} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathcal{A}^p_{SC} \end{bmatrix},$$

onde p é o número total de aminoácidos. Ou seja, $\mathcal{A}$ é uma matriz diagonal em blocos, cujos blocos são as matrizes $\mathcal{A}^i_{SC}$, $i = \{1, \dots, p\}$. A Figura 6.8 mostra exemplos destas matrizes.



(a) Matriz $\mathcal{A}$ para a sequência de aminoácidos LEU-GLU-LYS-VAL. Esta matriz tem sua diagonal formada pelos blocos $\mathcal{A}^1_{LEU}$, $\mathcal{A}^2_{GLU}$, $\mathcal{A}^3_{LYS}$ e $\mathcal{A}^4_{VAL}$.

(b) Matriz $\mathcal{A}^3_{LYS}$ referente ao aminoácido LYS. Note que esta matriz é um dos blocos de $\mathcal{A}$ na Figura 6.8a.

Figura 6.8: Exemplos das matrizes $\mathcal{A}$ e $\mathcal{A}^i_{SC}$. Nas Figuras, nnz equivale ao número de ligações covalentes presentes em toda a sequência, no caso da Figura 6.8a, ou presentes em um único aminoácido, no caso da Figura 6.8b.

Feito o mapeamento das ligações covalentes de todos os aminoácidos-padrão, é possível se determinar as distâncias relacionadas às arestas $\{i, j\} \in E_r$. Para isso, há quatro casos que devem ser analisados, dados dois vértices $i, j \in V_r$, como segue:

(i) Vértices que representam átomos separados por apenas uma ligação covalente. Neste caso, a distância $\hat{d}_{ij}$ é exata e é dada pelo comprimento desta ligação.

(ii) Vértices que representam átomos separados por duas ligações covalentes. Neste caso, sabendo-se que os ângulos formados por estas ligações são constantes, encontra-se o vértice k localizado entre i e j. Isso pode ser feito somando-se as linhas i e j de $\mathcal{A}$. A coluna que apresentar valor igual a 2 representa um vértice que tem ligação com i e j. A distância $\hat{d}_{ij}$ é dada pelas distâncias $\hat{d}_{ik}$ e $\hat{d}_{kj}$, e pelo ângulo $\widehat{ikj}$, utilizando-se a proposição 5.6.

(iii) Vértices que representam átomos separados por três ligações covalentes. Neste caso, a distância $\hat{d}_{ij}$ pertence ao intervalo $[\hat{d}^L_{ij}, \hat{d}^U_{ij}]$. Assim, deve-se encontrar os vértices k_1 e k_2 localizados entre i e j, aplicando-se o algoritmo 6.2. Então, as distâncias $\hat{d}^L_{ij}$ e $\hat{d}^U_{ij}$ serão calculadas através das distâncias $\hat{d}_{ik_1}$, $\hat{d}_{k_1k_2}$, $\hat{d}_{k_2j}$, utilizando-se a proposição 5.7 para os ângulos de torção iguais a 0 e π.

(iv) Quando não há ligações covalentes. Neste caso, pode ocorrer de os vértices i e j representarem o mesmo átomo e, portanto, $\hat{d}_{ij} = 0$.

Considerando-se os casos de (i) a (iv), é possível a criação de um algoritmo capaz de gerar instâncias que podem ser resolvidas pelo iBP. Neste trabalho, este será chamado iGen (de *Instance Generation*) e é descrito no algoritmo 6.3.

Algoritmo 6.2 Encontrando vértices entre i e j.

1: $t \leftarrow 0$.
2: $j' \leftarrow 0$.
3: Leia i, j, $\mathcal{A}$;
4: $Adj^i \leftarrow$ Todos os vértices ligados a i.
5: **for all** ($1 \leqslant k \leqslant |Adj^i|$ e $t \neq 1$) **do**
6: $v_1 \leftarrow Adj^i_k$.
7: **if** ($v_1 \neq j$) **then**
8: $Adj^{v_1} \leftarrow$ Todos os vértices ligados a v_1.
9: **for all** ($1 \leqslant p \leqslant |Adj^{v_1}|$ e $t \neq 1$) **do**
10: $v_2 \leftarrow Adj^{v_1}_p$.
11: **if** ($v_2 \neq j$) **then**
12: $Adj^{v_2} \leftarrow$ Todos os vértices ligados a v_2.
13: **for all** ($1 \leqslant s \leqslant |Adj^{v_2}|$ e $t \neq 1$) **do**
14: $j' \leftarrow Adj^{v_2}_s$
15: **if** ($j' = j$) **then**
16: $t \leftarrow 1$.
17: **end if**
18: **end for**
19: **end if**
20: **end for**
21: **end if**
22: **end for**
23: **if** ($t = 1$) **then**
24: Retorne v_1 e v_2.
25: **else**
26: Não há v_1 e v_2. Vértices têm número de ligações covalentes diferente de 3.
27: **end if**

Algoritmo 6.3 Algoritmo iGen.

1: Leia a sequências de aminoácidos $r_{SC} = \{r^1_{SC}, r^2_{SC}, \ldots, r^p_{SC}\}$, onde p é o número de aminoácidos.
2: Leia a matriz $\mathcal{A}$ que mapeia todas as ligações covalentes de r_{SC}.
3: **for all** $(i \in r_{SC})$ **do**
4: **for all** $(j \in r^i_{SC}$ and $j > 3)$ **do**
5: **for all** $(k \in \{3, 2, 1\})$ **do**
6: nCov $\leftarrow$ quantidade de ligações covalentes entre $j - k$ and j.
7: **if** (nCov $= 0$) **then**
8: $j - k$ and j são o mesmo vértice ou não têm ligação nenhuma.
9: $\hat{d}_{j-k,j} \leftarrow 0$
10: **end if**
11: **if** (nCov $= 1$) **then**
12: $\hat{d}_{j-k,j}$ é o comprimento de uma ligação covalente entre os átomos representados por $j - k$ e j.
13: **end if**
14: **if** (nCov $= 2$) **then**
15: Encontre o vértice v entre $j - k$ e j, usando $\mathcal{A}$.
16: Calcule $\hat{d}_{j-k,j}$ usando o ângulo de ligação $\widehat{ivj}$ e as distâncias de ligação $\hat{d}_{j-k,v}$ and $\hat{d}_{v,j}$.
17: **end if**
18: **if** (nCov $= 3$) **then**
19: $\hat{d}_{j-k,j} \in [\hat{d}^L_{j-k,j}, \hat{d}^U_{j-k,j}]$.
20: Encontre os vértices v_1 e v_2 entre $j - k$ e j, usando o algoritmo 6.2.
21: Calcule $\hat{d}^L_{j-k,j}$ e $\hat{d}^U_{j-k,j}$ usando as distâncias $\hat{d}_{j-k,v_1}$, $\hat{d}_{v_1,v_2}$ e $\hat{d}_{v_2,j}$ e seus respectivos ângulos de ligação. Aqui, usa-se a proposição 5.7 para o cálculo do ângulo de torção entre os planos definidos por $\{j - k, v_1, v_2\}$ e $\{v_1, v_2, j\}$.
22: **end if**
23: **end for**
24: **end for**
25: **end for**

Além das distâncias exatas e dos intervalos de distâncias calculados pelo iGen, há outra categoria que foi mencionada ao longo deste trabalho: distâncias oriundas de experimentos de ressonância magnética nuclear (RMN). Através de experimentos de RMN, obtêm-se intervalos reais $[\hat{d}^L_{ij}, \hat{d}^U_{ij}]$, referentes a pares de átomos de hidrogênio. Essa informação pode ser utilizada pelo iBP como critério de poda, sendo utilizado, por exemplo, pelo teste apresentado na definição 5.2. A fim de se simu-

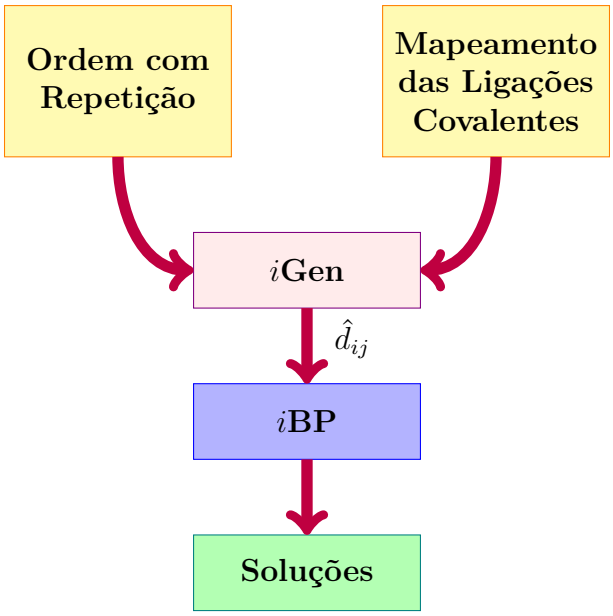


Figura 6.9: Esquema de obtenção de soluções dada uma sequência de aminoácidos.

lar esses intervalos, o *i*Gen calcula algumas distâncias entre hidrogênios (entre 4 – 5Å), a partir de um dos ramos da árvore de soluções, escolhido aleatoriamente. O conjunto de intervalos de distâncias entre hidrogênios causa impacto em termos de redução de soluções, principalmente após a inclusão das cadeias laterais. Existem aminoácidos que apresentam uma quantidade considerável de hidrogênios, como por exemplo a Lisina (LYS), a Leucina (LEU) e outros (veja Figura 2.3), o que aumenta a quantidade de informação a ser utilizada para este fim. A próxima seção abordará melhor este assunto, juntamente com experimentos feitos com a implementação aqui apresentada.

