Universidade Federal de Santa Catarina

Centro de Blumenau
Departamento de Matemática

PIBIC
Relatório Final

Geometria de Distâncias e Álgebras Geométricas: novas perspectivas geométricas, computacionais e aplicações

Geometria de Distâncias: uma aplicação na Geometria de Proteínas

Guilherme Philippi (g.philippi@grad.ufsc.br),

Orientador: Felipe Delfini Caetano Fidalgo (felipe.fidalgo@ufsc.br).

21 de agosto de 2019

Sumário

Abstract

In this paper, we study the Discretizable Molecular Distance Geometry Problem (DMDGP) applied to proteins, as well as the necessary tools for its comprehension, going from the graph theory to biomolecular structures. We, also, deal with some recent results on the ordering of a protein graph that composes the problem. The text concludes with a study of the algorithm described in the literature to solve the problem efficiently and a brief section of computer simulations.

Keywords: DMDGP, Distance geometry, Optimization.

Resumo

Neste trabalho, foram estudados o Discretizable Molecular Distance Geometry Problem (DMDGP) aplicado as proteínas, bem como as ferramentas necessárias para sua compreensão, passando da teoria de grafos às estruturas biomoleculares. Também lidamos com alguns resultados recentes sobre a ordenação do grafo da proteína que compõe o problema. O texto se encerra com um estudo sobre o algoritmo descrito na literatura para solucionar o problema de forma eficiente e uma brevê seção de simulações computacionais.

Palavras-chave: DMDGP, Geometria de Distâncias, Otimização.

1 Introdução

Existe uma relação muito forte entre a forma geométrica das moléculas orgânicas e suas funções em organismos vivos [9]. Outrora, em pesquisas sobre a molécula de DNA (ácido desoxirribonucleico), descobriu-se que essa era parte fundamental da produção de um dos pilares para a vida: a proteína. Esta é a estrutura básica que utilizamos para organizar nossas moléculas, gerando informação, ao possibilitarem um mecanismo funcional natural para a vida. Por exemplo, podemos citar o seu papel no transporte de oxigênio (hemoglobina), na proteção do corpo contra organismos patogênicos (imunoglobulina), com a catalização de reações químicas (apoenzima), além de outras inúmeras funções primordiais no nosso organismo [1].

Por conta dessa motivação tem-se esforços como o de Kurt Wüthrich, que propôs que utilizássemos experimentos de *Ressonância Magnética Nuclear* (RMN) para calcular a estrutura tridimensional de uma molécula de proteína (que lhe rendeu o premio Nobel da Química em 2002 [2]). Porém, a RMN não tem como resultado direto a estrutura tridimensional de uma proteína, mas sim distâncias entre átomos relativamente próximos que compõem a proteína — com inconvenientes erros associados, pois tratam-se de valores experimentais [12].

Para podermos calcular a estrutura de uma proteína a partir dessas distâncias, de forma estática, respeitando restrições de outras informações provenientes da física e química, surgira um novo problema na literatura conhecido como *Molecular Distance Geometry Problem* (MDGP), que é uma particularização do *Distance Geometry Problem* (DGP) [3]. Tal problema, munido de uma ordem conveniente para percorrer seus átomos (que garante uma discretização do espaço de buscas por soluções), pode ser discretizado, gerando o *Discretizable MDGP* (DMDGP).

Este último trata-se do nosso problema fundamental, que será melhor definido no Capítulo 4. Para podermos compreendê-lo, introduzimos a teoria de grafos (no Capítulo 2), seguido das principais informações sobre as estruturas biomoleculares das proteínas (Capítulo 3). Por último, apresentamos o principal algorítimo responsável pela solução do problema (Capítulo 5), contendo algumas simulações computacionais.

A revisão bibliográfica completa pode ser encontrados no fim do documento, sendo devidamente citada durante o texto.

2 Grafos: Aspectos Gerais

Esta seção tem como objetivo apresentar um breve resumo da *teoria de grafos*, tema muito estudado por diversos matemáticos e aplicado em diversas áreas do conhecimento para além da matemática.

Podemos dizer que em 1736 é que a teoria teve início, com base no artigo publicado por Leonhard Euler, sobre as 7 pontes de Königsberg [4] [5]. Esse é o problema que normalmente introduz quem está começando a trabalhar com grafos — se trata do desafio de ligar todos os pontos de um desenho sem tirar o lápis do papel e sem passar duas vezes no mesmo ponto. Segundo a história, os moradores daquela região perguntavam se era possível atravessar todas as pontes sem ter que repetir alguma delas. Euler provou que isso não era possível, ao formular matematicamente o problema, que deu origem a esta teoria.

Para isso, Euler abstraiu o problema, ao vê-lo de um ponto de vista matemático, como um conjunto de pontos intersectados por linhas (vide Figura 1). Essa representação facilitou a análise de que a solução do problema só seria possível se houvesse exatamente zero ou dois pontos de onde saísse um número ímpar de caminhos [4].

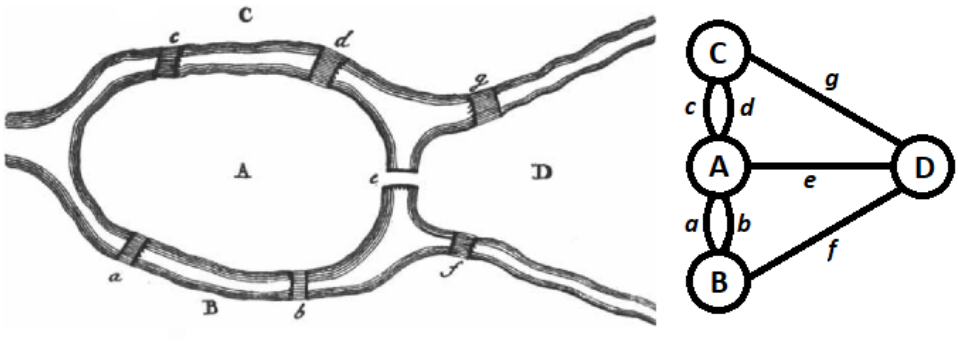

Figura 1: Ilustração original do problema [4] e sua representação em Grafos.

Além dessa, muitas outras situações reais podem ser convenientemente representadas por simples diagramas contendo um conjunto de pontos e linhas ligando pares desses pontos. Por exemplo, podemos definir o conjunto $P = \{a, b, c\}$ das pessoas a, b e c e um conjunto $A = \{\{a, b\}, \{b, c\}\}$ como o conjunto de amizades entre essas pessoas — no caso, a é amigo de b, que é amigo de c, porém a não é amigo de c.

> **Grafo:** Um grafo G é uma tripla ordenada da forma $(V(G), E(G), \psi_G)$, composto por um conjunto de *vértices* $V(G)$, de arestas $E(G)$ e uma *função de incidência* ψ_G que, por sua vez, associa a cada aresta de $V(G)$ um par não ordenado de vértices (nem sempre distintos) de $E(G)$. Costumamos dizer que as arestas ligam os vértices.

Existe, também, uma íntima relação entre Grafos e algoritmos. De fato, podemos inclusive representar um algoritmo por um grafo [6]. Na verdade, a definição de grafos é tão abrangente que podemos ver suas ligações com diversas áreas do conhecimento.

No que se segue, definiremos algumas características elementares que serão utilizadas durante esse texto. Para um estudo mais completo sobre essa teoria, vide [5] e [7].

2.1 Algumas Classificações Importantes

Existem duas definições de extrema importância para o nosso problema molecular (tema central desse texto): o conceito de grafo completo e o de estruturas k-cliques. Porém, seremos obrigado a definir alguns outros conceitos prévios, como segue.

Laço: Uma aresta $\{e_i, e_j\} \in E$ tal que $i = j$.

Também, caso existam duas arestas iguais ($\{e_i, e_j\}$ e $\{e_j, e_i\}$, por exemplo, lembrando que E é um conjunto de pares não ordenados), com as mesmas extremidades, estas recebem o nome de **arestas paralelas**.

Grafo simples: Um grafo que não possui laços ou arestas paralelas.

Grafo Completo: É um grafo simples em que todo vértice é conectado a todos os outros vértices.

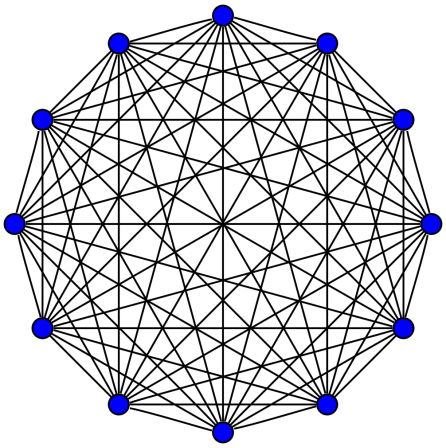

Figura 2: Diagrama de um grafo completo com 12 vértices ($|V| = 12$).

Outro conceito que nos será de grande utilidade é o de subgrafo.

Subgrafo: É um grafo resultante de um subconjunto de vértices e outro subconjunto de arestas de outro grafo. Isto é, seja $G = (V, E), G' = (V', E')$ é dito um subgrafo de G se (V', E') é um grafo tal que $V' \subseteq V$ e $E' \subseteq E$.

E, finalmente

k-Clique: é um subgrafo G' com k vértices tal que G' é completo.

Em especial, também podemos interpretar as arestas como *caminhos* e, se o fizermos, podemos pensar em alguma forma de métrica para esses caminhos. Esse pensamento da origem à nossa última definição de grafos, ditos *ponderados*.

Grafo Ponderado: É um grafo que possui uma função $d(E) \to \mathbb{R}$ associada, isto é, o grafo que possui valores numéricos atribuídos as suas arestas.

3 Um Passeio pela Bioquímica

A bioquímica é a ciência que estuda as formas e funções biológicas em termos químicos. Já no século XVIII, os químicos percebiam a grande diferença entre o mundo inanimado e o mundo vivo: Antoine-Laurent Lavoisier (1743-1794) constatou a relativa simplicidade do "mundo mineral" — não orgânico — comparada a complexidade dos "mundos animal e vegetal" [9]. Ele sabia que esses últimos eram constituídos de moléculas ricas nos elementos carbono, oxigênio, nitrogênio e fósforo, que, devido sua abundância na natureza somada com as suas características químicas, são ótimos para constituírem a complexidade da vida.

3.1 Carbono

A química dos organismos vivos está organizada em torno do carbono, pois este é muito comum na natureza e possuí uma ótima propriedade estrutural: O carbono pode formar ligações simples estáveis com até quatro outros átomos. De fato, o carbono constitui mais da metade do peso seco das células.

Sabe-se, através de experimentos de cristalografia [10], muito sobre a geometria das ligações dos átomos de uma proteína. Em particular, as quatro ligações simples do carbono formam um tetraedro (vide Figura 3, retirada de [9]) com ângulos de 109,5°entre duas ligações quaisquer e comprimento médio de ligação de 1,54Å[1]. Existe também uma outra característica muito importante para nós nas ligações do carbono: Sabe-se que as ligações simples podem rotacionar livremente (a menos que grupos muito grandes ou altamente carregados estejam ligados aos átomos de carbono, onde, neste caso — e, na verdade, esse é o caso comum —, a rotação é regida pelo equilíbrio de forças na molécula [11], que pode ser limitada), enquanto que as ligações duplas são mais curtas (em torno de 1,34Å) e não permitem rotação. Perceba também o plano formado pelos átomos A, B, X e Y na Figura 3.

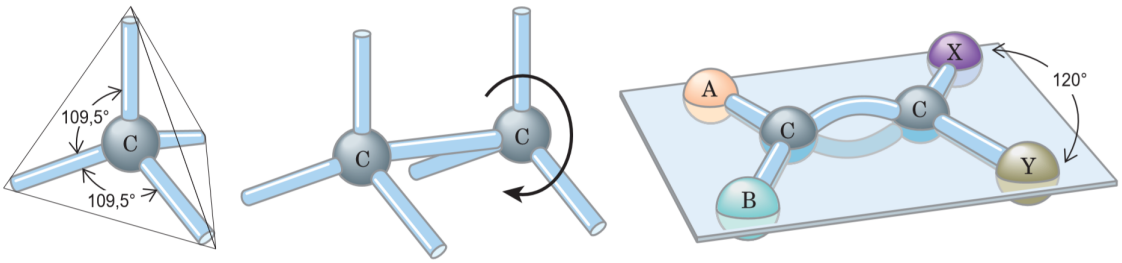

Figura 3: Geometria da ligação do carbono.

A versatilidade das ligações covalentes do carbono podem formar cadeias lineares, ramificadas e estruturas cíclicas. Nenhum outro elemento químico consegue formar moléculas com tanta diversidade de tamanhos, formas e composição.

[1]Unidade física para distâncias atômicas é o Ângstron (Å), onde equivale a 1Å= 10^{-10} m.

3.2 Classificação Macromolecular

As células contém um conjunto universal de moléculas pequenas. Mas como podemos discutir sobre o que é uma molécula pequena? Devemos definir uma forma de comparar os tamanhos moleculares. Na literatura existem duas medidas principais para esse fim, com uma relação bem definida entre si, tratam-se do *peso molecular* (ou *massa molecular relativa*), denominado M_r e da *massa molecular*, denotada simplesmente por m.

O peso molecular é definido como uma relação direta da massa da molécula da substância estudada com um duodécimo da massa do carbono-12 (^{12}C, em torno de $1,9926 \times 10^{-23}$ gramas), note que, como M_r é uma razão, não possui dimensão associada. Já a massa molecular é apenas a massa da molécula (ou massa molar) sobre o número de Avogadro — que é definida como sendo o número de átomos por mol de uma determinada substância. Esta, diferente da massa molecular relativa, possui dimensão e é expressa em dáltons (abreviado Da) e um dálton equivale a um duodécimo da massa do carbono-12 — donde deduze-se facilmente a relação entre massa molecular e peso molecular.

Os organismos vivos são constituídos por moléculas de características muito diversas. Existe uma coleção de aproximadamente mil moléculas consideradas pequenas (M_r ~100 a ~500) diferentes dissolvidas na fase aquosa das células [9]. Nessa coleção está contido os aminoácidos comuns, nucleotídeos, açúcares e seus derivados fosforilados e ácidos mono, di e tricarboxílicos. Porém, neste estudo, estaremos mais preocupados com moléculas significativamente maiores, chamadas *macromoléculas*.

Macromoléculas

As macromoléculas são as principais constituintes das células. São polímeros[1] com peso molecular acima de ~5.000. Polímeros menores são chamados de *oligômeros* — do grego, "oligos" significa "pouco". Proteínas (principal molécula do nosso estudo), ácidos nucleicos (DNA, RNA) e polissacarídeos são macromoléculas feitas de monômeros cujos pesos moleculares são de 500 ou menos, porém, como apresentam um grande número dessas subunidades, possuem um alto peso molecular — até 1 milhão para proteínas e até vários bilhões para ácidos nucleicos. A síntese de macromoléculas é a atividade mais custosa energeticamente das células.

Tanto as proteínas quanto os ácidos nucleicos são polímeros lineares (isto é, que não possuem ramos ligados as suas cadeias principais, agindo como um longo fio contínuo) feitos de subunidades monoméricas bem mais simples, donde esta sequência específica de meros é que dá as informações sobre a sua estrutura tridimensional e suas funções biológicas associadas [9].

Em especial, as proteínas são constituídas por um conjunto de monômeros muito bem conhecidos e catalogados, chamados *aminoácidos*. As proteínas constituem a segunda maior fração da célula, só perdendo para a água. Provavelmente são as mais versáteis de todas as biomoléculas: Algumas tem atividade catalítica e funcionam como enzimas, outras servem como elementos estruturais, receptoras de sinais, ou transportadoras que carregam substâncias específicas para dentro ou fora das células.

[1] Polímeros são moléculas formadas a partir de repetições de unidades estruturais menores, chamadas *meros* ou *monômeros*. Daí o nome, poli-meros ≈ vários-meros.

3.3 Configuração Molecular

No mundo biomolecular, toda a informação sobre uma molécula é dada pela sua estrutura (também chamada de *estereoquímica*), logo, suas ligações covalentes e seus grupos funcionais (subestruturas padrões associadas) são trivialmente importantes para definir seu bom funcionamento. Devido a característica rotacional das ligações simples do carbono, existem muitas moléculas (chamadas *estereoisômeros*) com a mesma fórmula molecular e ligações químicas, mas com diferentes configurações espaciais, o que pode mudar completamente suas funções.

De maneira simples, podemos identificar estereoisômeros pelo fato de que eles possuem as mesmas propriedades químicas, porém, não podem ser convertidos entre si sem que haja a quebra de uma ou mais ligações covalentes. Isto se dá pela presença de ligações duplas (devido a limitação na sua rotação) ou pela presença de *centros quirais*, onde a molécula rotacionada não pode corresponder a sua imagem especular (conforme Figura 4, extraída de [9]). Um átomo de carbono com quatro ligações diferentes é considerado assimétrico e é chamado de centro quiral — do grego, *chiros* quer dizer "mão", parafraseando estas estruturas com a relação da mão direita com a esquerda. Logo, se existir um centro quiral, sempre haverá pelo menos duas possibilidades para configuração.

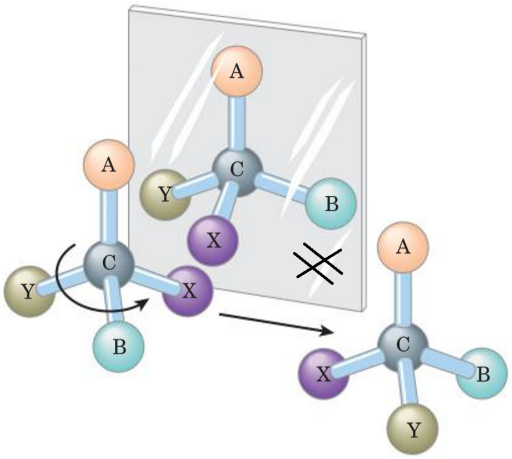

Figura 4: Ilustração de uma molécula quiral.

Outro conceito que nos será importante no futuro, a *conformação molecular* é a disposição dos átomos no espaço que pode ser mudada por rotação em torno de ligações simples, sem quebrar ligações covalentes. Estes ângulos possíveis tem posições mais estáveis e instáveis do ponto de vista energético, conforme mostra o gráfico da Figura 5. Podemos tentar descobrir a conformação mais provável de uma molécula minimizando a somatória de todas as forças atuantes na molécula [11].

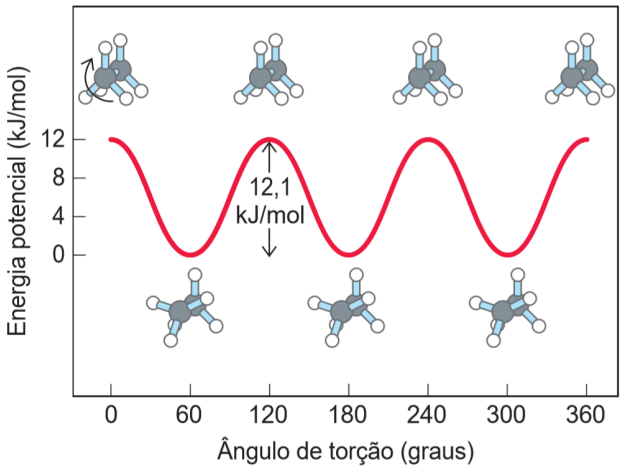

Figura 5: Conformações e Equilíbrio de Energia [9].

