

GEOMETRIA DE DISTÂNCIAS E ÁLGEBRAS GEOMÉTRICAS APLICADAS A CONFORMAÇÃO MOLECULAR

Guilherme Philippi

Departamento de Matemática
Universidade Federal de Santa Catarina, Blumenau

Orientado por Felipe Fidalgo

4º Seminário de Iniciação Científica
UFSC Campus Blumenau

23 de novembro de 2021



- 1 Motivação
- 2 Nosso problema
- 3 Objetivo
- 4 BP com Quatérnios
- 5 Simulações computacionais
- 6 Referências

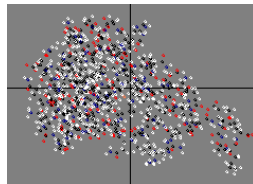
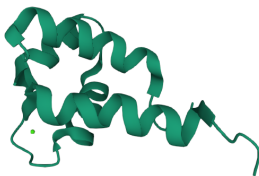
Início da seção 1

- 1 Motivação
- 2 Nosso problema
- 3 Objetivo
- 4 BP com Quatérnios
- 5 Simulações computacionais
- 6 Referências

Worldwide PDB

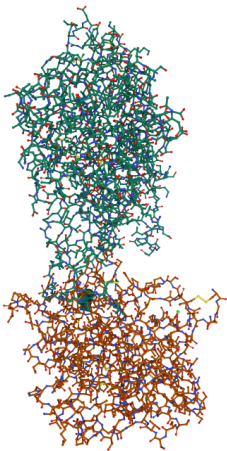


Todas as informações sobre a estrutura 3D de proteínas são concentradas no repositório Protein Data Bank (PDB) [1].

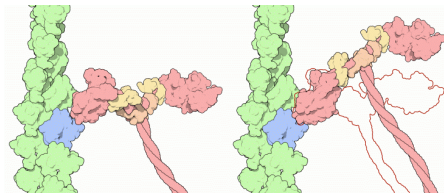
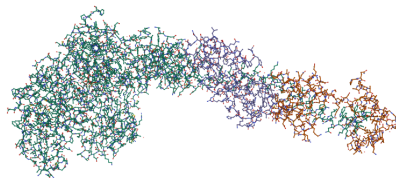


Motor muscular

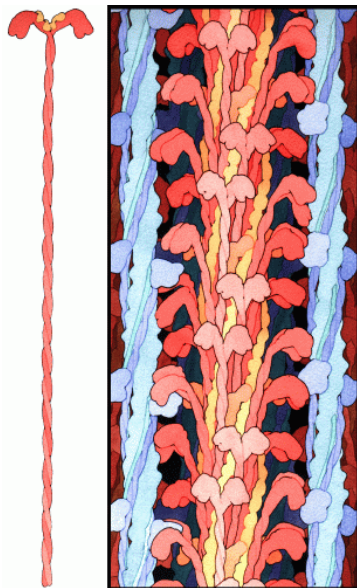
Actina



Miosina



Motor muscular



Início da seção 2

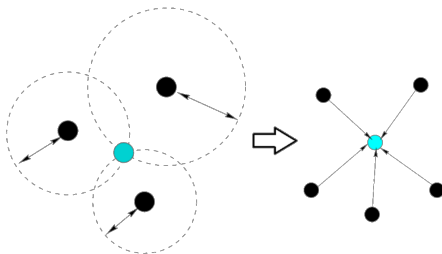
- 1 Motivação
- 2 **Nosso problema**
- 3 Objetivo
- 4 BP com Quatérnios
- 5 Simulações computacionais
- 6 Referências

Geometria de Distâncias

Definição (Distance Geometry Problem (DGP) [2])

Dados um grafo simples, ponderado e conectado $G = (V, E, d)$ e um inteiro $K > 0$, encontre uma realização $x : V \rightarrow \mathbb{R}^K$ tal que:

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



Chamamos G de grafo DGP. Esse é um problema **NP**-completo para $K = 1$ e **NP**-difícil para $K > 1$ [3].