6.4 Experimentos e Resultados

A seção 6.3 mostrou a implementação de ordenações para os átomos das cadeias laterais de proteínas. Nesta seção, serão apresentados experimentos computacionais que utilizam o algoritmo *i*Gen juntamente com o *i*BP para obtenção de conformações, dada uma sequência de aminoácidos. Na Figura 6.9, há uma ilustração de como esses programas interagem entre si. Os algoritmos serão executados respeitando esse esquema. O objetivo dos testes aqui apresentados é avaliar o *i*BP quanto ao seu tempo execução e mostrar a influência dos hidrogênios presentes nas cadeias laterais no número de soluções encontradas. Nos resultados presentes nas Tabelas 6.2-6.4, não foram ainda utilizados dados reais para distâncias e ângulos de ligação covalente. No lugar, consideram-se todos os comprimentos de ligações covalentes iguais a 1,5 Å e todos os ângulos entre ligações covalentes iguais a 110°. Os algoritmos *i*Gen e *i*BP foram implementados em Linguagem C e compilados utilizando o compilador

Sequência	min(D)	Soluções sem H's	Soluções com H's
GLY-GLY-GLY-GLY	4	11424	48
GLY-ALA-GLY-ALA	5	9792	256
ALA-ALA-ALA-ALA	5	11518848	1536
CYS-CYS-CYS-CYS	5	35840	512
CYS-GLY-GLY-CYS	5	3216256	768
GLY-ASN-GLY-GLY	5	320	64
SER-SER-SER-SER	5	35840	512
ALA-GLY-ASN-CYS	6	10240	3200
CYS-ASN-ALA-SER	8	3785264	784
ASN-ASN-ASN-ASN	6	264720	6800
GLY-ASP-SER-GLY	5	165120	128
GLY-ASP-GLY-ALA	4	244128	864
GLY-ALA-GLU-GLY	6	10175840	800
GLY-GLY-GLY-LYS	14	130282650	37856

Tabela 6.2: Influência dos hidrogênios das cadeias laterais na redução do número de soluções.

GCC (versão 4.7.2), em ambiente Linux. As sequências utilizadas foram escolhidas aleatoriamente.

A Tabela 6.2 faz uma comparação entre o número de soluções obtido considerando-se intervalos de distâncias entre pares de hidrogênios das cadeias laterais (conforme descrito na seção 6.3) e o número de soluções obtido removendo-se estes intervalos. A segunda coluna mostra a quantidade mínima de ramificações em que se divide cada intervalo (parâmetro D do algoritmo 6.1). A terceira coluna mostra a quantidade de soluções obtidas removendo-se das instâncias os intervalos de distâncias relacionados aos hidrogênios das cadeias laterais. Na quarta coluna, estes hidrogênios são considerados e é possível notar uma grande redução no número de soluções. Quanto aos resultados da Tabela 6.2, é interessante destacar o último caso, onde se tem um resíduo de Lisina (LYS), o qual apresenta uma quantidade considerável de hidrogênios. Neste caso, o número de soluções é reduzido de mais de 130 milhões para um pouco mais de 37 mil.

Os próximos resultados mostram a velocidade do algoritmo iBP. A Tabela 6.3 mostra o tempo de execução do iBP na resolução de sequências de 4 aminoácidos. Na terceira coluna tem-se o número de vértices de G_r, isto é, todas as repetições necessárias são consideradas. É importante mencionar, também, que os intervalos de distâncias entre os átomos de hidrogênios das cadeias laterais são utilizados.

Na Tabela 6.3, é válido destacar que, mesmo com um número grande de soluções, o iBP demonstra um bom desempenho. A próxima tabela (Tabela 6.4) mostra o tempo de execução do iBP na resolução de sequências longas, para somente uma so-

Sequência	Número de Soluções	$\|V_r\|$	Tempo de Execução
ALA-THR-CYS-ILE	68394	94	375.83s
ASP-TYR-CYS-ALA	315624	94	302.12s
VAL-ALA-ILE-GLY	99350	86	300.84s
TYR-ASP-PHE-ASN	34536	120	512.90s
CYS-ASP-VAL-LYS	8288	106	387.15s
GLN-SER-LEU-ASN	360	108	0.34s
LEU-GLU-LYS-VAL	71682	94	335.20s
ILE-ALA-SER-LYS	153322	109	301.18s
TRP-SER-CYS-GLY	30078	88	301.12s
PHE-ALA-GLY-LEU	8736	96	300.68s
PHE-VAL-ILE-VAL	209716	124	305.41s
MET-ALA-LEU-GLY	17182	91	300.85s
GLU-GLU-ASN-GLY	42834	85	303.02s
GLN-SER-LEU-LYS	99588	124	304.67s
CYS-SER-LYS-MET	112320	109	301.85s

Tabela 6.3: Tempo gasto pelo algoritmo iBP para encontrar soluções para sequências de 4 aminoácidos.

n_{aa}	$\|V_r\|$	Tempo de Execução
10	214	13.13s
20	467	16.13s
30	723	40.49s
40	911	52.15s
50	1211	64.86s
60	1534	80.29s
70	1749	88.65s

Tabela 6.4: Tempo gasto pelo algoritmo iBP para encontrar somente uma solução para sequências longas de aminoácidos.

lução. Nesta tabela, n_{aa} se refere ao número de aminoácidos presentes na sequência. Nela, encontram-se instâncias de até 1749 vértices resolvidas em até 1,5 min.

Para se obter os resultados aqui apresentados, foi utilizado um computador Intel Core i7, 2,30GHz com 8GB de memória RAM.

Capítulo 7

Conclusão e Trabalhos Futuros

A determinação da estrutura tridimensional de uma proteína está diretamente relacionada à determinação de sua função. Através de experimentos de ressonância magnética nuclear (RMN), é possível se obter informações estruturais que incluem algumas distâncias entre alguns pares de átomos. O problema de se calcular a estrutura de uma molécula dadas algumas distâncias entre alguns de seus átomos é conhecido como Problema Geométrico da Distância Molecular (MDGP), que, em geral, é resolvido por métodos de otimização global que fazem uma busca no espaço contínuo $\mathbb{R}^3$.

Neste trabalho, foi estudada uma abordagem discreta do MDGP, o Discretizable Distance Geometry Problem (DMDGP), juntamente com um algoritmo capaz de resolvê-la, o branch-and-prune (BP). Sobre este algoritmo, foi feito um estudo detalhado de seus aspectos matemáticos e uma implementação em MATLAB. Foram estudados também os aspectos matemáticos e computacionais do algoritmo geometric build-up (GB) e suas variações, e construída uma versão em MATLAB deste algoritmo. Assim, é possível consultar neste trabalho uma comparação, quanto à acurácia, desses dois algoritmos e constatar que o BP apresenta bom desempenho em casos onde existem limitações numéricas, advindas do mal condicionamento das matrizes envolvidas na solução do GB. Neste contexto, a pesquisa acerca da abordagem discreta do MDGP foi, então, intensificada.

Tendo em vista que experimentos de RMN fornecem, no lugar de distâncias exatas, intervalos de distâncias, em sua maioria, entre pares de átomos de hidrogênio, os estudos sobre a forma discreta do MDGP foram prolongados até a sua forma intervalar, conhecida como Interval DMDGP (iDMDGP), e até a versão intervalar do BP, o Interval BP (iBP). O iDMDP estabelece condições para que uma determinada molécula de proteína tenha as posições de seus átomos calculadas e, para satisfazer estas condições, é imprescindível a elaboração de uma ordem atômica. Em [1, 2], esta ordem foi proposta para as cadeias principais de proteínas. Neste trabalho, o conceito de ordem foi aplicado às cadeias laterais e os 20 aminoácidos padrão foram

descritos.

As distâncias fornecidas por RMN, que em geral são referentes a átomos de hidrogênio, podem não ser suficientes para se determinar a estrutura de uma proteína. Porém, com a ordem atômica estabelecida para os aminoácidos padrão e, sabendo-se que é possível se obter dados sobre distâncias e ângulos de ligação, um algoritmo capaz de gerar um conjunto de distâncias independentes daquelas fornecidas por RMN pôde ser criado. Este algoritmo, chamado de *i*Gen (Instance Generation) e proposto neste trabalho, se baseia em um mapeamento das ligações covalentes de cada aminoácido para calcular todas as distâncias necessárias para a execução do *i*BP.

Através da combinação do *i*Gen e do *i*BP, foram feitos experimentos onde se constatou a influência da quantidade de hidrogênios das cadeias laterais na redução do número de soluções obtidas pelo *i*BP. Também, foi possível observar o desempenho do *i*BP na obtenção de soluções para sequências de 4 aminoácidos e na obtenção de uma solução para sequências longas. Neste contexto, conclui-se que, com esses algoritmos, é possível se obter um grande número de conformações viáveis em um pequeno intervalo de tempo. É necessário, então, no que se refere a trabalhos futuros, direcionar atenção à qualidade das soluções, evitando-se soluções repetidas ou quimicamente inviáveis. Para tal, pesquisas para o desenvolvimento de testes de viabilidade (pruning tests) para o *i*BP podem ser feitas, utilizando, por exemplo, mais dados bioquímicos sobre a estrutura molecular de proteínas. Além disso, é necessário o aprimoramento do programa *i*Gen para que seja utilizado um banco de dados com comprimentos reais de ligações covalentes e com ângulos reais de ligação.

Referências Bibliográficas

[1] LAVOR, C., LIBERTI, L., MUCHERINO, A. "The *interval* Branch-and-Prune algorithm for the discretizable molecular distance geometry problem with inexact distances", *Journal of Global Optimization*, pp. 1–17, 2011. ISSN: 0925-5001. Disponível em: <http://dx.doi.org/10.1007/s10898-011-9799-6>. 10.1007/s10898-011-9799-6.

[2] LAVOR, C., LIBERTI, L., MUCHERINO, A. "The *i*BP algorithm for the discretizable molecular distance geometry problem with interval data", *Optimization Online*, January 2011. Disponível em: <http://www.optimization-online.org/DB_HTML/2011/01/2889.html>.