Para compreender melhor como serão as configurações das moléculas que trataremos nesse texto (proteínas), vale nos preocuparmos com as subestruturas do qual eles são formados.

3.4 Aminoácidos

As proteínas são longas cadeias lineares de aminoácidos ligados por um tipo específico de ligação (chamada *peptídica*), a qual é característica por ter como resíduo uma molécula de água. São vinte tipos diferentes de aminoácidos encontrados normalmente na natureza, sendo esses muito bem conhecidos e catalogados. O primeiro a ser descoberto foi a asparagina, em 1806; o ultimo foi a treonina, descoberto em 1938 [9]. Vale mencionar que, além destes vinte aminoácidos mais comuns, há vários outros menos frequentes, porém não constituem as proteínas.

Destes vinte aminoácidos comuns (disponíveis no Apêndice C), dezenove compartilham da mesma estrutura principal [1] — estes são chamados α-aminoácidos. Eles tem um grupo carboxílico e um grupo amina ligados ao mesmo átomo de carbono (o carbono α), além de mais um hidrogênio (chamado hidrogênio α) e, em sua última ligação, uma cadeia R que é o que diferencia cada aminoácido. Essa estrutura é ilustrada na Figura 6. O único aminoácido que difere disso é a Prolina, que possui como cadeia R um anel aromático que se fecha no nitrogênio (que no padrão mencionado há um grupo amina).

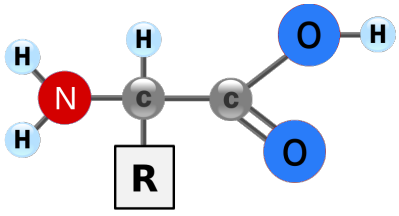

Figura 6: Estrutura padrão de um α-aminoácido.

Portanto, há uma noção prévia de qual tipo de estrutura esperar ao analisar uma molécula de proteína. Existe uma estrutura conhecida e repetitiva para os átomos.

Para todos os aminoácidos comuns, exceto a glicina, o carbono α está ligado com quatro outros átomos diferentes entre si (na glicina temos R como apenas mais um hidrogênio, sendo o aminoácido mais simples), o que transforma o carbono α em um centro quiral. Logo, cada aminoácido (menos glicina) tem sempre dois estereoisômeros possíveis. Porém, na verdade, apenas um destes ocorrê naturalmente nas proteínas [9].

Ligação Peptídica

A ligação entre dois aminoácidos é feita de modo covalente por meio de desidratação do grupo α-carboxílico de um com o grupo α-amina do outro — ou seja, ligar o carbono final de um no nitrogênio inicial do outro, liberando um oxigênio e dois hidrogênios, que formam uma molécula de água. Essa ligação, também chamada de resíduo (devido a liberação da água), forma um dipeptídeo.

Quando muitos aminoácidos se juntam, o produto é chamado de polipeptídeo. Perceba que os termos "polipeptídeo" e "proteína" parecem dirigir-se as mesmas moléculas, porém, a diferença está na massa molecular: As moléculas com massa abaixo de 10.000 são ditas polipeptídeos, enquanto as maiores que essas são consideradas proteínas. Os comprimentos dessas cadeias variam significativamente. O citocromo c humano tem apenas 104 aminoácidos, enquanto, no outro extremo, a titina (relacionada ao músculo de vertebrados) possui aproximadamente 27.000 aminoácidos e uma massa molecular de cerca de 3.000.000. No geral, as proteínas naturais contém menos de 2.000 aminoácidos [9].

Outra característica muito importante das ligações peptídicas é de que elas se comportam semelhantemente a ligações covalentes duplas dos carbonos. Estudos envolvendo difração de raios X em cristais de aminoácidos e polipeptídeos descobriram que a ligação peptídica $C-N$ é de alguma forma mais curta que a ligação de uma amina simples, e que os átomos associados a ligação peptídica estão todos coplanares (conforme Figura 7). Perceba que também são rígidos, não sendo possível a rotação. Essa é uma propriedade muito útil que também nos será importante, descoberta de 1930 que se deve a Linus Pauling e Robert Corey.

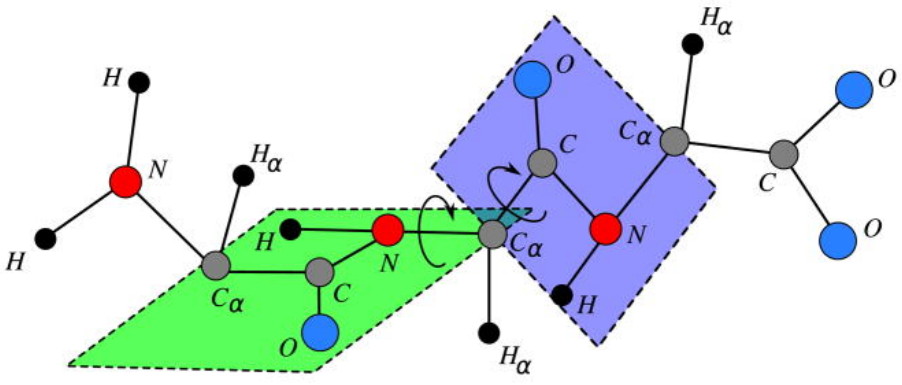

Figura 7: O grupo peptídico planar [12].

3.5 Estrutura das Proteínas

A estrutura de proteínas pode ser descrita em quatro níveis de importante hierarquia conceitual, conforme pode ser visto na Figura 8, retirado de [9]. A estrutura primaria consiste da mais detalhada, sendo de fato os polímeros de aminoácidos; Estes, por sua vez, formam alguns arranjos particularmente estáveis, que dão origem a padrões estruturais recorrentes, que chamamos de *estruturas secundárias* (como as hélices α, as duplas hélices etc..). A estrutura terciária descreve todos os aspectos do enovelamento tridimensional de um polipeptídeo, ou seja, define quais serão as forças atuantes na molécula — que da origem a sua conformação estável, que minimiza a energia livre de Gibbs do sistema. Quando existem mais estruturas terciárias em uma proteína, chamamos a junção destas de estrutura quaternária.

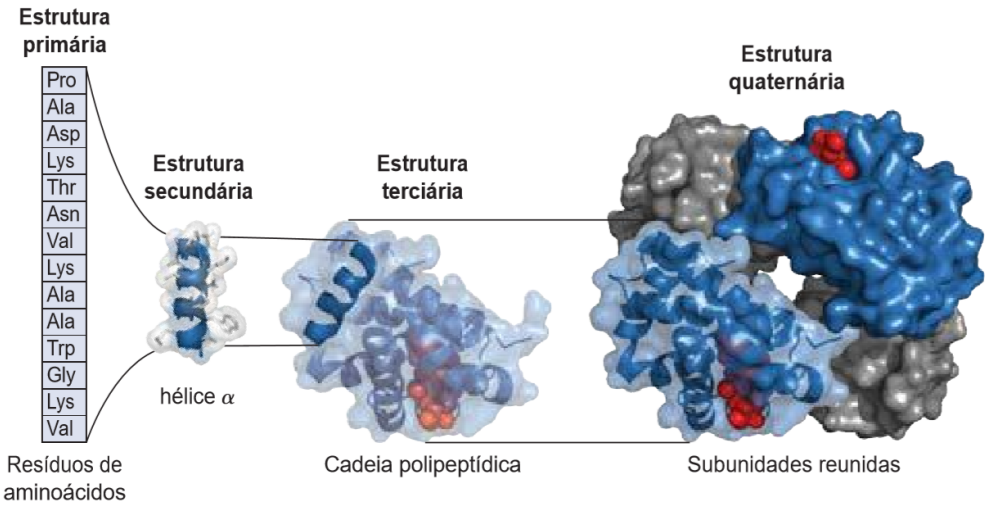

Figura 8: Níveis de estrutura das proteínas exemplificados na Hemoglobina.

Em especial, as diferentes configurações da estrutura primária — que pode mudar drasticamente entre estruturas primárias diferentes na mesma molécula — nos é mais informativa. A estrutura primária de uma proteína determina como ela se dobra em sua estrutura tridimensional, devido os ângulos e distâncias bem definidos de suas ligações entre átomos, que da a sua estrutura especial; o que, por sua vez, determina a função da proteína — como no exemplo da Figura 8, onde a estrutura da hemoglobina é que permite que átomos de oxigênio "encaixem" nela, possibilitando o transporte desse átomo pelo organismo, que é sua função (e só o é dado sua estrutura tridimensional).

Por sua relação com a estrutura tridimensional e, logo, função das proteínas, vamos nos concentrar em estudar a subdivisão de estruturas primárias.

A Cadeia Principal de uma Proteína

Quando se estuda proteínas a nível dos aminoácidos, não tardamos a perceber que elas possuem uma estrutura repetida muito interessante do ponto de vista bioquímico. Trata-se da *cadeia principal* de uma proteína, também chamada de *Backbone* — espinha dorsal, em tradução literal, fazendo alusão a importância desta estrutura. Perceba que os vinte aminoácidos que compõem as proteínas possuem sempre

os mesmos três átomos ligados em sequência (Figura 9): $N - C_\alpha - C$, através de ligações covalentes em torno do C_α e da ligação peptídica $C - N$ entre aminoácidos.

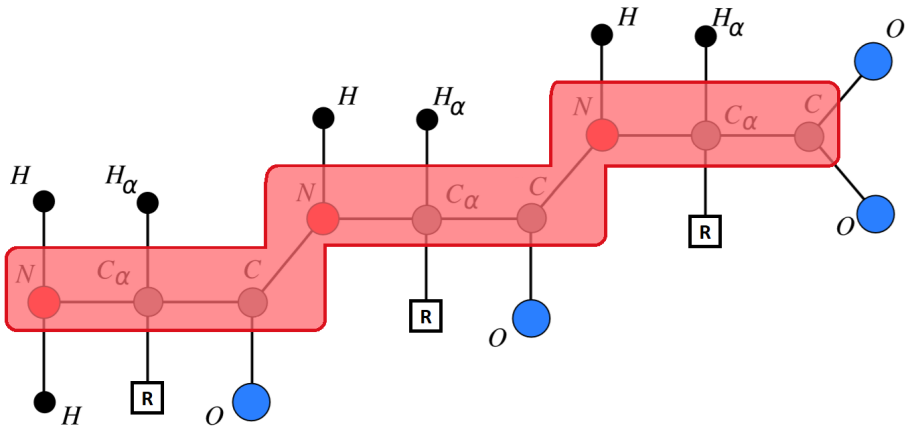

Figura 9: Representação da cadeia principal da proteína, adaptada de [12]

.

Outra informação bastante útil sobre esta cadeia principal é que, devido dados experimentais de cristalografia, sabe-se sobre a geometria média dessa subestrutura [10], onde os comprimentos e ângulos entre as ligações dos átomos que a formam são fixas, na média, a menos de erros de medida. Vide Figura 10, extraída do texto original de Ramachandran *et al*, um dos precursores deste estudo.

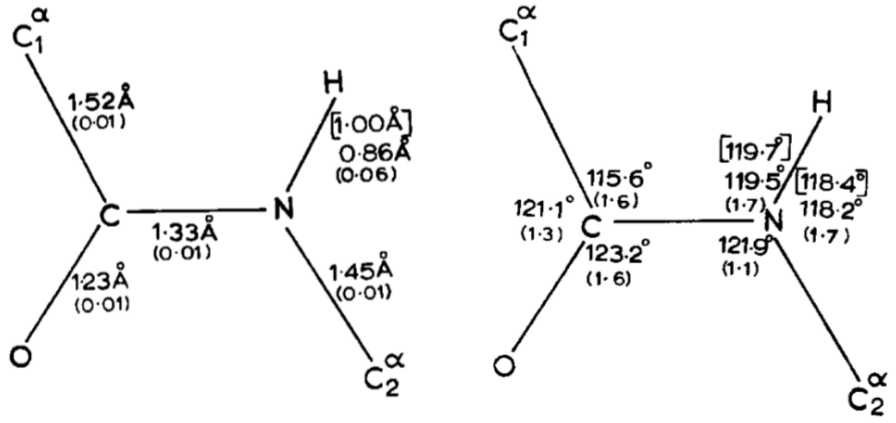

Figura 10: Dados de ângulos e distâncias médios de ligações em um aminoácido.

3.6 *Worldwide Protein Data Bank*

Já foi possível perceber a grande variedade de diferentes configurações possíveis para as proteínas. Com isso, há a necessidade de se estudar cada uma explicitamente, através de experimentos, catalogando e guardando essas informações. Esse grande esforço para entender o mundo das macromoléculas se deixa transparecer com o repositório *Worldwide Protein Data Bank* — ou simplesmente wwPDB [13].

Este é um repositório online e público onde estão guardadas todas os dados de proteínas e ácidos nucleicos já catalogados, em especial dados de suas estruturas

3D (posições x, y e z de cada um dos átomos que a constituem). Auxiliando tanto pesquisadores, quanto professores e estudantes, essa base de dados é um grande esforço em conjunto de físicos, biólogos, bioquímicos e vários outros profissionais de diversas áreas do conhecimento de todo o mundo.

Arquivo PDB

Quando se quer estudar uma proteína no repositório PDB, base fazer o *download* do arquivo PDB da molécula (extensão ".ent"). Esse é um arquivo de estruturas tridimensionais de macromoléculas biológicas determinadas experimentalmente, que descrevem as coordenadas espaciais de cada átomo cuja posição foi determinada (muitas das estruturas catalogadas não estão completas); também existem dados adicionais sobre informações de como as estruturas foram determinadas, os dados práticos dos experimentos, a precisão associada aos dados e tudo mais que quem estiver criando o documento achar necessário para aquela macromolécula.

Tecnicamente, o arquivo PDB trata-se de uma representação estruturada dos dados moleculares e experimentais da proteína. Ele é separado por seções, onde cada seção pode possuir subseções. São elas:

- **Seção Title** - Contem a descrição da molécula;
- **Seção Remark** - Vários comentários sobre anotações de entrada com mais profundidade que os registros padrões;
- **Seção Primary structure** - Sequências peptídicas ou nucleotídicas especificadas para serem posteriormente utilizadas, diminuindo a repetição do arquivo;
- **Seção Heterogen** - Descrição de grupos presentes não padronizados — Visto que proteínas também podem conter materiais inorgânicos, como o ferro presente na hemoglobina (vide Figura 8);
- **Seção Secondary structure** - Descrição das estruturas secundárias presentes na molécula;
- **Seção Connectivity annotation** - Descrição das conectividade químicas da molécula;
- **Seção Miscellaneous features** - Descrição dos recursos dentro da macromolécula;
- **Seção Crystallographic** - Descrição de parâmetros da cristalografia, quando o experimento utiliza esta metodologia;
- **Seção Coordinate transformation** - Matrizes como operadores de transformação das coordenadas;
- **Seção Coordinate** - Dados de coordenadas atômicas, a seção que mais vamos utilizar;
- **Seção Connectivity** - Citação das conexões químicas entre os átomos;
- **Seção Bookkeeping** - Resumo das características totais do arquivo e o marcador de fim de arquivo.

Como o arquivo é significativamente extenso, não entraremos em detalhes neste texto sobre as características detalhadas de cada uma das seções apresentadas. Porém, vale mencionar o tipo de entrada ATOM, presente na seção Coordinate, pois essa é a entrada que compõe a maior parte dos arquivos PDB, além de ser a de nosso interesse principal.

A entrada ATOM tem como objetivo descrever detalhes de cada átomo específico da molécula. Ela segue um padrão indentado, onde cada dado é caracterizado pela sua posição na linha (coluna). Segue principais dados da entrada e suas respectivas colunas na Tabela 1.

Código serial do átomo	7-11
Nome do átomo	13-16
Nome do resíduo que pertence	18-20
Identificador da cadeia	22
Código serial de dentro do resíduo	23-26
Coordenada x	31-38
Coordenada y	39-46
Coordenada z	47-54
Occupancy do átomo	55-60
Fator de temperatura	61-66
Simbolo do elemento	77-78

Tabela 1: Principais dados da entrada ATOM.

Segue exemplo de um conjunto de entradas do tipo ATOM na Figura 11.

```
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
ATOM      1  N   MET A   1     -10.885   6.773  13.357  1.00  0.00           N
ATOM      2  CA  MET A   1     -12.318   6.914  13.685  1.00  0.00           C
ATOM      3  C   MET A   1     -13.195   6.440  12.525  1.00  0.00           C
ATOM      4  O   MET A   1     -12.738   6.392  11.383  1.00  0.00           O
ATOM      5  CB  MET A   1     -12.654   8.361  14.078  1.00  0.00           C
ATOM      6  CG  MET A   1     -12.548   9.328  12.889  1.00  0.00           C
```

Figura 11: Conjunto de entradas do tipo ATOM.

Com esse conjunto de dados, pode-se, por exemplo, esboçar uma representação gráfica de uma molécula. Existem muitos softwares compatíveis com os arquivos PDB para este fim, por exemplo, o autor deste documento implementou uma visualização de uma projeção da molécula 3D no plano z = 0, como pode-se averiguar na Figura 12.

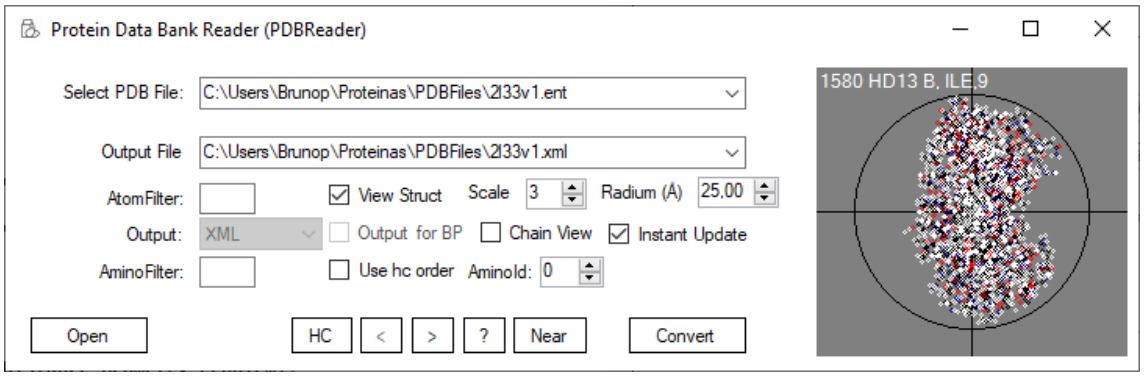

Figura 12: PDBReader com visualização a partir de um aquivo PDB.

Aqui vale um momento para uma introdução ao software desenvolvido, como parte dos resultados deste trabalho.

3.6.1 *Sotware PDBReader*

O arquivo PDB é denso, cheio de termos técnicos e pouco amigável, demandando um certo tempo para que alguém que esteja sendo introduzido nesta área se acostume com seu padrão. Por isso, surgiu a possibilidade de desenvolver uma aplicação que vise facilitar e automatizar a extração das informações das moléculas contidas nele. Este trabalho teve esse software, nomeado *Protein Data Bank Reader*, como primeiro resultado prático.

O software foi desenvolvido em C#, uma linguagem de programação multiparadigma, orientada a objetos e eventos, de tipagem forte, desenvolvida pela Microsoft como parte do *framework* .NET. A interface de usuário foi feita utilizando Windows Forms, como uma janela única, denominada fMain (que pode ser vista na Figura 12).

A aplicação pode ser usada de duas formas diferentes: Para gerar um arquivo bem formatado com os dados dos átomos contidos no arquivo PDB de entrada — isso pode ser feito em diversos formatos, como XML, JSON, Matriz (no padrão MatLAb) e MolConf (padrão para aplicar na biblioteca Julia Language MolecularConformation.jl [14]); Ou pode ser usado apenas como ferramenta de visualização da molécula (selecionando o *checkbox* struct view).

