UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE CIÊNCIAS
DEPARTAMENTO DE ESTATÍSTICA E MATEMÁTICA APLICADA
PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM E MÉTODOS QUANTITATIVOS

FERNANDO DO CARMO BATISTA

UMA BUSCA PROBABILÍSTICA PARA O PROBLEMA DE GEOMETRIA DE DISTÂNCIAS MOLECULARES

FORTALEZA
2024

FERNANDO DO CARMO BATISTA

UMA BUSCA PROBABILÍSTICA PARA O PROBLEMA DE GEOMETRIA DE DISTÂNCIAS MOLECULARES

Dissertação apresentada ao Programa de Pós-Graduação em Modelagem e Métodos Quantitativos, do Centro de Ciências, da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Modelagem e Métodos Quantitativos. Área de Concentração: Modelagem e Métodos Quantitativos.

Orientador: Prof. Dr. Michael Ferreira de Souza.

FORTALEZA

2024

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

B336b Batista, Fernando do Carmo.
Uma busca probabilística para o Problema de Geometria de Distâncias Moleculares / Fernando do Carmo Batista. – 2024.
49 f. : il. color.

Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Ciências, Programa de Pós-Graduação em Modelagem e Métodos Quantitativos, Fortaleza, 2024.
Orientação: Prof. Dr. Michael Ferreira de Souza.

1. Proteínas – Estrutura. 2. Árvore binária. 3. Geometria de distâncias. I. Título.

CDD 510

FERNANDO DO CARMO BATISTA

UMA BUSCA PROBABILÍSTICA PARA O PROBLEMA DE GEOMETRIA DE DISTÂNCIAS MOLECULARES

Dissertação apresentada ao Programa de Pós-Graduação em Modelagem e Métodos Quantitativos, do Centro de Ciências, da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Modelagem e Métodos Quantitativos. Área de Concentração: Modelagem e Métodos Quantitativos.

Aprovada em: 27/02/2024.

BANCA EXAMINADORA

Prof. Dr. Michael Ferreira de Souza (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Albert Einstein Fernandes Muritiba
Universidade Federal do Ceará (UFC)

Prof. Dr. Carlile Campos Lavor
Universidade Estadual de Campinas (UNICAMP)

À minha esposa e filhos (Eva, Benjamim e Ana Esther), meus maiores tesouros.

AGRADECIMENTOS

Agradeço a Deus pela vida e por ter me concedido a extraordinária oportunidade de cursar e concluir este mestrado. Sua orientação e graça foram fundamentais em cada passo desta jornada, guiando-me nos momentos desafiadores e iluminando meu caminho.

Expresso minha profunda gratidão ao meu orientador, Michael Souza, pela paciência infinita, pela empatia ao reconhecer minhas limitações e por estar sempre acessível. Suas inúmeras tentativas de me direcionar para um tema compatível com meu grau de maturidade não passam despercebidas, e reconheço que fui desafiado, mas também inspirado por seu comprometimento. Mesmo sem saber, por diversas vezes considerei desistir, temendo estar lhe causando demasiado trabalho, mas sua orientação constante e apoio inabalável foram cruciais para minha perseverança.

Quero expressar minha sincera gratidão ao professor Carlile Lavor, que colaborou indicando o tema de pesquisa e depositou confiança em meu projeto de dissertação. Seu apoio foi um incentivo valioso que impulsionou meu trabalho.

Um agradecimento especial ao professor Guilherme Barreto por também ter confiado em meu projeto de dissertação. Sua confiança motivou-me a superar desafios e aprimorar meu trabalho de maneiras que eu jamais imaginaria.

Aos professores do Programa de Pós-Graduação, agradeço sinceramente pela dedicação nas aulas e pela constante preocupação para com nós, alunos. Suas contribuições foram fundamentais para o meu crescimento acadêmico e profissional.

À minha esposa Clemilda, meu alicerce e fonte de apoio incondicional, agradeço do fundo do coração pelo companheirismo e por ter me incentivado a não desistir. Sua presença foi crucial para superar os desafios deste percurso.

Agradeço aos meus pais, por tudo que me tornei, e em especial ao meu pai, que mesmo não estando mais fisicamente presente, sempre me apoiou em todas as minhas escolhas. Sua influência continua a guiar-me, e seu legado é uma fonte constante de inspiração.

A todos que, de alguma forma, contribuíram para minha jornada acadêmica, expresso minha mais profunda gratidão. Este trabalho não teria sido possível sem o apoio inestimável de cada um de vocês.

RESUMO

A predição de estruturas tridimensionais de proteínas tem sido uma área de intensa pesquisa, abordada por diversas disciplinas representadas pela bioinformática. O presente trabalho propõe um método inovador de predição de estruturas de proteínas baseado em árvore binária, denominado Pesquisa Baseada em Frequência (FBS). Realizamos um comparativo estatístico da eficiência deste método em relação ao método Pesquisa em Profundidade (DFS), utilizando estruturas de proteínas obtidas por Ressonância Magnética Nuclear (NMR) disponíveis no Banco de Dados de Proteínas (PDB). O objetivo principal é avaliar a eficiência desses métodos em subsequências de átomos do backbone proteico de tamanhos 5, 10, 15, 20 e 25, enquanto investigamos se a natureza exibe preferências geométricas ao enovelar-se. Os resultados computacionais indicam que o método FBS supera o método DFS em pelo menos 70% das subsequências de átomos analisadas e sugerem a existência de preferências geométricas nas proteínas, conforme evidenciado pela amostra selecionada.

Palavras-chave: árvore binária; estrutura de proteína; geometria de distância.

ABSTRACT

The prediction of three-dimensional protein structures has been an area of intense research, addressed by several disciplines represented by bioinformatics. The present work proposes an innovative method for predicting protein structures based on a binary tree, called Frequency-Based Search (FBS). We carried out a statistical comparison of the efficiency of this method in relation to the Depth Search (DFS) method, using protein structures obtained by Nuclear Magnetic Resonance (NMR) available in the Protein Data Bank (PDB). The main objective is to evaluate the efficiency of these methods on subsequences of protein backbone atoms of sizes 5, 10, 15, 20 and 25, while investigating whether nature exhibits geometric preferences when folding. The computational results indicate that the FBS method outperforms the DFS method in at least 70% of the atom subsequences analyzed and suggest the existence of geometric preferences in proteins, as evidenced by the selected sample.

Keywords: binary tree; protein structure; distance geometry.

SUMÁRIO

1 INTRODUÇÃO

O Problema de Enovelamento Proteico (PFP, do inglês *Protein Folding Problem*), é um desafio crucial na biologia molecular e bioinformática, envolvendo o intricado processo pelo qual uma cadeia polipeptídica se organiza tridimensionalmente em uma estrutura funcional e estável, conhecida como sua conformação nativa. Compreender esse processo é essencial para desvendar os mistérios da vida e desenvolver tratamentos eficazes para uma variedade de doenças, muitas das quais estão associadas a disfunções proteicas (Lesk, 2008).

As origens do Problema de Enovelamento Proteico estão ligadas aos resultados dos trabalhos de Max Perutz e John Kendrew agraciados com o Nobel de Química de 1962 pela determinação de estruturas proteicas globulares (Anfinsen, 1973; Dill; MacCallum, 2012; Nobel-Prize.org, 1962). Esse problema foi colocado pela primeira vez há pouco mais de meio século e refere-se a três grandes questões: (i) *o código físico*: quais são as propriedades físico-químicas pelas quais uma sequência de aminoácidos dita a estrutura tridimensional de uma proteína? (ii) *o mecanismo de enovelamento*: uma cadeia polipeptídica possui inúmeras possibilidades de conformação, então como as proteínas podem se enovelar tão rápido? (iii) *o código computacional*: podemos conceber um algoritmo computacional para prever estruturas proteicas a partir de suas sequências de aminoácidos?

A complexidade computacional do PFP é evidente diante do surgimento acelerado de novas proteínas, muitas das quais permanecem sem estruturas conhecidas por longos períodos após sua descoberta (Dill; MacCallum, 2012). Algumas das principais dificuldades na predição dessas estruturas são atribuídas às limitações experimentais (VVÜTHRICH, 1989), a complexidade das interações moleculares e a diversidade de conformações que uma proteína pode adotar (Lesk, 2008).

Esse desafio de determinar a estrutura de proteínas por meio de algoritmos computacionais pode ser abordado como uma das aplicações do Problema de Geometria de Distâncias (DGP, do inglês *Distance Geometry Problem*), que envolve a determinação de conformações proteicas a partir de distâncias interatômicas (Lavor *et al.*, 2017). A resolução eficiente desse problema é essencial para a reconstrução precisa das estruturas moleculares e contribui significativamente para a compreensão das funções biológicas das proteínas.

Neste cenário, diversos algoritmos, como DGSOL (Moré; Wu, 1999), *Branch and Prune* (BP) (Liberti *et al.*, 2008a), *Geometric Build-Up* (Dong; Wu, 2003) e Suavização Hiperbólica (Souza *et al.*, 2011), foram desenvolvidos para abordar um subproblema do DGP, o

Problema Geométrico de Distâncias Moleculares (MDGP, do inglês *Molecular Distance Geometry Problem*), oferecendo estratégias distintas para superar desafios computacionais e limitações experimentais.

Conhecer as estruturas das proteínas não apenas aprimora nossa compreensão fundamental da biologia molecular, mas também desempenha um papel direto na descoberta de fármacos (Dill; MacCallum, 2012). A compreensão das conformações tridimensionais das proteínas é crucial para identificar sítios ativos, entender interações com ligantes e projetar moléculas terapêuticas eficazes. Assim, avanços na resolução do Problema de Enovelamento Proteico e do Problema de Geometria de Distâncias têm implicações significativas na pesquisa de novos medicamentos e na busca por terapias mais precisas e eficientes.

Nessa perspectiva de desenvolver algoritmos mais eficientes, fazemos um comparativo entre o algoritmo de Pesquisa em Profundidade (DFS, do inglês *Depth-First Search*) (Cormen *et al.*, 2022) e o algoritmo, desenvolvido neste trabalho, de Pesquisa Baseada em Frequência (FBS, do inglês *Frequency-Based Search*).

Baseado no algoritmo DFS, o BP avança sistematicamente através da árvore de busca associada ao DDGP, podando o espaço de solução satisfazendo as restrições de distância dadas como entrada para o problema (Lavor *et al.*, 2012). O DFS é um algoritmo com baixo consumo de memória, mas não incorpora informações bioquímicas sobre proteínas. Neste trabalho, pela primeira vez, utilizamos dados do Banco de Dados de Proteínas (PDB, do inglês *Protein Data Bank*) (Berman *et al.*, 2000) para propor uma estratégia de busca alternativa ao DFS, o FBS.

No Capítulo 2, abordamos o Problema de Geometria de Distâncias, detalhando suas origens e relevância para a determinação de estruturas moleculares. Discutimos também as metodologias e desafios associados à discretização do espaço de solução do DGP e DDGP, bem como as aplicações práticas. No Capítulo 3, mergulhamos no universo das proteínas, analisando sua composição química, níveis estruturais e a importância dos bancos de dados proteicos para a pesquisa científica. A Metodologia é explorada no Capítulo 4, onde comparamos as abordagens tradicionais de pesquisa em árvore, como o algoritmo DFS, com a nossa proposta de algoritmo FBS, informado por dados do PDB. No Capítulo 5, apresentamos os resultados obtidos, demonstrando a eficácia de nossa abordagem. Por fim, no Capítulo 6, refletimos sobre as conclusões e possibilidades futuras de pesquisa, ampliando o horizonte de aplicações práticas e teóricas.

2 GEOMETRIA DE DISTÂNCIAS

Inicialmente, gostaríamos de destacar que este capítulo teve como fonte predominante a obra "Um Convite à Geometria de Distâncias" dos pesquisadores Carlile Lavor (UNICAMP) e Leo Liberty (École Polytechnique), publicada pela Sociedade Brasileira de Matemática Aplicada e Computacional (SBMAC) em 2014.

Geometria de Distâncias (DG - do inglês *Distance Geometry*) é uma área de estudo consolidada da Matemática Aplicada que faz uso, em sua essência, da Matemática e da Computação. A DG investiga as relações existentes entre três situações (Lavor; Liberti, 2014):

- Distâncias entre objetos relacionados a um determinado problema,
- Distâncias entre pontos (representando tais objetos) em um dado espaço geométrico,
- Localização desses pontos, possivelmente em um espaço geométrico distinto.

Atualmente, o problema fundamental da DG é determinar um conjunto de pontos em um dado espaço geométrico, cujas distâncias entre alguns deles são conhecidas. A primeira menção explícita a esse problema e com essas restrições foi feita por Yemini, em 1978 (Yemini,).

Considera-se que a DG surgiu em 1928, quando Menger (Menger, 1928) caracterizou vários conceitos geométricos usando a ideia de distância. Entretanto, apenas com os resultados de Blumenthal (Blumenthal, 1970) em 1953, o tema se tornou uma nova área do conhecimento chamada de Geometria de Distâncias.

Naquela época, a principal questão da DG era encontrar condições necessárias e suficientes para decidir se uma dada matriz é uma matriz de distâncias. Mais especificamente, a questão era decidir se, dada uma matriz simétrica, existe um número inteiro positivo K e um conjunto de pontos em $\mathbb{R}^K$, onde as distâncias euclidianas entre esses pontos são iguais às entradas dessa matriz. Note que, nesse caso, todas as distâncias são consideradas conhecidas.

Em 1988, o lançamento do livro "*Distance Geometry and Molecular Conformation*" de Crippen e Havel (Crippen *et al.*, 1988), considerados os pioneiros na aplicação da DG ao cálculo de estrutura de proteínas, marcou consolidação dessa aplicação como tema de estudo.

Em 2013, publicado pela Springer, aparece o livro "*Distance Geometry: Theory, Methods, and Applications*" (Mucherino *et al.*, 2012), o primeiro integralmente dedicado à DG reunindo diferentes aplicações e pesquisadores da área. Em junho do mesmo ano, na cidade de Manaus (Brasil) foi realizado o primeiro *workshop* internacional dedicado ao tema: *Workshop on Distance Geometry and Applications*. A participação de universidades, como Princeton

University (EUA), o apoio de sociedades acadêmicas, como a *The International Federation of Operational Research Societies* e de agências de fomento à pesquisa brasileira, como o Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), marcaram a grandiosidade e importância desse evento.