[3] LAVOR, C., LEE, J., LEE-ST. JOHN, A., et al. "Discretization orders for distance geometry problems.(Report)", *Optimization Letters*, v. 6, n. 4, pp. 783(14), 2012-04-01.

[4] ABRAGAM, A. "The principles of nuclear magnetism (Clarendon Press: Oxford University Press, London, 1961. xvi-599 p. 84 s.)", *Nuclear Physics*, v. 28, n. 1, pp. 692 – 693, 1961. ISSN: 0029-5582. doi: DOI:10.1016/0029-5582(61)91110-5. Disponível em: <http://www.sciencedirect.com/science/article/B73DR-4D5P5V1-2P/2/b4bddba3ffb44fb1e2f8baf230231980>.

[5] WÜTHRICH, K. *NMR of Proteins and Nucleic Acids*. New York, John Wiley & Sons, 1986.

[6] GÜNTERT, P. "Structure calculation of biological macromolecules from NMR data", *Quarterly Reviews of Biophysics*, v. 31, n. 02, pp. 145–237, 1998. doi: null. Disponível em: <http://dx.doi.org/10.1017/S0033583598003436>.

[7] CRIPPEN, G. M., HAVEL, T. F. "Distance geometry and molecular conformation", *Research Studies Press, Journal of Computational Chemistry*, v. 11, n. 2, pp. 265–266, 1988-1990. ISSN: 1096-987X. doi:

10.1002/jcc.540110212. Disponível em: <http://dx.doi.org/10.1002/jcc.540110212>.

[8] LIBERTI, L., TSIAKIS, P., KEEPING, B., et al. *ooOPS*. Relatório técnico, Centre for Process Systems Engineering, Chemical Engineering Department, Imperial College, London, UK, 2001.

[9] ZASLAVSKI, A., LIBERTI, L. "Writing Global Optimization Software". In: Pardalos, P., Liberti, L., Maculan, N. (Eds.), *Global Optimization*, v. 84, *Nonconvex Optimization and Its Applications*, Springer US, pp. 211–262, 2006. ISBN: 978-0-387-30528-8. Disponível em: <http://dx.doi.org/10.1007/0-387-30528-9_8>. 10.1007/0-387-30528-9_8.

[10] HANSEN, P., MLADENOVIC, N. "Variable neighborhood search: Principles and applications", *European Journal of Operational Research*, v. 130, n. 3, pp. 449 – 467, 2001. ISSN: 0377-2217. doi: DOI:10.1016/S0377-2217(00)00100-4. Disponível em: <http://www.sciencedirect.com/science/article/B6VCT-42FS7FG-1/2/f5892eb143b79d617667bb98eee4e88e>.

[11] LIBERTI, L., LAVOR, C., MACULAN, N., et al. "Double variable neighbourhood search with smoothing for the molecular distance geometry problem", *J. of Global Optimization*, v. 43, pp. 207–218, March 2009. ISSN: 0925-5001. doi: 10.1007/s10898-007-9218-1. Disponível em: <http://portal.acm.org/citation.cfm?id=1502593.1502607>.

[12] KUCHERENKO, S., SYTSKO, Y. "Application of Deterministic Low-Discrepancy Sequences in Global Optimization", *Comput. Optim. Appl.*, v. 30, pp. 297–318, March 2005. ISSN: 0926-6003. doi: 10.1007/s10589-005-4615-1. Disponível em: <http://portal.acm.org/citation.cfm?id=1061896.1061903>.

[13] HOAI AN, L. T. "Solving Large Scale Molecular Distance Geometry Problems by a Smoothing Technique via the Gaussian Transform and D.C. Programming", *J. of Global Optimization*, v. 27, pp. 375–397, December 2003. ISSN: 0925-5001. doi: 10.1023/A:1026016804633. Disponível em: <http://portal.acm.org/citation.cfm?id=945565.945567>.

[14] HOAI AN, L. T., TAO, P. D. "Large-Scale Molecular Optimization from Distance Matrices by a D. C. Optimization Approach", *SIAM J. on Optimization*, v. 14, pp. 77–114, January 2003. ISSN: 1052-6234. doi: 10.1137/S1052623498342794. Disponível em: <http://portal.acm.org/citation.cfm?id=782134.942242>.

[15] REAMS, R., CHATHAM, G., GLUNT, W., et al. "Determining protein structure using the distance geometry program APA", *Computers & Chemistry*, v. 23, n. 2, pp. 153 – 163, 1999. ISSN: 0097-8485. doi: DOI:10.1016/S0097-8485(99)00003-0. Disponível em: <http://www.sciencedirect.com/science/article/B6TFV-3WH670J-6/2/0451b07131857fa0d93e7c9b65865263>.

[16] DONG, Q., WU, Z. "A Geometric Build-Up Algorithm for Solving the Molecular Distance Geometry Problem with Sparse Distance Data", *J. of Global Optimization*, v. 26, pp. 321–333, July 2003. ISSN: 0925-5001. doi: 10.1023/A:1023221624213. Disponível em: <http://portal.acm.org/citation.cfm?id=762830.762844>.

[17] WILLIAMS, G. A., DUGAN, J. M., ALTMAN, R. B. "Constrained Global Optimization for Estimating Molecular Structure from Atomic Distances", *Journal of Computational Biology*, v. 8, pp. 2001, 2001.

[18] GROSSO, A., LOCATELLI, M., SCHOEN, F. "Solving molecular distance geometry problems by global optimization algorithms", *Computational Optimization and Applications*, v. 43, pp. 23–37, 2009. ISSN: 0926-6003. Disponível em: <http://dx.doi.org/10.1007/s10589-007-9127-8>. 10.1007/s10589-007-9127-8.

[19] BISWAS, P., TOH, K.-C., YE, Y. "A Distributed SDP Approach for Large-Scale Noisy Anchor-Free Graph Realization with Applications to Molecular Conformation", *SIAM Journal on Scientific Computing*, v. 30, pp. 1251–1277, March 21, 2008. ISSN: 1064-8275. doi: http://dx.doi.org/10.1137/05062754X. Disponível em: <http://epubs.siam.org/sisc/resource/1/sjoce3/v30/i3/p1251_s1?isAuthorized=no>.

[20] XU, H., IZRAILEV, S., AGRAFIOTIS, D. K. "Conformational Sampling by Self-Organization", *Journal of Chemical Information and Computer Sciences*, v. 43, pp. 1186–1191, 2003.

[21] HENDRICKSON, B. "The Molecule Problem: Exploiting Structure In Global Optimization", *SIAM Journal on Optimization*, v. 5, pp. 835–857, 1995.

[22] LIBERTI, L., LAVOR, C., MACULAN, N. *A Branch-and-Prune Algorithm for the Molecular Distance Geometry Problem.* Technical report q-bio/0608012, arXiv, 2006.

[23] LIBERTI, L., LAVOR, C., MACULAN, N. "A Branch-and-Prune Algorithm for the Molecular Distance Geometry Problem", *International Transactions in Operational Research*, v. 15, pp. 1–17, 2008.

[24] BERMAN, H. M., WESTBROOK, J., FENG, Z., et al. "The Protein Data Bank", *Nucleic Acids Research*, v. 28, n. 1, pp. 235–242, January 2000. Disponível em: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC102472/>.

[25] FRUTON, J. S. "Molecules and Life – Historical Essays on the Interplay of Chemistry and Biology", *Wiley-Interscience*, v. 18, n. 4, pp. 471–472, 1972 (1974, por W. Rödel). ISSN: 1521-3803. doi: 10.1002/food.19740180423. Disponível em: <http://dx.doi.org/10.1002/food.19740180423>.

[26] SANGER, F. "The arrangement of amino acids in proteins", *Adv. Protein Chem.*, v. 7, pp. 1–67, 1952.

[27] TAYLOR, W. R. "The classification of amino acid conservation", *Journal of Theoretical Biology*, v. 119, n. 2, pp. 205 – 218, 1986. ISSN: 0022-5193. doi: DOI:10.1016/S0022-5193(86)80075-3. Disponível em: <http://www.sciencedirect.com/science/article/B6WMD-4KDGR7F-8/2/ee71c08df50de12adb74946aa15f7c3e>.

[28] CAMPBELL, M. K., FARRELL, S. O. *Biochemistry*. Brooks Cole, 2007.

[29] PAULING, L. *The Nature of the Chemical Bond*. Cornell University Press, Ithaca, New York, 1960.

[30] RAMACHANDRAN, G., SASISEKHARAN, V. "Conformation of Polypeptides and Proteins", *Advances in Protein Chemistry*, v. 23, pp. 283 – 437, 1968. ISSN: 0065-3233. doi: DOI:10.1016/S0065-3233(08)60402-7. Disponível em: <http://www.sciencedirect.com/science/article/B7CTK-4S987VX-B/2/eb14612bba5a8774eacf4cd6e2763241>.

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

[32] LEHNINGER, A., NELSON, D. L., COX, M. M. *Lehninger Principles of Biochemistry*. W. H. Freeman, Jun 2008. Disponível em: <http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20%255C&path=ASIN/1429224169>.

[33] DONG, Q., WU, Z. "A linear-time algorithm for solving the molecular distance geometry problem with exact inter-atomic distances", *Journal of Global Optimization*, v. 22, pp. 365–375, 2002. ISSN: 0925-5001. Disponível em: <http://dx.doi.org/10.1023/A:1013857218127>. 10.1023/A:1013857218127.

[34] LIBERTI, L., LAVOR, C., MUCHERINO, A., et al. "Molecular distance geometry methods: from continuous to discrete", *International Transactions in Operational Research*, v. 18, n. 1, pp. 33–51, 2010. ISSN: 1475-3995. doi: 10.1111/j.1475-3995.2009.00757.x. Disponível em: <http://dx.doi.org/10.1111/j.1475-3995.2009.00757.x>.