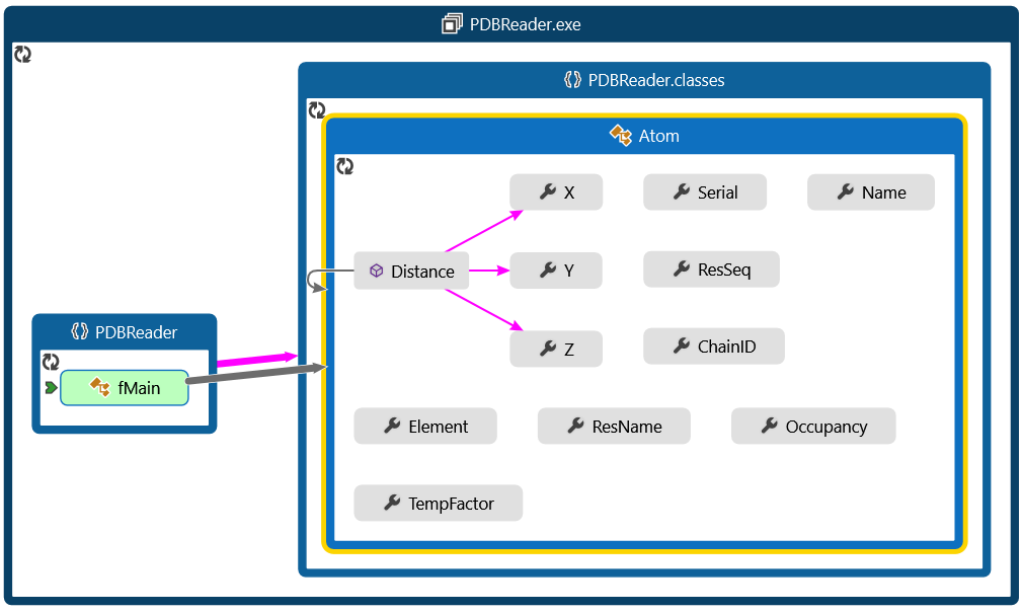

Figura 13: Relação entre classes do PDBReader.

Observe na Figura 13 a existência de um conjunto de classes em *PDBReader.class* que o formulário fMain utiliza. Em especial, a classe Atom, que representa um átomo, contendo todos as suas propriedades (retiradas do arquivo PDB, como as posições x, y e z) e uma função muito importante, chamada *Distance*, que retorna a distância euclidiana entre dois átomos.

O formulário fMain possui um conjunto de eventos, disparados por interações com o usuário (como mostrado na Figura 14). Como pode-se perceber pelo diagrama, a grande maioria dos eventos chamam o método *updateView*, que tem a função de atualizar a tela de visualização da proteína. Perceba que isso só acontece quando se está com o checkbox *structView* selecionado, uma vez que o updateView só funciona nesse caso.

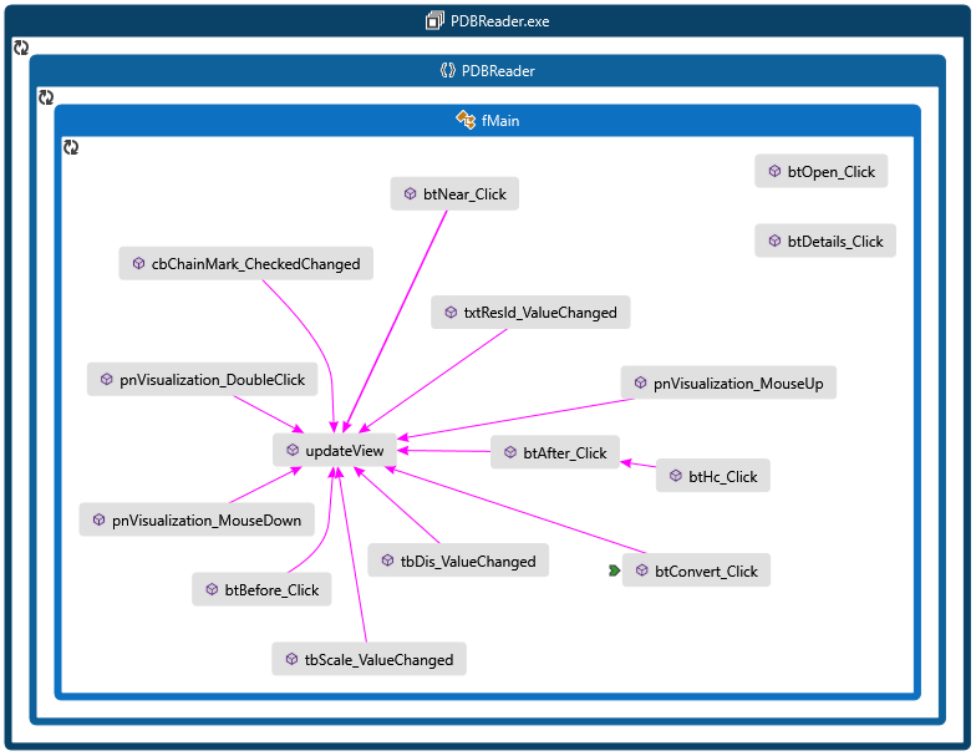

Figura 14: Respostas do formulário ás ações do usuário.

Definição das funções do PDBReader

Segue abaixo uma descrição dos principais componentes do software que permitem interação com o usuário. Verifique a presença dos identificador *id* de cada componente na Figura 15, que são referenciados na Tabela 2 com suas respectivas descrições.

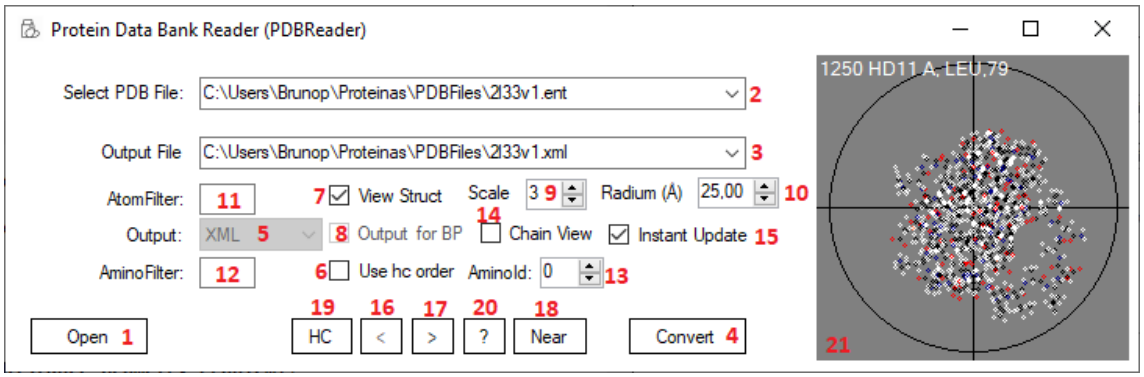

Figura 15: Interação do PDBReader.

id	Tipo	Descrição
1	Botão	Abre uma caixa de diálogo para selecionar o arquivo PDB de entrada
2	Caixa de Texto	Diretório onde está o arquivo de entrada
3	Caixa de Texto	Diretório para onde será gerado o arquivo de saída
4	Botão	Realiza a leitura do arquivo e faz a conversão
5	Combo de Seleção	Seleciona o formato do arquivo de saída
6	Caixa de Seleção	Seleciona se é para usar o Ordenação HC durante a conversão
7	Caixa de Seleção	Seleciona se o objetivo é ter uma visualização da molécula
8	Caixa de Seleção	Seleciona se será usado o padrão Branch-and-Prune na conversão
9	Entrada Numérica	Informa qual a escala para ser usada na visualização
10	Entrada Numérica	Informa o raio a ser considerado na conversão e visualização
11	Caixa de Texto	Filtrar por algum átomo específico (e.g. “C” para carbono)
12	Caixa de Texto	Filtrar por algum aminoácido específico (e.g. “ALA” para Alanina)
13	Entrada Numérica	Se > 0 filtra para o aminoácido de identificador específico
14	Caixa de Seleção	Se selecionado pinta de rosa todas as cadeias que não forem a primeira
15	Caixa de Seleção	Se deseja que o software atualize o painel sempre que houver alterações.
16	Botão	Permite movimentação entre átomos, centraliza a tela no átomo anterior
17	Botão	Permite movimentação entre átomos, centraliza a tela próximo átomo
18	Botão	Centraliza o painel no átomo com menor distância para o atual
19	Botão	Tenta percorrer o aminoácido atual usando a ordem HC de forma empírica
20	Botão	Abre uma janela com informações sobre o átomo atual e os próximos
21	Painel Visual	Centraliza o painel em um átomo clicando duas vezes nele

Tabela 2: Descrição dos componentes do software.

Por exemplo, pode-se estudar apenas o segundo aminoácido (Alanina) da proteína Calcyclin (codigo PDB 1A03) — uma proteína do tipo ligante de cálcio — apenas setando o AminoId (componente 13 da Figura 15) para 2 e selecionando o View Sctruct (componente 7). Também podemos alterar a escala de exibição (componente 9) e a distância radial de visão (componente 10), para facilitar a visualização nessa dimensão pequenina. O resultado se vê na Figura 16.

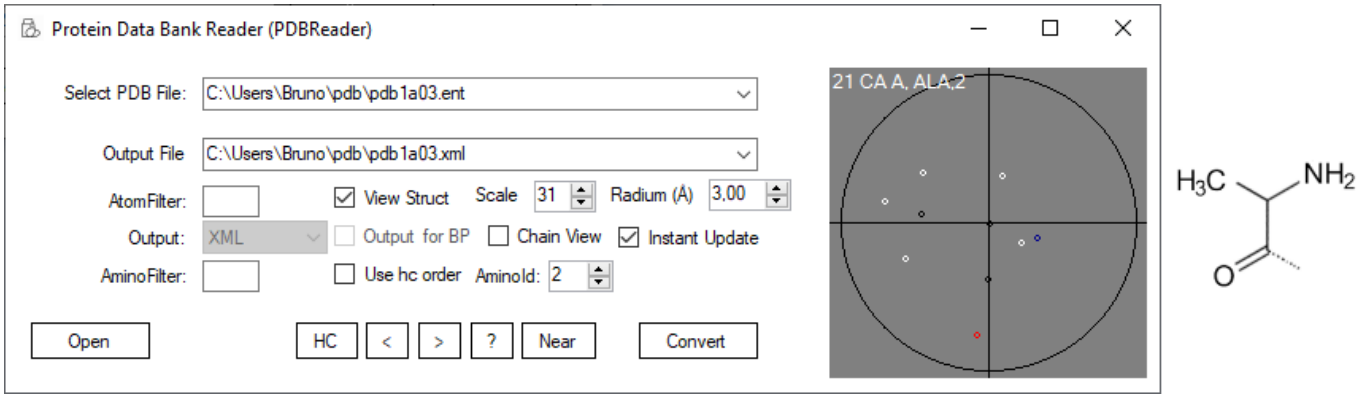

Figura 16: Visualização de uma Alanina utilizando o PDBReader.

4 *Molecular Distance Geometry Problem*

Até o momento, provavelmente, já deve estar mais que claro a importância de se estudar a estrutura tridimensional de proteínas e a grande complexidade relacionada a esta tarefa. De fato, a bioquímica é um universo de estudos imenso e, se não tomarmos cuidado, facilmente nos perderemos entre seus labirintos. Porém, a partir daqui voltaremos nosso olhar para uma perspectiva um pouco mais matemática, o que nos dá uma capacidade de abstração elevada.

Sempre perdemos alguma coisa quando trazemos um problema da vida real para um modelo matemático. Com a experiência, adquirimos a habilidade de optar pelas hipóteses corretas para solucionar o problema mais próximo possível do real, porém, ao fazê-lo, essas hipóteses podem dificultar demasiadamente a busca por uma solução. Esse é um dilema recorrente na matemática aplicada.

Nesse capitulo, estudaremos algumas definições importantes que nos darão base e estrutura para discutir sobre as possíveis hipóteses do problema, onde tentaremos construir, juntos, a sua real definição.

4.1 Geometria de Distâncias

O principal conteúdo de estudo da Geometria de Distâncias, como o próprio nome sugere, rodeia a ideia de distância entre objetos de determinada estrutura geométrica. Assim, o problema fundamental da Geometria de Distâncias consiste em determinar um conjunto de pontos, pertencentes a um espaço geométrico, com base nas distâncias conhecidas entre tais pontos. Note que nem sempre sabemos todas as distâncias entre todos os pontos.

O surgimento da Geometria de Distâncias se deu por volta de 1928, com o matemático Karl Menger, que caracterizou uma série de conceitos geométricos (e.g. congruência; conjuntos convexos) em termos de distâncias [3]. Entretanto, somente em 1953, com Leonard Blumenthal, que a Geometria de Distâncias tornou-se uma nova área de conhecimento [8]. E foi graças a Yemini, em 1978, que o problema fundamental de Geometria de Distâncias foi enunciado.

> **Distance Geometry Problem (DGP):** Dado $K \in \mathbb{Z}_+$ e um grafo simples e não-direcionado $G = (V, E)$ cujas arestas são ponderadas pelos valores da função não-negativa $d : E \longrightarrow \mathbb{R}_+$, determine se existe imersões $x : V \longrightarrow \mathbb{R}^K$ (realizações do grafo) tais que
>
> $$\forall \{u, v\} \in E, \quad \|x_u - x_v\| = d_{u,v}. \tag{1}$$

Uma da características mais interessantes desse problema é sua simplicidade — tem como entrada apenas um conjunto de distâncias entre pontos não localizados e procura-se localizá-los —, que o leva a ser aplicado em muitas diferentes áreas do conhecimento [3].

Tendo em mente nosso problema fundamental, podemos ajustar nossos dados moleculares para aplicá-lo.

4.2 Ressonância Magnética Nuclear

A ressonância magnética nuclear é um processo físico que analisa a interação da radiação eletromagnética com a matéria. Neste experimento é escolhida uma faixa de

radiofrequência para bombardear uma amostra que está imersa em um campo magnético bastante intenso. Dependendo da radiofrequência utilizada, alguns núcleos atômicos irão absorver energia e outros não. Caso atinja-se uma frequência exata de ressonância dentro destes núcleos atômicos, é possível medir essa ressonância como um sinal de radiofrequência enviado dos núcleos atômicos — para calcular distâncias entre átomos próximos, com distâncias menores que 5Å. No nosso problema, a frequência utilizada é para a ressonância dos núcleos de hidrogênio e um computador capta essas respostas eletromagnéticas dos núcleos atômicos para utilizar como dados do problema. [15]

Assim, esse procedimento fornece vários *intervalos* de distâncias possíveis relativas associadas a átomos de hidrogênio próximos (sendo que por vezes também é capaz de captar átomos de um isótopo específico de carbono, devido a proximidade de sua frequência com a do hidrogênio), esses intervalos também são chamados de *distancias intervalares*. Pode-se representar essas distancias intervalares matematicamente por intervalos de números reais $[d^i_{i,j}, d^f_{i,j}]$. Isto é, existe um real $d_{i,j}$ que representa a distância real tal que

$$0 \leq d^i_{i,j} \leq d_{i,j} \leq d^f_{i,j}$$

Porém, devido a grande complexidade gerada ao trabalhar com estas distâncias intervalares, nesse texto nós só trabalharemos com as distâncias $d_{i,j}$ exatas, supondo-as como hipótese.

4.3 Modelagem Matemática

Quando um pesquisador de outra área se depara com um problema que não consegue resolver e precisa recorrer aos matemáticos, quase sempre se torna uma tarefa muito complicada para quem for tentar desenvolver o problema. Não pela complexidade matemática do assunto, isto os matemáticos dominam. A dificuldade está em *entender* o problema de outras áreas para resolve-los.

A boa interpretação de um problema de matemática aplicada deve se ater a algumas perguntas importantes:

- **Quais são as hipóteses?** Ou seja, fatos dos quais devemos nos basear e nos limitar para propor, de forma dialética, uma solução para o problema. Nos importaremos com seis hipóteses que advém de informações já discutidas aqui:

  Hipótese 1: as distâncias fornecidas pelos experimentos de RMN estão associados a pares de átomos conhecidos: nós sabemos a quais átomos as distâncias se referem (isso não é bem verdade, mas supomos que seja assim para desenvolver o problema);

  Hipótese 2: todos os átomos da molécula da proteína cuja estrutura 3D queremos calcular são conhecidos: conhecermos a estrutura química da molécula;

  Hipótese 3: todos os átomos da molécula de proteína estão ligadas a algum átomo cuja distância é conhecida: não há átomos soltos, afinal, se existisse, seria outra molécula;

  Hipótese 4: existe uma ordem, dada a priori, entres os átomos da cadeia principal da proteína cuja estrutura 3D queremos calcular: conhecemos o

esqueleto padrão — backbone —, formado de aminoácidos, da molécula de proteína examinada;

Hipótese 5: as distâncias entre os átomos de uma molécula de proteína separados por duas ligações covalentes são conhecidas: existem esses resultados na literatura;

Hipótese 6: as distâncias fornecidas pela RMN são representandos por intervalos de números reais que contêm o valor correto associado.

- **Qual resultado deseja-se obter?** De forma simplificada, qual é nossa *tese*? Dificilmente se chega no lugar ideal se não há o conhecimento da direção a seguir. Cabe-se uma definição formal dos nossos objetivos:

  Deve-se determinar os pontos $x_i \in \mathbb{R}^3$, $i = 1, \ldots, n$ (n é o número de átomos da molécula), satisfazendo as equações

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

  onde $E \subset \{1, ..., n\} \times \{1, ..., n\}$ e d_{ij} *são os valores de distâncias fornecidas pela RMN.*

  Tentar resolver os sistemas de equações acima parece não ser uma boa ideia, já que existem evidências de que não seja possível obter uma fórmula fechada para isso [8]. Podemos até tentar resolver numericamente, porém as soluções seriam infinitas para as equações.

  A abordagem mais usual é a representação do problema usando otimização. Para isso devemos resolver todas as equações do problema. Podemos, dessa maneira, considerar uma única expressão com todas elas, dada por

  $$f(x_1, ..., x_n) = \sum_{(i,j) \in E} (\|x_i - x_j\| - d_{ij})^2$$

  Para resolvermos tal otimização, basta encontrar valores de $x_i \in \mathbb{R}^3$, $i = 1, ..., n$, tal que $f(x_1, ..., x_n) = 0$. Assim temos

  $$\min_{x_i \in \mathbb{R}^n} f(x_1, ..., x_n).$$

  A dificuldade nesse caso está em encontrar o mínimo global, pois existem vários mínimos locais e estes *crescem exponencialmente com a quantidade de átomos da molécula*. Outro ponto delicado é que pode ser muito complicado distinguir um mínimo local de um global, uma vez que os métodos de otimização continua só se referem a informações locais. Isso tudo torna o problema muito custoso para resolver.

4.3.1 MDGP: Uma definição formal

Agora que temos uma boa base, vamos definir o Problema de Geometria de Distâncias Moleculares formalmente [8].