Discretização do problema molecular

Definição (Discretizable Molecular DGP (DMDGP) [4])

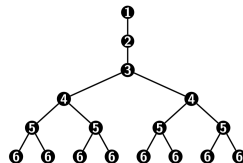
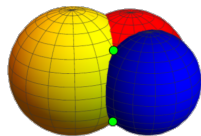
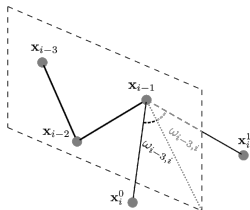
Dado um grafo DGP e uma ordenação nos vértices v_1, \dots, v_n tal que

- Existe uma realização válida para v_1, v_2, v_3 e
- Para todo $i \geq 4$, o conjunto $\{v_{i-3}, v_{i-2}, v_{i-1}, v_i\}$ é um clique com

$$d_{i-3,i-2} + d_{i-2,i-1} > d_{i-3,i-1},$$

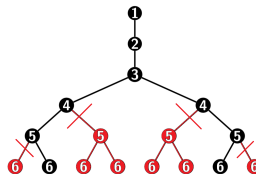
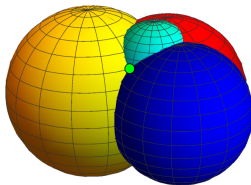
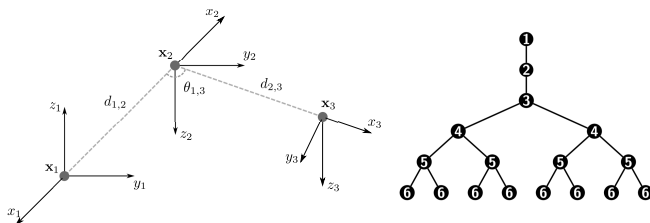
encontre uma realização $x : V \rightarrow \mathbb{R}^3$ tal que

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



Algoritmo Branch-&-Prune (BP)

É definido em três partes: Inicialização, branch e prune.



Algoritmo Branch-&Prune (BP)

Considerando $\mathbf{x}_i \in \mathbb{R}^3, i = 1, \dots, n$ da forma (x_{i1}, x_{i2}, x_{i3}) , temos:

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

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

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

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

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

Início da seção 3

- 1 Motivação
- 2 Nosso problema
- 3 Objetivo**
- 4 BP com Quatérnios
- 5 Simulações computacionais
- 6 Referências

Objetivo

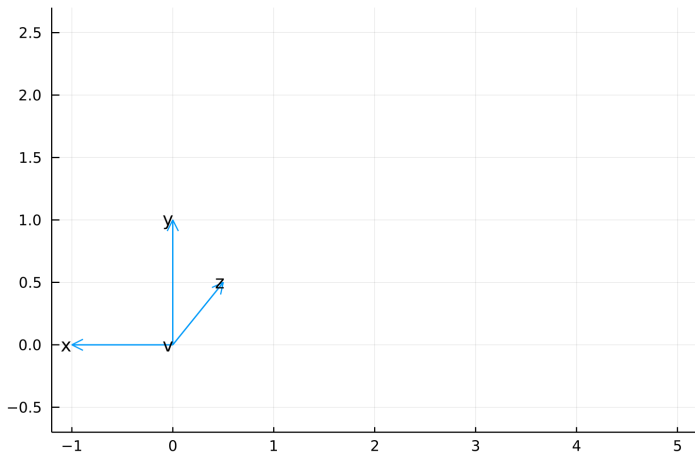
Queremos reescrever o BP usando a álgebra de quatérnios.

Início da seção 4

- 1 Motivação
- 2 Nosso problema
- 3 Objetivo
- 4 BP com Quatérnios**
- 5 Simulações computacionais
- 6 Referências

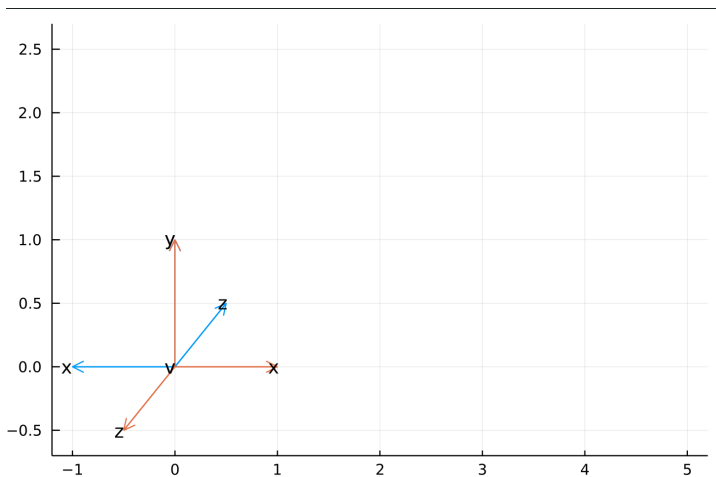
Inicialização [5]