[35] GOLUB, G. H., VAN LOAN, C. F. *Matrix computations (3rd ed.)*. Baltimore, MD, USA, Johns Hopkins University Press, 1996. ISBN: 0-8018-5414-8.

[36] TREFETHEN, L. N., BAU, D. *Numerical Linear Algebra*. SIAM: Society for Industrial and Applied Mathematics, jun. 1997. ISBN: 0898713617. Disponível em: <http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0898713617>.

[37] DAVIS, R., ERNST, C., WU, D. "Protein structure determination via an efficient geometric build-up algorithm", *BMC Structural Biology*, v. 10, n. Suppl 1, pp. S7, 2010. ISSN: 1472-6807. doi: 10.1186/1472-6807-10-S1-S7. Disponível em: <http://www.biomedcentral.com/1472-6807/10/S1/S7>.

[38] DAVIS, R. T. *Geometric Build-up Solutions for Protein Determination via Distance Geometry*. Master theses & specialist projects, Western Kentucky University, 2009. Disponível em: <http://digitalcommons.wku.edu/theses/102>. Paper 102.

[39] WU, D., WU, Z. "An updated geometric build-up algorithm for solving the molecular distance geometry problems with sparse distance data", *Journal of Global Optimization*, v. 37, pp. 661–673, 2007. ISSN: 0925-5001. Disponível em: <http://dx.doi.org/10.1007/s10898-006-9080-6>. 10.1007/s10898-006-9080-6.

[40] LAVOR, C., LIBERTI, L., MACULAN, N., et al. "The discretizable molecular distance geometry problem", *Computational Optimization and Applications*, v. 52, pp. 115–146, 2012. ISSN: 0926-6003. doi: 10.1007/s10589-011-9402-6. Disponível em: <http://dx.doi.org/10.1007/s10589-011-9402-6>.

[41] MUCHERINO, A., LAVOR, C., MALLIAVIN, T., et al. "Influence of Pruning Devices on the Solution of Molecular Distance Geometry Problems". In: Pardalos, P., Rebennack, S. (Eds.), *Experimental Algorithms*, v. 6630, *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 206–217, 2011. ISBN: 978-3-642-20661-0. doi: 10.1007/978-3-642-20662-7_18. Disponível em: <http://dx.doi.org/10.1007/978-3-642-20662-7_18>.

[42] SEMPLE, J. G., KNEEBONE, G. T. Algebraic Projective Geometry. Oxford University Press, London, 1952.

[43] ENGH, R. A., HUBER, R. "Accurate bond and angle parameters for X-ray protein structure refinement", *Acta Crystallographica Section A*, v. 47, n. 4, pp. 392–400, Jul 1991. doi: 10.1107/S0108767391001071. Disponível em: <http://dx.doi.org/10.1107/S0108767391001071>.

[44] SCHLICK, T. *Molecular Modeling and Simulation: An Interdisciplinary Guide*. Secaucus, NJ, USA, Springer-Verlag New York, Inc., 2002. ISBN: 038795404X.

Apêndice A

Problema Geométrico da Distância Molecular

Teorema A.1 (Decomposição em Valores Singulares [35])**.** *Se A é uma matriz real $m \times n$, então existem matrizes ortogonais*

$$U = [u_1, \ldots, u_m] \in \mathbb{R}^{m\times m} \quad e \quad V = [v_1, \ldots, v_n] \in \mathbb{R}^{n\times n}$$

tais que

$$U^T A V = \begin{bmatrix} \Sigma \\ \mathbf{0} \end{bmatrix} \in \mathbb{R}^{m\times n},$$

onde $\Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_p)$ é a matriz diagonal formada pelos valores singulares $\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_p \geqslant 0$ de A, $p = \min\{m, n\}$, e $\mathbf{0}$ é a matriz nula $(m-p) \times n$ (veja [35] para maiores detalhes).

Teorema A.2 (Decomposição Reduzida em Valores Singulares [35])**.** *Se $A = U\Sigma V^T \in \mathbb{R}^{m\times n}$ $(m \geqslant n)$ é a decomposição em valores singulares de A, conforme o teorema A.1, então*

$$A = U_1 \Sigma_1 V^T$$

onde

$$U_1 = U(:, 1:n) = [u_1 \ u_2 \ \ldots \ u_n] \in \mathbb{R}^{m\times n}$$

e

$$\Sigma_1 = \Sigma(1:n, 1:n) = \operatorname{diag}(\sigma_1, \ldots, \sigma_n) \in \mathbb{R}^{n\times n}.$$

Apêndice B

Descrição Matemática do Algoritmo Branch-and-Prune

B.1 Matriz homogênea de translação em $\mathbb{R}^3$

Seja um vetor qualquer $x_0 = (x_{01}\ x_{02}\ x_{03})^T \in \mathbb{R}^3$ e sua representação em coordenadas homogêneas

$$x_0^h = (x_{01}\ x_{02}\ x_{03}\ w)^T = (x_{01}\ x_{02}\ x_{03}\ 1)^T \in \mathbb{R}^4, \text{ com } w = 1.$$

Multiplicando a matriz

$$T_i = \begin{pmatrix} 1 & 0 & 0 & d_{i-1,i} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

por x_0^h, têm-se

$$T_i x_0^h = \begin{pmatrix} 1 & 0 & 0 & d_{i-1,i} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{01} \\ x_{02} \\ x_{03} \\ 1 \end{pmatrix} = \begin{pmatrix} x_{01} + d_{i-1,i} \\ x_{02} \\ x_{03} \\ 1 \end{pmatrix},$$

que é a representação em coordenadas homogêneas do vetor $(x_{01} + d_{i-1,i}\ \ x_{02}\ \ x_{03}\ \ 1)^T \in \mathbb{R}^3$ que, por sua vez, é uma translação de x_0 por $(d_{i-1,i}\ 0\ 0)^T$.

B.2 Rotação em $\mathbb{R}^3$ de um ângulo $\pi - \theta_{i-2,i}$

Sabe-se que uma rotação de $\pi - \theta_{i-2,i}$ em torno do eixo z em $\mathbb{R}^3$ é dada por

$$R_z(\pi - \theta_{i-2,i}) = \begin{pmatrix} \cos(\pi - \theta_{i-2,i}) & -\operatorname{sen}(\pi - \theta_{i-2,i}) & 0 \\ \operatorname{sen}(\pi - \theta_{i-2,i}) & \cos(\pi - \theta_{i-2,i}) & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Porém, considerando-se a seção 5.3.1, itens de 1 a 4 da página 57, e a figura 5.5b, tem-se que o eixo z é representado pelo vetor normal $\eta_{i-2,i}$. Assim, $R_z(\pi - \theta_{i-2,i}) = R_{\eta_{i-2,i}}(\pi - \theta_{i-2,i}) = R$, simplificando a notação.

A matriz homogênea

$$R_{\pi-\theta_{i-2,i},\eta_{i-2,i}} = \begin{pmatrix} \cos(\pi - \theta_{i-2,i}) & -\operatorname{sen}(\pi - \theta_{i-2,i}) & 0 & 0 \\ \operatorname{sen}(\pi - \theta_{i-2,i}) & \cos(\pi - \theta_{i-2,i}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

pode ser reescrita como

$$R_{\pi-\theta_{i-2,i},\eta_{i-2,i}} = \begin{pmatrix} R & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix}.$$

Seja o vetor $x_0 = (x_{01}\ x_{02}\ x_{03})^T$, respeitando os eixos aqui considerados, e sua representação em coordenadas homogêneas $x_0^h = (x_{01}\ x_{02}\ x_{03}\ 1)^T = (x_0\ 1)^T$, tem-se

$$R_{\pi-\theta_{i-2,i},\eta_{i-2,i}} x_0^h = \begin{pmatrix} R & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \begin{pmatrix} x_0 \\ 1 \end{pmatrix} = \begin{pmatrix} Rx_0 \\ 1 \end{pmatrix}$$

que é a representação em coordenadas homogêneas de Rx_0 que, por sua vez, é o vetor x_0 rotacionado de $\pi - \theta_{i-2,i}$ em torno de $\eta_{i-2,i}$.

B.3 Rotação em $\mathbb{R}^3$ de um ângulo $\omega_{i-3,i}$

Análogo ao item B.2, porém, neste caso, tem-se uma rotação em torno do eixo x, que, segundo os eixos descritos na seção 5.3.1, itens de 1 a 4 da página 57, tem sua direção dada por $v_{i-2,i-1}$.

Apêndice C

Propriedades de Vetores Tridimensionais

Neste capítulo, serão apresentadas algumas propriedades de produtos vetoriais utilizadas nas demonstrações.

Proposição C.1. *O produto vetorial $u \times v$, onde $u = (u_1, u_2, u_3)$ e $v = (v_1, v_2, v_3)$ pode ser representado por um produto de matriz por vetor do tipo $M_u v$ onde*

$$M_u = \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix}.$$

Demonstração. Pela multiplicação da matriz pelo vetor, chega-se ao resultado. □

Corolário C.1. *Sejam e_i, $i = 1, 2, 3$, os vetores da base canônica de $\mathbb{R}^3$, então a matriz M_u associada ao produto vetorial $u \times v$ pode ser decomposta como:*

$$u_1 \begin{bmatrix} 0 \\ -e_3^T \\ e_2^t \end{bmatrix} + u_2 \begin{bmatrix} e_3^T \\ 0 \\ -e_1^t \end{bmatrix} + u_3 \begin{bmatrix} -e_2^T \\ e_1^t \\ 0 \end{bmatrix},$$

neste caso, 0 é uma matriz 1×3 formada por zeros.