Molecular Distance Geometry Problem: Dado um grafo simples não-direcionado $G = (V, E)$, de modo que suas arestas sejam valoradas por uma função não-negativa $d : E \to \mathbb{R}_+$, considere a seguinte aplicação:

$$x : V \to \mathbb{R}^3$$

De modo que para todo $\{u, v\} \in E$ temos:

$$\|x(u) - x(v)\| = d(u, v)$$

É fácil ver que essa se trata de uma especificação do DGP, onde $k = 3$. Logo, nossa tarefa é encontrar uma aplicação x que satisfaça todas as distâncias e dados do grafo que temos. Esta função é denominada *realização* de G. Quando a realização satisfaz todas as equações da forma "$\|x(u)-x(v)\| = d(u, v)$", dizemos que ela é uma *realização válida*.

4.4 Modelagem Computacional

Uma vez que existe uma introdução a modelagem matemática do problema, pode-se pensar em uma abordagem computacional. Não é segredo para ninguém que a computação veio e vem melhorando muito a forma como se faz matemática, introduzindo novas ferramentas e campos de estudo de grande importância. Vamos utilizar dessa evolução para a solução desse problema que, de outra forma, não seria viável. Na verdade, uma das dúvidas que estamos motivados a responder aqui é se, mesmo com toda nossa capacidade computacional atual, o problema tem uma solução viável. Definiremos melhor adiante o que é uma solução dita *viável*.

Dados de Entrada e Saída

É um paradigma de computação se preocupar em deixar claro quais dados devem ser de entrada (*input*), para serem utilizados, processados, modificados e todo o resto que necessitar para gerar os dados de saída (*output*).

Existem dois tipos de dados que serão tratados aqui, os dados *teóricos* e os dados *reais*. Todos os dados reais que utilizamos são provenientes dos experimentos de RMN. Já os teóricos vem de conhecimento da Química, Física e Biologia.

Dados de Entrada [8]:

- quantidade de átomos: n;
- sequências de átomos:

$$N^1, C_\alpha^1, C^1, N^2, C_\alpha^2, C^2, ..., N^{n/3}, C_\alpha^{n/3}, C^{n/3};$$

- distância entre os átomos separados por uma ligação covalente: $\{d_{1,2}, d_{2,3}, ..., d_{n-1,n}\}$, onde $i = 2, ..., n$;
- distância entre os átomos separados por duas ligações covalentes: $\{d_{1,3}, d_{2,4}, ..., d_{n-2,n}\}$, $i = 3, ..., n$;
- distância entre átomos próximos, com no máximo 5 Å, fornecidas pela RMN - *único dado de entrada real*;

Dados de Saída:

- posições $x_1, ..., x_n \in \mathbb{R}^3$ dos n átomos da proteína.

Como as distâncias fornecidas pela RMN são um dado real, não são um valor exato. Dados reais são incertos, dependem de muitas variáveis, como a precisão da ferramenta que a mediu, o tipo de medida, quantas vezes fora medido, entre outras coisas. Esta incerteza é calculada e pode-se saber mais sobre em [16].

Complexidade do MDGP

É de extrema importância verificarmos o custo computacional de se tentar resolver o MDGP, pois, só com esta analise, pode-se dizer se o problema tem ou não uma solução viável. Caso não tenha, o trabalho pode ser encerrado por aqui. De nada nos adianta solução que não pode ser calculada.

Para poder calcular o custo computacional de uma solução matemática, deve-se verificar quantas vezes o "*núcleo*" do programa é computado, ou seja, se verifica quantas vezes a operação central do programa é realizada. Caso for realizada apernas uma vez, dizemos que o custo é baixo. Caso essa quantidade cressa proporcionalmente a medida que aumentamos os dados de entrada do problema (por exemplo, a medida que aumentamos a quantidade n de átomos da molécula), então dizemos que a dificuldade é linear. Para mais detalhes sobre custo computacional, leia [17]. Resolvemos, então, um MDGP simples — retirado de [8] — e verificamos qual seria seu custo.

Considere um MDGP restrito ao plano, onde temos um grafo $G = (V(G), E(G))$ tal que $V(G) = \{u, v, r, s\}$ e $E(G) = \{\{u, v\}, \{u, r\}, \{v, r\}, \{v, s\}, \{r, s\}, \{u, s\}\}$. Fixando u, v, r e s, ou seja, determinando $x_u, x_v, x_r \in \mathbb{R}^2$ de tal modo que $\|x_u - x_v\| = d_{uv}$, $\|x_u - x_r\| = d_{ur}$ e $\|x_v - x_r\| = d_{vr}$, podemos montar o seguinte sistema quadrático:

$$\|x_s - x_u\| = d_{us}$$

$$\|x_s - x_v\| = d_{vs}$$

$$\|x_s - x_r\| = d_{rs}$$

Elevando os termos ao quadrado,

$$\|x_s\|^2 - 2(x_s.x_u) + \|x_u\|^2 = d_{us}^2$$

$$\|x_s\|^2 - 2(x_s.x_v) + \|x_v\|^2 = d_{vs}^2$$

$$\|x_s\|^2 - 2(x_s.x_r) + \|x_r\|^2 = d_{rs}^2$$

subtraindo a primeira equação das outras duas, temos:

$$2(x_s.x_v) - 2(x_s.x_u) = \|x_v\|^2 - \|x_u\|^2 + d_{us}^2 - d_{vs}^2$$

$$2(x_s.x_r) - 2(x_s.x_u) = \|x_r\|^2 - \|x_u\|^2 + d_{us}^2 - d_{rs}^2$$

Colocando x_s em evidência para ficar claro as incógnitas

$$2(x_v - x_u)x_s = \|x_v\|^2 - \|x_u\|^2 + d_{us}^2 - d_{vs}^2$$

$$2(x_r - x_u)x_s = \|x_r\|^2 - \|x_u\|^2 + d_{us}^2 - d_{rs}^2$$

Por tanto, temos um sistema linear $Ax = b$, onde

$$A = 2\begin{bmatrix} x_{v1} - x_{u1} & x_{v2} - x_{u2} \\ x_{r1} - x_{u1} & x_{r2} - x_{u2} \end{bmatrix},$$

$$b = \begin{bmatrix} \|x_v\|^2 - \|x_u\|^2 + d_{us}^2 - d_{vs}^2 \\ \|x_r\|^2 - \|x_u\|^2 + d_{us}^2 - d_{rs}^2 \end{bmatrix},$$

e

$$x = \begin{bmatrix} x_{s1} \\ x_{s2} \end{bmatrix}.$$

Se a matriz A for inversível, temos uma única solução $x^* = A^{-1}b$. Logo, podemos concluir que se o grafo G de um MDGP for completo, podemos resolver o problema através da resolução de n sistemas lineares, sendo n proporcional ao número de vértices (átomos da molécula). Diz-se, então, que o problema pode ser resolvido em tempo *linear*.

No entanto, isso não acontece. Sabemos que o grafo molecular não é completo pois algumas distâncias não são informadas, assim, teremos apenas parte do grafo.

Admitindo que todas as distâncias dadas são valores *precisos* e que certamente representam as distâncias entre os átomos, o problema então terá uma solução. Para encontrarmos a solução devemos achar o mínimo global discutido na modelagem matemática, que, como vimos anteriormente, é inviável. A quantidade de mínimos locais cresce exponencialmente com a quantidade de vértices. Ou seja, o custo computacional para resolver um MDGP onde, suponha, todas as distâncias da RMN são realmente precisas, pode ser proporcional a 2^n, onde n é a quantidade de átomos. O que faz esse problema entrar na classificação *NP-difícil*. [17]

Por tanto, precisamos encontrar outra abordagem para solucionar nosso problema. Nosso próximo passo será tentar garantir que, ao menos, não tenhamos uma quantidade não enumerável de soluções do problema.

4.5 Estudando o Conjunto Solução de um MDGP

Com base no que temos, podemos analisar condições para garantir a finitude do conjunto solução do MDGP — conforme em [8]. A cardinalidade do conjunto solução de um MDGP (elementos são funções) pode ajudar na solução do problema. Sabemos que um MDGP pode ter conjunto solução vazio, solução única ou ser não-enumerável. Para essa analise, consideremos o mesmo problema da seção anterior, porém, nesse caso, os vértices não estão restritos ao plano. Temos, então, um MDGP com $V = \{u, v, r, s\}$ e $E = \{\{u, v\}, \{u, r\}, \{v, r\}, \{v, s\}, \{r, s\}, \{u, s\}\}$. Fixamos u, v, r e s; obtemos o mesmo sistema quadrático:

$$\|x_s - x_u\| = d_{us}$$

$$\|x_s - x_v\| = d_{vs}$$

$$\|x_s - x_r\| = d_{rs}$$

Fazendo o mesmo procedimento anterior de elevar ao quadrado e subtrair a primeira das outras duas, obtemos:

$$2(x_v - x_u)x_s = \|x_v\|^2 - \|x_u\|^2 + d_{us}^2 - d_{vs}^2$$

$$2(x_r - x_u)x_s = \|x_r\|^2 - \|x_u\|^2 + d_{us}^2 - d_{rs}^2$$

Até agora, nada mudou, porém, agora escrevendo explicitamente, temos:

$$\begin{bmatrix} x_{v1} - x_{u1} & x_{v2} - x_{u2} & x_{v3} - x_{u3} \\ x_{r1} - x_{u1} & x_{r2} - x_{u2} & x_{r3} - x_{u3} \end{bmatrix} \begin{bmatrix} x_{s1} \\ x_{s2} \\ x_{s3} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \|x_v\|^2 - \|x_u\|^2 + d_{us}^2 - d_{vs}^2 \\ \|x_r\|^2 - \|x_u\|^2 + d_{us}^2 - d_{rs}^2 \end{bmatrix}$$

Ou seja, não temos mais uma matriz 2×2, mas sim, uma matriz 2×3.

Reescrevendo o sistema de outra maneira

$$\begin{bmatrix} x_{s1} \\ x_{s2} \end{bmatrix} + \begin{bmatrix} x_{v3} - x_{u3} \\ x_{r3} - x_{u3} \end{bmatrix} \begin{bmatrix} x_{s3} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \|x_v\|^2 - \|x_u\|^2 + d_{us}^2 - d_{vs}^2 \\ \|x_r\|^2 - \|x_u\|^2 + d_{us}^2 - d_{rs}^2 \end{bmatrix}$$

E supondo que a matriz

$$A = \begin{bmatrix} x_{v1} - x_{u1} & x_{v2} - x_{u2} \\ x_{r1} - x_{u1} & x_{r2} - x_{u2} \end{bmatrix}$$

seja inversível, obtemos

$$\begin{bmatrix} x_{s1} \\ x_{s2} \end{bmatrix} = \frac{1}{2} A^{-1} \begin{bmatrix} \|x_v\|^2 - \|x_u\|^2 + d_{us}^2 - d_{vs}^2 \\ \|x_r\|^2 - \|x_u\|^2 + d_{us}^2 - d_{rs}^2 \end{bmatrix} - A^{-1} \begin{bmatrix} x_{v3} - x_{u3} \\ x_{r3} - x_{u3} \end{bmatrix} \begin{bmatrix} x_{s3} \end{bmatrix}$$

isso agora implica que não temos mais solução, pois, para cada valor de $x_{s3} \in \mathbb{R}$, obtemos valores para x_{s1} e x_{s2}.

Para encontramos uma solução, devemos retornar ao sistema quadrático e resolver uma equação do sistema linear acima.

Geometricamnete, temos a interseção de uma reta, dada por

$$\begin{bmatrix} x_{s1} \\ x_{s2} \end{bmatrix} = A - B \begin{bmatrix} x_{s3} \end{bmatrix}$$

onde

$$A = \frac{1}{2} \begin{bmatrix} x_{v1} - x_{u1} & x_{v2} - x_{u2} \\ x_{r1} - x_{u1} & x_{r2} - x_{u2} \end{bmatrix}^{-1} \begin{bmatrix} \|x_v\|^2 - \|x_u\|^2 + d_{us}^2 - d_{vs}^2 \\ \|x_r\|^2 - \|x_u\|^2 + d_{us}^2 - d_{rs}^2 \end{bmatrix}$$

$$B = \begin{bmatrix} x_{v1} - x_{u1} & x_{v2} - x_{u2} \\ x_{r1} - x_{u1} & x_{r2} - x_{u2} \end{bmatrix}^{-1} \begin{bmatrix} x_{v3} - x_{u3} \\ x_{r3} - x_{u3} \end{bmatrix} \begin{bmatrix} x_{s3} \end{bmatrix}$$

e uma esfera tal que

$$\|x_s - x_u\| = d_{us}$$

resultando em 3 possibilidades: conjunto vazio (a reta não intersepta a esfera); apenas um ponto (a reta é tangente à esfera); dois pontos (a reta é secante à esfera).

A discussão sugere dois aspectos importantes sobre a finitude:

1. Para cada vértice $s \in V$ a ser realizado em $\mathbb{R}$, devem existir arestas $u, s, v, s, r, s \in E$ tais que os vértices $u, v, r \in V$ já tenham sido realizados, para que se possa gerar um sistema quadrático, com $x_s \in \mathbb{R}^3$ como única incógnita, dado por

$$\|x_s - x_u\| = d_{us}$$

$$\|x_s - x_v\| = d_{vs}$$

$$\|x_s - x_r\| = d_{rs}$$

2. Para que esse sistema tenha no máximo duas soluções, a matriz linear do sistema obtido subtraindo uma equação das outras duas deve ter posto completo.

As duas informações cruciais, relacionadas ao vértice $s \in V$, são:

- Existem $u, v, r \in V$ tais que $\{u, s\}, \{v, s\}, \{r, s\} \subset E$,
- $x_u, x_v, x_r \in \mathbb{R}^3$ fazem parte de uma realização *parcial* válida.

A ideia que conecta os pontos anteriores está ligada ao conceito de *ordem nos vértices* do grafo $G = (V, E, d)$ do MDGP. Se existir uma ordem dos vértices que satifaz as condições 1 e 2 acima, podemos garntir, a menos de rotações e translações, que o cojunto solução do problema é finito.

Relembremos que resolver um MDGP é conseguir uma realização válida $x : V \to \mathbb{R}^3$ do grafo associado, o que implica, por sua vez, definir um ponto $x_s \in \mathbb{R}^3$, para cada $s \in V$, satisfazendo todas as equações do sistema

$$\forall u, v \in E, \|x_u - x_v\| = d_{uv}.$$

Portanto uma solução do problema pode ser representada, então, como um elemento do $\mathbb{R}^{3\|V\|}$

4.6 Ordenação Conveniente dos Vértices

Caso exista uma ordenação que respeite essas condições, podemos tornar a busca por uma solução para o problema factível. Porém, encontrar tal ordem, no geral, não é trivial e depende muito da natureza do problema. Encontrar essa ordem trata-se do *Problema da Ordem de Vértices Discretizáveis* — ou simplesmente DVOP (do inglês, *Discretization Vertex Order Problem*). Segue sua definição formal:

> ***Discretization Vertex Order Problem*:** Dado um grafo simples não direcionado $G = (V, E)$ e um escalar positivo K, estabelecer se existe uma ordem $<$ em V tal que:
>
> (a) o subconjunto $\{v \in V \mid \rho(<, v) \leq K\} \subset V$ é um K-clique em G, e
>
> (b) para todo $v \in V$ com posto $\rho(<, v) > K$, temos $|\delta(v) \cap \gamma(<, v)| \geq K$.
>
> Onde $\delta(v) = \{u \in V \mid \{u, v\} \in E\}$; para a ordem $<$ em V, tem-se $\gamma(<, v) = \{u \in V \mid u < v\}$ o conjunto de predecessores de v na ordem $<$ e $\rho(<, v) = |\gamma(v)| + 1$ o posto de v em $<$ (a ordem é total porque V é finito).

É importante mencionar que o DVOP é um problema NP-Completo [18] — o que não nos faz fugir da infactibilidade vista no MDGP. Entretanto, talvez uma ordem possa ser construída manualmente utilizando as características próprias das proteínas. Felizmente, já existe uma solução para esse problema [12]. Essa ordem se chama *Hand-crafted vertex order* — ou HC Order — e possui esse nome justamente por ter sido feita a mão, após um grande esforço desempenhado por Carlile Lavor e colaboradores, estudando as configurações dos aminoácidos na cadeia principal.

HC Order

Definiremos assa ordenação conforme [12]. Seja $G = (V, E, d)$ o grafo associado a cadeia principal de uma proteína ($\{N^k, C^k_\alpha, C^k\}$, para $k = 1, \ldots, p$), incluindo os átomos de oxigênio O^k, ligados ao C^k, e átomos de hidrogênio H^k e H^k_α, ligados ao N^k e C^k_α, respectivamente (conforme Figura 17, onde $p = 3$).

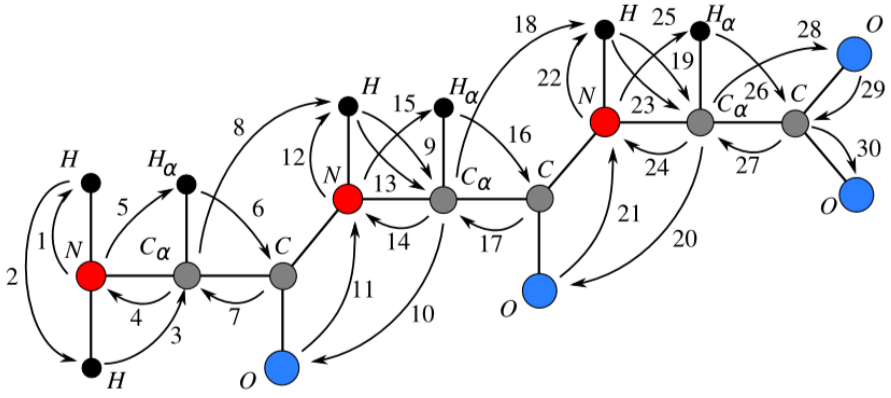

Figura 17: Esboço da ordenação HC [12].

Define-se a ordem HC como:

$$hc = \{N^1, H^1, H^{1'}, C^1_\alpha, N^1, H^1_\alpha, C^1, C^1_\alpha, \ldots,$$
$$H^i, C^i_\alpha, O^{i-1}, N^i, H^i, C^i_\alpha, N^i, H^i_\alpha, C^i, C^i_\alpha, \ldots,$$
$$H^p, C^p_\alpha, O^{p-1}, N^p, H^p, C^p_\alpha, N^p, H^p_\alpha, C^p, C^p_\alpha, O^p, C^p, O^{p'}\}$$

Onde, como na Figura 17, $i = 2, \ldots, p-1$, $H^{1'}$ é o segundo hidrogênio ligado ao N^1 e $O^{p'}$ é o segundo oxigênio ligado ao C^p.

Dado nossas hipóteses do problema, fica trivial perceber que essa ordenação respeita as condições (a) e (b), a menos de duas informações que podem não ser tão óbvias. Primeiramente: Toda distância entre hidrogênios relativamente próximos — inclusive entre H^{i-1}_α e H^i — são dados pela RMN. A segunda informação pertinente é que, para respeitar a condição (a) no átomo H^i, precisamos ter conhecimento do plano mostrado na Figura 7, que diz que os átomos em torno da ligação peptídica são coplanares — e, por tanto, pode-se descobrir a distância entre o H^i e C^{i-1}_α através das leis de senos e cossenos.

Nesse momento, é importante mencionar que o software PDBReader implementa a ordenação HC (como pode ser visto no componente 6 da Figura 15), ou seja, este trabalho também tem como resultado final uma automação do processo de criação de instâncias de proteínas ordenadas a partir de dados de moléculas reais vindas do wwPDB. Vale mencionar que essa é uma ferramenta valiosa do ponto de vista acadêmico, uma vez que possibilita a entrada de dados reais para testes — é comum, na literatura, a utilização de instâncias artificiais nos testes [19] [20].