$$x_1 = (0, 0, 0)$$



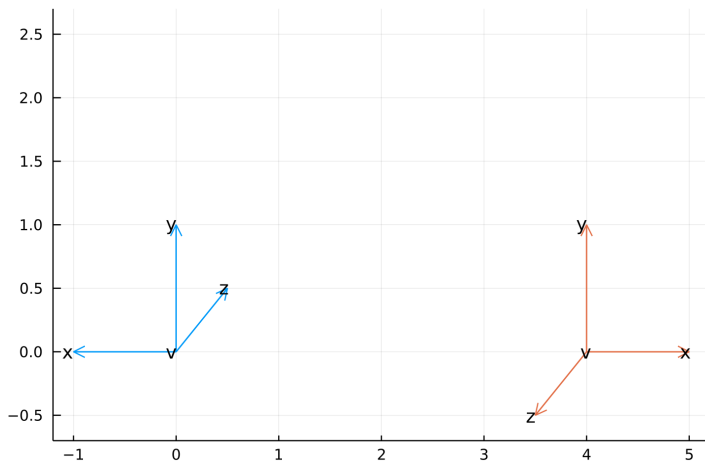
Inicialização

$$x_2 = (\mathbf{j})x_1(\mathbf{j})^*$$



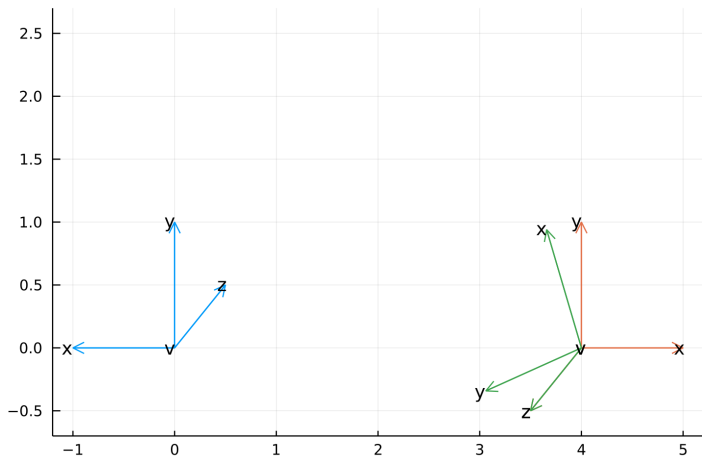
Inicialização

$$x_2 = (\mathbf{j})(x_1 + d_{1,2}\mathbf{i})(\mathbf{j})^*$$



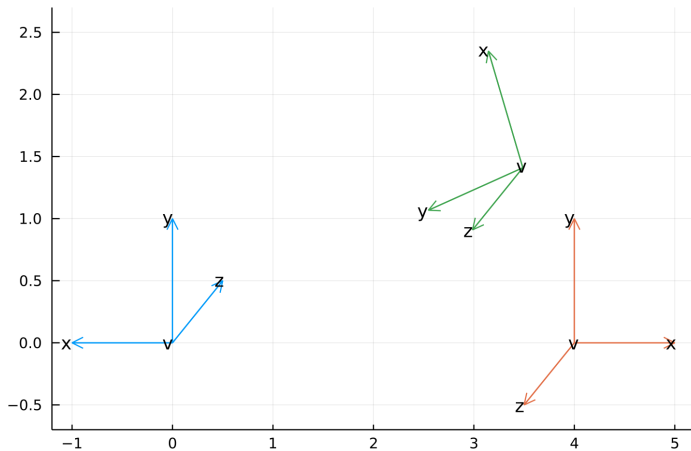
Inicialização

$$x_3 = (\sin(\frac{\theta}{2}) + \cos(\frac{\theta}{2})\mathbf{k})(x_2)(\sin(\frac{\theta}{2}) + \cos(\frac{\theta}{2})\mathbf{k})^*$$