Proposição C.2. *O produto interno de dois produtos vetoriais entre quatro vetores distintos de $\mathbb{R}^3$ atende a:*

$$\Big((u \times w), (y \times v)\Big) = (u, y)(v, w) - (u, v)(y, w). \tag{C.1}$$

Demonstração. Usando a versão matricial do produto vetorial, tem-se

$$\Big[(u\times w),(y\times v)\Big]=\Big[M_u w\Big]^T M_y v = w^T M_u^T M_y v =$$
$$= w^T \begin{bmatrix} 0 & u_3 & -u_2 \\ -u_3 & 0 & u_1 \\ u_2 & -u_1 & 0 \end{bmatrix} \begin{bmatrix} 0 & -y_3 & y_2 \\ y_3 & 0 & -y_1 \\ -y_2 & y_1 & 0 \end{bmatrix} v =$$
$$= w^T \begin{bmatrix} y_2u_2+y_3u_3 & -y_1u_2 & y_1u_3 \\ -y_2u_1 & y_1u_1+y_3u_3 & -y_2u_3 \\ -y_3u_1 & -y_3u_2 & y_1u_1+y_2u_2 \end{bmatrix} v =$$
$$= w^T \begin{bmatrix} -y_1u_1+y_1u_1+y_2u_2+y_3u_3 & -y_1u_2 & y_1u_3 \\ -y_2u_1 & y_1u_1+y_2u_2-y_2u_2+y_3u_3 & -y_2u_3 \\ -y_3u_1 & -y_3u_2 & y_1u_1+y_2u_2+y_3u_3-y_3u_3 \end{bmatrix} v =$$
$$= w^T\Big[(-y)u^T+u^T y I_3\Big]v = -w^T y u^T v + u^T y w^T v = (u,y)(v,w)-(u,v)(y,w),$$

onde I_3 é a matriz identidade de ordem 3. □

Os seguintes corolários são úteis em demonstrações.

Corolário C.2. $\Big((u\times w),(u\times v)\Big)=\|u\|_2^2(v,w)-(u,v)(u,w).$

Corolário C.3. $\Big((u\times w),(u\times w)\Big)=\|u\|_2^2\|w\|_2^2-(u,w)^2.$

Proposição C.3 (Fórmula de Grassmann)**.** *Sejam os vetores u, v e w em $\mathbb{R}^3$ então*

$$u\times(v\times w)=(vw^T-wv^T)u,$$

onde as multiplicações são multiplicações matriciais usuais.

Demonstração. $v\times w=\Big[v_2w_3-v_3w_2,\ v_3w_1-v_1w_3,\ v_1w_2-v_2w_1\Big]$, logo aplicando o

corolário C.1 para $u \times (v \times w)$, tem-se

$$\left[u_1 \begin{bmatrix} 0 \\ -e_3^T \\ e_2^t \end{bmatrix} + u_2 \begin{bmatrix} e_3^T \\ 0 \\ -e_1^t \end{bmatrix} + u_3 \begin{bmatrix} -e_2^T \\ e_1^t \\ 0 \end{bmatrix} \right] \begin{bmatrix} v_2w_3 - v_3w_2 \\ v_3w_1 - v_1w_3 \\ v_1w_2 - v_2w_1 \end{bmatrix} =$$

$$= u_1 \begin{bmatrix} 0 \\ -(v_1w_2 - v_2w_1) \\ v_3w_1 - v_1w_3 \end{bmatrix} + u_2 \begin{bmatrix} v_1w_2 - v_2w_1 \\ 0 \\ -(v_2w_3 - v_3w_2) \end{bmatrix} + u_3 \begin{bmatrix} -(v_3w_1 - v_1w_3) \\ v_2w_3 - v_3w_2 \\ 0 \end{bmatrix} =$$

$$= \begin{bmatrix} 0 & v_1w_2 - v_2w_1 & -(v_3w_1 - v_1w_3) \\ -(v_1w_2 - v_2w_1) & 0 & v_2w_3 - v_3w_2 \\ v_3w_1 - v_1w_3 & -(v_2w_3 - v_3w_2) & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} =$$

$$= \begin{bmatrix} v_1w_1 - v_1w_1 & v_1w_2 - v_2w_1 & v_1w_3 - v_3w_1 \\ v_2w_1 - v_1w_2 & v_2w_2 - v_2w_2 & v_2w_3 - v_3w_2 \\ v_3w_1 - v_1w_3 & v_3w_2 - v_2w_3 & v_3w_3 - v_3w_3 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} =$$

$$= \left[\begin{bmatrix} v_1w_1 & v_1w_2 & v_1w_3 \\ v_2w_1 & v_2w_2 & v_2w_3 \\ v_3w_1 & v_3w_2 & v_3w_3 \end{bmatrix} - \begin{bmatrix} v_1w_1 & v_2w_1 & v_3w_1 \\ v_1w_2 & v_2w_2 & v_3w_2 \\ v_1w_3 & v_2w_3 & v_3w_3 \end{bmatrix} \right] \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} =$$

$$= (vw^T - wv^T)u$$

□

Apêndice D

Ordenação para os Átomos das Cadeias Laterais de Proteínas

$$
\begin{aligned}
r^1_{ALA} &= \{N^1, H^{1,1}_N, H^{1,2}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, H^{1,2}_\beta, C^1_\beta, H^{1,3}_\beta, C^1_\alpha, C^1_\beta, C^1, C^1_\alpha\}.\\
r^i_{ALA} &= \{N^1, H^{1,1}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, H^{1,2}_\beta, C^1_\beta, H^{1,3}_\beta, C^1_\alpha, C^1_\beta, C^1, C^1_\alpha\}.\\
r^p_{ALA} &= \{N^1, H^{1,1}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, H^{1,2}_\beta, C^1_\beta, H^{1,3}_\beta, C^1_\alpha, C^1_\beta, C^1, C^1_\alpha,\\
&\quad O^1, C^1, O^2\}.
\end{aligned}
$$

$$
\begin{aligned}
r^1_{PRO} &= \{N^1, H^{1,1}_N, C^1_\delta, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, H^{1,2}_\beta, C^1_\gamma, C^1_\beta, H^{1,1}_\gamma, H^{1,2}_\gamma, C^1_\gamma, C^1_\delta,\\
&\quad H^{1,1}_\delta, H^{1,2}_\delta, N^1, C^1_\delta, C^1_\alpha, N^1, C^1, H^{1,1}_\alpha, C^1_\alpha, C^1\}.\\
r^i_{PRO} &= \{N^1, H^{1,1}_N, C^1_\delta, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, H^{1,2}_\beta, C^1_\gamma, C^1_\beta, H^{1,1}_\gamma, H^{1,2}_\gamma, C^1_\gamma, C^1_\delta,\\
&\quad H^{1,1}_\delta, H^{1,2}_\delta, N^1, C^1_\delta, C^1_\alpha, N^1, C^1, H^{1,1}_\alpha, C^1_\alpha, C^1\}.\\
r^p_{PRO} &= \{N^1, H^{1,1}_N, C^1_\delta, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, H^{1,2}_\beta, C^1_\gamma, C^1_\beta, H^{1,1}_\gamma, H^{1,2}_\gamma, C^1_\gamma, C^1_\delta,\\
&\quad H^{1,1}_\delta, H^{1,2}_\delta, N^1, C^1_\delta, C^1_\alpha, N^1, C^1, H^{1,1}_\alpha, C^1_\alpha, C^1, O^1, C^1, O^2\}.
\end{aligned}
$$

$$
\begin{aligned}
r^1_{GLY} &= \{N^1, H^{1,1}_N, H^{1,2}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, H^{1,2}_\alpha, C^1, C^1_\alpha\}.\\
r^i_{GLY} &= \{N^1, H^{1,1}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, H^{1,2}_\alpha, C^1, C^1_\alpha\}.\\
r^p_{GLY} &= \{N^1, H^{1,1}_N, H^{1,2}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, H^{1,2}_\alpha, C^1, C^1_\alpha, O^1, C^1, O^2\}.
\end{aligned}
$$

$$
\begin{aligned}
r^1_{LEU} &= \{N^1, H^{1,1}_N, H^{1,2}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, C^1_\gamma, C^1_\beta, H^{1,1}_\gamma, C^1_\gamma, C^1_\delta, H^{1,1}_\delta, H^{1,2}_\delta, \\
&\quad C^1_\delta, H^{1,3}_\delta, C^1_\gamma, C^1_\delta, C^2_\delta, C^1_\gamma, H^{2,1}_\delta, H^{2,2}_\delta, H^{2,3}_\delta, C^2_\delta, C^1_\gamma, C^1_\beta, H^{1,2}_\beta, C^1_\alpha, C^1_\beta, C^1, C^1_\alpha\}. \\
r^i_{LEU} &= \{N^1, H^{1,1}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, C^1_\gamma, C^1_\beta, H^{1,1}_\gamma, C^1_\gamma, C^1_\delta, H^{1,1}_\delta, H^{1,2}_\delta, \\
&\quad C^1_\delta, H^{1,3}_\delta, C^1_\gamma, C^1_\delta, C^2_\delta, C^1_\gamma, H^{2,1}_\delta, H^{2,2}_\delta, H^{2,3}_\delta, C^2_\delta, C^1_\gamma, C^1_\beta, H^{1,2}_\beta, C^1_\alpha, C^1_\beta, C^1, C^1_\alpha\}. \\
r^p_{LEU} &= \{N^1, H^{1,1}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, C^1_\gamma, C^1_\beta, H^{1,1}_\gamma, C^1_\gamma, C^1_\delta, H^{1,1}_\delta, H^{1,2}_\delta, \\
&\quad C^1_\delta, H^{1,3}_\delta, C^1_\gamma, C^1_\delta, C^2_\delta, C^1_\gamma, H^{2,1}_\delta, H^{2,2}_\delta, H^{2,3}_\delta, C^2_\delta, C^1_\gamma, C^1_\beta, H^{1,2}_\beta, C^1_\alpha, C^1_\beta, C^1, C^1_\alpha, \\
&\quad O^1, C^1, O^2\}.
\end{aligned}
$$