A grande variedade de aplicações da DG como, por exemplo, em astronomia, bioquímica, estatística, nanotecnologia, robótica e telecomunicações, transcende ao mero desenvolvimento de teorias matemáticas. Dentre essas áreas, destacamos a bioquímica por ser a motivação desse trabalho determinar a estrutura tridimensional das proteínas a partir das distâncias entre os átomos que as compõem.

2.1 Problema de geometria de distâncias

A estrutura matemática mais utilizada para a representação do Problema de Geometria de Distâncias (DGP - do inglês *Distance Geometry Problem*) e de suas aplicações é o grafo. A depender da aplicação, os vértices podem representar estrelas, átomos, entre outros e as distâncias conhecidas representam as arestas entre os vértices correspondentes. Quando falamos de distâncias conhecidas, estamos nos referindo àquelas previamente dadas na formulação de uma instância do problema.

Definição 2.1.1 (Problema de Geometria de Distâncias (DGP)) *Dado um inteiro $K > 0$ e um grafo simples não orientado $G = (V, E)$, cujas arestas são ponderadas pela função não negativa $d : E \to \mathbb{R}^*_+$, determinar, caso exista, uma função (imersão) $x : V \to \mathbb{R}^K$ tal que*

$$||x(u) - x(v)||_2 = d(u,v), \quad \forall \{u,v\} \in E. \tag{2.1}$$

O DGP possui ligação com a aplicação da Ressonância Magnética Nuclear (NMR - do inglês *Nuclear Magnetic Resonance*) na determinação de estruturas proteicas, pois em vez das posições dos átomos, a NMR fornece informações (limites) para as distâncias entre determinados pares de átomos. Por exemplo, a NMR é capaz de estimar a distância entres pares de átomos de hidrogênio que estejam a menos de 5Å (1Å equivale 10^{-10} m) um do outro. Se considerarmos $V = \{v_1, \ldots, v_n\}$ como sendo o conjunto de átomos de uma proteína e $E \subset V \times V$ como sendo o conjunto de pares $\{v_i, v_j\}$ para os quais conhecemos a distância d_{ij}, então o objetivo do DGP é determinar as coordenadas em $\mathbb{R}^3$ de cada um dos átomos de modo a satisfazer as distâncias conhecidas.

Na prática, os experimentos de NMR fornecem apenas o subconjunto de distâncias entre átomos que estão espacialmente próximos e a precisão dos dados é limitada. Assim, em um cenário real, o conjunto E é esparso e, em vez das igualdades da Eq.2.1, temos desigualdades da forma $\underline{d}(u,v) \leq ||x(u) - x(v)||_2 \leq \bar{d}(u,v), \quad \forall \{u,v\} \in E$. Esta forma do DGP parece ser mais fácil de resolver, uma vez que as restrições são relaxadas. Porém, na prática, os limites inferior e superior estão próximos e, portanto, o problema ainda é difícil de resolver. De fato, quando os limites superior e inferior estiverem próximos, o DGP com restrições relaxadas pertence à classe NP-difícil (Moré; Wu, 1997).

2.1.1 Sobre o número de soluções

É fácil ver que, se tivermos uma solução do DGP, então poderemos obter infinitas outras soluções por meio de rotações e translações. Utilizando a Álgebra Geométrica, podemos inclusive provar que o conjunto solução do DGP será vazio, finito ou infinito não enumerável mesmo desconsiderando as soluções obtidas via isometrias (Lavor; Liberti, 2014). Em particular para o Problema Discretizável de Geometria de Distância (DDGP - do inglés *Discretizable Distance Geometry Problem*) é possível demonstrar que o número de soluções será par e, mais ainda, quase sempre uma potência de dois (Liberti *et al.*, 2014b).

2.2 Métodos de resolução do DDGP

Há diversos métodos que podem ser utilizados para a resolução do DDGD, dentre eles os algoritmos DGSOL, Suavização Hiperbólica e *Branch-and-Prune* (BP). Iremos descrever superficialmente os dois primeiros algoritmos citados anteriormente e descrever o último com um pouco mais de detalhes.

2.3 Métodos de suavização e penalização

Uma alternativa para a resolução do DDGP é reformulá-lo como um problema de otimização irrestrita onde as restrições são penalizadas. Uma destas alternativas é penalizar o resíduo de cada distância, ou seja, a diferença entre a distância conhecida e a distância calculada. A função objetivo é a soma dos quadrados dos resíduos. A função objetivo é então minimizada por meio de métodos de otimização não-linear.

$$\min_x f(x) = \sum_{\{u,v\}(\in E} (||x(u) - x(v)||_2 - d(u,v))^2, \tag{2.2}$$

onde $x : V \to \mathbf{R}^k$.

Esse modelo para o DDGP têm dois desafios que atrapalham o uso dos métodos de otimização não-linear: a falta de diferenciabilidade e o excesso de minimizadores locais que não são globais. O primeiro desafio vem da norma Euclidiana na função objetivo, e o segundo vem da combinação do problema, que faz o número de minimizadores crescer junto com as quantidades de vértices e arestas (Lavor; Liberti, 2014).

2.3.1 Algoritmo DGSOL

Para tratar destes desafios, no algoritmo DGSOL (Moré; Wu, 1997), a função original f é suavizada pela transformada gaussiana. A transformada gaussiana é um tipo de transformada de suavização que usa um parâmetro λ para ajustar o nível de suavização. Quando $\lambda = 0$, a função original é recuperada, e quando λ cresce, a função fica mais suave. A transformada gaussiana $\langle f \rangle_\lambda$ de uma função $f : \mathbb{R}^n \to \mathbb{R}$ é definida como

$$\langle f \rangle_\lambda (x) = \frac{1}{\pi^{n/2}\lambda^n} \int_{\mathbb{R}^n} f(y) e^{-\frac{||y-x||^2}{\lambda^2}} dy,$$

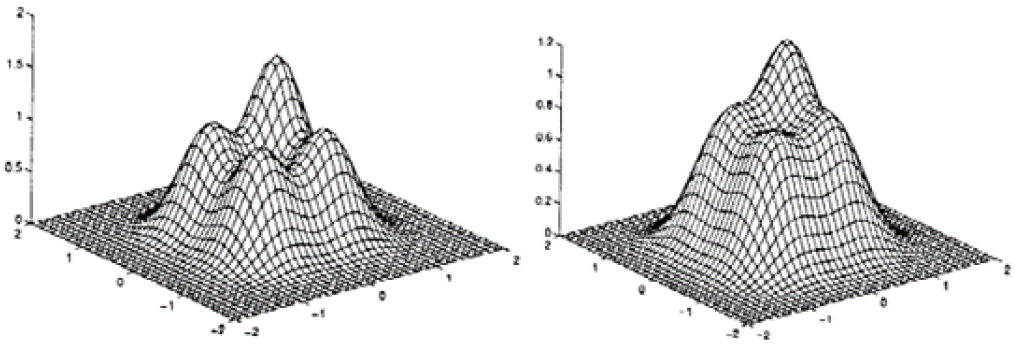
O valor $\langle f \rangle_\lambda (x)$ é um cálculo de f em um entorno de x, com o tamanho proporcional desse entorno determinado pelo parâmetro λ. O entorno fica menor quando λ fica menor, de forma que quando $\lambda = 0$, o entorno é o ponto x. A transformada gaussiana $\langle f \rangle_\lambda$ também pode ser entendida como a convolução de f com a função densidade gaussiana.

A função suavizada elimina minimizadores pequenos e estreitos e preserva a forma geral da função (Moré; Wu, 1999). Isso faz com que o algoritmo de otimização não se distraia com minimizadores locais pouco relevantes e foque em áreas com valores médios da função onde há mais chances de encontrar um minimizador global.

A transformada gaussiana é um operador linear que mantém a ordem e que diminui os componentes de alta frequência de f. Além disso, a transformada gaussiana troca de lugar com a diferenciação, de modo que a transformada gaussiana do gradiente (Hessiana) é o gradiente (Hessiana) da transformada gaussiana. Essas características da transformada gaussiana normalmente não são encontradas em outras técnicas de suavização.

Mostramos o processo de transformação na Figura 1 com uma função que é a soma de quatro Gaussianas. A função original ($\lambda = 0$) está do lado esquerdo enquanto $\lambda = 0,3$ está do lado direito. Veja que a função original tem quatro maximizadores, mas dois desses maximizadores sumiram em $\lambda = 0,3$, e outro minimizador provavelmente sumirá se λ for aumentado mais ainda. A Figura 1 mostra que a função original é aos poucos transformada em uma função mais suave com menos maximizadores locais e que a suavização cresce conforme λ cresce.

Figura 1 – Gráfico mostrando a transformada Gaussiana de uma função



Fonte: Moré; Wu (1999, p. 222).

Legenda: Na esquerda a função original ($\lambda = 0$) e na direita a função suavizada ($\lambda = 0.3$)

2.3.2 Algoritmo suavização hiperbólica

Assim como o método DGSOL, a técnica da suavização hiperbólica busca tanto adicionar a diferenciabilidade nos modelos feitos via programação matemática, quanto diminuir o número de minimizadores. A diferença é a simplicidade da suavização hiperbólica, que é uma técnica mais recente e que não requer a transformada gaussiana.

Na Suavização Hiperbólica, a diferenciabilidade será alcançada pela troca da norma Euclidiana $||\cdot||$ pela função a seguir:

$$\theta_t(x_i) = \sqrt{t^2 + < x_i, x_i >}, \tag{2.3}$$

onde $< \cdot >$ representa o produto interno e $t > 0$.

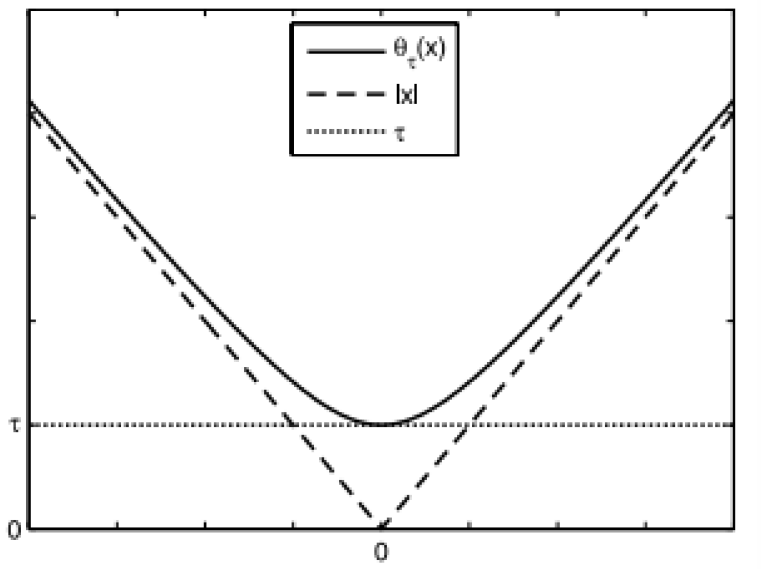
A função $\theta_t(x_i)$ possui as seguintes propriedades (Souza *et al.*, 2011):

1. $\lim_{t \to 0} \theta_t(x_i) = ||x_i||$, ou seja, a função é uma boa aproximação para a norma Euclidiana;
2. θ_t é infinitamente diferenciável;

3. $\theta_t(x_i) > ||x_i||, \forall x_i$;
4. $t_1 > t_2 \rightarrow \theta_{t_1}(x_i) > \theta_{t_2}(x_i), \forall x_i \in \mathbf{R}^n$.

O parâmetro t é o responsável por aproximar a função θ_t da norma Euclidiana, conforme figura 2

Figura 2 – Gráfico da suavização hiperbólica θ_t é uma hipérbole equilateral



Fonte: Souza *et al* (2011, p. 462).

Aplicando-a na função objetivo 2.2, temos

$$f(x) = \sum(\theta_t(x_u - x_v) - d(x_u, x_v))^2, \forall u, v \in E \tag{2.4}$$

Essa nova função objetivo é da classe C^∞, o que permite o uso dos métodos tradicionais, mas ela é diferente da função objetivo do problema original (2.2). Para conseguir a solução de 2.2 é sugerida então a solução de uma série de problemas parametrizados (Souza *et al.*, 2011), onde a série das soluções obtidas vai se aproximar, no final, da solução do problema original (Souza, 2010. 86 p.).

2.3.3 Algoritmo Branch and Prune (BP)

Esse algoritmo baseia-se na ordem que pode ser estabelecida para os vértices (v_1, v_2, ..., v_n) e na utilização das distâncias obtidas pelos experimentos de Ressonância Magnética Nuclear (NMR). Inicialmente, fixamos as posições dos três primeiros vértices:

$v_1 = (0,0,0)$,

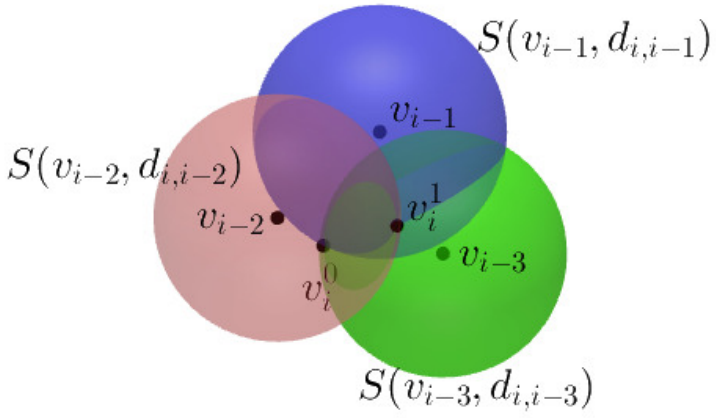
$v_2 = (d_{12}, 0, 0)$ e

$v_3 = (\frac{d_{13}^2 + d_{12}^2 - d_{23}^2}{2d_{12}}, \frac{\sqrt{-{d_{23}}^4 + (2{d_{13}}^2 + 2{d_{12}}^2){d_{23}}^2 - {d_{13}}^4 + 2{d_{12}}^2{d_{13}}^2 - {d_{12}}^4}}{2d_{12}}, 0)$.

As coordenadas de v_3 são obtidas da seguinte forma: tomamos as esferas $S(v_1, d_{13})$ e $S(v_2, d_{23})$; a interseção das duas é uma circunferência C paralela ao plano yz e centro pertencente ao eixo x; consideramos x_3 como o ponto de coordenadas não negativas pertencente a interseção do plano xy e a circunferência C. Como v_3 está no plano xy, sua coordenada z é igual a zero.