A ordenação HC gera uma estrutura muitíssimo interessante para se trabalhar. De fato, por garantir que sempre tenhamos pelo menos um 3-clique em todo átomo com posto maior que três, podemos tentar encontrar a realização do próximo átomo

da sequência v_i com $i \geq 4$, utilizando a interseção das 3 esferas centradas nos três átomos anteriores v_{i-3}, v_{i-2} e v_{i-1} (já realizados) e com os respectivos raios iguais as distâncias $d_{i,i-3}, d_{i,i-2}$ e $d_{i,i-1}$ para o átomo i que se está tentando localizar.

Essas intersecções tem três possibilidades associadas (veja Figura 18): Ou não temos nenhum ponto de interseção entre elas — e isso não acontece, pois fere as hipóteses do problema, logo, só ocorre se existe alguma informação incorreta; Ou existe um ponto — donde os átomos são colineares e isso também nunca acontece, devido aos ângulos típicos das ligações; ou existem dois pontos onde as esferas se interceptam — este é o caso geral.

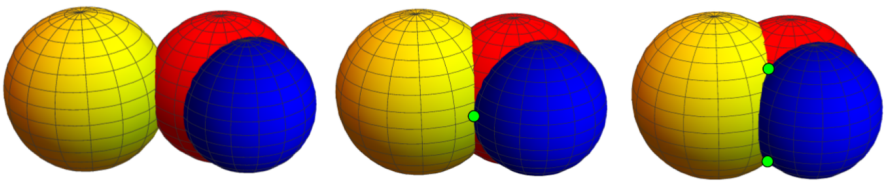

Figura 18: Interseção de três esferas [8].

Perceba, então, que além da ordenação dos vértices garantir a finitude do conjunto solução do problema, ela também organizar o espaço onde devemos fazer a busca por uma solução. Na verdade, a ordem induz uma estrutura de *árvore binária* no espaço de busca [1]. De fato, conforme discussão anterior entorno da Equação 2, sempre temos duas possibilidades para posicionar o próximo átomo da molécula [3] — pois, como já dito, possuímos dois $\text{sen}(\omega_{ijkl})$ associados, um positivo e outro negativo. Com isso, ganhamos uma discretização do problema (pois saímos espaço contínuo de soluções), nos permitindo definir uma nova variante para ele.

4.7 *Discretizable Molecular Distance Geometry Problem*

Esse é o problema central desse texto. Trata-se do MDGP monido de uma ordenação conveniente que permite sua discretização, como segue formalmente definido.

> ***Discretizable Molecular Distance Geometry Problem:*** Dados um grafo ponderado e não-direcionado $G = (V, E, d)$ associado a um MDGP, onde $d : E \longrightarrow \mathbb{R}_+$, o subconjunto de vértices iniciais $U_0 = \{v_1, v_2, v_3\}$ e uma relação de ordem total em V que satisfaça a seguinte relação de axiomas:
>
> 1. $G[U_0]$ é um clique em três vértices (iniciando a configuração);
> 2. para todo vértice v_i com posto $i = \rho(v_i) > 3$ nesta ordem, $G[U_i]$ é uma clique com quatro vértices (ordem de discretização, dada anteriormente) e
> 3. para cada vértice v_i, com posto $i = \rho(v_i) > 3$, juntamente com $\{v_{i-3}, v_{i-2}, v_{i-1}\}$, vale a desigualdade
>
> $$d_{i-3,i-1} < d_{i-3,i-2} + d_{i-2,i-1}, \qquad \text{(Desigualdade Triangular Estrita)}$$
>
> encontre uma imersão $x : V \longrightarrow \mathbb{R}^3$ tal que valha $\|x(v_i) - x(v_j)\| = d_{i,j}$, $\forall \{v_i, v_j\} \in E$.

4.7.1 Representações de Átomos em Coordenadas Internas

Como as partículas que estamos interessados estão localizadas no espaço tridimensional, podemos representa-las utilizando coordenadas cartesianas tridimensionais $x_1, ..., x_n \in \mathbb{R}^3$, onde x_n é a *realização* do n-ésimo átomo da molécula analisada.

Além da utilização das coordenadas cartesianas, também podemos usar outro sistema de coordenadas, mais condizente com os dados que temos a priori sobre a molécula — rever Figura 10. Tal sistema denomina-se *coordenadas internas* (e tem forte relação com as coordenadas esféricas).

As coordenadas internas de uma proteína são definidas pela distância entre os átomos $d_{1,2}, ..., d_{n-1,n}$, pelo ângulo planar $\theta_{1,3}, ..., \theta_{n-2,n}$ (formados por 3 átomos consecutivos) e pelos ângulos de torção $\omega_{1,4}, ..., \omega_{n-3,n}$ (formado por 4 átomos consecutivos), conforme ilustrado na Figura 19. O ângulo de torção é definido entre os planos formados pelos átomos $i-3, i-2, i-1$ e $i-2, i-1, i$, respectivamente. Assim, temos que ω varia no intervalo $[0, 2\pi]$ e θ de $[0, \pi]$ — assim como em um sistema de coordenadas esféricas — e também já vimos que as distâncias entre os átomos unidos por ligações covalentes são da ordem de 1,5Å.

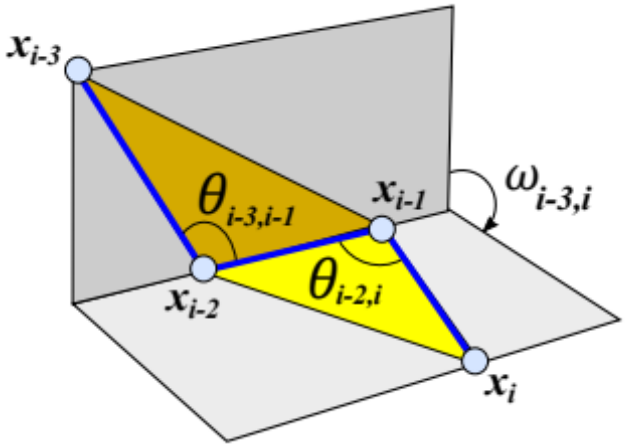

Figura 19: Ângulos planos e de torção [8]

Pode-se obter facilmente os ângulos planos pela *Lei dos Cossenos*, tendo em vista que conhece-se todas as distâncias que, por construção, representam os lados do triângulo — para mais detalhes, recorrer ao Apêndice A. Porém, descobrir o ângulo de torção associado (também chamado de ângulo diedral) pode se mostrar um grande problema.

O cosseno de um ângulo diedral pode ser dado em função de distâncias euclidianas e ângulos planos [11]. Sejam $x_i, x_j, x_k, x_l \in \mathbb{R}^3$ quaisquer quatro átomos consecutivos com coordenadas $(x_{i_1}, x_{i_2}, x_{i_3})$, $(x_{j_1}, x_{j_2}, x_{j_3})$, $(x_{k_1}, x_{k_2}, x_{k_3})$ e $(x_{l_1}, x_{l_2}, x_{l_3})$, respectivamente; Também r_{ab}, com $a, b \in \{i, j, k, l\}$ as distâncias entre os átomos x_a e x_b; e por último θ_{ijk}, o ângulo definido pelos átomos x_i, x_j, x_k e θ_{kji}, o ângulo definido pelos átomos x_k, x_j, x_i. Então, o cosseno do ângulo diedral ω_{ijkl} é dado por:

$$\cos(\omega_{ijkl}) = \frac{r_{ij}^2 + r_{jl}^2 - 2r_{ij}r_{jl}\cos(\theta_{ijk})\cos(\theta_{kjl} - r_{il}^2)}{2r_{ij}r_{jl}\text{sen}(\theta_{ijk})\text{sen}(\theta_{kjl})}. \tag{2}$$

Porém, não temos como definir exatamente o seno desse ângulo. Só podemos fazer $\text{sen}(\omega_{ijkl}) = \pm\sqrt{1-\cos(\omega_{ijkl})^2}$. Mas vamos deixar esse problema do ângulo de torção para depois — por enquanto supomos que temos os valores exatos de ω_{ijkl}.

Por tanto, conseguimos definir todo o problema a partir das coordenadas internas. Como nosso objetivo é obter as posições tridimensionais de cada átomo, nosso problema se restringiu em transformar as coordenadas internas em cartesianas. Para isso, utiliza-se uma série de operações lineares que estão descritas pelas matrizes abaixo.

Consideremos que as coordenadas do ponto $x_i \in \mathbb{R}^3, i = 1, ..., n$ são dadas por (x_{i1}, x_{i2}, x_{i3}), temos:

$$\begin{bmatrix} x_{i1} \\ x_{i2} \\ x_{i3} \\ 1 \end{bmatrix} = B_1 B_2 \cdots B_i \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix},$$

onde

$$B_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad B_2 = \begin{bmatrix} -1 & 0 & 0 & -d_{1,2} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$B_3 = \begin{bmatrix} -\cos\theta_{1,3} & -\text{sen}\theta_{1,3} & 0 & -d_{2,3}\cos\theta_{1,3} \\ \text{sen}\theta_{1,3} & -\cos\theta_{1,3} & 0 & d_{2,3}\text{sen}\theta_{1,3} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

e

$$B_i = \begin{bmatrix} -c_{\theta_i} & -s_{\theta_i} & 0 & -d_{i-1,i}c_{\theta_i} \\ s_{\theta_i}c_{\omega_i} & -c_{\theta_i}c_{\omega_i} & -s_{\omega_i} & d_{i-1,i}s_{\theta_i}c_{\omega_i} \\ s_{\theta_i}s_{\omega_i} & -c_{\theta_i}s_{\omega_i} & c_{\omega_i} & d_{i-1,i}s_{\theta_i}s_{\omega_i} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

dado $s_{\theta_i} = \text{sen}(\theta_{i-2,i})$, $c_{\theta_i} = \cos(\theta_{i-2,i})$, $s_{\omega_i} = \text{sen}(\omega_{i-3,i})$, $c_{\omega_i} = \cos(\omega_{i-3,i})$. Em B_i, $i = 4, ..., n$.

Perceba que B_i (chamada *Matriz de Torção*) é a matriz que engloba todas as operações necessárias para encontrar a i-ésima realização do i-ésimo átomo da molécula, tendo conhecimento de todas as matrizes $B_j \ \forall j < i$. Como é de grande importância o entendimento de tais operações que formam a B_i, deixa-se o Apêndice B para esta discussão.

Note também que fixando os comprimentos das ligações covalentes $d_{1,2}, d_{2,3}$ e o valor do ângulo plano $\theta_{1,3}$, os três primeiros átomos terão as coordenadas dadas por

$$x_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad x_2 = \begin{bmatrix} -d_{1,2} \\ 0 \\ 0 \end{bmatrix}, \quad x_3 = \begin{bmatrix} -d_{1,2} + d_{2,3}\cos\theta_{1,3} \\ d_{2,3}\text{sen}\theta_{1,3} \\ 0 \end{bmatrix}.$$

Dadas essas três realizações dos primeiros átomos, fixamos a base do sistema, evitando estruturas obtidas por meio de rotações e translações a partir de uma mesma estrutura.

Além dos dados teóricos sobre as estruturas dos aminoácidos, também teremos dados de distâncias entre átomos dadas de forma experimental, através de experimentos de RMN.

4.7.2 Espaço de Busca por Soluções

Note que, usando os valores de distâncias das cliques $\{v_{i-3}, v_{i-2}, v_{i-1}, v_i\}$ garantidas pelo DMDGP, temos gratuitamente todas as distâncias entre átomos consecutivos $d_{1,2}, \ldots, d_{n-1,n}$, donde podemos obter facilmente os ângulos planos $\theta_{1,3}, \ldots, \theta_{n-2,n}$ (Apêndice A).

Logo, sabendo que só precisamos das coordenadas internas para definir a estrutura 3D das proteínas — visto que podemos aplicar o conjunto de matrizes B_i para realizar a conversão dos sistemas de coordenadas — como temos todas as distâncias e ângulos planos, só nos falta verificar os ângulos de torção $\omega_{1,4}, \ldots, \omega_{n-3,n}$, que, como vimos, sempre possuímos dois possíveis ângulos associados ($\omega^1_{i-3,i}$ e $\omega^2_{i-3,i}$). Isso induz uma estrutura binária de decisão que é ilustrado na Figura 20, onde temos as duas posições possíveis (i e i') para o último vértice.

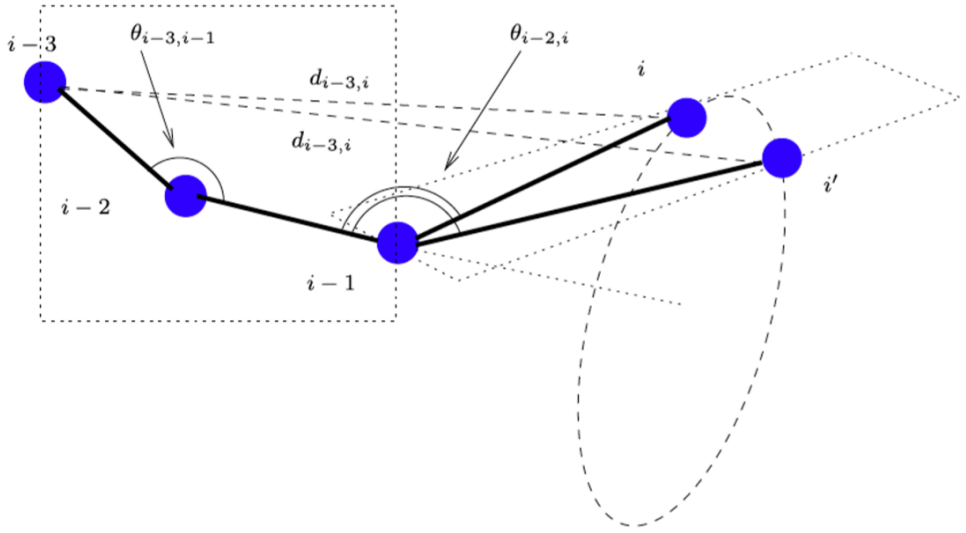

Figura 20: Duas possibilidades para o ângulo de torção [21].

Uma das conclusões mais interessantes que partem dessa análise é que não precisamos mais resolver os sistemas quadráticos que definem o DGP, problema fundamental de Geometria de Distâncias — que antes envolviam otimização contínua, complicada do ponto de vista computacional, com a quantidade de mínimos locais explodindo para moléculas demasiadamente grandes —, pois já temos definidas as posições possíveis para cada vértice. Logo, nosso problema se restringiu apenas a decidir quais são os ângulos de torção corretos, diminuindo drasticamente nosso espaço de busca por soluções.

5 *Branch-and-Prune*

Introduzimos o capítulo anterior com a definição do problema fundamental da nossa área, o *Distance Geometry Problem*, que não tardou ao ceder seu lugar de destaque para o *Molecular Distance Geometry Problem* — sendo exatamente o mesmo problema, porém, com o espaço fixado em $\mathbb{R}^3$. Em seguida fomos apresentados ao *Discretization Vertex Order Problem*, que, de posse de uma de suas soluções, nos permitiu construir uma versão discreta do nosso problema, chegando, finalmente, no *Discretizable Molecular Distance Geometry Problem* — que se trata do MDGP munido de uma ordem esperta para seus vértices.

A ordenação nos átomos da molécula, além de reduzir infinitamente o conjunto solução do problema [8], ainda induziu uma estrutura de **árvore binária** para os pontos que sobraram [1], que logo deu origem a um algorítimo que se aproveita dessa estrutura para reduzir ainda mais o conjunto solução do problema — o algorítimo Branch-and-Prune, tema desse capítulo.

5.1 O Algorítimo

Apresentado em 2007 [19], por Leo Liberti, Carlile Lavor e Nelson Maculan, este algorítimo (também chamado BP) consiste em uma estratégia numérica recursiva, que resolve o DMDGP eficientemente utilizando uma busca combinatória no espaço de busca de soluções, onde realiza-se vértice por vértice do sistema, seguindo a ordem dada, "podando" — isto é, descartando — todo sub-conjunto solução do sistema que não esteja de acordo com as informações pré-estabelecidas. Desde que ele foi publicado, tem se verificado tanto sua beleza matemática, quanto a sua eficiência numérica-computacional para resolver problemas em Geometria de Distâncias.

Como todo algorítimo, esse possui um conjunto de entradas e saídas.

Entradas: O grafo $G(V, E, d)$ que define o DMDGP — onde possuímos uma ordenação para V — e mais um escalar $\varepsilon \in \mathbb{R}$ que da a tolerância aceita no algoritmo (pois o BP não é um método exato, encontrando apenas soluções distantes a menos de ε das reais).

Para facilitar a utilização do algorítimo, também faremos uma distinção dos vértices que compõem o conjunto E: Sejam os subconjuntos $E_d, E_p \subset E$, tal que $E = E_d \cup E_p$ — denominados como, respectivamente, *arestas de discretização* e *arestas de poda* —, onde

$$E_d = \{\{v_i, v_j\} \in E : |i - j| \leq 3\} \text{ e } E_p = E - E_d.$$

Saídas: Uma árvore binária T, onde cada nó de nível i da árvore é uma realização possível do vértice $v_i \in V$, de tal forma que o caminho $C \subset T$ partindo da raiz (primeiro nó) até uma folha de nível $n = |V|$ da árvore seja uma solução para o problema, isto é, um conjunto de realizações de todos os átomos da molécula.

São três fases que definem o algoritmo: Inicialização, *Branching* e *Pruning* [1].

Inicialização

Esta etapa se preocupa com a inicialização da estrutura. Ela define a realização dos três primeiros átomos v_1, v_2 e $v_3 \in V$ da sequência, que são posicionados nas respectivas posições x_1, x_2 e $x_3 \in \mathbb{R}^3$, utilizando as operações contidas nas matrizes B_1, B_2 e B_3 (ver Capitulo 4.7.1). Obtendo os pontos

$$x_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad x_2 = \begin{bmatrix} -d_{1,2} \\ 0 \\ 0 \end{bmatrix}, \quad x_3 = \begin{bmatrix} -d_{1,2} + d_{2,3}\cos\theta_{1,3} \\ d_{2,3}\text{sen}\theta_{1,3} \\ 0 \end{bmatrix}.$$

Essas três primeiras posições estão associados biunivocamente com os três primeiros nós da arvore de busca T (representada por um grafo), que também é iniciada, conforme Figura 21

Figura 21: Inicialização de T [1].

Branching

Essa etapa está associada com o processo de "ramificação" de T [1]. Ou seja, supondo que já foram realizados os vértices $v_1, \ldots, v_{i-1}$ (onde $3 < i < |V|$), repeitando a ordenação em V, nosso objetivo é obter a realização $x_i = (x_{i1}, x_{i2}, x_{i3}) \in \mathbb{R}^3$ do vértice $v_i \in V$.