$$
\begin{aligned}
r^1_{VAL} &= \{N^1, H^{1,1}_N, H^{1,2}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, C^1_\gamma, C^1_\beta, H^{1,1}_\gamma, C^1_\gamma, H^{1,2}_\gamma, H^{1,3}_\gamma, C^1_\gamma, \\
&\quad C^1_\beta, C^2_\gamma, H^{2,1}_\gamma, H^{2,2}_\gamma, H^{2,3}_\gamma, C^2_\gamma, C^1_\beta, C^1_\alpha, H^{1,1}_\alpha, C^1, C^1_\alpha\}. \\
r^i_{VAL} &= \{N^1, H^{1,1}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, C^1_\gamma, C^1_\beta, H^{1,1}_\gamma, C^1_\gamma, H^{1,2}_\gamma, H^{1,3}_\gamma, C^1_\gamma, \\
&\quad C^1_\beta, C^2_\gamma, H^{2,1}_\gamma, H^{2,2}_\gamma, H^{2,3}_\gamma, C^2_\gamma, C^1_\beta, C^1_\alpha, H^{1,1}_\alpha, C^1, C^1_\alpha\}. \\
r^p_{VAL} &= \{N^1, H^{1,1}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, C^1_\gamma, C^1_\beta, H^{1,1}_\gamma, C^1_\gamma, H^{1,2}_\gamma, H^{1,3}_\gamma, C^1_\gamma, \\
&\quad C^1_\beta, C^2_\gamma, H^{2,1}_\gamma, H^{2,2}_\gamma, H^{2,3}_\gamma, C^2_\gamma, C^1_\beta, C^1_\alpha, H^{1,1}_\alpha, C^1, C^1_\alpha, O^1, C^1, O^2\}.
\end{aligned}
$$

$$
\begin{aligned}
r^1_{ILE} &= \{N^1, H^{1,1}_N, H^{1,2}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, C^1_\gamma, C^1_\beta, H^{1,1}_\gamma, C^1_\gamma, C^1_\delta, H^{1,1}_\delta, H^{1,2}_\delta, \\
&\quad C^1_\delta, H^{1,3}_\delta, C^1_\gamma, C^1_\delta, H^{1,2}_\gamma, C^1_\gamma, C^1_\beta, C^2_\gamma, H^{2,1}_\gamma, H^{2,2}_\gamma, H^{2,3}_\gamma, C^2_\gamma, C^1_\beta, C^1_\alpha, C^1, \\
&\quad H^{1,1}_\alpha, C^1_\alpha, C^1\}. \\
r^i_{ILE} &= \{N^1, H^{1,1}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, C^1_\gamma, C^1_\beta, H^{1,1}_\gamma, C^1_\gamma, C^1_\delta, H^{1,1}_\delta, H^{1,2}_\delta, \\
&\quad C^1_\delta, H^{1,3}_\delta, C^1_\gamma, C^1_\delta, H^{1,2}_\gamma, C^1_\gamma, C^1_\beta, C^2_\gamma, H^{2,1}_\gamma, H^{2,2}_\gamma, H^{2,3}_\gamma, C^2_\gamma, C^1_\beta, C^1_\alpha, C^1, \\
&\quad H^{1,1}_\alpha, C^1_\alpha, C^1\}. \\
r^p_{ILE} &= \{N^1, H^{1,1}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, C^1_\gamma, C^1_\beta, H^{1,1}_\gamma, C^1_\gamma, C^1_\delta, H^{1,1}_\delta, H^{1,2}_\delta, \\
&\quad C^1_\delta, H^{1,3}_\delta, C^1_\gamma, C^1_\delta, H^{1,2}_\gamma, C^1_\gamma, C^1_\beta, C^2_\gamma, H^{2,1}_\gamma, H^{2,2}_\gamma, H^{2,3}_\gamma, C^2_\gamma, C^1_\beta, C^1_\alpha, C^1, \\
&\quad H^{1,1}_\alpha, C^1_\alpha, C^1, O^1, C^1, O^2\}.
\end{aligned}
$$

$$
\begin{aligned}
r^1_{MET} &= \{N^1, H^{1,1}_N, H^{1,2}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, C^1_\gamma, C^1_\beta, H^{1,1}_\gamma, S^1, C^1_\gamma, C^1_\delta, S^1, H^{1,1}_\delta, \\
&\quad H^{1,2}_\delta, H^{1,3}_\delta, C^1_\delta, S^1, C^1_\gamma, H^{1,2}_\gamma, C^1_\beta, C^1_\gamma, H^{1,2}_\beta, C^1_\beta, C^1_\alpha, C^1, H^{1,1}_\alpha, C^1_\alpha, C^1\}. \\
r^i_{MET} &= \{N^1, H^{1,1}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, C^1_\gamma, C^1_\beta, H^{1,1}_\gamma, S^1, C^1_\gamma, C^1_\delta, S^1, H^{1,1}_\delta, \\
&\quad H^{1,2}_\delta, H^{1,3}_\delta, C^1_\delta, S^1, C^1_\gamma, H^{1,2}_\gamma, C^1_\beta, C^1_\gamma, H^{1,2}_\beta, C^1_\beta, C^1_\alpha, C^1, H^{1,1}_\alpha, C^1_\alpha, C^1\}. \\
r^p_{MET} &= \{N^1, H^{1,1}_N, H^{1,2}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, C^1_\gamma, C^1_\beta, H^{1,1}_\gamma, S^1, C^1_\gamma, C^1_\delta, S^1, H^{1,1}_\delta, \\
&\quad H^{1,2}_\delta, H^{1,3}_\delta, C^1_\delta, S^1, C^1_\gamma, H^{1,2}_\gamma, C^1_\beta, C^1_\gamma, H^{1,2}_\beta, C^1_\beta, C^1_\alpha, C^1, H^{1,1}_\alpha, C^1_\alpha, C^1, O^1, C^1, O^2\}.
\end{aligned}
$$

$$
\begin{aligned}
r^1_{PHE} &= \{N^1, H^{1,1}_N, H^{1,2}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, C^1_\gamma, C^1_\beta, C^1_\delta, C^1_\gamma, H^{1,1}_\delta, C^1_\varepsilon, C^1_\delta, H^{1,1}_\varepsilon, \\
&\quad C^1_\varphi, C^1_\varepsilon, H^{1,1}_\varphi, C^2_\varepsilon, C^1_\varphi, H^{2,1}_\varepsilon, C^2_\delta, C^2_\varepsilon, H^{2,1}_\delta, C^1_\gamma, C^2_\delta, C^1_\beta, C^1_\gamma, H^{1,2}_\beta, C^1_\beta, C^1_\alpha, \\
&\quad C^1, H^{1,1}_\alpha C^1_\alpha, C^1\}. \\
r^i_{PHE} &= \{N^1, H^{1,1}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, C^1_\gamma, C^1_\beta, C^1_\delta, C^1_\gamma, H^{1,1}_\delta, C^1_\varepsilon, C^1_\delta, H^{1,1}_\varepsilon, \\
&\quad C^1_\varphi, C^1_\varepsilon, H^{1,1}_\varphi, C^2_\varepsilon, C^1_\varphi, H^{2,1}_\varepsilon, C^2_\delta, C^2_\varepsilon, H^{2,1}_\delta, C^1_\gamma, C^2_\delta, C^1_\beta, C^1_\gamma, H^{1,2}_\beta, C^1_\beta, C^1_\alpha, \\
&\quad C^1, H^{1,1}_\alpha C^1_\alpha, C^1\}. \\
r^p_{PHE} &= \{N^1, H^{1,1}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, C^1_\gamma, C^1_\beta, C^1_\delta, C^1_\gamma, H^{1,1}_\delta, C^1_\varepsilon, C^1_\delta, H^{1,1}_\varepsilon, \\
&\quad C^1_\varphi, C^1_\varepsilon, H^{1,1}_\varphi, C^2_\varepsilon, C^1_\varphi, H^{2,1}_\varepsilon, C^2_\delta, C^2_\varepsilon, H^{2,1}_\delta, C^1_\gamma, C^2_\delta, C^1_\beta, C^1_\gamma, H^{1,2}_\beta, C^1_\beta, C^1_\alpha, \\
&\quad C^1, H^{1,1}_\alpha C^1_\alpha, C^1, C^1, O^1, C^1, O^2\}.
\end{aligned}
$$

$$
\begin{aligned}
r^1_{TRP} &= \{N^1, H^{1,1}_N, H^{1,2}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, C^1_\gamma, C^1_\beta, C^1_\delta, C^1_\gamma, H^{1,1}_\delta, C^1_\delta, N^2, C^1_\varepsilon, C^1_\varphi, \\
&\quad H^{1,1}_\varphi, C^1_\eta, C^1_\varphi, H^{1,1}_\eta, C^1_\eta, C^2_\varphi, H^{2,1}_\varphi, C^2_\varepsilon, C^2_\varphi, C^2_\delta, C^2_\varepsilon, C^1_\gamma, C^2_\delta, C^1_\beta, C^1_\gamma, H^{1,2}_\beta, C^1_\beta, C^1_\alpha, C^1, \\
&\quad H^{1,1}_\alpha, C^1_\alpha, C^1\}. \\
r^i_{TRP} &= \{N^1, H^{1,1}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, C^1_\gamma, C^1_\beta, C^1_\delta, C^1_\gamma, H^{1,1}_\delta, C^1_\delta, N^2, C^1_\varepsilon, C^1_\varphi, \\
&\quad H^{1,1}_\varphi, C^1_\eta, C^1_\varphi, H^{1,1}_\eta, C^1_\eta, C^2_\varphi, H^{2,1}_\varphi, C^2_\varepsilon, C^2_\varphi, C^2_\delta, C^2_\varepsilon, C^1_\gamma, C^2_\delta, C^1_\beta, C^1_\gamma, H^{1,2}_\beta, C^1_\beta, C^1_\alpha, C^1, \\
&\quad H^{1,1}_\alpha, C^1_\alpha, C^1\}. \\
r^p_{TRP} &= \{N^1, H^{1,1}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, C^1_\gamma, C^1_\beta, C^1_\delta, C^1_\gamma, H^{1,1}_\delta, C^1_\delta, N^2, C^1_\varepsilon, C^1_\varphi, \\
&\quad H^{1,1}_\varphi, C^1_\eta, C^1_\varphi, H^{1,1}_\eta, C^1_\eta, C^2_\varphi, H^{2,1}_\varphi, C^2_\varepsilon, C^2_\varphi, C^2_\delta, C^2_\varepsilon, C^1_\gamma, C^2_\delta, C^1_\beta, C^1_\gamma, H^{1,2}_\beta, C^1_\beta, C^1_\alpha, C^1, \\
&\quad H^{1,1}_\alpha, C^1_\alpha, C^1, O^1, C^1, O^2\}.
\end{aligned}
$$