O restante dos vértices v_i, com $i = 4, 5, ..., n$, são obtidos a partir dos três vértices imediatamente anteriores v_{i-3}, v_{i-2} e v_{i-1} através da interseção das três esferas $S(v_{i-3}, d_{i,i-3})$, $S(v_{i-2}, d_{i,i-2})$ e $S(v_{i-1}, d_{i,i-1})$, onde $S(u, d)$ representa a esfera de centro u e raio d (Figura 3). Pelas condições do problema, essa interseção são dois pontos, que são as duas possíveis posições para v_i. Essa ideia induz uma estrutura de árvore binária com 2 possibilidades para v_4, 4 possibilidades para v_5, 8 possibilidades para v_6, ..., 2^{n-3} possibilidades para v_{n-3}. (Figura 4)

Figura 3 – Desenho indicando que a interseção de três esferas são exatamente dois pontos, v_i^0 e v_i^1

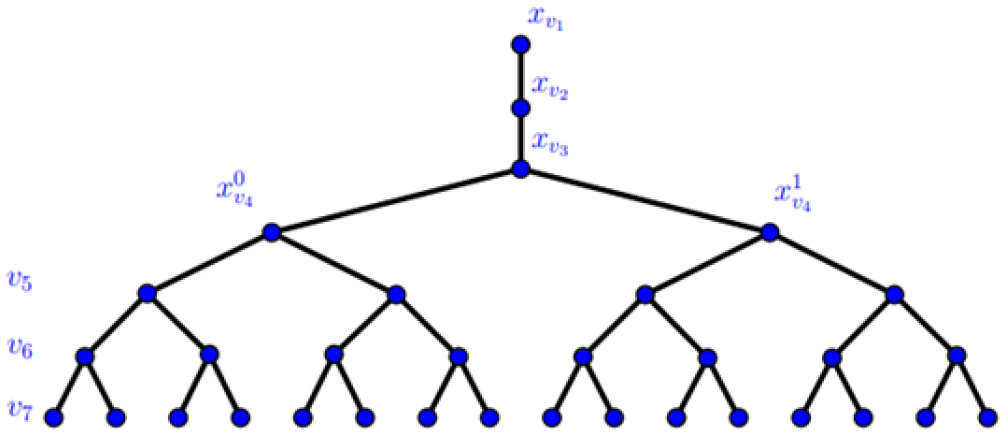


Fonte: elaborada pelo autor.

Como ponto de partida decidimos fazer uma "busca pela esquerda", ou seja, quando não temos arestas adicionais, optamos por escolher a solução v_i^0 do sistema quadrático associado. Algumas vezes necessitamos fazer um retorno na árvore. Esse retorno deve ser feito com cuidado para não nos perdemos.

Essencialmente, o procedimento de resolução do DDGP pode ser definido como uma sequência de sistemas quadráticos e testes de viabilidade (quando existem arestas adicionais). Com esses dois subproblemas em mente, podemos dividir as arestas E do grafo do DDGP em

Figura 4 – Imagem de uma árvore binária



Fonte: Lavor; Liberti (2014, p. 28).

dois conjuntos disjuntos:

$$E = E_d \cup E_p$$

onde

$$E_d = \{\{a_4, v_4\}, \{b_4, v_4\}, \{c_4, v_4\}, ..., \{a_n, v_n\}, \{b_n, v_n\}, \{c_n, v_n\}\}$$

é o conjunto das arestas de discretização e

$$E_p = E - E_d$$

é o conjunto das arestas de poda.

As arestas de discretização "moldam" o espaço de busca como uma árvore binária e as arestas de poda "indicam" o caminho que deve ser percorrido, ao reduzir o espaço de busca.

Para encontrar uma solução, devemos tentar descer pela árvore, da raiz até o último nível, passando por todos os testes de viabilidade. Chegando no último nível, a solução é dada pelo caminho percorrido entre o primeiro e o último nível da árvore, considerando apenas os nós viáveis.

Podemos descer pela árvore, sem nenhum retorno, em dois casos extremos:

- $E_p = \emptyset$, pois todos os vértices da árvore binária serão viáveis.
- $\forall v_i, i = 5, ..., n$, existem pelo menos 4 vértices que geram pontos (não coplanares) anteriores a v_i, a_i, b_i, c_i, d_i, com $\{\{a_i, v_i\}, \{b_i, v_i\}, \{c_i, v_i\}, \{d_i, v_i\}\} \subset E$, pois existirá exatamente um único nó viável em cada nível.

3 PROTEÍNAS

Este capítulo foi largamente inspirado no livro "Introdução à Bioinformática"de autoria do professor Arthur M. Lesk (Lesk, 2008), da *Pennsylvania State University*, EUA, uma tradução em português publicada pela editora Artmed em 2008, e no livro "Bioinformática: da biologia à flexibilidade molecular"de autoria do professor Hugo Verli (Verli, 2014), da Universidade Federal do Rio Grande do Sul, Brasil.

Um dos campos do conhecimento que auxilia a desvendar os “mistérios” das proteínas é a bioinformática. Esta área pode ser compreendida como uma ciência interdisciplinar que combina técnicas de computação e teoria da informação aplicadas à biologia para construir, utilizar e acessar dados biológicos. Dentro desses dados biológicos, estão inclusos aqueles relacionados às proteínas. A bioinformática é, portanto, uma área interdisciplinar, englobando biologia, ciência da computação, química, física, bioquímica, engenharia da informação, matemática e estatística (Notari *et al.*, 2020).

O momento chave para a bioinformática foi no início da década de 1950, quando a revista *Nature* publicou o trabalho clássico sobre a estrutura em hélice da molécula de DNA por James Watson e Francis Crick (Watson; Crick, 1953).

Outras contribuições importantes foram apresentadas nos trabalhos de Linus Pauling e Robert Corey (Cao; Mezzenga, 2019), no início da década de 1950, e de Gopalasamudram N. Ramachandran (Ramachandran, 2013), no início da década de 1960, que estabeleceram as bases para a compreensão da estrutura tridimensional de proteínas.

Em 1966, Cyrus Levinthal (Levinthal, 1966) publicou na revista *Scientific American* o trabalho desenvolvido no *Massachusetts Institute of Technology* por John Ward e Robert Stotz (Stotz; Ward, 1965). Esse trabalho marcou o início da utilização de programas de computadores para visualização de estruturas tridimensionais de moléculas.

Ainda na década de 1960, ocorreu o primeiro esforço de sistematização do conhecimento sobre a estrutura tridimensional das proteínas, os efetores da informação genética. Em 1965, foi publicado o “*Atlas of Protein Sequence and Structure*”, organizado por diversos autores, com destaque para Margaret Dayhoff (Chang *et al.*, 1965).

Os primeiros estudos sobre a dinâmica e o enovelamento de proteínas por meio de simulações de dinâmica molecular foram conduzidos por Michael Levitt e Arieh Warshel (Levinthal, 1966) nos anos de 1970, resultando no Prêmio Nobel de Química em 2013 (Thiel; Hummer, 2013).

3.1 Composição química das proteínas

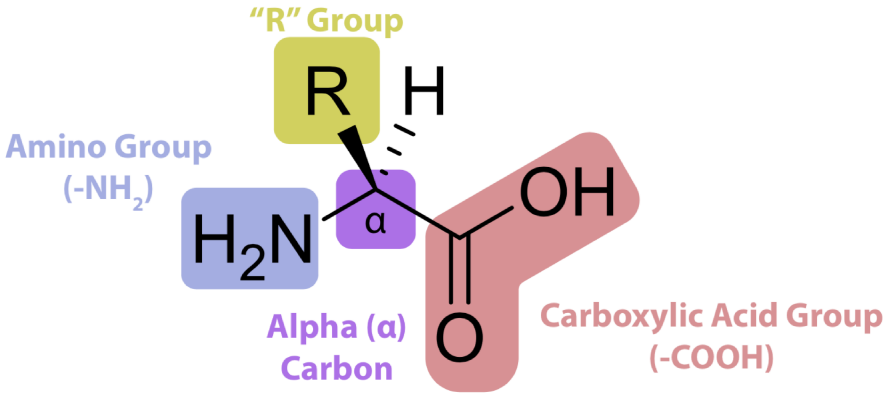
As proteínas constituem as macromoléculas mais abundantes, e cada célula de um organismo pode conter milhares delas, cada uma desempenhando uma função única. A função de uma proteína é definida pela disposição dos átomos presentes na sequência de aminoácidos em sua estrutura tridimensional (Wolynes, 2015).

Em muitos casos, apenas uma pequena porção da estrutura, conhecida como **sítio ativo**, funciona de maneira precisa, enquanto o restante da estrutura existe principalmente para criar e manter as relações espaciais entre os resíduos do sítio ativo. As proteínas evoluem por meio de alterações estruturais originadas por mutações nas sequências de aminoácidos e rearranjos gênicos, que integram diferentes combinações de subunidades estruturais.

Bioquimicamente, as proteínas desempenham uma variedade de papéis nos processos vitais. Existem proteínas estruturais, como as proteínas capsídeo viral, a camada mais externa e áspera da pele humana e de outros animais, e as proteínas do citoesqueleto. Há também proteínas que catalisam reações químicas, como as enzimas, além de proteínas de transporte e armazenamento, como a hemoglobina e a ferritina. Proteínas reguladoras, incluindo hormônios e proteínas sinalizadoras ou receptoras de sinais, também desempenham papéis essenciais, assim como as proteínas que controlam a transcrição gênica e as envolvidas em reconhecimento, como moléculas de adesão celular, anticorpos e outras do sistema imune.

Os monômeros de uma proteína, os aminoácidos (Figura 5), diferem significativamente dos monômeros do DNA e RNA. Enquanto DNA e RNA possuem apenas 4 tipos de nucleotídeos, existem 20 tipos de aminoácidos (Figura 1). Cada aminoácido possui a mesma estrutura básica, à qual está ligado, de forma padronizada, um grupo lateral que confere a cada aminoácido uma característica química única. As moléculas de proteína são polipeptídeos, formados pela ligação dos aminoácidos em uma sequência específica. Essas ligações ocorrem quando o grupo carboxila (COOH) de um aminoácido reage com o grupo amina (NH2) de outro aminoácido, liberando uma molécula de água no processo, em uma reação conhecida como condensação. Ao longo de bilhões de anos de evolução, essa sequência específica foi selecionada para conferir à proteína uma função útil. Assim, ao se dobrar em uma forma tridimensional precisa, com sítios reativos em sua superfície, esses polímeros de aminoácidos podem se ligar com alta especificidade a outras moléculas, atuando como enzimas que catalisam reações cruciais nas células (Alberts *et al.*, 2017).

Figura 5 – Desenho da estrutura química de um aminoácido



Fonte: Cabeen *et al* (2020, p. 16).

Tabela 1 – Representações dos aminoácidos

Aminoácido	**Representação de 3 Letras**	**Representação de 1 Letra**
Alanina	Ala	A
Cisteína	Cys	C
Ác. aspártico	Asp	D
Ác. glutâmico	Glu	E
Fenilalanina	Phe	F
Glicina	Gly	G
Histidina	His	H
Isoleucina	Ile	I
Leucina	Leu	L
Lisina	Lys	K
Metionina	Met	M
Asparagina	Asn	N
Prolina	Pro	P
Glutamina	Gln	Q
Arginina	Arg	R
Serina	Ser	S
Treonina	Thr	T
Valina	Val	V
Triptofano	Trp	W
Tirosina	Tyr	Y

Fonte: Lesk (2008, p. 27)

3.2 Níveis de estruturas das proteínas

As proteínas de modo geral possuem quatro níveis de estrutura, são eles:

– **Estrutura primária**

O primeiro nível de complexidade, a estrutura primária, é um padrão de letras (ou grupos pequenos de letras) que mostra a composição do biopolímero. Essa sequência de letras é uma informação de tipo unidimensional (1D), em que a única dimensão mostrada é a ordem de ocorrência dos monômeros. No caso das proteínas em uma sequência de aminoácidos, Figura 6.

Mesmo sendo menos complexa, a estrutura primária nos dá muitas informações sobre o

Figura 6 – Exemplo de sequência de aminoácidos de uma proteína, no qual cada letra representa um aminoácido.

Fonte: Verli (2014, p.22).

formato natural da biomolécula e, por isso, sobre suas funções. Essas informações vêm principalmente da análise de sequências de biomoléculas (aminoácidos ou nucleotídeos) em procura de padrões específicos ligados a certas características ou funções. Depois de achados, esses padrões ou marcas podem ser usados na procura das mesmas características em outras proteínas, desconhecidas. Essas análises também nos permite estudar a evolução dessas biomoléculas e de seus organismos, ajudando a compreender como a vida se formou e chegou ao seu nível atual de complexidade.

– **Estrutura secundária**

  A partir da sequência de monômeros descritos em uma ordem específica na estrutura primária, surgem interações entre monômeros vizinhos e com as moléculas de solvente circundantes.

  Essas interações resultam na formação de padrões repetitivos de organização espacial, conhecidos como estrutura secundária. Esses padrões ou elementos aparecem em um número relativamente pequeno de tipos, e a estrutura tridimensional das biomoléculas pode ser descrita como uma combinação desses elementos.

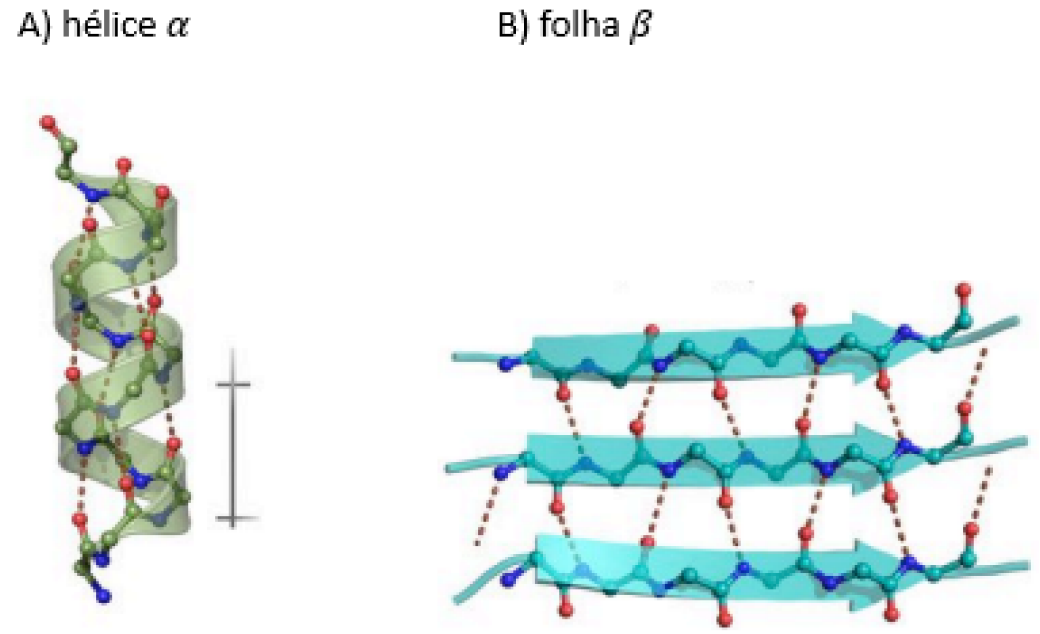
  Diferentes composições de estrutura primária podem gerar o mesmo tipo de estrutura secundária. Não é coincidência que as propriedades dessas estruturas secundárias, mesmo formadas por sequências diferentes, apresentem semelhanças. Por exemplo, uma alça em proteínas é frequentemente uma estrutura secundária bastante flexível, enquanto folhas e hélices tendem a ser mais rígidas (Verli, 2014).