Para isso, basta calcularmos o produto matricial mostrado no Capitulo 4.7.1:

$$\begin{bmatrix} x_{i1} \\ x_{i2} \\ x_{i3} \\ 1 \end{bmatrix} = C_i \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix},$$

onde $C_i = C_{i-1}B_i = \prod_{j=1}^{i} B_j$ é dita o *produto acumulado* das matrizes de torção B_j, que, para $j > 3$, são dadas por

$$B_j = \begin{bmatrix} -\cos(\theta_{j-2,j}) & -\text{sen}(\theta_{j-2,j}) & 0 & -d_{j-1,j}\cos(\theta_{j-2,j}) \\ \text{sen}(\theta_{j-2,j})\cos(\omega_{j-3,j}) & -\cos(\theta_{j-2,j})\cos(\omega_{j-3,j}) & -\text{sen}(\omega_{j-3,j}) & d_{j-1,j}\,\text{sen}(\theta_{j-2,j})\cos(\omega_{j-3,j}) \\ \text{sen}(\theta_{j-2,j})\,\text{sen}(\omega_{j-3,j}) & -\cos(\theta_{j-2,j})\,\text{sen}(\omega_{j-3,j}) & \cos(\omega_{j-3,j}) & d_{j-1,j}\,\text{sen}(\theta_{j-2,j})\,\text{sen}(\omega_{j-3,j}) \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Porém, como $\text{sen}(\omega_{j-3,j}) = \pm\sqrt{1 - \cos(\omega_{j-3,j})^2}$, sempre temos duas matrizes de torção B_j^1 e B_j^2 associadas a cada vértice v_j. É a este fato que está associado o conceito de ramificação, pois, a cada matriz de torção temos como resultado um novo conjunto de realizações, que se visualiza como um novo ramo de T.

Ilustramos esse processo na Figura 22, que presume um DMDGP com $|V| = 6$. Ou seja, obtemos $2^{6-3} = 2^3 = 8$ possíveis soluções, dadas pelas suas três ramificações.

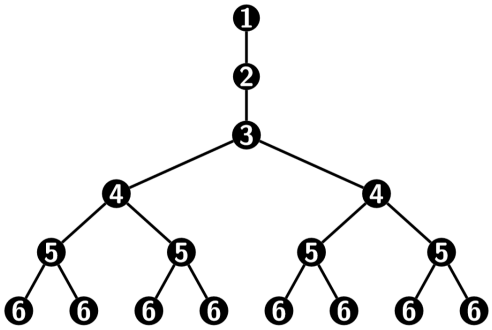

Figura 22: Árvore T completa de uma instância DMDGP com 6 vértices [1].

Fica claro aqui que o processo de ramificação garante que, no máximo, existem 2^{n-3} soluções possíveis para o problema. Isso colabora com a enumerabilidade e finitude do conjunto solução do problema, porém, também mostra que o número de soluções cresce de uma forma exponencial com o crescimento da molécula.

Pruning

Essa etapa tem como função diminuir drasticamente o conjunto solução do problema. Conseguimos isso ao classificar os diferentes ramos de T (gerados pelos diferentes ângulos de torção $\omega^1_{i-3,i}$ e $\omega^2_{i-3,i}$) como factíveis ou não e, então, "podando" os infactíveis.

Como já discutimos antes, para calcular as matrizes de torção da etapa anterior só precisamos dos dados do 3-clique garantido pelas hipóteses do DMDGP, ou seja, das distâncias $d_{i,i-3}, d_{i,i-2}$ e $d_{i,i-1}$ associadas aos elementos de E_d. Com isso, todas as distâncias associados aos elementos de E_p podem ser consideradas como dados adicionais para o problema.

Perceba que, para todo v_j tal que $\{v_i, v_j\} \in E_p$, se conhecemos a realização de v_j, a distância extra $d_{i,j}$ pode ser enxergada, do ponto de vista geométrico, como uma esfera extra àquelas outras três dadas pelo 3-clique. Isso gera a interseção de quatro esferas no $\mathbb{R}^3$, donde podemos ter apenas uma das duas possibilidades: Ou elas se interceptam em um único ponto (veja Figura 23) ou em nenhum.

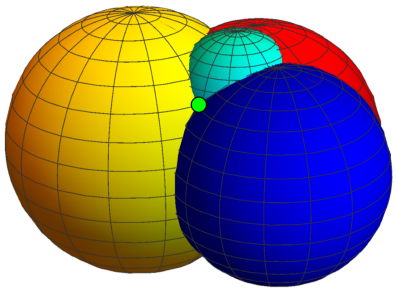

Figura 23: Interseção de quatro esferas no $\mathbb{R}^3$ [8].

E é este o princípio da factibilidade de um ramo: Sempre que estamos na etapa de *Branching* — ou seja, calculando uma realização x_i de v_i a partir de uma matriz de torção —, podemos verificar se existe uma distância extra, associada a algum

elemento $\{v_i, v_j\} \in E_p$ tal que $j < i$ e, caso exista, podemos verificar se $|x_i - x_j| < \varepsilon$. Caso for, significa que o ramo é factível e, caso não for, podemos descartar ("podar") todas as soluções associadas a sub-árvore definida por aquele ramo.

A etapa de *pruning* é ilustrada na Figura 24 (adaptada de [1]). Dando continuidade ao exemplo da Figura 22, agora temos $E_p = \{\{v_1, v_5\}, \{v_2, v_6\}\}$, o que nos permitiu testar a factibilidade dos ramos associados ao v_5 e ao v_6 e podar os infactíveis, diminuindo consideravelmente o conjunto solução.

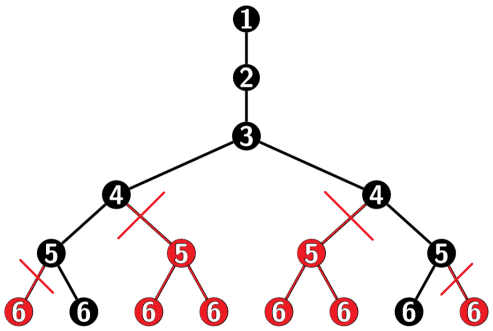

Figura 24: Árvore T de um DMDGP de 6 vértices com a poda evidenciada.

Perceba que o crescimento exponencial de soluções (2^{n-3}) da etapa de *branching* está associado com o DMDGP ser um problema NP-Difícil. Por conta disso é que esse algorítimo tem tanta beleza associada, isto é, apesar da grande complexidade de se resolver o problema original, o BP pode encontrar rapidamente essas soluções.

Mas uma dúvida ainda é importante: O conjunto $E_p = \varnothing$? Neste caso, $E = E_d$ e, portanto, todas as posições finais são factíveis trivialmente. Dessa forma, todas as 2^{n-3} realizações encontradas durante a etapa de *branching* representam soluções possíveis para o DMDGP, o que, é claro, é o nosso pior caso, pois o BP precisa continuar a exploração de todos os 2^{n-3} nós de T.

Por outro lado, se utilizarmos a ordenação hc apresentada no Capitulo 4.6 como ordem que define o DMDGP, nós garantimos que $E_p \neq \varnothing$. Essa é uma das vantagens dessa ordenação. Na verdade, muito mais do que não vazio, graças as propriedades geométricas das proteínas estudadas em [12] nós podemos enunciar o seguinte resultado (extraído de [12]):

Teorema 5.1 *Usando a ordenação hc, considerando que todos os ângulos e distâncias das ligações atômica estão fixadas nos seus valores de equilíbrio energético (essa é conhecida como hipótese de geometria rígida), que os átomos ao redor das ligações peptídicas formam um plano, que as posições possíveis para C_α^1 e C^j (com $j = 1, \ldots, p$) são únicas — devido a propriedade quiral do tetraedro formado por $\{N^1, H^1, H^{1\prime}, C_\alpha^1\}$ e $\{C_\alpha^j, N^j, H_\alpha^j, C^j\}$ — e, dado o conjunto de distâncias entre os pares de átomos de hidrogênio*

$$\{H^{1\prime}, H_\alpha^1\}, \ldots, \{H_\alpha^{i-1}, H^i\}, \{H^i, H_\alpha^i\}, \{H_\alpha^i, H^{i+1}\}, \ldots, \{H^p, H_\alpha^p\}$$

(onde $i = 2, \ldots, p-1$ e p é o número de aminoácidos que compõem a proteína), as ramificações na árvore de busca só ocorrem em átomos de hidrogênio dados por

$$\{H_\alpha^1, \ldots, H^i, H_\alpha^i, \ldots, H^p, H_\alpha^p\}.$$

5.2 Estrutura Algorítmica

Agora que a ideia por trás do BP está desenvolvida nos três passos anteriores (inicialização, *branching* e *pruning*) podemos apresenta-lo formalmente como um algorítimo de uma função recursiva, como segue.

Algorithm 1: Algoritmo BP [1] [19]

```
1  BranchAndPrune(T, v, i)
2  if i ≤ n − 1 then
3  |  Calcule as matrizes de torção B_i^1 e B_i^2;
4  |  Recupere a matriz de torção acumulada C_{i−1} referente ao nó-pai P(v);
5  |  Calcule as próximas matrizes de torção acumuladas C_i = C_{i−1}B_i^1 e
   |   C'_i = C_{i−1}B_i^2;
6  |  Utilize-as para calcular as posições x_i = C_i y e x'_i = C'_i y;
7  |  Seja λ = 1, ρ = 1;
8  |  foreach {v_j, v_i} ∈ E_p com j < i do
9  |  |  if (||x_j − x_i||^2 − d_ij^2)^2 > ε then
10 |  |  |  λ = 0;
11 |  |  end
12 |  |  if (||x_j − x'_i||^2 − d_ij^2)^2 > ε then
13 |  |  |  ρ = 0;
14 |  |  end
15 |  end
16 |  if λ = 1 then
17 |  |  Crie um nó z, armazenando C_i e x_i e fazendo P(z) = v e L(v) = z;
18 |  |  Faça T ← T ∪ {z};
19 |  |  BranchAndPrune(T, z, i + 1);
20 |  else
21 |  |  Faça L(v) = PRUNED;
22 |  end
23 |  if ρ = 1 then
24 |  |  Crie um nó z', armazenando C'_i e x'_i e fazendo P(z') = v e R(v) = z';
25 |  |  Faça T ← T ∪ {z'};
26 |  |  BranchAndPrune(T, z', i + 1);
27 |  else
28 |  |  Faça R(v) = PRUNED;
29 |  end
30 else
31 |  Solução está armazenada nos nós-pais de n a 1, em busca retrocedida.
32 end
```

5.3 Simulações Computacionais

Afim de ilustrar o que foi apresentado aqui, realizamos alguns experimentos computacionais, implementando o algorítimo BP em Linguagem C.

Utilizamos a biblioteca BLAS (*Basic Linear Algebra Subprograms*), onde encontra-se algumas boas implementações essenciais para que trabalhemos, como o produto matricial.

Inicialmente, precisamos conhecer um pouco sobre matrizes de distâncias [3]:

5.3.1 Representando distâncias com matrizes

É evidente a necessidade de uma forma de representar o conjunto de dados de entrada para o algorítimo BP. A definição a seguir nos ajuda com isso.

> **Matriz de Distâncias Euclidianas (EDM)**: Uma matriz $D_{n\times n}$ é dita MDE se existe um inteiro $k > 0$ e um conjunto $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^k$ tal que para $i, j \leq n$ temos
>
> $$D_{i,j} = \|x_i - x_j\|.$$

Perceba que utilizando uma matriz desse tipo podemos representar de forma completa o grafo que define um DMDGP: O conjunto de vértices V tem uma relação biunívoca com o conjunto de linhas da matriz e, além disso, a ordem dos vértices é dado pela ordem das linhas; Também temos o conjunto de arestas E, pois se $d_{ij} \in D_{n\times n} \neq 0$, então existe $\{v_i, v_j\} \in E$;

Inclusive, utilizando esse conceito, podemos definir um problema paralelo da área [1] que funcione a partir de uma extensão e um completamento das lacunas dessa matriz, tentando transformar-la em uma matriz de distâncias euclidianas quadrada, simétrica, completa e com diagonal nula. Esse é o *Euclidean Distance Matrix Completion Problem* (EDMCP). Perceba também que resolve-lo significa ter o grafo completo da análise do capítulo 4.4, deixando a solução do nosso problema com complexidade linear.

5.3.2 Medida de distância entre resultados

A qualidade das soluções aqui apresentadas foram averiguadas utilizando a medida *Mean Distance Error*, que é definida como se segue, a partir de [22].

> ***Mean Distance Error* (MDE)**: Para uma conformação $x = \{x_1, \ldots, x_n\}$ de uma instância com n vértices, cujo conjunto de distâncias disponíveis é dado por $m = |d|$, o MDE é definido por
>
> $$MDE(x) = \frac{1}{m} \sum_{i,j} \frac{|\|x_i - x_j\| - d_{i,j}|}{d_{i,j}}.$$

5.3.3 Experimentos

Para os primeiros testes, utilizamos como moléculas sintéticas as chamadas Lavor Instances [20], que possuem uma estrutura minimamente parecida com as reais (pois são baseadas nas posições de equilíbrio energético da molécula), garantindo sua validade acadêmica. Todos os testes foram rodados em um Intel Core 5 CPU 8600k @ 3.6 GHz com 8GB de RAM DDR4 2666Mhz (em *single channel*, logo 1333Mhz), rodando Linux e utilizamos $\varepsilon = 0.01$.

Um exemplo de matriz de distâncias euclidianas de uma Lavor Instance, de tamanho $|V| = 10$, é mostrado como se segue:

$$\begin{bmatrix}
0 & 1.526 & 2.4923 & 2.8914 & 3.4897 & 4.5857 & 4.3763 & 3.9408 & 2.6081 & 3.0647 \\
1.526 & 0 & 1.526 & 2.4923 & 2.9339 & 4.3520 & 4.6600 & 4.4179 & 3.0301 & 3.6325 \\
2.4923 & 1.526 & 0 & 1.5259 & 2.4923 & 3.8396 & 4.3445 & 3.8890 & 2.5309 & 2.9777 \\
2.8914 & 2.4923 & 1.5259 & 0 & 1.526 & 2.4923 & 2.9231 & 2.5068 & 1.4901 & 2.4843 \\
3.4897 & 2.9339 & 2.4923 & 1.526 & 0 & 1.5259 & 2.4923 & 2.9339 & 2.4827 & 3.8434 \\
4.5857 & 4.3520 & 3.8396 & 2.4923 & 1.5259 & 0 & 1.526 & 2.4923 & 2.9019 & 4.3030 \\
4.3763 & 4.6600 & 4.3445 & 2.9231 & 2.4923 & 1.526 & 0 & 1.5259 & 2.4923 & 3.8409 \\
3.9408 & 4.4179 & 3.8890 & 2.5068 & 2.9339 & 2.4923 & 1.5259 & 0 & 1.5259 & 2.4923 \\
2.6081 & 3.0301 & 2.5309 & 1.4901 & 2.4827 & 2.9019 & 2.4923 & 1.5259 & 0 & 1.5259 \\
3.0647 & 3.6325 & 2.9777 & 2.4843 & 3.8434 & 4.3030 & 3.8409 & 2.4923 & 1.5259 & 0
\end{bmatrix}$$

Ao utilizar nossa implementação do BP para calcular as possíveis instâncias dessa EDM, obtemos como resultado duas moléculas válidas. São elas:

x	y	z
0.000000	0.000000	0.000000
-1.526000	0.000000	0.000000
-2.035389	1.438471	0.000000
-1.466917	2.180162	-1.206403
-1.932825	1.498175	-2.489563
-1.447230	2.296543	-3.695997
0.077180	2.364681	-3.681643
0.543635	3.069462	-2.411061
0.016528	2.320614	-1.190381
0.482983	3.025395	0.080201

Tabela 3: Exemplar 1.

x	y	z
0.000000	0.000000	0.000000
-1.526000	0.000000	0.000000
-2.035389	1.438471	0.000000
-1.466917	2.180162	1.206403
-1.932825	1.498175	2.489563
-1.447230	2.296543	3.695997
0.077180	2.364681	3.681643
0.543635	3.069462	2.411061
0.016528	2.320614	1.190381
0.482983	3.025395	-0.080201

Tabela 4: Exemplar 2.

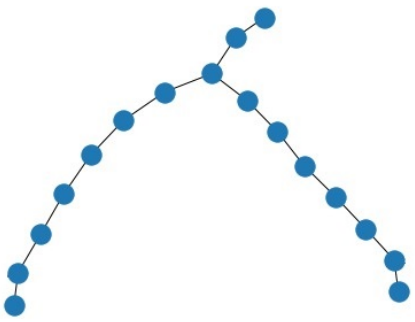

Figura 25: Árvore T do calculo da Lavor Instance com $|V| = 10$.

Como forma de verificar se o resultado condiz com o correto, calculou-se os MDE dos exemplares acima, que são mostrados na Tabela 5. O tempo para resolver esse BP foi da ordem de 0,616ms.

Exemplar	**MDE**
1	10^{-23}
2	10^{-23}

Tabela 5: Exemplar 2.

Também calculou-se as instâncias de outras Larvor Instances, com $|V|$ = 20, 50, 100, 200, 500 e 1000. Os resultados são mostrados na Tabela 6, onde, MDE média = $\frac{1}{n}\sum_{i=1}^{n} MDE_i$ e n é o número de soluções encontradas.

$\|V\|$	n	**MDE**	**time (ms)**
20	2	10^{-23}	0.231
50	4	10^{-13}	4.381
100	8	10^{-13}	26.181
200	8	10^{-23}	24.103
500	64	10^{-12}	431.113
1000	4608	10^{-11}	80851.054

Tabela 6: Exemplar 2.

Vale mencionar que a instância com $|V|$ = 2000 também foi gerada, porém, infelizmente a memória RAM do computador (8Gb) não foi suficiente para que o algorítimo chegasse ao fim, deixando instâncias com ordens de grandeza superiores para trabalhos futuros.

Dominado esta etapa, também utilizamos instâncias reais retiradas do repositório PDB (lidas com o auxílio do software PDBReader) para testar a nossa implementação do BP, feita em Linguagem C. Porém, mais uma vez nós não conseguimos terminar de executar o algorítimo para nenhuma das instâncias reais, pois, em média, elas possuem $|V|$ = 2000. Como visto nas Lavor Instances, solucionar exemplares desta ordem costuma extrapolar os 8Gb de memória que tínhamos a disposição.

6 Resultados e Discussão

Como vimos, não é possível, dada nosso suporte computacional atual, dar continuidade aos testes do software sem que haja alguma alteração. Uma boa estratégia seria um estudo sobre a otimização de memória necessária para implementar o BP. Isso pode ser observado no desenvolvimento do MD-Jeep [22], uma implementação em C feita por Antonio Mucherino, Leo Liberti e Carlile Lavor em 2010.

Um solução trivial pensada para contornar essa situação foi tentar manipular o valor de ε, de forma a produzir um filtro manual que diminuísse a quantidade de soluções (que estava crescendo exponencialmente). Porém, não obtivemos resultados satisfatórios. Pequenas oscilações em torno de um certo valor de ε (intrínsecos de cada molécula) faziam que, ou os resultados explodissem a memória, ou não fossem nenhum.

Outra alternativa para solucionar esse problema pode ser encontrada estudando as simetrias do DMDGP [1] [3]. Perceba, nas Tabelas 3 e 4, que os resultados possuem uma similaridade. Isso se dá devido as simetrias nas soluções de cada ramificação da arvore T, pois os resultados são simétricos (espelhados ao plano formado pelos três átomos anteriores [8]). Com isso, não precisamos buscar por todas as soluções da árvore de busca, pois, tendo uma solução, consegue-se a sua simétrica em tempo linear [1]. Isso é implementado em uma variação do BP, chamado *SymBP* [1].

Outras otimizações do algorítimo BP também podem ser encontrados na literatura, como uma versão que utiliza um paradigma Dividir e Conquistar [1], onde se constrói uma implementação paralela (Multithreading), que se utiliza das simetrias para produzir vários SymBP em paralelo. Um estudo sobre essas implementações poderiam ser úteis para otimizar nosso algorítimo.