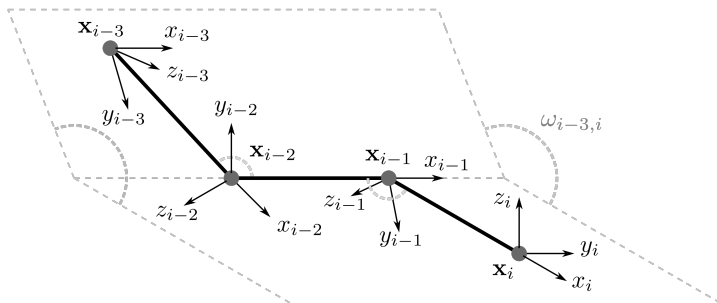
Inicialização

$$x_3 = (\sin(\frac{\theta}{2}) + \cos(\frac{\theta}{2})\mathbf{k})(x_2 + d_{2,3}\mathbf{i})(\sin(\frac{\theta}{2}) + \cos(\frac{\theta}{2})\mathbf{k})^*$$



Branch

Ângulo diedral (ou, de torção)

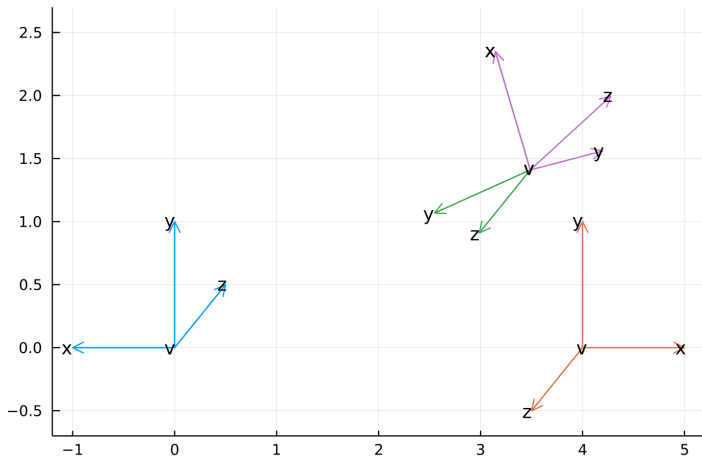


$$\sin(\omega_{i-3,i}) = \sqrt{1 - \cos^2(\omega_{i-3,i})}$$

Branch

Fazendo $q_{i,i} = \cos\left(\frac{\omega_{i-3,i}}{2}\right) + \sin\left(\frac{\omega_{i-3,i}}{2}\right)\mathbf{i}$, temos que

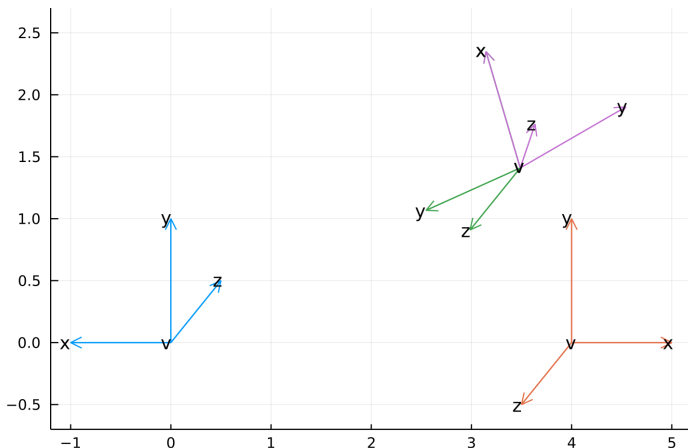
$$x_i^1 = (q_{i,i})(x_{i-1})(q_{i,i})^*$$



Branch

Ou, fazendo $\cos(-\frac{\omega_{i-3,i}}{2}) + \sin(-\frac{\omega_{i-3,i}}{2})\mathbf{i} = q_{i,i}^*$, temos que

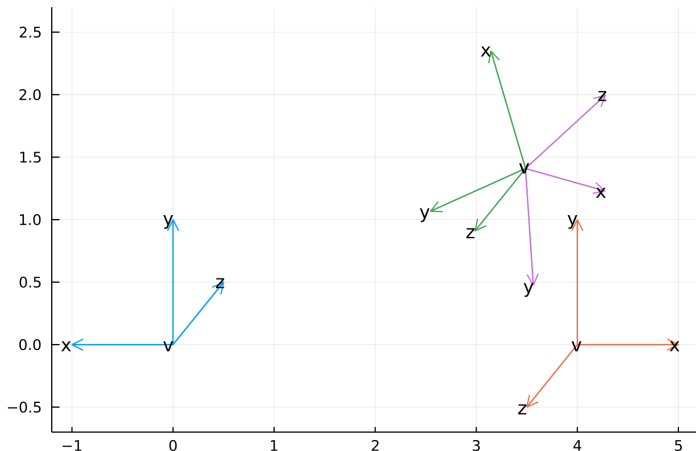
$$x_i^0 = (q_{i,i})^*(x_{i-1})(q_{i,i})$$