  As estruturas secundárias mais comuns estão relacionadas a proteínas e incluem três grupos principais de elementos: alças, hélices α e folhas β (Figura 7). A hélice α e as folhas β foram inicialmente descritas por Linus Pauling e Robert B. Corey em 1951 (Pauling; Corey, 1951), embora as primeiras propostas para as estruturas em folhas datem de décadas anteriores, em 1933, por Astbury e Bell (Astbury; Bell, 1938).

– **Estrutura terciária**

  A importância do conhecimento da estrutura secundária de biomoléculas reside princi-

Figura 7 – Representação dos tipos mais comuns de estrutura secundária encontrados em proteínas



Fonte: Verli (2014, p.25).

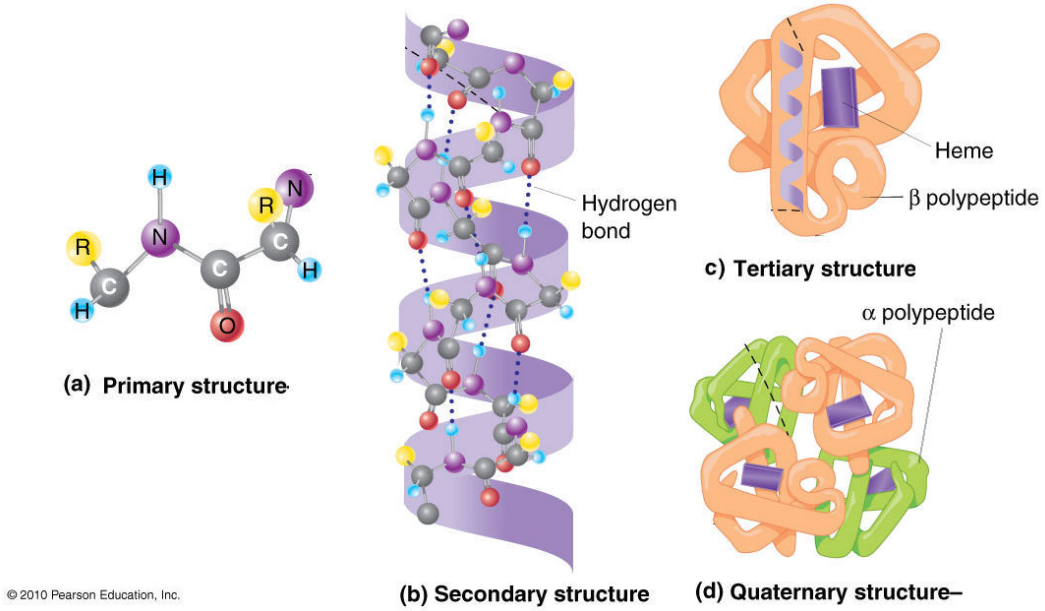
palmente no fato de que esses elementos se organizam no espaço tridimensional, dando origem ao que chamamos de estrutura terciária. Em outras palavras, a estrutura terciária de uma biomolécula específica corresponde à montagem de seus elementos de estrutura secundária. Por outro lado, é a estrutura terciária (ou quaternária, como veremos a seguir) que desempenhará a função biológica da molécula em questão.

Os diversos elementos de estrutura secundária de uma molécula específica se organizam em uma estrutura terciária por meio de um fenômeno conhecido como enovelamento (também chamado de dobramento em português, do termo em inglês "*folding*"). Nesse processo, uma combinação de forças converge para que a biomolécula adote uma conformação mais estável no meio biológico alvo.

– **Estrutura quaternária**

A estrutura quaternária consiste em agregados macromoleculares, predominantemente compostos por proteínas. Essas biomoléculas podem adotar estados oligoméricos, formando dímeros (2 subunidades), trímeros (3 subunidades), tetrâmeros (4 subunidades), pentâmeros (5 subunidades), hexâmeros (6 subunidades) ou mais, conforme necessário para desempenhar funções específicas em condições nativas. Vale ressaltar que nem todas as biomoléculas exibem esse grau de organização. A Figura 8 ilustra os quatro níveis de estruturas.

Figura 8 – Diferentes níveis estruturais de representação proteica



Fonte: Carr (2022, p. 1).

À primeira vista, poderíamos considerar redundante lidar com estruturas 3D ao manipular sequências, conjuntos de informações 1D, uma vez que, em geral, as estruturas de proteínas são determinadas por seus genes. No entanto, essa percepção é limitada e não reflete a verdade em diversas situações. De fato, existem aspectos únicos em cada conjunto de informações, que não são diretamente transferíveis entre eles.

Outro ponto crucial é que o enovelamento de proteínas, em muitas circunstâncias, vai além de sua sequência de aminoácidos. Envolve considerações sobre o ambiente e o local onde a proteína estará na célula ou organismo, a presença de modificações co- ou pós-traducionais, e a interação com chaperonas, proteínas especializadas que desempenham um papel fundamental na correta dobragem e montagem de outras proteínas. Para ilustrar a complexidade desse fenômeno, embora diversas sequências com identidade mínima possam apresentar estruturas 3D extremamente semelhantes, em alguns casos, a substituição de um ou poucos resíduos de aminoácidos pode modificar completamente a função, chegando até a influenciar na forma tridimensional que uma proteína adota.

3.3 Bancos de dados de proteínas

Este capítulo foi largamente inspirado no livro "Introdução à Bioinformática"de autoria do professor Arthur M. Lesk (Lesk, 2008), da Pennsylvania State University, EUA, uma tradução em português publicada pela editora Artmed em 2008.

Com o aumento do número de bancos de dados, a comunicação entre eles tornou-se uma prioridade, facilitando a interatividade entre bancos de dados de biologia molecular. As atividades de bancos de dados em bioinformática podem ser classificadas em arquivamento, focado na conservação e organização, e interpretação, que compila informações biológicas de forma mais útil para apoiar a pesquisa. Diferentes bancos de dados especializam-se em diversos tipos de informações, como sequências de ácidos nucleicos, sequências de proteínas, estruturas, funções de proteínas, entre outros.

O desenvolvimento de bancos de dados em bioinformática revela dois aspectos notáveis. O primeiro é o crescimento expressivo de projetos de bancos de dados individuais que reorganizam os dados armazenados de maneiras diferentes. O segundo é a fusão de muitos bancos individuais em sítios "guarda-chuva"(portais abrangentes), como o Interpro, que inclui bancos de dados como PROSITE, Pfam, PRINTS, SMART e ProDom.

Bancos de dados primários relacionados a macromoléculas biológicas abrangem sequências de ácidos nucleicos, sequências de aminoácidos, estruturas de proteínas e ácidos nucleicos, estruturas cristalográficas de pequenas moléculas, funções de proteínas, padrões de expressão de genes, vias metabólicas, redes de interação e controle, e publicações. Esses bancos desempenham papéis cruciais na pesquisa em bioinformática, fornecendo acesso a uma ampla gama de informações essenciais.

Um dos bancos de dados mais conhecido relacionado a proteínas é o Banco de Dados de Proteínas (PDB, do inglês Protein Data Bank). Um registro no PDB referente a uma proteína inclui, entre outras informações (Lesk, 2008):

- A identificação da proteína e a espécie à qual pertence;
- O responsável pela determinação da estrutura, com referências a publicações descrevendo o processo de determinação;
- Detalhes experimentais relacionados à determinação da estrutura, incluindo informações sobre a qualidade geral do resultado;
- A sequência de aminoácidos;
- Moléculas adicionais presentes na estrutura, como co-fatores, inibidores e solventes, incluindo moléculas de água;
- Atribuições da estrutura secundária;
- Coordenadas atômicas.

O PDB é mantido por uma colaboração global de instituições de pesquisa e orga-

nizações em todo o mundo. Algumas das principais instituições que colaboram com o PDB incluem:

- Protein Data Bank Europe (PDBe): É uma das principais instalações de arquivo de dados biomoleculares da Europa e uma parte do European Bioinformatics Institute (EBI), que por sua vez é parte do European Molecular Biology Laboratory (EMBL). O PDBe fornece acesso a uma variedade de recursos, incluindo a base de dados do PDB, bem como ferramentas e serviços relacionados à análise de estruturas biomoleculares.
- Japan Protein Data Bank (PDBj): É um dos membros fundadores do PDB e opera como um centro de dados para a comunidade biomolecular no Japão. O PDBj fornece acesso a estruturas de proteínas e ácidos nucleicos, bem como a ferramentas e serviços para análise estrutural. Além disso, o PDBj desenvolve e mantém recursos exclusivos, como o Protein Databank Japan-Structural Motif Database (PDBj-MMDB) e o PDBj-BMRB Integrated Database (PDBj-BMRB).

Essas instituições, juntamente com outras ao redor do mundo, desempenham um papel fundamental na coleta, curadoria e disponibilização de estruturas biomoleculares no PDB 9. Sua colaboração garante que a comunidade científica tenha acesso a uma ampla gama de dados estruturais que são essenciais para o avanço da pesquisa em biologia molecular, bioquímica, biomedicina e muitas outras áreas relacionadas.

Figura 9 – Imagem contendo as instituições que integram o Worldwide Protein Data Bank

Fonte: https://www.rcsb.org.

Nas Figuras 10, 11 e 12 podemos visualizar uma parte de um arquivo do PDB referente a proteína E.coli expressed scFv (6QF6).

Uma das vantagens desses bancos de dados reside na capacidade de extrair informações, como as coordenadas x, y e z dos átomos, de milhares de proteínas. Esses dados podem ser processados por pacotes na linguagem Python, por exemplo, para alimentar o treinamento de redes neurais. Esse treinamento visa prever estruturas tridimensionais de proteínas que ainda não

Figura 10 – Imagem contendo a parte superior do arquivo PDB

```
data_6QF6
#
_entry.id   6QF6
#
_audit_conform.dict_name       mmcif_pdbx.dic
_audit_conform.dict_version    5.318
_audit_conform.dict_location   http://mmcif.pdb.org/dictionaries/ascii/mmcif_pdbx.dic
#
loop_
_database_2.database_id
_database_2.database_code
PDB   6QF6
WWPDB D_1200013616
#
loop_
_pdbx_database_related.db_name
_pdbx_database_related.details
_pdbx_database_related.db_id
_pdbx_database_related.content_type
PDB 'CHO expressed'                    6qb9 unspecified
PDB 'CHO expressed in complex with Mcl1' 6qb6 unspecified
#
_pdbx_database_status.status_code                     REL
_pdbx_database_status.status_code_sf                  REL
_pdbx_database_status.status_code_mr                  ?
_pdbx_database_status.entry_id                        6QF6
_pdbx_database_status.recvd_initial_deposition_date   2019-01-09
_pdbx_database_status.SG_entry                        N
_pdbx_database_status.deposit_site                    PDBE
_pdbx_database_status.process_site                    PDBE
_pdbx_database_status.status_code_cs                  ?
_pdbx_database_status.methods_development_category    ?
_pdbx_database_status.pdb_format_compatible           Y
#
_audit_author.name              'Luptak, J.'
_audit_author.pdbx_ordinal      1
_audit_author.identifier_ORCID  0000-0002-9527-8755
```

Fonte: https://www.rcsb.org.
Legenda: Nesse trecho vemos informações de identificação, como nome da proteína, autor e data em que foi depositada.

foram descobertas.

Figura 11 – Imagem mostrando o trecho contendo os 25 primeiros aminoácidos, identificados por três caracteres, de um total de 247 que compõe a sequência de aminoácidos da proteína

```
#
loop_
_entity_poly_seq.entity_id
_entity_poly_seq.num
_entity_poly_seq.mon_id
_entity_poly_seq.hetero
1 1   GLY n
1 2   SER n
1 3   GLN n
1 4   VAL n
1 5   THR n
1 6   LEU n
1 7   LYS n
1 8   GLU n
1 9   SER n
1 10  GLY n
1 11  GLY n
1 12  GLY n
1 13  LEU n
1 14  VAL n
1 15  LYS n
1 16  PRO n
1 17  GLY n
1 18  GLY n
1 19  SER n
1 20  LEU n
1 21  ARG n
1 22  LEU n
1 23  SER n
1 24  CYS n
1 25  ALA n
```

Fonte: https://www.rcsb.org.