$$
\begin{aligned}
r^1_{GLN} &= \{N^1, H^{1,1}_N, H^{1,2}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, C^1_\gamma, C^1_\beta, H^{1,1}_\gamma, C^1_\gamma, O^1, N^2, C^1_\delta, H^{2,1}_N, \\
&\quad H^{2,2}_N, N^2, C^1_\delta, C^1_\gamma, H^{1,2}_\gamma, C^1_\beta, C^1_\gamma, H^{1,2}_\beta, C^1_\beta, C^1_\alpha, C^1, H^{1,1}_\alpha, C^1_\alpha, C^1\}. \\
r^i_{GLN} &= \{N^1, H^{1,1}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, C^1_\gamma, C^1_\beta, H^{1,1}_\gamma, C^1_\gamma, O^1, N^2, C^1_\delta, H^{2,1}_N, \\
&\quad H^{2,2}_N, N^2, C^1_\delta, C^1_\gamma, H^{1,2}_\gamma, C^1_\beta, C^1_\gamma, H^{1,2}_\beta, C^1_\beta, C^1_\alpha, C^1, H^{1,1}_\alpha, C^1_\alpha, C^1\}. \\
r^p_{GLN} &= \{N^1, H^{1,1}_N, H^{1,2}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, C^1_\gamma, C^1_\beta, H^{1,1}_\gamma, C^1_\gamma, O^1, N^2, C^1_\delta, H^{2,1}_N, \\
&\quad H^{2,2}_N, N^2, C^1_\delta, C^1_\gamma, H^{1,2}_\gamma, C^1_\beta, C^1_\gamma, H^{1,2}_\beta, C^1_\beta, C^1_\alpha, C^1, H^{1,1}_\alpha, C^1_\alpha, C^1, O^1, C^1, O^2\}.
\end{aligned}
$$

$$
\begin{aligned}
r^1_{TYR} &= \{N^1, H^{1,1}_N, H^{1,2}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, C^1_\gamma, C^1_\beta, C^1_\delta, C^1_\gamma, H^{1,1}_\delta, C^1_\varepsilon, C^1_\delta, H^{1,1}_\varepsilon, \\
&\quad C^1_\varphi, C^1_\varepsilon, O^1, C^1_\varphi, H^{1,1}_O, O^1, C^1_\varphi, C^2_\varepsilon, H^{2,1}_\varepsilon, C^2_\delta, C^2_\varepsilon, H^{2,1}_\delta, C^1_\gamma, C^2_\delta, C^1_\beta, C^1_\gamma, H^{1,2}_\beta, C^1_\beta, \\
&\quad C^1_\alpha, C^1, H^{1,1}_\alpha, C^1_\alpha, C^1\}. \\
r^i_{TYR} &= \{N^1, H^{1,1}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, C^1_\gamma, C^1_\beta, C^1_\delta, C^1_\gamma, H^{1,1}_\delta, C^1_\varepsilon, C^1_\delta, H^{1,1}_\varepsilon, \\
&\quad C^1_\varphi, C^1_\varepsilon, O^1, C^1_\varphi, H^{1,1}_O, O^1, C^1_\varphi, C^2_\varepsilon, H^{2,1}_\varepsilon, C^2_\delta, C^2_\varepsilon, H^{2,1}_\delta, C^1_\gamma, C^2_\delta, C^1_\beta, C^1_\gamma, H^{1,2}_\beta, C^1_\beta, \\
&\quad C^1_\alpha, C^1, H^{1,1}_\alpha, C^1_\alpha, C^1\}. \\
r^p_{TYR} &= \{N^1, H^{1,1}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, C^1_\gamma, C^1_\beta, C^1_\delta, C^1_\gamma, H^{1,1}_\delta, C^1_\varepsilon, C^1_\delta, H^{1,1}_\varepsilon, \\
&\quad C^1_\varphi, C^1_\varepsilon, O^1, C^1_\varphi, H^{1,1}_O, O^1, C^1_\varphi, C^2_\varepsilon, H^{2,1}_\varepsilon, C^2_\delta, C^2_\varepsilon, H^{2,1}_\delta, C^1_\gamma, C^2_\delta, C^1_\beta, C^1_\gamma, H^{1,2}_\beta, C^1_\beta, \\
&\quad C^1_\alpha, C^1, H^{1,1}_\alpha, C^1_\alpha, C^1, O^1, C^1, O^2\}.
\end{aligned}
$$

$$
\begin{aligned}
r^1_{ASN} &= \{N^1, H^{1,1}_N, H^{1,2}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, C^1_\gamma, C^1_\beta, O^1, N^2, C^1_\gamma, H^{2,1}_N, H^{2,2}_N, N^2, \\
&\quad C^1_\gamma, C^1_\beta, H^{1,2}_\beta, C^1_\alpha, C^1_\beta, C^1, C^1_\alpha\}. \\
r^i_{ASN} &= \{N^1, H^{1,1}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, C^1_\gamma, C^1_\beta, O^1, N^2, C^1_\gamma, H^{2,1}_N, H^{2,2}_N, N^2, \\
&\quad C^1_\gamma, C^1_\beta, H^{1,2}_\beta, C^1_\alpha, C^1_\beta, C^1, C^1_\alpha\}. \\
r^p_{ASN} &= \{N^1, H^{1,1}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, C^1_\gamma, C^1_\beta, O^1, N^2, C^1_\gamma, H^{2,1}_N, H^{2,2}_N, N^2, \\
&\quad C^1_\gamma, C^1_\beta, H^{1,2}_\beta, C^1_\alpha, C^1_\beta, C^1, C^1_\alpha, O^1, C^1, O^2\}.
\end{aligned}
$$

$$
\begin{aligned}
r^1_{SER} &= \{N^1, H^{1,1}_N, H^{1,2}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, O^1, C^1_\beta, H^{1,1}_O, O^1, C^1_\beta, H^{1,2}_\beta, C^1_\alpha, \\
&\quad C^1_\beta, C^1, C^1_\alpha\}. \\
r^i_{SER} &= \{N^1, H^{1,1}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, O^1, C^1_\beta, H^{1,1}_O, O^1, C^1_\beta, H^{1,2}_\beta, C^1_\alpha, \\
&\quad C^1_\beta, C^1, C^1_\alpha\}. \\
r^p_{SER} &= \{N^1, H^{1,1}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, O^1, C^1_\beta, H^{1,1}_O, O^1, C^1_\beta, H^{1,2}_\beta, C^1_\alpha, \\
&\quad C^1_\beta, C^1, C^1_\alpha, O^1, C^1, O^2\}.
\end{aligned}
$$

$$
\begin{aligned}
r^1_{CYS} &= \{N^1, H^{1,1}_N, H^{1,2}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, S^1, C^1_\beta, H^{1,1}_S, S^1, C^1_\beta, H^{1,2}_\beta, C^1_\alpha, \\
&\quad C^1_\beta, C^1, C^1_\alpha\}.
\end{aligned}
$$

$$
\begin{aligned}
r^i_{CYS} &= \{N^1, H^{1,1}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, S^1, C^1_\beta, H^{1,1}_S, S^1, C^1_\beta, H^{1,2}_\beta, C^1_\alpha, \\
&\quad C^1_\beta, C^1, C^1_\alpha\}.
\end{aligned}
$$

$$
\begin{aligned}
r^p_{CYS} &= \{N^1, H^{1,1}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, S^1, C^1_\beta, H^{1,1}_S, S^1, C^1_\beta, H^{1,2}_\beta, C^1_\alpha, \\
&\quad C^1_\beta, C^1, C^1_\alpha, O^1, C^1, O^2\}.
\end{aligned}
$$

$$
\begin{aligned}
r^1_{THR} &= \{N^1, H^{1,1}_N, H^{1,2}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, C^1_\gamma, C^1_\beta, H^{1,1}_\gamma, H^{1,2}_\gamma, C^1_\gamma, H^{1,3}_\gamma, C^1_\beta, C^1_\gamma, \\
&\quad O^1, C^1_\beta, H^{1,1}_O, O^1, C^1_\beta, C^1_\alpha, C^1, H^{1,1}_\alpha, C^1_\alpha, C^1\}.
\end{aligned}
$$

$$
\begin{aligned}
r^i_{THR} &= \{N^1, H^{1,1}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, C^1_\gamma, C^1_\beta, H^{1,1}_\gamma, H^{1,2}_\gamma, C^1_\gamma, H^{1,3}_\gamma, C^1_\beta, C^1_\gamma, \\
&\quad O^1, C^1_\beta, H^{1,1}_O, O^1, C^1_\beta, C^1_\alpha, C^1, H^{1,1}_\alpha, C^1_\alpha, C^1\}.
\end{aligned}
$$

$$
\begin{aligned}
r^p_{THR} &= \{N^1, H^{1,1}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, C^1_\gamma, C^1_\beta, H^{1,1}_\gamma, H^{1,2}_\gamma, C^1_\gamma, H^{1,3}_\gamma, C^1_\beta, C^1_\gamma, \\
&\quad O^1, C^1_\beta, H^{1,1}_O, O^1, C^1_\beta, C^1_\alpha, C^1, H^{1,1}_\alpha, C^1_\alpha, C^1, O^1, C^1, O^2\}.
\end{aligned}
$$