7 Considerações Finais

Com isso concluímos um estudo elementar sobre o Discretizable Molecular Distance Geometry Problem. Acreditamos que nos cabe, no fim de um projeto como esse, olhar para as propostas levantadas inicialmente e verificar se elas foram cumpridas. Seguem o conjunto de objetivos específicos desse projeto, monidos de breve conclusão:

1. Entender as estruturas básicas de proteínas:

   Este fora feito de forma intensa, resultando no capítulo 3 deste documento;

2. Relacionar-se eficientemente com o PDB (*Protein Data Bank*) - como extrair os dados computacionais que servirão de insumos:

   Apresentou-se este repositório junto do capítulo 3, devido sua proximidade temática. Vale mencionar que lá também fora apresentado o software PDBReader, implementado pelo autor deste documento, que visa automatizar o processo de extração dos dados de distâncias do repositório PDB;

3. Compreender o DMDGP e sua estrutura de ordenamento dos vértices:

   Podemos considerar essa frase como um resumo do capítulo 4. Peço atenção especial no estudo da ordenação HC, feita no fim do capítulo, devido sua grande importância no tema;

4. Conhecer todos os passos do algoritmo BP:

   Feito no capítulo 5, onde estudou-se todos os passos referentes a esse algorítimo: Inicialização, *branching* e *pruning*;

5. Simular, computacionalmente, o algoritmo BP com instâncias artificialmente geradas, como descrito na Literatura, dominando cada passo utilizado:

   Feito no fim do capítulo 5, com resultados condizentes com a literatura;

6. Aplicar o Algoritmo BP estudado em estruturas proteicas como instâncias reais do problema:

   Infelizmente não conseguimos poder computacional para que nosso algorítimo conseguisse calcular estas instâncias. Há uma pequena discussão, no Capitulo 7, sobre como contornar essa situação. Mas nos cabe aqui frisar que um estudo sobre otimização de memória computacional poderia ser útil para que nossa implementação realizasse os testes de forma adequada.

   Vale lembrar, porém, que uma parte importante desse resultado foi desenvolvido ao criar o software PDBReader, possibilitando a extração dos dados de moléculas reais do repositório PDB.

Como implementações futuras, deseja-se estudar mais sobre as distâncias intervalares que aparecem devido a ordenação HC [12] (detalhe este que não fora considerado na nossa implementação). Uma ferramenta que tem demonstrado potencial para auxiliar nessa passagem é a Geometria Conforme [8].

Também deseja-se poder fazer um novo estudo envolvendo o problema de geometria de distâncias aplicado a outras áreas do conhecimento, como localização de sensores [3].

Referências

[1] Felipe Delfini Caetano Fidalgo. *Dividindo e conquistando com simetrias em geometria de distâncias*. PhD thesis, UNICAMP, Campinas, SP, Fevereiro 2015.

[2] K. Wüthrich. Protein structure determination in solution by nuclear magnetic resonance spectroscopy. *American Association for the Advancement of Science*, 243(4887):45–50, Jan 1989.

[3] Leo Liberti, Carlile Lavor, Nelson Maculan, and Antonio Mucherino. Euclidean distance geometry and applications. *Society for Industrial and Applied Mathematics*, 56(1):3–69, February 2014.

[4] Leonhard Euler. Leonhard euler and the königsberg bridges. *Scientific American*, 189(1):66–72, 1953.

[5] J. A. Bondy and U. S. R. Murty. *Graph Theory With Applications*. Elsevier Science Publishing, New York, 5 edition, 1982.

[6] Paulo Oswaldo Boaventura Netto. *Grafos*. Blucher, São Paulo, 5 edition, 2012.

[7] Jayme Luiz Szwarcfiter. *Teoria computacional de grafos: Os algoritmos*. Elsevier Brasil, 2018.

[8] C. Lavor, N. Maculan, M. Souza, and R. Alves. *Álgebra e Geometria no Cálculo de Estrutura Molecular*. IMPA, Rio de Janeiro, RJ, 31º colóquio brasileiro de matemática edition, 2017.

[9] David L Nelson and Michael M Cox. *Lehninger principles of biochemistry*. W.H.Freeman and Company, 2013.

[10] GN Ramachandran, AS Kolaskar, C Ramakrishnan, and V Sasisekharan. The mean geometry of the peptide unit from crystal structure data. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, 359(2):298–302, 1974.

[11] CC Lavor. *Uma abordagem determinística para minimização global da energia potencial de moléculas*. PhD thesis, PhD thesis, COPPE/UFRJ, Rio de Janeiro, 2001.

[12] Carlile Lavor, Leo Liberti, Bruce Donald, Bradley Worley, Benjamin Bardiaux, Thérèse E Malliavin, and Michael Nilges. Minimal nmr distance information for rigidity of protein graphs. *Discrete Applied Mathematics*, 256:91–104, 2019.

[13] wwPDB.org. Worldwide protein data bank.

[14] Emerson Castelani; Repositório da MolecularConformation.jl; Acesso em 11/08/2019. Url = "https://github.com/evcastelani/molecularconformation.jl".

[15] Peter A. Rinck. *The history of MR imaging*. The Round Table Foundation, Germany, 11th edition, 2017.

[16] Não lembro o autor. *Um livro de tratamento de dados estatístico Tenho que verificar certinho o titulo na ufsc, é um livro de lá*. Favor relevar, Germany, whatever edition, 2017.

[17] Não lembro o autor. *Custo computacional, teoria da computação,Tenho que verificar certinho o titulo na ufsc, é um livro de lá.* Favor relevar, Germany, whatever edition, 2017.

[18] Andrea Cassioli, Oktay Günlük, Carlile Lavor, and Leo Liberti. Discretization vertex orders in distance geometry. *Discrete Applied Mathematics*, 197:27–41, 2015.

[19] Leo Liberti, Carlile Lavor, and Nelson Maculan. A branch-and-prune algorithm for the molecular distance geometry problem. *International Transactions in Operational Research*, 15(1):1–17, 2008.

[20] Carlile Lavor. On generating instances for the molecular distance geometry problem. In *Global optimization*, pages 405–414. Springer, 2006.

[21] Carlile Lavor, Leo Liberti, Nelson Maculan, and Antonio Mucherino. The discretizable molecular distance geometry problem. *Computational Optimization and Applications*, 52(1):115–146, 2012.

[22] Antonio Mucherino, Leo Liberti, and Carlile Lavor. Md-jeep: an implementation of a branch and prune algorithm for distance geometry problems. In *International Congress on Mathematical Software*, pages 186–197. Springer, 2010.

[23] Alfredo Steinbruch and Paulo Winterle. *Geometria Analítica.* Makron Books, São Paulo, SP, 2a edition, 1987.

[24] Elon Lages Lima. *Álgebra Linear.* SBM, Rio de Janeiro : IMPA, 1a edition, 2014.

Apêndice A

Lei dos Cossenos e Ângulos Entre dois Vetores no $\mathbb{R}^3$

Leis dos Cossenos

A lei dos cossenos é uma propriedade trigonométrica válida para qualquer triângulo, permitindo encontrar o valor de um dos seus lados conhecendo apenas os outros lados e um ângulo. Porém, aqui utilizaremos a ideia reversa, onde, nesse caso, saberemos os lados e queremos descobrir os ângulos.

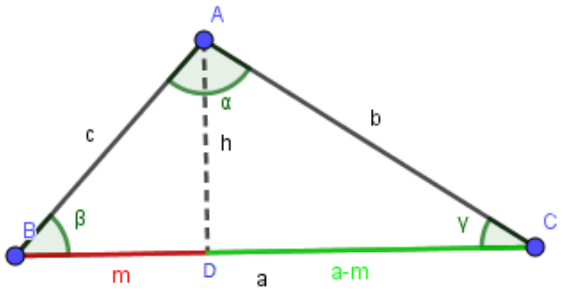

Figura 26: Triângulo para ilustrar a lei dos cossenos.

- **Demonstração Leis dos Cossenos:**

  Dado um triângulo qualquer, traça-se uma altura relativa ao lado a. Aplicando o *Teorema de Pitágoras* no ΔABD:

  $$c^2 = m^2 + h^2 \rightarrow h^2 = c^2 - m^2 \tag{3}$$

  Aplicando novamente *Pitágoras*, porém, em ΔADC, obtemos:

  $$b^2 = h^2 + (a - m)^2 \tag{4}$$

  Substituindo na equação 4 o valor de h^2 obtido em 3:

  $$b^2 = c^2 - m^2 + a^2 - 2am + m^2$$

  $$b^2 = c^2 + a^2 - 2am$$

  Analisando a Figura 26, pode-se perceber que $\frac{m}{c} = \cos\beta$, então:

  $$b^2 = c^2 + a^2 - 2ac\cos\beta$$

  Analogamente, obtém-se:

  $$c^2 = a^2 + b^2 - 2ab\cos\gamma$$

  $$a^2 = b^2 + c^2 - 2bc\cos\alpha$$

  Note também que se o argumento dos cossenos for $\frac{\pi}{2}$ recaímos no Teorema de Pitágoras. ■

- **Ângulos Entre 2 Vetores:**

  Sejam dois vetores $\vec{u}$ e $\vec{v} \in \mathbb{R}^2$, representados na Figura 27

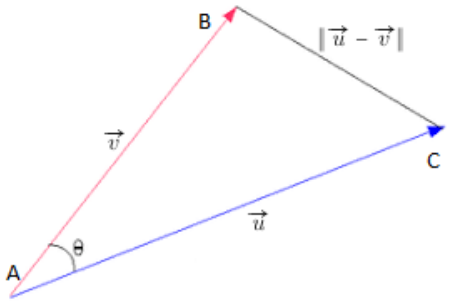

Figura 27: Diferença entre vetores u e v

Para encontrarmos o angulo θ utilizaremos a lei dos cossenos aplicada a ΔABC:

$$\|\vec{u} - \vec{v}\|^2 = \|\vec{u}\|^2 + \|\vec{v}\|^2 - 2\|\vec{u}\|\|\vec{v}\|\cos\theta \tag{5}$$

Utilizando a definição do produto escalar [23]

$$\|\vec{u} - \vec{v}\|^2 = \|\vec{u}\|^2 + \|\vec{v}\|^2 - 2\vec{u}\,\vec{v} \tag{6}$$

Comparando a equação 5 com a 6, obtemos trivialmente

$$\|\vec{u}\|^2 + \|\vec{v}\|^2 - 2\|\vec{u}\|\|\vec{v}\|\cos\theta = \|\vec{u}\|^2 + \|\vec{v}\|^2 - 2\vec{u}\,\vec{v}$$

$$\vec{u}\,\vec{v} = \|\vec{u}\|\|\vec{v}\|\cos\theta$$

Logo,

$$\cos\theta = \frac{\vec{u}\,\vec{v}}{\|\vec{u}\|\|\vec{v}\|}$$

■

Apêndice B

Matrizes como Transformações Lineares e Sobre B_i

Quando se começa a aprender sobre matrizes, no ensino médio, normalmente este é um assunto que lhe é apresentado como algo que caiu do céu, difícil de engolir. Na verdade, não é raro um graduando de matemática ter dificuldades para entende-las.

Matrizes

Uma matriz real $\mathbf{a} = [a_{ij}]$ de dimensões $m \times n$ é definida como uma lista de números a_{ij}, onde $i \mid 1 < i < m$ e $j \mid 1 < j < n$ são os índices que, juntos, identificam unicamente cada elemento. Costuma-se representar a matriz $\mathbf{a}$ como um quadro de $m \cdot n$ elementos, onde m é o número de linhas e n é o número de colunas, de forma que o elemento a_{ij} situa-se no cruzamento entre a i-ésima linha e a j-ésima coluna, como segue:

$$\mathbf{a} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

Seja $M(m \times n)$ um conjunto de matrizes de tal forma que todas as matrizes de dimensão $m \times n$ estejam dentro dele. Podemos definir algumas operações neste conjunto. São elas:

- **Soma de Matrizes:** Sejam duas matrizes $\mathbf{a}, \mathbf{b} \in M(m \times n)$, define-se a soma de $\mathbf{a} = [a_{ij}]$ com $\mathbf{b} = [b_{ij}]$ como

$$\mathbf{a} + \mathbf{b} = [a_{ij} + b_{ij}]$$

- **Multiplicação por Escalar:** Sejam $\mathbf{a} \in M(m \times n)$ e $\lambda \in \mathbb{R}$ um escalar qualquer, o produto de $\mathbf{a} = [a_{ij}]$ por λ é definido como

$$\lambda \cdot \mathbf{a} = \lambda[a_{ij}] = [\lambda a_{ij}]$$

Tendo estas duas operações construídas, podemos definir também:

- **Existência de elemento nulo:** Define-se como *matriz nula* $\mathbf{0} \in M(m \times n)$ a matriz formada exclusivamente por $m \cdot n$ zeros.

- **Existência do elemento oposto para adição:** Seja $\mathbf{a} \in M(m \times n)$ uma matriz da forma $\mathbf{a} = [a_{ij}]$, define-se o elemento oposto de $\mathbf{a}$ como $-\mathbf{a} = [-a_{ij}]$.

Tendo tais definições, chegamos em um resultado interessante.

Proposição: O conjunto $M(m \times n)$ é um espaço vetorial. [24]

Para chegar na conclusão acima, basta tomar as definições dadas e perceber que podemos escrever cada linha e coluna de uma matriz como vetores, ou seja, $\forall \mathbf{a} \in M(m \times n)$, $\exists$ um *vetor-linha* $a_i^l = (a_{i1}, a_{i2}, \ldots, a_{in})$ e um *vetor-coluna* $a_j^c = (a_{1j}, a_{2j}, \ldots, a_{mj})$ que representam a i-ésima linha e j-ésima coluna de $\mathbf{a}$, respectivamente.

Para que possamos continuar, necessitamos introduzir mais um grande tópico de álgebra linear, as *transformações lineares*.

Transformações Lineares

Sejam E, F espaços vetoriais. Uma transformação linear $A : E \longrightarrow F$ é uma correspondência que associa a cada vetor $v \in E$ um vetor $A(v) = A \cdot v = Av \in F$ de modo que valham, para quaisquer $u, v \in E$ e $\alpha \in \mathbb{R}$, as relações:

$$A(u + v) = A(u) + A(v),$$
$$A(\alpha \cdot v) = \alpha Av.$$

Denominamos o vetor $A \cdot v$ como imagem (ou transformado) de v pela transformação A [24]. Perceba que transformação linear é uma classificação de um conjunto especial de funções, ou seja, aquelas que respeitam os dois requisitos acima.

Teorema: Sejam E, F espaços vetoriais e B uma base de E. A cada vetor $x \in B$, façamos corresponder (de maneira arbitrária) um vetor $b \in F$. Então existe uma única transformação linear $A : E \longrightarrow F$ tal que $A \cdot x = b$ para cada $x \in B$.

A demonstração deste teorema se encontra em [24].

Graças a este enunciado, para podermos definir uma transformação linear $A : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ basta escolher para cada $j = 1, \ldots, n$ um vetor $v_j = (a_{1j}, a_{2j}, \ldots, a_{mj}) \in \mathbb{R}^m$ de tal forma que $v_j = A \cdot e_j$ é a imagem do j-ésimo vetor da base canônica, $e_j = (0, \ldots, 1, \ldots, 0)$, pela transformação linear A. Logo, fica determinada a imagem $A \cdot v$ de qualquer vetor $v = (x_1, \ldots, x_n) \in \mathbb{R}^n$. Com efeito, tem-se $v = x_1 e_1 + \cdots + x_n e_n$, por tanto podemos escrever

$$A \cdot v = A(\sum_{j=1}^{n} x_j e_j) = \sum_{j=1}^{n} x_j A \cdot e_j = \sum_{j=1}^{n} (a_{1j}x_j, a_{2j}x_j, \ldots, a_{mj}x_j)$$

Aplicando então o somatório em cada elemento, obtemos

$$A \cdot v = (\sum_{j=1}^{n} a_{1j}x_j, \sum_{j=1}^{n} a_{2j}x_j, \ldots, \sum_{j=1}^{n} a_{mj}x_j)$$

Ou seja, cada componente $\sum_{j=1}^{n} a_{ij}x_j$, onde $i \in \mathbb{N} \mid 1 < i < m$, representa a componente imagem de v pela transformação A. Podemos chamar cada uma dessas componentes de y_i, concluindo que

$$A(x_1, x_2, \ldots, x_n) = (y_1, y_2, \ldots, y_m)$$

onde

$$\begin{array}{ccccccccc} y_1 & = & a_{11}x_1 & + & a_{12}x_2 & + & \cdots & + & a_{1n}x_n \\ y_2 & = & a_{21}x_1 & + & a_{22}x_2 & + & \cdots & + & a_{2n}x_n \\ \vdots & & \vdots & & \vdots & & \vdots & & \vdots \\ y_m & = & a_{m1}x_1 & + & a_{m2}x_2 & + & \cdots & + & a_{mn}x_n \end{array}$$

Interessante, não? Note: Acaba-se de concluir que uma transformação linear $A : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ pode ser inteiramente representada pela matriz $\mathbf{a} = [a_{ij}] \in M(m \times n)$, onde os vetores-coluna dessa matriz são as imagens $A \cdot e_j$ dos vetores da base canônica de $\mathbb{R}^n$ e os vetores-linha são dados pelas componentes da imagem $A \cdot v$ de um vetor arbitrário $v = (x_1, \ldots, x_n) \in \mathbb{R}^n$, que podemos representar por um vetor $w = (y_1, \ldots, y_m) \in \mathbb{R}^m$. [24] Diz-se que $\mathbf{a}$ é *matriz da transformação* A relativa as bases canônicas de $\mathbb{R}^n$ e $\mathbb{R}^m$.

Operando Matrizes

As matrizes como vistas até aqui não apresentam muitas dificuldades, são apenas operações de elemento a elemento, sem grandes complicações. Pode-se até imagina-las apenas como uma metodologia de organizar uma grande quantidade de dados, de forma a expressa-los mais facilmente em um padrão de quadro bidimensional, ou seja, contradizendo os argumentos apresentados na introdução deste apêndice. Nesta seção introduziremos a operação que traz para as matrizes esse ar de mistério, estimulando dúvidas: *O produto de matrizes.*

Definição: Sejam $\mathbf{a} = [a_{ij}] \in M(m \times n)$ e $\mathbf{b} = [b_{ij}] \in M(n \times p)$ matrizes quaisquer de foma que o número de colunas n da matriz $\mathbf{a}$ seja o mesmo que o número de linhas n da matriz $\mathbf{b}$. O produto de $\mathbf{a}$ por $\mathbf{b}$ (nessa ordem) é definido como $\mathbf{ab} = \mathbf{c} = [c_{ij}] \in M(m \times p)$, onde c_{ij} é o ij-ésimo elemento da matriz $\mathbf{c}$ e é dado por

$$c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{in}b_{nj} = \sum_{k=1}^{n} a_{ik}b_{kj}$$

onde $i, j \in \mathbb{N}$ são tais que $1 < i < m$ e $1 < j < p$.