Figura 12 – Imagem de mais um trecho do arquivo PDB

```
ATOM   1    N N   . THR A 1 5   ? -78.569 -30.893 -10.226 1.00 42.69  ? 3   THR A N   1
ATOM   2    C CA  . THR A 1 5   ? -77.438 -31.199 -11.111 1.00 42.75  ? 3   THR A CA  1
ATOM   3    C C   . THR A 1 5   ? -77.290 -32.765 -11.098 1.00 47.52  ? 3   THR A C   1
ATOM   4    O O   . THR A 1 5   ? -76.499 -33.342 -10.332 1.00 48.60  ? 3   THR A O   1
ATOM   5    C CB  . THR A 1 5   ? -76.191 -30.265 -10.768 1.00 43.88  ? 3   THR A CB  1
ATOM   6    O OG1 . THR A 1 5   ? -75.283 -30.156 -11.890 1.00 28.95  ? 3   THR A OG1 1
ATOM   7    C CG2 . THR A 1 5   ? -75.438 -30.649 -9.463  1.00 41.51  ? 3   THR A CG2 1
ATOM   8    N N   . LEU A 1 6   ? -78.151 -33.436 -11.911 1.00 41.68  ? 4   LEU A N   1
ATOM   9    C CA  . LEU A 1 6   ? -78.277 -34.902 -12.089 1.00 40.07  ? 4   LEU A CA  1
ATOM   10   C C   . LEU A 1 6   ? -79.015 -35.609 -10.904 1.00 42.05  ? 4   LEU A C   1
ATOM   11   O O   . LEU A 1 6   ? -78.404 -35.991 -9.897  1.00 39.47  ? 4   LEU A O   1
ATOM   12   C CB  . LEU A 1 6   ? -76.938 -35.615 -12.432 1.00 39.35  ? 4   LEU A CB  1
ATOM   13   C CG  . LEU A 1 6   ? -76.206 -35.166 -13.693 1.00 42.86  ? 4   LEU A CG  1
ATOM   14   C CD1 . LEU A 1 6   ? -75.008 -36.009 -13.924 1.00 42.52  ? 4   LEU A CD1 1
ATOM   15   C CD2 . LEU A 1 6   ? -77.093 -35.273 -14.932 1.00 45.24  ? 4   LEU A CD2 1
ATOM   16   N N   . LYS A 1 7   ? -80.344 -35.773 -11.063 1.00 39.38  ? 5   LYS A N   1
ATOM   17   C CA  . LYS A 1 7   ? -81.222 -36.435 -10.101 1.00 39.87  ? 5   LYS A CA  1
ATOM   18   C C   . LYS A 1 7   ? -81.841 -37.679 -10.751 1.00 44.78  ? 5   LYS A C   1
ATOM   19   O O   . LYS A 1 7   ? -82.565 -37.541 -11.734 1.00 44.96  ? 5   LYS A O   1
ATOM   20   C CB  . LYS A 1 7   ? -82.330 -35.482 -9.596  1.00 42.53  ? 5   LYS A CB  1
ATOM   21   C CG  . LYS A 1 7   ? -83.006 -35.994 -8.318  1.00 60.80  ? 5   LYS A CG  1
ATOM   22   C CD  . LYS A 1 7   ? -84.473 -35.563 -8.155  1.00 70.10  ? 5   LYS A CD  1
```

Fonte: https://www.rcsb.org.

Legenda: Podemos identificar na coluna 2 o número de identificação do átomo, na coluna, na coluna 3 o nome do átomo, na coluna 5 o nome do resíduo de aminoácido e nas colunas 9, 10 e 11 as coordenadas x, y, z do átomo em *angstroms*.

4 METODOLOGIA

4.1 O DDGP pesquisa em árvore

O DDGP é uma subclasse específica do Problema de Geometria de Distância Molecular (MDGP) (Liberti *et al.*, 2014a), definido a seguir.

Dado um grafo não direcionado ponderado $G = (V, E, d)$, onde V representa o conjunto de átomos na molécula e E é o conjunto de pares de átomos com distâncias conhecidas, dado por $d : E \to \mathbb{R}^+$, resolver o MDGP envolve encontrar uma função $x : V \to \mathbb{R}^3$ tal que

$$\|x(v_i) - x(v_j)\|_2 = d(v_i, v_j), \ \ \forall \{v_i, v_j\} \in E.$$

O DDGP é um MDGP com uma ordenação particular dos vértices de G. Inicialmente, deve-se identificar um subconjunto de V composto por três vértices que formam uma clique (um subgrafo totalmente conectado). Para cada vértice $v \in V$ fora deste subconjunto inicial, deve haver três vértices que precedem v na ordem estabelecida, conectados a v (ou seja, com distâncias conhecidas para v). Portanto, as condições para a ordenação $v_1, \ldots, v_n \in V$ são as seguintes:

H_1**:** Os três primeiros vértices estão conectados, ou seja, $\{v_1, v_2\}, \{v_1, v_3\}, \{v_2, v_3\} \in E$.

H_2**:** Para qualquer vértice v_i com $i > 3$, existem três outros vértices v_j, v_k, v_l (referidos como *vértices de referência* para v_i), com $j, k, l < i$, tal que $\{v_j, v_i\}, \{v_k, v_i\}, \{v_l, v_i\}$ *em* E.

Uma notação simplificada é usada onde os vértices v_i são representados apenas por seus índices i. Assim, x_i representa as coordenadas do vértice v_i e d_{ij} representa a distância entre os vértices v_i e v_j.

Para eliminar soluções obtidas meramente por rotações e translações de uma determinada solução, as posições dos três primeiros átomos podem ser fixadas da seguinte forma:

$$x_1 = (0,0,0), \ \ x_2 = (d_{1,2}, 0, 0), \ \ x_3 = (d_{1,3}\cos(\theta_{2,3}), d_{1,3}\sin(\theta_{2,3}), 0),$$

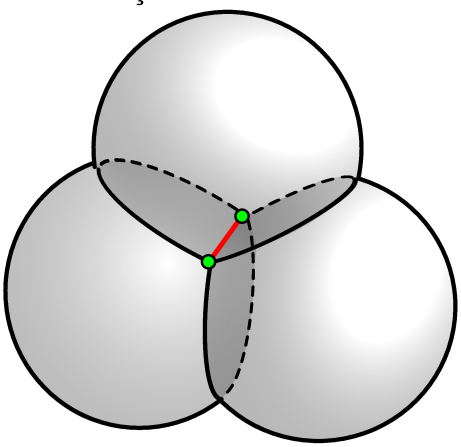
formando os vértices de um triângulo com lados $d_{1,2}, d_{1,3}, d_{2,3}$.

Numa abordagem de construção iterativa para resolver o DDGP, começamos fixando os três primeiros vértices. Posteriormente, sob a hipótese H_2, com os vértices de referência $\{j, k, l\}$ para o vértice $i > 3$ fixos, precisamos encontrar a solução para o seguinte sistema:

$$\begin{aligned} \|x_i - x_j\|_2 &= d_{ij}, \\ \|x_i - x_k\|_2 &= d_{ik}, \\ \|x_i - x_l\|_2 &= d_{il}. \end{aligned} \tag{4.1}$$

Cada uma dessas restrições define uma esfera centrada em um dos vértices de referência, com raio igual à distância entre este centro e o ponto x_i. Portanto, x_i deve estar na interseção dessas três esferas. Assumindo que existe uma solução para o DDGP e que os pontos x_j, x_k, e x_l não são colineares, a intersecção destas três esferas consistirá em no máximo dois pontos (ver Figura 13).

Figura 13 – Desenho de dois pontos na interseção das três esferas



Fonte: elaborada pelo autor.

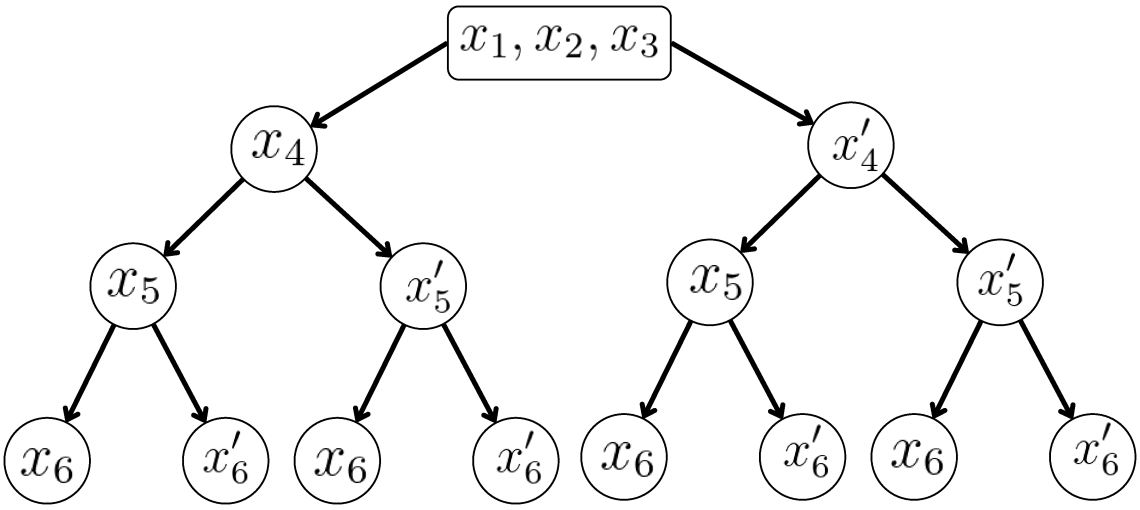
Através deste procedimento construtivo, uma vez fixos os átomos $j < i$, o átomo i terá duas posições possíveis, x_i ou x_i' em $\mathbb{R}^3$, obtido do sistema (4.1). Uma vez escolhido x_i entre as duas possibilidades, podemos continuar o processo fixando o ponto x_{i+1}.

Uma representação natural para todas as configurações possíveis $x = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^{3\times n}$ é uma árvore binária, onde x_1, x_2, x_3 são representados como um único nó raiz (já que são fixos) e seus dois filhos representam as duas possibilidades para x_4, e para cada um deles, as duas possibilidades para x_5 e assim por diante (veja a Figura 14).

Se designarmos o nó esquerdo como filho 0 e o da direita como filho 1, também podemos mapear a posição relativa de cada realização de x_i em relação ao plano π_i definido por x_j, x_k, x_l, onde j, k, l são os átomos de referência do i-ésimo átomo. Especificamente, podemos definir vetores

$$\vec{u} = x_j - x_l, \quad \vec{v} = x_k - x_l, \quad \vec{w} = x_i - x_l,$$

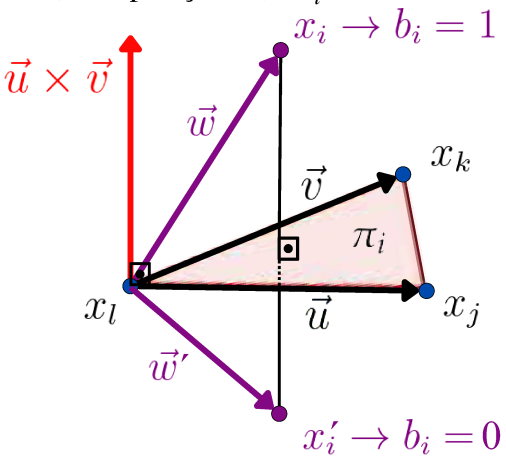
Figura 14 – Desenho de uma árvore binária associada a uma instância DDGP com seis átomos



Fonte: elaborada pelo autor.

e atribua orientação 0 se $\vec{w} \cdot (\vec{u} \times \vec{v}) < 0$, e 1 caso contrário (ver Figura 15).

Figura 15 – Imagem do plano π_i e as posições x_i e x'_i associada com as orientações 0 and 1



Fonte: elaborada pelo autor.

A busca por soluções dentro da árvore binária DDGP pode ser tentada usando força bruta. No entanto, esta abordagem torna-se impraticável para moléculas com centenas de átomos, uma vez que o tamanho da árvore cresce exponencialmente com o número de átomos.

Quando distâncias adicionais, nomeadamente d_{ij} onde j não é um dos átomos de referência do átomo i, são conhecidas, as configurações viáveis são reduzidas pela presença da restrição adicional $(\|x_i - x_j\|_2 = d_{ij})$. Essas restrições adicionais são chamadas de *restrições de poda*, pois tornam inviáveis as ramificações da árvore binária DDGP.

O algoritmo BP (Liberti *et al.*, 2008b; Carvalho *et al.*, 2008) desenvolvido para resolver o problema percorre de forma inteligente a árvore de busca DDGP. Utilizando as restrições de poda, poda galhos identificados como inviáveis, eliminando assim a necessidade de explorar toda a árvore.

Em sua busca por soluções para o DDGP, o algoritmo BP explora caminhos na árvore binária associada usando uma estratégia DFS que favorece arbitrariamente os nós 0 da árvore binária. Por exemplo, em uma árvore binária de comprimento dois, os caminhos explorados seriam $00,\ 01,\ 10,\ 11$.

DFS é um algoritmo fundamental usado na travessia de árvores (Cormen *et al.*, 2022), caracterizado por explorar um ramo o mais profundamente possível antes de retroceder para explorar outros ramos. Embora sempre encontre soluções em árvores finitas (dependentes do tamanho), o DFS é notável por sua eficiência de memória, pois só precisa armazenar uma pilha de nós no caminho atual do nó raiz. Entretanto, é importante observar que o DFS não garante o caminho mais curto para a solução.

Em contraste com o DFS, propomos uma estratégia de busca do melhor primeiro, chamada *Frequency - Based Search* (FBS), um algoritmo que percorre a árvore selecionando qual caminho seguir com base em uma função de avaliação que estima quais nós são mais prováveis para levar a uma solução.

4.2 Uma abordagem FBS definida pelo PDB

O Banco de Dados de Proteínas (PDB) é um recurso crucial para o avanço científico, contendo mais de 1 terabyte de dados estruturais para proteínas, DNA e RNA. O arquivo cresce quase 10% ao ano e facilita mais de 2 milhões de downloads diários de arquivos de dados de estrutura (ABOUT...,). Existem 184.318 entradas relacionadas a proteínas no PDB, das quais 14.134 foram obtidas através de RMN (RCSB...,).

Para montar nosso repositório de dados, selecionamos todas as estruturas proteicas derivadas de experimentos de RMN presentes no PDB, considerando a relevância de tal técnica para o escopo de nossa pesquisa. De cada arquivo PDB selecionado, extraímos as seguintes informações para os átomos do backbone da proteína em questão: seu índice único, seu nome (N, C_α, C, H, H_α) e suas coordenadas em $\mathbb{R}^3$, bem como o índice e a abreviatura de três letras do resíduo ao qual pertence.

É importante destacar que alguns arquivos PDB não descrevem completamente o

esqueleto da proteína. Por exemplo, existem arquivos em que faltam alguns resíduos e em outros faltam alguns átomos. Também optamos por remover resíduos de prolina e glicina, pois esses aminoácidos apresentam características geométricas únicas e são frequentemente estudados separadamente na literatura (Lovell *et al.*, 2003).

Referimo-nos aos trechos da estrutura formada por resíduos contíguos após a remoção de prolinas e glicinas como *segmentos de proteína*. Assim, associados a cada arquivo PDB, geramos vários arquivos, um para cada segmento proteico. O número total de segmentos de proteína obtidos de todos os arquivos de RMN no PDB é 72.983.

4.3 Uma ordenação DDGP para o FBS

Para estabelecer uma estratégia FBS para explorar a árvore de busca DDGP, inspirada nos dados do PDB, deve-se primeiro determinar uma ordem DDGP para os átomos no backbone. Apresentamos uma ordenação DDGP que agrupa átomos em resíduos, utiliza o comprimento das ligações covalentes e os ângulos de ligação, bem como as propriedades geométricas dos planos peptídicos.