$$
\begin{aligned}
r^1_{ARG} &= \{N^1, H^{1,1}_N, H^{1,2}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, C^1_\gamma, C^1_\beta, H^{1,1}_\gamma, C^1_\gamma, C^1_\delta, H^{1,1}_\delta, H^{1,2}_\delta, C^1_\delta, \\
&\quad N^2, H^{2,1}_N, C^1_\varepsilon, N^2, N^3, C^1_\varepsilon, H^{3,1}_N, H^{3,2}_N, N^3, C^1_\varepsilon, N^4, H^{4,1}_N, H^{4,2}_N, N^4, C^1_\varepsilon, N^2, C^1_\delta, C^1_\gamma, \\
&\quad H^{1,2}_\gamma, C^1_\beta, C^1_\gamma, H^{1,2}_\beta, C^1_\beta, C^1_\alpha, C^1, H^{1,1}_\alpha, C^1_\alpha, C^1\}.
\end{aligned}
$$

$$
\begin{aligned}
r^i_{ARG} &= \{N^1, H^{1,1}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, C^1_\gamma, C^1_\beta, H^{1,1}_\gamma, C^1_\gamma, C^1_\delta, H^{1,1}_\delta, H^{1,2}_\delta, C^1_\delta, \\
&\quad N^2, H^{2,1}_N, C^1_\varepsilon, N^2, N^3, C^1_\varepsilon, H^{3,1}_N, H^{3,2}_N, N^3, C^1_\varepsilon, N^4, H^{4,1}_N, H^{4,2}_N, N^4, C^1_\varepsilon, N^2, C^1_\delta, C^1_\gamma, \\
&\quad H^{1,2}_\gamma, C^1_\beta, C^1_\gamma, H^{1,2}_\beta, C^1_\beta, C^1_\alpha, C^1, H^{1,1}_\alpha, C^1_\alpha, C^1\}.
\end{aligned}
$$

$$
\begin{aligned}
r^p_{ARG} &= \{N^1, H^{1,1}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, C^1_\gamma, C^1_\beta, H^{1,1}_\gamma, C^1_\gamma, C^1_\delta, H^{1,1}_\delta, H^{1,2}_\delta, C^1_\delta, \\
&\quad N^2, H^{2,1}_N, C^1_\varepsilon, N^2, N^3, C^1_\varepsilon, H^{3,1}_N, H^{3,2}_N, N^3, C^1_\varepsilon, N^4, H^{4,1}_N, H^{4,2}_N, N^4, C^1_\varepsilon, N^2, C^1_\delta, C^1_\gamma, \\
&\quad H^{1,2}_\gamma, C^1_\beta, C^1_\gamma, H^{1,2}_\beta, C^1_\beta, C^1_\alpha, C^1, H^{1,1}_\alpha, C^1_\alpha, C^1, O^1, C^1, O^2\}.
\end{aligned}
$$

$$
\begin{aligned}
r^1_{HIS} &= \{N^1, H^{1,1}_N, H^{1,2}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, C^1_\gamma, C^1_\beta, N^2, C^1_\gamma, H^{2,1}_N, C^1_\varepsilon, N^2, H^{1,1}_\varepsilon, N^3, \\
&\quad C^1_\varepsilon, C^1_\delta, N^3, H^{1,1}_\delta, C^1_\delta, C^1_\gamma, C^1_\beta, H^{1,2}_\beta, C^1_\alpha, C^1_\beta, C^1, C^1_\alpha\}.
\end{aligned}
$$

$$
\begin{aligned}
r^i_{HIS} &= \{N^1, H^{1,1}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, C^1_\gamma, C^1_\beta, N^2, C^1_\gamma, H^{2,1}_N, C^1_\varepsilon, N^2, H^{1,1}_\varepsilon, N^3, \\
&\quad C^1_\varepsilon, C^1_\delta, N^3, H^{1,1}_\delta, C^1_\delta, C^1_\gamma, C^1_\beta, H^{1,2}_\beta, C^1_\alpha, C^1_\beta, C^1, C^1_\alpha\}.
\end{aligned}
$$

$$
\begin{aligned}
r^p_{HIS} &= \{N^1, H^{1,1}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, C^1_\gamma, C^1_\beta, N^2, C^1_\gamma, H^{2,1}_N, C^1_\varepsilon, N^2, H^{1,1}_\varepsilon, N^3, \\
&\quad C^1_\varepsilon, C^1_\delta, N^3, H^{1,1}_\delta, C^1_\delta, C^1_\gamma, C^1_\beta, H^{1,2}_\beta, C^1_\alpha, C^1_\beta, C^1, C^1_\alpha, O^1, C^1, O^2\}.
\end{aligned}
$$

$$
\begin{aligned}
r^1_{LYS} &= \{N^1, H^{1,1}_N, H^{1,2}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, C^1_\gamma, C^1_\beta, H^{1,1}_\gamma, C^1_\gamma, C^1_\delta, H^{1,1}_\delta, C^1_\varepsilon, C^1_\delta, H^{1,1}_\varepsilon, \\
&\quad C^1_\varepsilon, N^2, H^{2,1}_N, H^{2,2}_N, N^2, H^{2,3}_N, C^1_\varepsilon, N^2, H^{1,2}_\varepsilon, C^1_\varepsilon, C^1_\delta, H^{1,2}_\delta, C^1_\gamma, C^1_\delta, H^{1,2}_\gamma, C^1_\gamma, C^1_\beta, H^{1,2}_\beta, \\
&\quad C^1_\alpha, C^1_\beta, C^1, C^1_\alpha\}. \\
r^i_{LYS} &= \{N^1, H^{1,1}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, C^1_\gamma, C^1_\beta, H^{1,1}_\gamma, C^1_\gamma, C^1_\delta, H^{1,1}_\delta, C^1_\varepsilon, C^1_\delta, H^{1,1}_\varepsilon, \\
&\quad C^1_\varepsilon, N^2, H^{2,1}_N, H^{2,2}_N, N^2, H^{2,3}_N, C^1_\varepsilon, N^2, H^{1,2}_\varepsilon, C^1_\varepsilon, C^1_\delta, H^{1,2}_\delta, C^1_\gamma, C^1_\delta, H^{1,2}_\gamma, C^1_\gamma, C^1_\beta, H^{1,2}_\beta, \\
&\quad C^1_\alpha, C^1_\beta, C^1, C^1_\alpha\}. \\
r^p_{LYS} &= \{N^1, H^{1,1}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, C^1_\gamma, C^1_\beta, H^{1,1}_\gamma, C^1_\gamma, C^1_\delta, H^{1,1}_\delta, C^1_\varepsilon, C^1_\delta, H^{1,1}_\varepsilon, \\
&\quad C^1_\varepsilon, N^2, H^{2,1}_N, H^{2,2}_N, N^2, H^{2,3}_N, C^1_\varepsilon, N^2, H^{1,2}_\varepsilon, C^1_\varepsilon, C^1_\delta, H^{1,2}_\delta, C^1_\gamma, C^1_\delta, H^{1,2}_\gamma, C^1_\gamma, C^1_\beta, H^{1,2}_\beta, \\
&\quad C^1_\alpha, C^1_\beta, C^1, C^1_\alpha, O^1, C^1, O^2\}.
\end{aligned}
$$

$$
\begin{aligned}
r^1_{GLU} &= \{N^1, H^{1,1}_N, H^{1,2}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, C^1_\gamma, C^1_\beta, H^{1,1}_\gamma, C^1_\gamma, C^1_\delta, O^1, O^2, C^1_\delta, C^1_\gamma, \\
&\quad H^{1,2}_\gamma, C^1_\beta, C^1_\gamma, H^{1,2}_\beta, C^1_\beta, C^1_\alpha, C^1\}. \\
r^i_{GLU} &= \{N^1, H^{1,1}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, C^1_\gamma, C^1_\beta, H^{1,1}_\gamma, C^1_\gamma, C^1_\delta, O^1, O^2, C^1_\delta, C^1_\gamma, \\
&\quad H^{1,2}_\gamma, C^1_\beta, C^1_\gamma, H^{1,2}_\beta, C^1_\beta, C^1_\alpha, C^1\}. \\
r^p_{GLU} &= \{N^1, H^{1,1}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, C^1_\gamma, C^1_\beta, H^{1,1}_\gamma, C^1_\gamma, C^1_\delta, O^1, O^2, C^1_\delta, C^1_\gamma, \\
&\quad H^{1,2}_\gamma, C^1_\beta, C^1_\gamma, H^{1,2}_\beta, C^1_\beta, C^1_\alpha, C^1, O^1, C^1, O^2\}.
\end{aligned}
$$

$$
\begin{aligned}
r^1_{ASP} &= \{N^1, H^{1,1}_N, H^{1,2}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, C^1_\gamma, C^1_\beta, O^1, O^2, C^1_\gamma, C^1_\beta, H^{1,2}_\beta, C^1_\alpha, C^1_\beta, \\
&\quad C^1, C^1_\alpha\}. \\
r^i_{ASP} &= \{N^1, H^{1,1}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, C^1_\gamma, C^1_\beta, O^1, O^2, C^1_\gamma, C^1_\beta, H^{1,2}_\beta, C^1_\alpha, C^1_\beta, \\
&\quad C^1, C^1_\alpha\}. \\
r^p_{ASP} &= \{N^1, H^{1,1}_N, C^1_\alpha, N^1, H^{1,1}_\alpha, C^1_\alpha, C^1_\beta, H^{1,1}_\beta, C^1_\gamma, C^1_\beta, O^1, O^2, C^1_\gamma, C^1_\beta, H^{1,2}_\beta, C^1_\alpha, C^1_\beta, \\
&\quad C^1, C^1_\alpha, O^1, C^1, O^2\}.
\end{aligned}
$$