Também podemos definir o elemento c_{ij} usando como perspectiva o *produto interno de vetores* [24], onde a_i^l é o i-ésimo vetor-linha de $\mathbf{a}$ e b_j^c é o j-ésimo vetor-coluna de $\mathbf{b}$, então

$$c_{ij} = \langle a_i^l, b_j^c \rangle = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{in}b_{nj}$$

Resultando em

$$\mathbf{c} = \begin{bmatrix} \langle a_1^l, b_1^c \rangle & \langle a_1^l, b_2^c \rangle & \cdots & \langle a_1^l, b_p^c \rangle \\ \langle a_2^l, b_1^c \rangle & \langle a_2^l, b_2^c \rangle & \cdots & \langle a_2^l, b_p^c \rangle \\ \vdots & \vdots & \vdots & \vdots \\ \langle a_m^l, b_1^c \rangle & \langle a_m^l, b_2^c \rangle & \cdots & \langle a_m^l, b_p^c \rangle \end{bmatrix}$$

Perceba que o produto de matrizes, no geral, não é comutativo, ou seja, $\mathbf{ab} \neq \mathbf{ba}$, salvo exceções, e aqui está o problema no seu estudo. Não é fácil digerir a ideia de um produto não comutativo, ainda mais sem entender de onde ele vem. Neste terreno não há intuição geométrica que ajude e nem tente extrapolar este conceito para a dimensão infinita, caso contrário perderá agradáveis noites de sono.

Mas nem tudo está perdido, existem algumas aplicações que nos ajudam a entender melhor esta poderosa ferramenta, como segue.

Exemplo: Suponha uma transformação linear $A : \mathbb{R}^n \longrightarrow \mathbb{R}^m$, sabemos que esta transformação pode ser inteiramente representada por uma matriz $\mathbf{a} = [a_{ij}] \in M(m \times n)$, conforme visto na seção anterior, logo, a equação $Ax = b$ pode ser inteiramente representada como o produto de matrizes $\mathbf{ax} = \mathbf{b}$, onde os vetores $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$ e $b = (b_1, \ldots, b_m) \in \mathbb{R}^m$ passam a ser considerados como matrizes $n \times 1$ e $m \times 1$, respectivamente, ou seja, como vetores-coluna, logo

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

Com isso concluí-se que podemos aplicar algumas funções especiais, as quais respeitam os dois requisitos que definem transformações lineares, utilizando matrizes. Isso se torna deveras interessante quando se necessita resolver problemas de forma computacional que utilizem transformações lineares, pois a utilização de matrizes traz ganhos computacionais significativos [24].

Outra aplicação que demonstra a verdadeira importância da utilização das matrizes é quando necessita-se concatenar transformações lineares, ou seja compor funções.

Definição: Dadas as transformações lineares $A : E \longrightarrow F$, $B : F \longrightarrow G$, onde o domínio de B coincide com a imagem de A, define-se o *produto* $BA : E \longrightarrow G$ colocando, $\forall v \in E$, $(BA)v = B(Av)$, como representado na Figura 28.

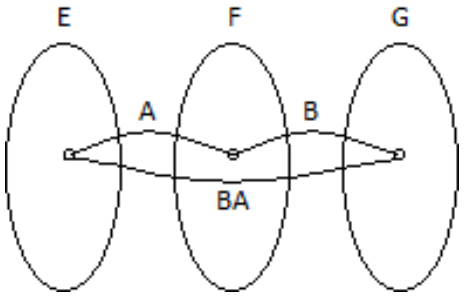

Figura 28: Transformação linear composta BA

Assim, podemos enunciar o seguinte teorema.

Teorema: Sejam $A : E \longrightarrow F$ e $B : F \longrightarrow G$, $U = \{u_1, \ldots, u_p\} \subset E$, $V = \{v_1, \ldots, v_n\} \subset F$ e $W = \{w_1, \ldots, w_m\} \subset G$ tal que $\mathbf{a} \in M(m \times p)$ é a matriz de A nas bases U, V e $\mathbf{b} \in M(m \times n)$ é a matriz de B nas bases V, W, então a matriz de $BA : E \longrightarrow G$ nas bases U, W é o produto $\mathbf{ba} \in M(m \times p)$ das matrizes $\mathbf{b}$ e $\mathbf{a}$. [24]

Ou seja, se tivermos duas transformações lineares $A : E \longrightarrow F$ e $B : F \longrightarrow G$, podemos multiplicar suas representações matriciais $\mathbf{b}$ e $\mathbf{a}$ afim de obter o equivalente matricial da função composta $BA : E \longrightarrow G$. Este teorema é muito importante, podendo simplificar um algorítimo de n operações lineares $\mathbf{a_i} \mid 1 < i < n$ em apenas uma aplicação da forma $\mathbf{a_r x} = \mathbf{b}$, onde $\mathbf{a_r}$ é a matriz resultante de $\prod_{i=1}^{n} \mathbf{a}_i$.

Assim concluímos o estudo sobre transformações lineares e suas representações como matrizes. Podemos aplicar estes conhecimentos em um caso específico de forte interesse neste documento, a matriz B_i.

Sobre a Matriz B_i

Ao trabalhar-se com dados vindos de experimentos de RMN, é fácil perceber que o sistema de coordenadas cartesianas não é a melhor forma para representar estes dados, pois não se sabe de antemão qual a posição de cada átomo estudado. Como alternativa, utiliza-se as coordenadas internas que, de forma muito semelhante ao

sistema de coordenadas esféricas (normalmente visto em calculo no $\mathbb{R}^3$), faz uso de distâncias e ângulos para representar localizações.

Infelizmente temos um problema: as coordenadas internas não são muito agradáveis. Necessitam de constantes manipulações trigonométricas e possuem complicada interpretação geométrica, pois não dependem de um referencial fixo, ou seja, esse sistema de coordenadas não possui uma origem bem definida como em um plano cartesiano., diferentemente, nesse sistema a referencia se dá sempre partindo da localização atual. Pode-se dizer que a referência é sempre de dentro para fora. Talvez seja daí que venha o termo "coordenadas *internas*", pois o referencial é o interior do ponto que se estuda.

Dados os motivos citados, salvo a liberdade poética do autor, temos forte interesse de apresentar os resultados finais de um experimento de RMN em coordenadas cartesianas, ou seja, deve-se transformar coordenadas internas em cartesianas. Infelizmente isso não é uma tarefa fácil e depende muito da configuração espacial que compõe a molécula estudada. Pode-se dizer que o PGDM resume-se nesta transformação de coordenadas. Perceba a importância de se estudar a matriz que possui este papel, no caso, a B_i.

Conforme apresentado na seção 4.7.1, a B_i é a i-ésima matriz 4×4 em um produtório de i matrizes 4×4 que, multiplicadas pelo vetor-coluna $(0, 0, 0, 1)$ dão origem a realização do i-ésimo átomo x_i da molécula estudada. Ou seja,

$$B_i \in M(4 \times 4) \mid B_1 B_2 \cdots B_i \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} x_{i1} \\ x_{i2} \\ x_{i3} \\ 1 \end{bmatrix} \ \forall\ 1 < i < N,$$

onde N é o número de átomos na molécula e o vetor-coluna $(x_{i1}, x_{i2}, x_{i3}, 1)$ é a realização do i-ésimo átomo da molécula, ou seja, tal vetor possui componentes que coincidem com as coordenadas do ponto onde está localizado tal átomo no $\mathbb{R}^3$.

Como dito anteriormente, o produto de matrizes é uma forma eficiente de concatenar funções e sabemos também que cada matriz B_i, necessária para encontrar a i-ésima realização na molécula, pode ser vista como representação de uma transformação linear, ou seja, cada B_i pode ser vista como a matriz que compõe todas as operações necessárias para passar da x_{i-1} até a x_i realização, logo

$$(B_1 B_2 \cdots B_{i-1}) B_i \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} x_{i1} \\ x_{i2} \\ x_{i3} \\ 1 \end{bmatrix},$$

onde $(B_1, B_2, \cdots, B_{i-1})$ é a matriz que representa todas as operações necessárias para chegar da x_1 até a x_{i-1}. Perceba que é como se caminhássemos pelas realizações das moléculas, onde sabemos exatamente para onde apontar (pelos ângulos θ e ω) e exatamente o quanto caminhar (a distância d) para ir de uma molécula a outra.

Logo, sabemos que a B_i será composta por movimentos de translações e rotações, nos bastando descobrir quais para entende-la. Nos será útil definir estes dois tipos de transformações lineares, como segue:

- **Translação:** Podemos resumir esta operação em um deslocamento.

Seja o vetor $v = (a) \in \mathbb{R}$. Perceba que há uma relação biunívoca entre o espaço vetorial $\mathbb{R}$ e o conjunto C de pontos na reta real, ou seja, $\forall\, v \in \mathbb{R}\ \exists!\, u \in C \mid v \equiv u$, logo, podemos movimentar o vetor v de forma a fazer o ponto por ele representado na reta também se movimentar.

Como faremos isso? É intuitivo imaginar que a solução seja uma função do tipo $f : \mathbb{R} \longrightarrow \mathbb{R} \mid f(x) = (x + 1)$, ou seja, uma função que leva um ponto x para uma unidade mais longe da origem. Infelizmente isso não se trata de uma movimentação, mas sim de um teleporte. Veja, o ponto mudou de lugar instantaneamente! Isso fere a linearidade:

$$f(x + a) = x + a + 1 \neq f(x) + f(a) = x + a + 2$$

Isso se deve pois quando algo é deslocado ele possui uma velocidade $\frac{\partial}{\partial t}$ associada, ou seja, precisamos deslocá-lo em relação a alguma dimensão específica do espaço vetorial (no caso t), logo, precisamos incrementar uma dimensão ao universo de nossa reta para encontrar a transformação linear que desloca um elemento v por ela.

Fazemos então

$$f : \mathbb{R}^2 \longrightarrow \mathbb{R} \mid f(x, t) = x + t,$$

ou seja, agora o ponto x ficará uma unidade mais distante da origem assim que a dimensão t também o ficar. Perceba que agora x movimenta-se a medida que t movimenta, sem saltos, logo, respeitando a linearidade

$$f((x, t) + (a, a)) = f(x + a, t + a) = x + t + a + a = f(x, t) + f(a, a).$$

Como toda transformação linear pode ser escrita como uma matriz, segue que se

$$f(x, t) = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} x \\ t \end{bmatrix} = \begin{bmatrix} x + t \end{bmatrix} = x + t$$

então $\begin{bmatrix} 1 & 1 \end{bmatrix}$ é a matriz que representa tal transformação, nossa translação.

- **Rotação:** Quando se quer rotacionar um objeto, sem alterar o formato dele, normalmente recorremos a matrizes de rotação.

  Tentemos definir um operador que rotacione um vetor em um ângulo θ em torno da origem [24], ou seja, precisamos de uma transformação linear $R : \mathbb{R}^2 \longrightarrow \mathbb{R}^2$ que, dado um vetor $v = (x, y) \in \mathbb{R}^2$, seja $Rv = (x', y')$. Como

  $$Rv = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x' \\ y' \end{bmatrix},$$

  então $x' = ax + by$ e $y' = cx + dy$.

  Perceba que nosso objetivo é encontrar a, b, c e d. Sabemos que se $B_{\mathbb{R}^2} = \{(1, 0), (0, 1)\}$ é a base canônica do $\mathbb{R}^2$, então $R(1, 0) = (a, c)$ e $R(0, 1) = (b, d)$. Por construção, se rotacionarmos os versores $(1, 0)$ e $(0, 1)$ em um ângulo θ, conforme Figura 29, seguindo as definições de senos e cossenos, obtemos que $R(1, 0) = (cos(\theta), sen(\theta))$ e $R(0, 1) = (-sen(\theta), cos(\theta))$.

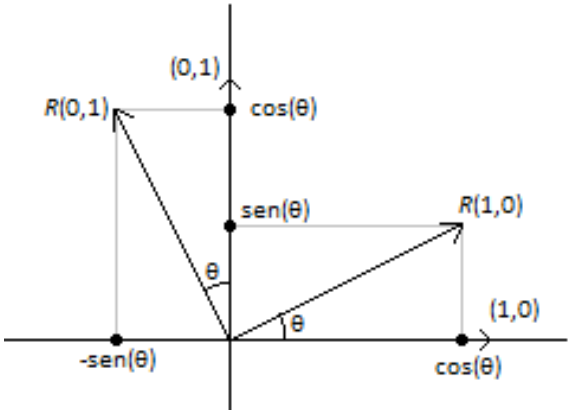

Figura 29: Transformação linear composta BA

Logo, $x' = xcos(\theta) - ysen(\theta)$ e $y' = xsen(\theta) + ycos(\theta)$. Nossa matriz de rotação em torno da origem fica:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} cos(\theta) & -sen(\theta) \\ sen(\theta) & cos(\theta) \end{bmatrix}.$$

Podemos enxergar esta rotação como uma projeção de uma transformação no espaço tridimensional $\mathbb{R}^3$, onde a rotação acontece ao redor do eixo z (que sai do papel), mantendo o eixo fixo. Para tentarmos ampliar esta matriz para a terceira dimensão, basta que nós entendamos que as componentes da transformação devem ser tais que não modifiquem componentes no eixo z, logo:

$$\begin{bmatrix} cos(\theta) & -sen(\theta) \\ sen(\theta) & cos(\theta) \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x' \\ y' \end{bmatrix} \longrightarrow \begin{bmatrix} cos(\theta) & -sen(\theta) & 0 \\ sen(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix},$$

onde $z' = z$.

A mesma lógica pode ser aplicada para deduzir as rotações em torno dos demais eixos, onde todas são enunciadas a seguir.

$$R_x(\theta) \cdot (x, y, z) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\theta) & -sen(\theta) \\ 0 & sen(\theta) & cos(\theta) \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix},$$

$$R_y(\theta) \cdot (x, y, z) = \begin{bmatrix} cos(\theta) & 0 & -sen(\theta) \\ 0 & 1 & 0 \\ sen(\theta) & 0 & cos(\theta) \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}.$$

$$R_z(\theta) \cdot (x, y, z) = \begin{bmatrix} cos(\theta) & -sen(\theta) & 0 \\ sen(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix},$$

Agora que possuímos tais definições, podemos voltar a analisar nossa B_i. Então, anunciando-a, sendo $i = 4, \ldots, N$:

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

Pode-se decompor esta matriz em produtos das matrizes de translação e rotações descritas, donde

$$B_i = R_x(w_{i-3,i}) \cdot R_x(\pi) \cdot R_z(\theta_{i-2,i}) \cdot R_y(\pi) \cdot T_x(d_{i-1,i}),$$

onde, perceba, as operações são compostas da direita para a esquerda, ou seja, primeiro há uma translação e depois um conjunto de quatro rotações. Dentre essas, duas são de um angulo fixo π. Isto se deve ao fato de que, como estamos caminhando de um átomo a outro, toda vez que chegamos em um novo átomo deve-se primeiro "olhar para trás" (para o átomo anterior) e só então definir aonde está a próxima direção à se percorrer, pois os ângulos assim são definidos (vide Figura 19).

Apêndice C

É comum dividirmos os aminoácidos proteicos em cinco classes, como segue.

Grupos R apolares, alifáticos

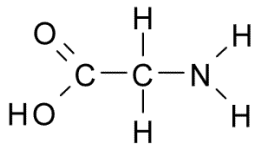

Figura 30: Glicina

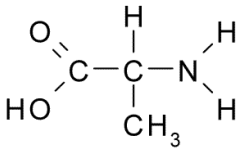

Figura 31: Alanina

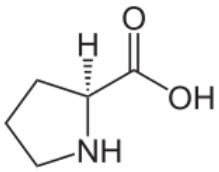

Figura 32: Prolina

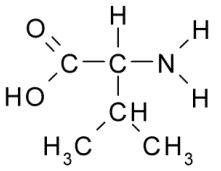

Figura 33: Valina

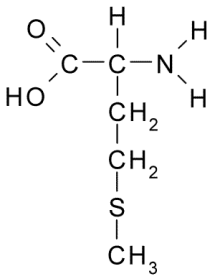

Figura 34: Metionina

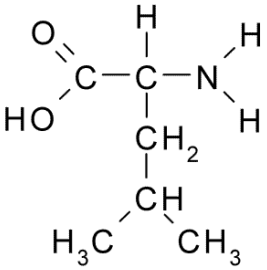

Figura 35: Leucina

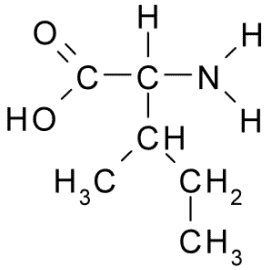

Figura 36: Isoleucina

Grupos R polares, não carregados

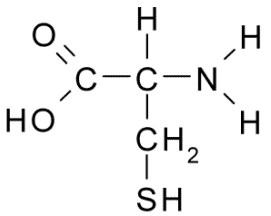

Figura 37: Cisteína

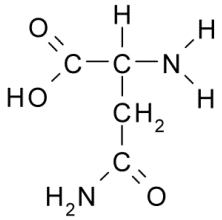

Figura 38: Asparagina

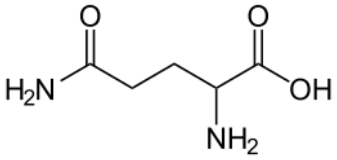

Figura 39: Glutamina

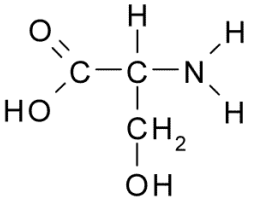

Figura 40: Serina

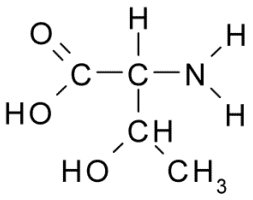

Figura 41: Treonina

Grupos R aromáticos

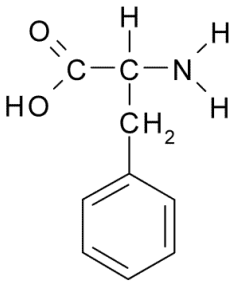

Figura 42: Fenilalanina

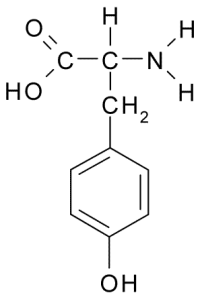

Figura 43: Tirosina

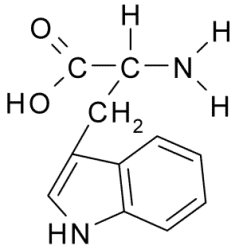

Figura 44: Triptofano

Grupos R carregados positivamente

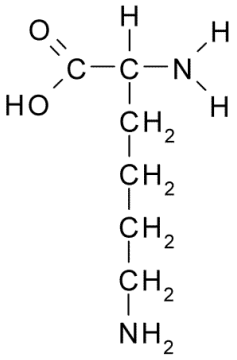

Figura 45: Lisina

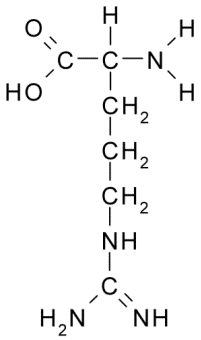

Figura 46: Arginina

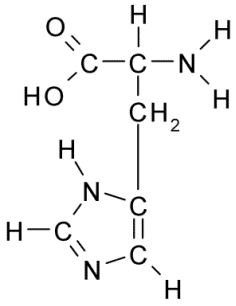

Figura 47: Histidina

Grupos R carregados negativamente

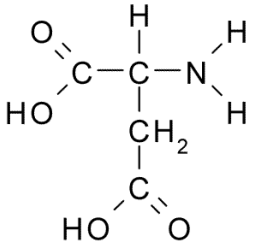

Figura 48: Aspartato

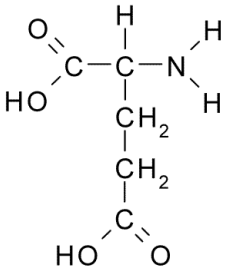

Figura 49: Glutamato