No contexto da geometria das proteínas, considera-se que os comprimentos e ângulos das ligações são fixos, apesar dos movimentos internos naturais das proteínas. Esta suposição é conhecida como *hipótese da geometria rígida* (Gibson; Scheraga, 1997). Consequentemente, as distâncias entre átomos conectados por uma ou duas ligações covalentes são conhecidas. Além desta informação de distância, está bem estabelecido na literatura biológica que os átomos "ao redor"de uma ligação peptídica pertencem ao mesmo plano, implicando que todas as distâncias entre estes átomos também são conhecidas (Lavor *et al.*, 2019). Como a ligação peptídica conecta o carbono carboxila do i-ésimo resíduo (C^i) ao nitrogênio amina do i+1-ésimo resíduo (N^{i+1}), os átomos no i-ésimo plano peptídico são C^i_α, C^i, N^{i+1}, C^{i+1}_α (veja a Figura 16). Consideramos também que as distâncias entre H^i, H^i_α e H^i_α, H^{i+1} podem ser detectadas por NMR (Güntert, 1998; Rowland; Taylor, 1996; BILLETER *et al.*, 1982).

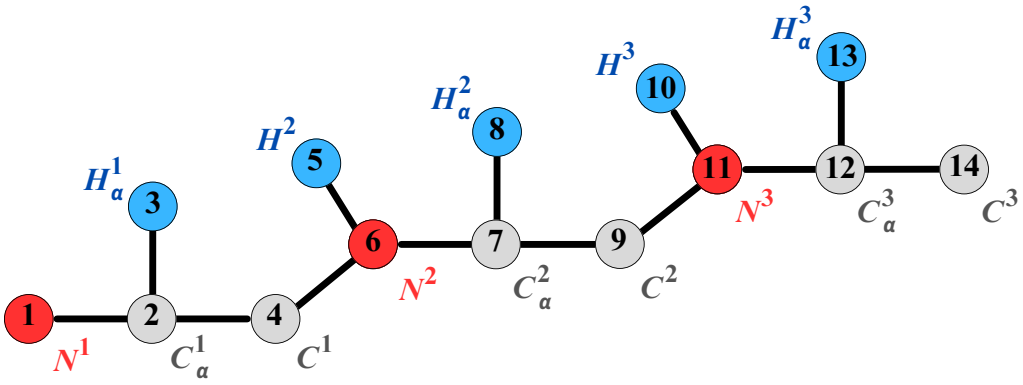
Com base nessas propriedades, usamos a seguinte ordem DDGP para átomos na estrutura da proteína:

$$\rho = (N^1, C^1_\alpha, H^1_\alpha, C^1, \ldots, H^i, N^i, C^i_\alpha, H^i_\alpha, C^i, \ldots, H^n, N^n, C^n_\alpha, H^n_\alpha, C^n). \tag{4.2}$$

A figura 16 ilustra a ordenação ρ para um peptídeo de três resíduos. Os números

dentro dos círculos indicam o índice dos átomos na ordenação.

Figura 16 – Ilustração de 3 resíduos peptídicos em ρ



Fonte: elaborada pelo autor.

A tabela 2 mostra, para cada elemento v em ρ, os três átomos de referência de v cujas coordenadas e distâncias até v são usadas para imergir v em $\mathbb{R}^3$ (coluna “Átomos Predecessores”), bem como suas posições em ρ (coluna "Posições Predecessoras"). Observe que todos os resíduos são compostos por cinco átomos, exceto o primeiro, que é composto por quatro átomos.

Tabela 2 – A ordem do DDGP relacionada ao ρ

Ordem	Átomo	Posições Predecessoras			Átomos Predecessores		
1	N^1						
2	C_α^1						
3	H_α^1						
4	C^1	3	2	1	H_α^1	C_α^1	N^1
$\vdots$							
$5(i-1)$	H^i	$5(i-2)+4$	$5(i-2)+3$	$5(i-2)+2$	C^{i-1}	H_α^{i-1}	C_α^{i-1}
$5(i-1)+1$	N^i	$5(i-1)$	$5(i-2)+4$	$5(i-2)+2$	H^i	C^{i-1}	C_α^{i-1}
$5(i-1)+2$	C_α^i	$5(i-1)+1$	$5(i-1)$	$5(i-2)+4$	N^i	H^i	C^{i-1}
$5(i-1)+3$	H_α^i	$5(i-1)+2$	$5(i-1)+1$	$5(i-1)$	C_α^i	N^i	H^i
$5(i-1)+4$	C^i	$5(i-1)+3$	$5(i-1)+2$	$5(i-1)+1$	H_α^i	C_α^i	N^i
$\vdots$							
$5(n-1)$	H^n	$5(n-2)+4$	$5(n-2)+3$	$5(n-2)+2$	C^{n-1}	H_α^{n-1}	C_α^{n-1}
$5(n-1)+1$	N^n	$5(n-1)$	$5(n-2)+4$	$5(n-2)+2$	H^n	C^{n-1}	C_α^{n-1}
$5(n-1)+2$	C_α^n	$5(n-1)+1$	$5(n-1)$	$5(n-2)+4$	N^n	H^n	C^{n-1}
$5(n-1)+3$	H_α^n	$5(n-1)+2$	$5(n-1)+1$	$5(n-1)$	C_α^n	N^n	H^n
$5(n-1)+4$	C^n	$5(n-1)+3$	$5(n-1)+2$	$5(n-1)+1$	H_α^n	C_α^n	N^n

Fonte: elaborada pelo autor.

Deve-se enfatizar que, para cada elemento de ρ, exceto para os átomos de nitrogênio

(N^i), os átomos de referência são os três predecessores imediatos (ver Tabela 2).

4.4 Uma representação binária para o *backbone* proteico

Existem diferentes representações para proteínas. Em princípio, eles podem ser representados por *strings* formadas por 23 caracteres, cada um representando um dos possíveis resíduos de aminoácidos. Esta é uma representação única, mas não é geométrica. Outra representação possível é uma lista de coordenadas tridimensionais que definem a localização de cada átomo que compõe a proteína.

Como o espaço de soluções de um DDGP pode ser organizado como uma árvore binária, uma solução pode ser representada como uma cadeia binária que incorpora informações geométricas. Formalmente, seguindo a ordem ρ (dada em (4.2)), cada átomo na posição x_i pode ser associado a um bit b_i com base em sua posição relativa ao plano formado por seu átomos de referência (ver 4.1). Assim, as coordenadas cartesianas de cada segmento de proteína podem ser mapeadas, em uma relação um-para-um, para uma *sequência binária* $b = (b_1, \ldots, b_n)$.

Dado que os três primeiros átomos de ρ são facilmente fixados em $\mathbb{R}^3$ e não possuem átomos de referência (ver Figura 16), consideramos $b_1 = b_2 = b_3 = 0$, sem perda de generalidade. Nota-se que para o quarto átomo de ρ, as duas posições possíveis para imersão dele em $\mathbb{R}^3$, resultantes da intersecção das esferas associadas a x_1, x_2 e x_3, são sempre viáveis, já que v_4 nunca tem um quarto vizinho anterior. Isto implica que para cada solução x onde $b_4 = 0$, existe outra solução x', onde $b_4 = 1$, obtida refletindo x em relação ao plano π_4 que passa pelos pontos x_1, x_2 e x_3.

Como consequência direta desta característica única de v_4, sabemos que a representação binária de x' é a inversão total dos bits de b a partir da quarta posição. Portanto, para reduzir a representação das estruturas obtidas desta forma, normalizamos nosso conjunto de dados invertendo todas as representações binárias onde $b_4 = 1$. Com esta escolha, as representações binárias em nosso banco de dados têm o formato $b = (0,0,0,0,b_5,\ldots,b_n)$. Adotamos a representação reduzida $s = (b_5,\ldots,b_n) \in \{0,1\}^{n-4}$, removendo os bits fixos.

4.5 Extraindo sequências binárias

As distâncias conhecidas *a priori* relacionam átomos dentro do mesmo resíduo ou átomos de resíduos consecutivos. Além disso, sabe-se que pequenos trechos de proteínas podem

seguir padrões geométricos. Particularmente, os gráficos de Ramachandran destacam alguns desses padrões para sequências de três resíduos (Ramakrishnan; Ramachandran, 1965). Para incorporar as relações geométricas entre resíduos adjacentes, extraímos todas as *subsequências binárias* associadas a grupos de até cinco resíduos consecutivos em uma cadeia proteica.

Deve-se notar que, com exceção do primeiro resíduo em um segmento de proteína, cada resíduo é composto por cinco átomos distintos a serem considerados, a saber, N, C_α, C, H e H_α, conforme ilustrado na Figura 16. Consequentemente, as subsequências binárias obtidas variam em tamanho com o número de resíduos consecutivos considerados, sendo 5, 10, 15, 20 e 25 bits, respectivamente.

Cada subsequência extraída está relacionada aos quatro bits imediatamente anteriores. Como apenas a primeira subsequência tem garantia de normalização, ou seja, o bit que a precede é definido como 0, pode ser necessário inverter todos os bits da subsequência analisada para manter a normalização das strings em consideração. Portanto, em todos os casos, é feita uma verificação do valor binário do átomo imediatamente anterior ao primeiro átomo da subsequência binária em questão. Se este valor for 1, todos os valores binários na subsequência serão invertidos.

Por exemplo, considere um segmento de cinco resíduos com a seguinte sequência binária reduzida,

$$s = (1,0,0,1,1,0,1,1,1,0,0,0,1,1,0,1,0,1,0,1), \tag{4.3}$$

onde a sequência completa $b = (0,0,0,0,s)$ inclui os bits do primeiro resíduo.

As seções de s associadas a cada um de seus resíduos são:

$$\begin{aligned} \textit{Residue}\ 2: &\quad (1,0,0,1,1,\\ \textit{Resduo}\ 3: &\quad 0,1,1,1,0,\\ \textit{Resduo}\ 4: &\quad 0,0,1,1,0,\\ \textit{Resduo}\ 5: &\quad 1,0,1,0,1). \end{aligned} \tag{4.4}$$

Em relação à sequência reduzida representada por s, coletamos quatro subsequências de um resíduo, três subsequências de dois resíduos, duas subsequências de três resíduos e uma subsequência de quatro resíduos. Observe que a sequência s é composta por bits associados a apenas quatro resíduos, o que implica na ausência de subsequências contendo cinco resíduos.

As subsequências binárias de três resíduos consecutivos a serem coletados são aquelas associadas aos tripletos de resíduos (2, 3, 4) e (3, 4, 5), com $s_1 = (1,0,0,1,1,0,1,1,1,0,0,0,1,1,0)$ e $s_2 = (0,1,1,1,0,0,0,1,1,0,1,0,1,0,1)$, respectivamente. O valor binário na posição em s imediatamente antes da seção associada aos resíduos (3, 4, 5) é 1 (ou seja, o binário da última posição de *Resíduo 2*). Portanto, todos os binários de s_2 devem ser invertidos: $s_2 \leftarrow (1,0,0,0,1,1,1,0,0,1,0,1,0,1,0)$.

Nosso conjunto de dados completo com suas 72.983 configurações tridimensionais e as respectivas codificações binárias pode ser gerado automaticamente pelo script Python disponível no repositório https://github.com/romulomarques/proteinGeometryData.

4.6 A função de avaliação da FBS

Uma solução para o DDGP é representada por um caminho viável da raiz até uma folha na árvore binária do seu espaço de solução. No DFS, a busca por tal caminho é realizada em profundidade, preferindo o nó 0 em cada nível. Na nova estratégia de busca, a FBS busca identificar um caminho viável baseado na frequência de subsequências binárias de comprimentos 5, 10, 15, 20 e 25. Mais especificamente, agrupamos as subsequências por tamanho, contamos a frequência de cada ocorrência e as ordenamos em frequência decrescente dentro de cada grupo.

Embora a árvore DDGP possa ter uma altura maior que 25, restringimos nossa análise a subsequências de tamanhos 5, 10, 15, 20 e 25 por dois motivos: o primeiro é que subsequências de comprimentos que são múltiplos de cinco representam resíduos com todos os seus cinco átomos; e a segunda é que a frequência de subsequências maiores que 25 é relativamente baixa comparada ao número de possíveis subsequências de tamanhos menores. Em situações onde a árvore DDGP possui altura maior que as subsequências consideradas, a estratégia FBS pode ser empregada para construir incrementalmente um caminho preferencial concatenando subsequências consecutivas.

Idealmente, a estratégia ideal de busca em árvore é aquela que requer o menor número de nós visitados. Para DFS, assumindo que a solução binária é $b = (b_1, \ldots, b_n)$, o número de nós visitados é dado por

$$DFS(b) = n + \sum_{i=0}^{n-1} b_{n-i}(2^{i+1} - 1), \tag{4.5}$$

onde n é o número de nós no caminho que representa a solução, e o termo b_{n-i} indica se a

subárvore enraizada no nó à esquerda do nó associado a b_{n-i} foi visitada (valor 1 neste caso e 0 caso contrário). Quando $b_{n-i} = 1$, deve-se somar o número total de nós na subárvore associada, que é dado pelo fator $(2^{i+1} - 1)$.

No FBS, os caminhos associados às sequências mais frequentes são testados primeiro. Ao contrário do DFS, o FBS não aproveita caminhos já percorridos. Em vez disso, cada caminho alternativo é testado separadamente. Assim, se a posição de b na ordem FBS for k, então k caminhos de comprimento n devem ser testados, totalizando um custo de

$$FBS(b) = k \times n. \tag{4.6}$$

A figura 17 ilustra uma árvore binária de altura quatro, composta por nós numerados de 1 a 15, seguindo a ordem de visitação do DFS. Abaixo da representação da árvore, os quadrados exibem a ordenação FBS dos 8 caminhos da raiz às folhas (ordenação aprendida no PDB). Suponha que o caminho $(1, 9, 13, 14)$, destacado com setas em azul, seja a representação binária da solução. Além disso, suponha que b esteja na terceira posição na ordem FBS. Com essas suposições, apenas o nó 15 não seria visitado pelo DFS na busca pela solução b, e o algoritmo FBS avaliaria três caminhos de comprimento quatro, envolvendo um total de 12 nós: (1,9,10,11) , (1,2,6,7) e (1,9,13,14).

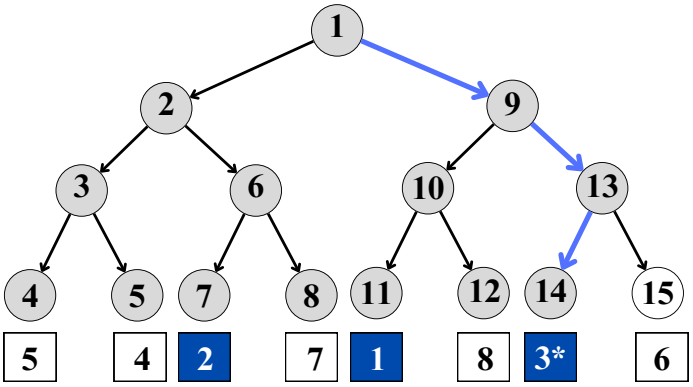
Aplicando as equações (4.5) e (4.6), obtemos

$$DFS(b) = 4 + 0 \times (2^1 - 1) + 1 \times (2^2 - 1) + 1 \times (2^3 - 1) = 14$$

e

$$FBS(b) = 3 \times 4 = 12.$$

Figura 17 – Diagrama ilustrando os nós da árvore binária visitados pelo DFS e pelo FBS



Fonte: elaborada pelo autor.

5 RESULTADOS

Nesta seção, apresentamos uma análise estatística descritiva das subsequências binárias extraídas do PDB. Destacamos que a distribuição dessas subsequências binárias não é uniforme. Além disso, comparamos o desempenho do DFS e do FBS em termos do número de nós visitados na busca por soluções na árvore binária do DDGP.

Na tabela 3, para cada comprimento, a coluna *quantidade* denota o número de subsequências binárias presentes em nosso conjunto de dados, a coluna k representa o número de subsequências binárias distintas e a coluna $k_{\max}$ indica o número máximo de subsequências binárias possíveis, definido como $k_{\max} = 2^{tamanho}$. A coluna $k/k_{\max}$ revela a fração de subsequências observadas em relação ao universo de subsequências matematicamente possíveis. Finalmente, a coluna $quantidade/k_{\max}$ é a razão entre o número de subsequências binárias e o número máximo de subsequências possíveis.

Tabela 3 – Número de trechos binárias exclusivos de cada tamanho

$tamanho$	$quant.$	k	k_{max}	k/k_{max}	$quant./k_{max}$
5	758,977	32	32	1.000	23,718.03
10	686,020	396	1,024	0.387	669.94
15	613,063	4,268	32,768	0.130	18.71
20	540,106	54,742	1,048,576	0.052	0.52
25	475,732	189,904	33,554,432	0.006	0.01

Fonte: elaborada pelo autor.

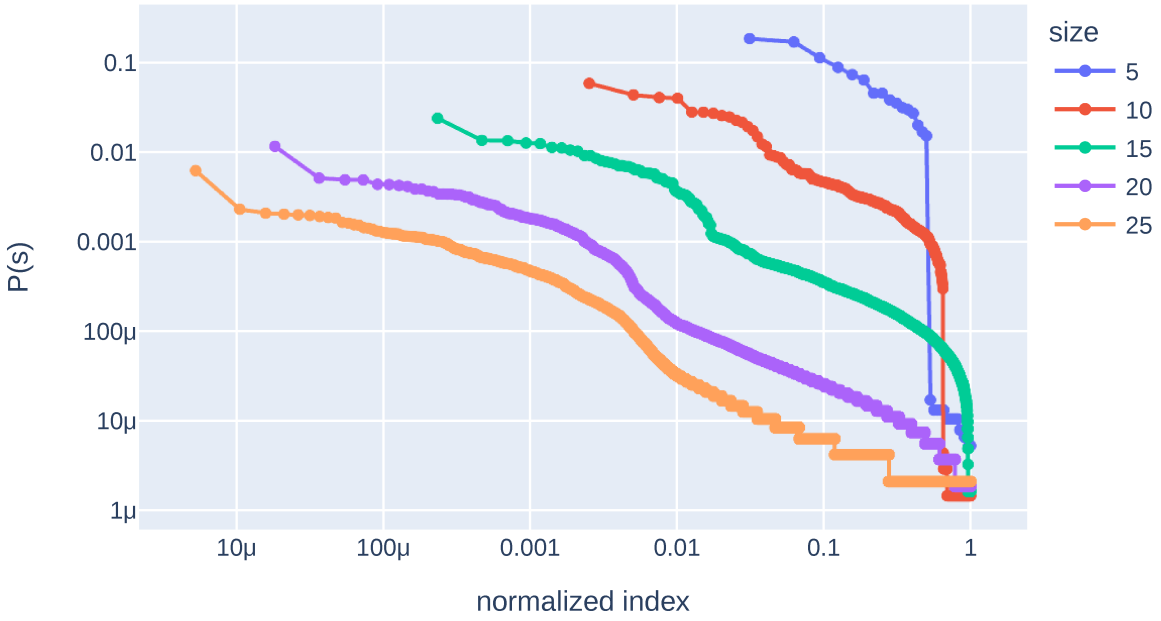
Na Tabela 3, exceto para comprimento 5, a fração de subsequências observadas ($k/k_{\max}$) permanece bem abaixo de 1. Por outro lado, para comprimentos 10 e 15, o tamanho da amostra (*quantidade*) é significativamente maior que o número máximo de subsequências possíveis ($k_{\max}$). Por exemplo, para tamanho 15, a amostra é quase 19 vezes maior que $k_{\max}$. Ou seja, nesses dados, a baixa fração de subsequências observadas em uma amostra relativamente grande, sugerem pelo menos três interpretações possíveis: ou temos uma amostra não representativa (mesmo que tenhamos usado todos os dados de RMN do PDB), ou a "Natureza" não gera todas as subsequências binárias matematicamente possíveis ou algumas subsequências têm uma probabilidade de ocorrência extremamente baixa. No entanto, este raciocínio não é aplicável a subsequências de comprimento 20 e 25, pois o tamanho da amostra é consideravelmente menor que o número de subsequências possíveis: para tamanho 20, a razão entre *quantidade* e $k_{\max}$ é 0,52 ; para o tamanho 25, a proporção cai para apenas 0,01.

Para a aplicação da estratégia FBS, atribuímos índices às subsequências binárias,

ordenando-as em ordem decrescente em relação às suas probabilidades de ocorrência. Depois disso, normalizamos os índices dividindo-os pelo número total de subsequências de cada tamanho. Como resultado, os índices normalizados possuem valores no intervalo $(0, 1]$. Ou seja, se tivermos sequências s_1, s_2 e s_3 com probabilidades de ocorrência de $0,1$, $0,6$ e $0,3$, respectivamente, então os índices normalizados seriam $1/3$ para s_2, $2/3$ para s_3 e 1 para s_1.

A figura 18 mostra a distribuição de probabilidade (frequências relativas) para subsequências binárias de comprimentos 5, 10, 15, 20 e 25 identificadas por seus índices normalizados ($\mu = 10^{-6}$). Dado que as distribuições de probabilidade apresentam um declínio considerável e monótono, podemos concluir que estas distribuições não são uniformes. Isto reforça que, no PDB, algumas subsequências binárias são favorecidas.

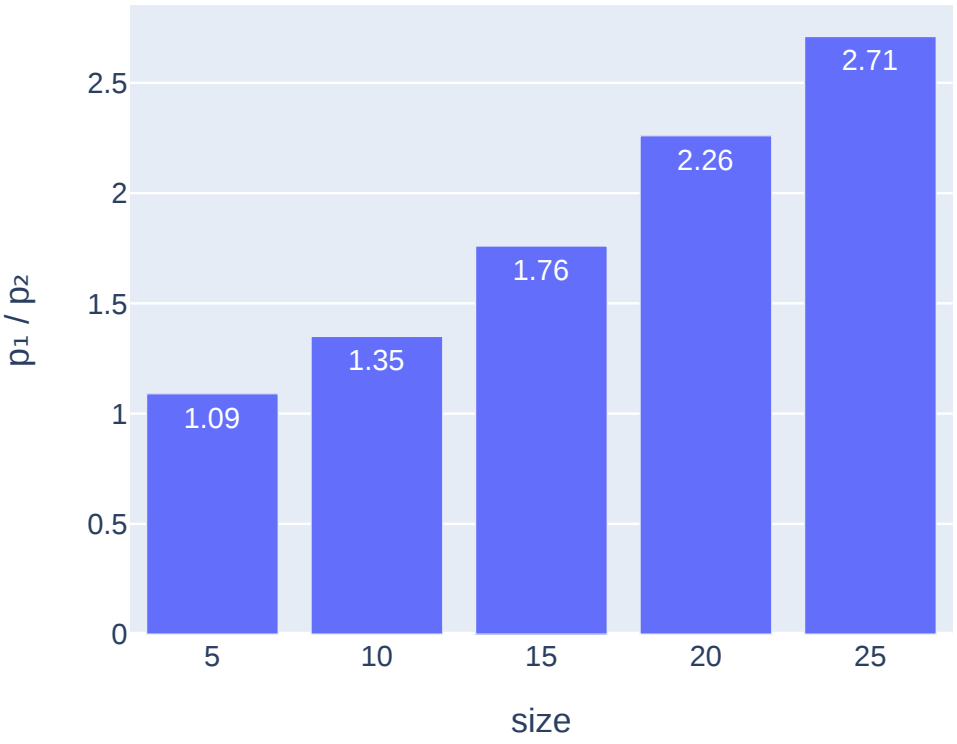
Figura 18 – Informações de probabilidade de ocorrência de subsequências binárias de proteínas



Fonte: elaborada pelo autor.
Legenda: Probabilidade, em escala logarítmica, de cada configuração de subsequência binária extraída do PDB. As subsequências estão em ordem decrescente de probabilidades.

Na verdade, as subsequências mais frequentes têm probabilidades de ordens de grandeza maiores que as outras. Ainda na Figura 18, vê-se claramente a diferença entre as duas maiores probabilidades para cada comprimento. Na Figura 19 são mostradas as razões entre essas probabilidades. Por exemplo, para subsequências de comprimento 20, a razão entre as duas probabilidades mais altas é $2,26$, e para comprimento 25, esta razão é $2,71$.

Figura 19 – Gráfico com informações de probabilidade de ocorrência de subsequências binárias de proteínas

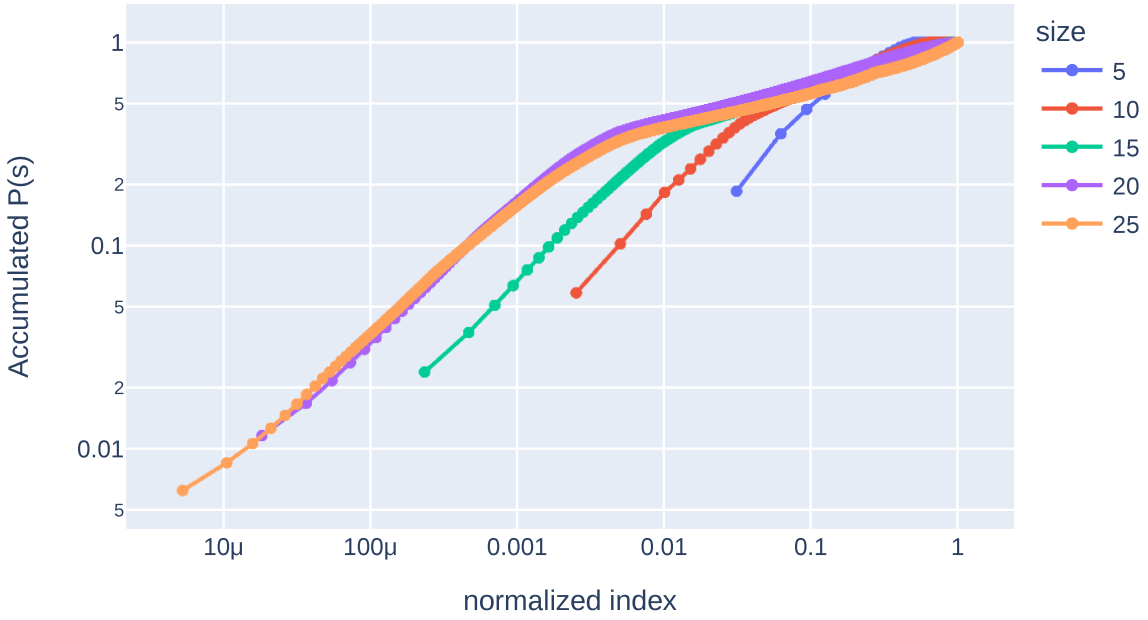


Fonte: elaborada pelo autor.
Legenda: Razão entre as probabilidades da subsequência binária mais frequente e da segunda mais frequente, para cada tamanho de subsequência.

A Figura 20 apresenta as probabilidades acumuladas para subsequências com comprimentos de 5, 10, 15, 20 e 25, utilizando os mesmos índices normalizados mostrados na Figura 18. É observável que, para todos os comprimentos, os primeiros 10% dos índices normalizados, que representam as subsequências mais frequentes, acumulam pelo menos 50% das probabilidades. Esta observação destaca as diferenças na distribuição das probabilidades de ocorrência de subsequências binárias no PDB.

Nas Figuras 21 e 22 é apresentada a distribuição de probabilidade acumulada, descrevendo a relação entre os custos associados à execução do DFS e do FBS na busca por todas as subsequências binárias contidas em nosso conjunto de dados. Os custos considerados são o número de nós visitados durante a exploração da árvore binária do DDGP e são calculados através das equações (4.5) e (4.6). Neste contexto, valores maiores que 1 indicam desempenho superior do algoritmo FBS em comparação ao DFS. Como pode ser visto, o desempenho relativo

Figura 20 – Gráfico da probabilidade acumulada das subsequências ordenadas em escala logarítmica



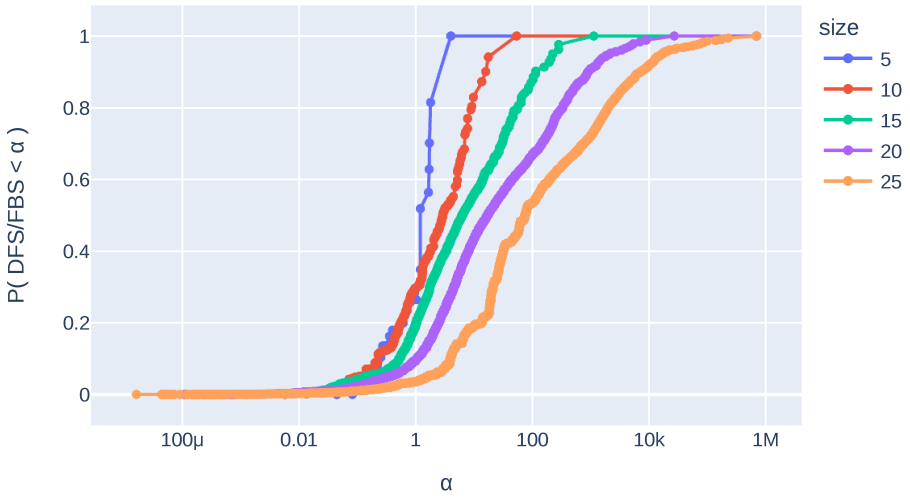
Fonte: elaborada pelo autor.

do FBS aumenta com o tamanho das subsequências. Além disso, para todos os comprimentos, o DFS tem melhor desempenho em, no máximo, 30% das subsequências (ou seja, para cada comprimento de subsequência, $P(DFS/FBS < 1) \leq 0,3$).

A figura 22 exibe a fração de instâncias em que o custo do algoritmo DFS é pelo menos α vezes maior que o custo do algoritmo FBS. Quando $\alpha = 1.5$, esta fração é 50% em instâncias envolvendo subsequências de tamanho 5. Porém, para subsequências de tamanho 25, o FBS é menos oneroso em mais de 90% das instâncias. Notavelmente, para instâncias maiores, o FBS é até milhares de vezes mais eficiente que o DFS ($\alpha = 1000$). Estes dados sugerem que o desempenho do FBS em relação ao DFS aumenta consideravelmente com o tamanho das instâncias.

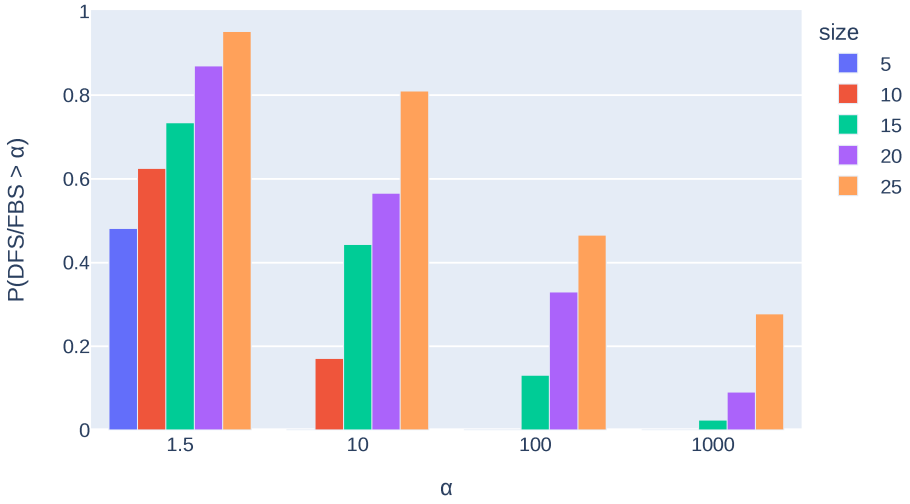
Apesar dos experimentos sugerirem uma clara vantagem do FBS sobre o DFS, é importante notar que as subsequências binárias extraídas do PDB constituem apenas uma amostra do possível espaço de subsequências.

Figura 21 – Gráfico da probabilidade do custo relativo (custo DFS dividido pelo custo FBS) para cada tamanho de subsequência binária



Fonte: elaborada pelo autor.
Legenda: Probabilidade de custo relativo sendo menor que α, com α atingindo até 1 milhão. Valores de α maiores que 1 significam que o FBS é mais eficiente que o DFS.

Figura 22 – Gráfico da probabilidade do custo relativo (custo DFS dividido pelo custo FBS) para cada tamanho de subsequência binária



Fonte: elaborada pelo autor.
Legenda: Probabilidade de custo relativo ser maior que 1. 5, 10, 100 e 1000.

6 CONCLUSÕES E TRABALHOS FUTUROS

Até onde sabemos, esta é a primeira vez que informações do PDB foram utilizadas para resolver o Problema de Geometria de Distância Discretizável (DDGP), um modelo para determinar estruturas de proteínas usando dados de Ressonância Magnética Nuclear (NMR), que se baseia nas propriedades combinatórias de o problema.

Explorando essa estrutura combinatória, definimos uma representação de cadeia binária para as coordenadas dos átomos do esqueleto da proteína. Observamos que substrings binárias exibem padrões com distribuição de frequência não uniforme, o que nos motivou a propor uma nova busca no espaço de soluções DDGP, denominada Busca Baseada em Frequência (FBS).

Os resultados obtidos mostram que a FBS pode ser mais eficiente do que a busca em profundidade comumente utilizada na literatura, sendo pelo menos 50% mais eficiente em 70% dos casos testados. A comparação de desempenho é baseada no número de nós visitados na árvore de busca.

Destacamos que enquanto a busca em profundidade explora todo o espaço matematicamente possível, a FBS, com base nas informações estatísticas do PDB, explora apenas uma parte do espaço, mas que tem maior probabilidade de conter soluções viáveis.

A evidência numérica apresentada neste trabalho sugere, portanto, que uma adaptação do método Branch-and-Prune (BP) que utiliza FBS tem o potencial de ser mais eficiente do que a abordagem convencional de BP.

Futuramente podemos utilizar o FBS em algoritmos de busca, como por exemplo, o Branch and Prune e assim prever a estrutura completa da proteína.

REFERÊNCIAS

ABOUT Us - Research Collaboratory for Structural Bioinformatics Protein Data Bank (RCSB PDB). Disponível em: https://www.rcsb.org/pages/about-us/index. Acesso em: 25 nov. 2023.

ALBERTS, B. *et al.* **Biologia molecular da célula**. 6. ed. Porto Alegre: Artmed, 2017.

ANFINSEN, C. B. Principles that govern the folding of protein chains. **Science**, [*s.l.*], v. 181, n. 4096, p. 223–230, 1973.

ASTBURY, W. T.; BELL, F. O. Some recent developments in the x-ray study of proteins and related structures. **Cold Spring Harbor symposia on quantitative biology, [***s.l.*], v. 6, p. 109–121, 1938.

BERMAN, H. M. *et al.* The Protein Data Bank. **Nucleic Acids Research**, v. 28, n. 1, p. 235–242, 01 2000. ISSN 0305-1048.

BILLETER, M.; BRAUN, W.; WÜTHRICH, K. Sequential resonance assignments in protein h nuclear magnetic resonance spectra: computation of sterically allowed proton-proton distances and statistical analysis of proton-proton distances in single crystal protein conformations. **Journal of Molecular Biology**, v. 155, p. 321–346, 1982.

BLUMENTHAL, L. M. **Theory and applications of distance geometry**. Bronx, N.Y.: Chelsea Pub. Co., 1970.

CABEEN, M. *et al.* **An integrated introduction to chemistry and biology**, [*s.l.*]. 2020, p. 27. Disponível em: https://www.labxchange.org/library/books/d16810cd-172e-4c2d-b82a-ba21e5dfbe0f. Acesso em: 25 nov. 2023.

CAO, Y.; MEZZENGA, R. Food protein amyloid fibrils: origin, structure, formation, characterization, applications and health implications. **Advances in colloid and interface science,** [*s.l.*], v. 269, p. 334–356, 2019.

CARR, S. M. **Four levels of protein structure**, [*s.l.*], 2022. Disponível em: https://www.mun.ca/biology/scarr/iGen3_06-04.html. Acesso em: 25 nov. 2023.

CARVALHO, R. S.; LAVOR, C.; PROTTI, F. Extending the geometric build-up algorithm for the molecular distance geometry problem. **Information Processing Letters**, v. 108, p. 234–237, 2008.

CHANG, M. *et al.* **Atlas of protein sequence and structure**. Silver Spring, Marylan: National Biomedical Research Foundation, 1965.

CORMEN, T. H. *et al.* **Introduction to algorithms**. 4. ed. Cambridge, Massachusett: The MIT Press, 2022.

CRIPPEN, G. M. *et al.* **Distance geometry and molecular conformation**. Taunton: Research Studies Press, 1988.

DILL, K. A.; MACCALLUM, J. L. The protein-folding problem, 50 years on. **Science**, [*s.l.*], v. 338, n. 6110, p. 1042–1046, 2012.

DONG, Q.; WU, Z. A geometric build-up algorithm for solving the molecular distance geometry problem with sparse distance data. **Journal of Global Optimization**, [*s.l.*], v. 26, p. 321–333, 2003.

GIBSON, K. D.; SCHERAGA, H. A. Energy minimization of rigid-geometry polypeptides with exactly closed disulfide loops. **Journal of Computational Chemistry**, v. 18, n. 3, p. 403–415, 1997.

GÜNTERT, P. Structure calculation of biological macromolecules from nmr data. **Quaterly Reviews of Biophysics**, v. 31, p. 145–237, 1998.

LAVOR, C.; LIBERTI, L. **Um convite à geometria de distâncias**. São Carlos: SBMAC, 2014.

LAVOR, C. *et al.* The discretizable molecular distance geometry problem. **Computational Optimization and Applications**, v. 52, p. 115–146, 2012.

LAVOR, C. *et al.* **An introduction to distance geometry applied to molecular geometry**. Cham: Springer, 2017.

LAVOR, C. *et al.* Minimal nmr distance information for rigidity of protein graphs. **Discrete Applied Mathematics**, v. 256, p. 91–104, 2019.

LESK, A. M. **Introdução à bioinformática**. 2. ed. Porto Alegre: Artmed, 2008.

LEVINTHAL, C. Molecular model-building by computer. **Scientific american**, [*s.l.*], v. 214, n. 6, p. 42–53, 1966.

LIBERTI, L.; LAVOR, C.; MACULAN, N. A branch-and-prune algorithm for the molecular distance geometry problem. **International Transactions in Operational Research**, [*s.l.*], v. 15, n. 1, p. 1–17, 2008.

LIBERTI, L.; LAVOR, C.; MACULAN, N. A branch-and-prune algorithm for the molecular distance geometry problem. **International Transactions in Operational Research**, v. 15, p. 1–17, 2008.

LIBERTI, L. *et al.* Euclidean distance geometry and applications. **SIAM review**, [*s.l.*], v. 56, n. 1, p. 3–69, 2014.

LIBERTI, L. *et al.* On the number of realizations of certain henneberg graphs arising in protein conformation. **Discrete Applied Mathematics**, [*s.l.*], v. 165, p. 213–232, 2014.

LOVELL, S. C. *et al.* Structure validation by c_α geometry: ϕ, ψ and c_β deviation. **Proteins:** Structure, Function, and Genetics, v. 50, n. 3, p. 437–450, 2003.

MENGER, K. Untersuchungen über allgemeine metrik. **Mathematische Annalen**, Springer, v. 100, n. 1, p. 75–163, 1928.

MORÉ, J. J.; WU, Z. Global continuation for distance geometry problems. **SIAM Journal on Optimization**, SIAM, v. 7, n. 3, p. 814–836, 1997.

MORÉ, J. J.; WU, Z. Distance geometry optimization for protein structures. **Journal of Global Optimization**, [*s.l.*], v. 15, p. 219–234, 1999.

MUCHERINO, A. *et al.* **Distance geometry:** theory, methods, and applications. New York: Springer, 2012.

NOBELPRIZE.ORG. **The Nobel Prize in Chemistry 1962**, [*s.l.*], 2022. 1962. Disponível em: https://www.nobelprize.org/prizes/chemistry/1962/summary/. Acesso em: 25 nov. 2023.

NOTARI, D. L.; ALBA, G. D.; SILVA, S. d. A. o. **Bioinformática** : contexto computacional e aplicações. Caxias do Sul, RS: Educs, 2020.

PAULING, L.; COREY, R. B. Configurations of polypeptide chains with favored orientations around single bonds: two new pleated sheets. **Proceedings of the National Academy of Sciences**, [*s.l.*], v. 37, n. 11, p. 729–740, 1951.

RAMACHANDRAN, G. N. **Conformation of Biopolymers:** Papers Read at an International Symposium Held at the University of Madras, 18-21 January 1967. [*S. l.*]: Elsevier, 2013.

RAMAKRISHNAN, C.; RAMACHANDRAN, N. Stereochemical criteria for polypeptide and protein chain conformations: Ii. allowed conformations for a pair of peptide units. **Biophysical Journal**, v. 5, n. 6, p. 909–933, 1965.

RCSB PDB Statistics: growth of structures from NMR experiments released per year. Disponível em: https://www.rcsb.org/stats/growth/growth-nmr. Acesso em: 25 nov. 2023.

ROWLAND, R. S.; TAYLOR, R. Intermolecular nonbonded contact distances in organic crystal structures: comparison with distances expected from van der waals radii. **The Journal of Physical Chemistry**, v. 100, p. 7384–7391, 1996.

SOUZA, M. *et al.* Hyperbolic smoothing and penalty techniques applied to molecular structure determination. **Operations Research Letters**, [*s.l.*], v. 39, n. 6, p. 461–465, 2011.

SOUZA, M. F. **Suavização hiperbólica aplicada à otimização de geometria molecular**. 2010. 86 p. Tese (Doutorado em Engenharia de Sistemas e Computação) - Universidade Federal do Rio de Janeiro, 2010.

STOTZ, R. H.; WARD, J. E. **Operating manual for the ESL display console**. [*S. l.*]: Massachusetts Institute of Technology, Electronic Systems Laboratory, 1965.

THIEL, W.; HUMMER, G. Methods for computational chemistry. **Nature**, [*s.l.*], v. 504, n. 7478, p. 96–97, 2013.

VERLI, H. **Bioinformática:** da biologia à flexibilidade molecular. São Paulo: SBBq, 2014.

VVÜTHRICH, K. Protein structure determination in solution by nuclear magnetic resonance spectroscopy. **Science**, v. 243, p. 4887, 1989.

WATSON, J. D.; CRICK, F. H. Molecular structure of nucleic acids: a structure for deoxyribose nucleic acid. **Nature**, [*s.l.*], v. 171, n. 4356, p. 737–738, 1953.

WOLYNES, P. G. Evolution, energy landscapes and the paradoxes of protein folding. **Biochimie**, [*s.l.*], v. 119, p. 218–230, 2015.

YEMINI, Y. The positioning problem: a draft of an intermediate summary. *In*: PROCEEDINGS of the conference on distributed sensor networks, [*S.l., s.n.*], 1978. p. 137–145.