Branch

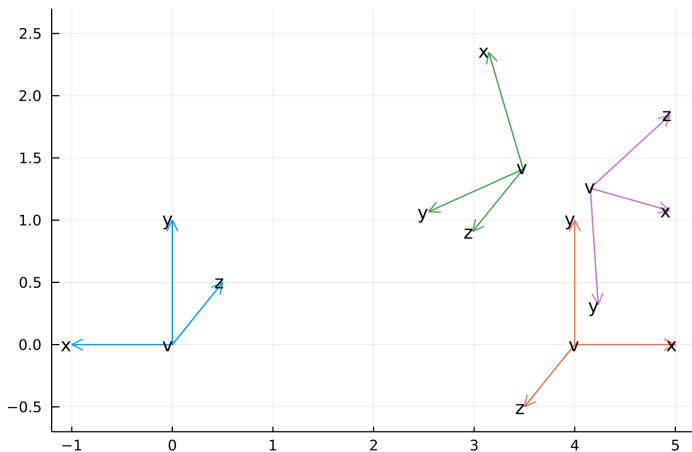
E fazendo $q_{i,k} = \sin(\frac{\theta_{i-2,i}}{2}) + \cos(\frac{\theta_{i-2,i}}{2})\mathbf{k}$, temos que

$$x_i^1 = (q_{i,i})(q_{i,k})(x_{i-1})(q_{i,k})^*(q_{i,i})^*$$



Branch

$$x_i^1 = (q_{i,i})(q_{i,k})(x_{i-1} + d_{i,i-1}\mathbf{i})(q_{i,k})^*(q_{i,i})^*$$



Calculo iterativo

Perceba que, se chamarmos $q_1 = 1$, $q_2 = -\mathbf{j}$, podemos reescrever o quatérnio x_2 como

$$x_2 = \mathbf{j}(x_1 + d_{1,2}\mathbf{i})(-\mathbf{j}) = q_1(q_2(\mathbf{0} + d_{1,2}\mathbf{i})q_2^* + \mathbf{0})q_1^* = q_1q_2(\mathbf{0} + d_{1,2}\mathbf{i})q_1^*q_2^* + q_1\mathbf{0}q_1^*,$$

e se chamarmos $\mathbf{t}_2 = \mathbf{0} + d_{1,2}\mathbf{i}$ e $\mathbf{0} = \mathbf{t}_1$ para identificar as translações envolvendo, respectivamente, a x_1 e a x_2 , temos

$$x_2 = q_1q_2\mathbf{t}_2q_2^*q_1^* + q_1\mathbf{t}_1q_1^* = (q_1q_2)\mathbf{t}_2(q_1q_2)^* + x_1.$$

Se renomearmos os quatérnios de rotação e rearranjarmos os termos,

$$\begin{aligned} \mathbf{x}_1 &= q_1(\mathbf{t}_1)q_1^* \\ \mathbf{x}_2 &= (q_1q_2)\mathbf{t}_2(q_1q_2)^* + \mathbf{x}_1 \\ \mathbf{x}_3 &= (q_1q_2q_3)\mathbf{t}_3(q_1q_2q_3)^* + \mathbf{x}_2 \end{aligned}$$

e, para $i \geq 4$,

$$\begin{aligned} \mathbf{x}_i &= q_1(q_2(q_3(\cdots q_i(\mathbf{t}_i)q_i^* \cdots + \mathbf{t}_3)q_3^* + \mathbf{t}_2)q_2^* + \mathbf{t}_1)q_1^* = \\ &= (q_1q_2 \dots q_i)\mathbf{t}_i(q_1q_2 \dots q_i)^* + \mathbf{x}_{i-1}. \end{aligned}$$

Prune

Assim,

$$\begin{aligned} \mathbf{x}_i^0 &= (q_1 q_2 \dots q_i^*) \mathbf{t}_i (q_1 q_2 \dots q_i^*)^* + \mathbf{x}_{i-1} \\ \mathbf{x}_i^1 &= (q_1 q_2 \dots q_i) \mathbf{t}_i (q_1 q_2 \dots q_i)^* + \mathbf{x}_{i-1} \end{aligned} .$$

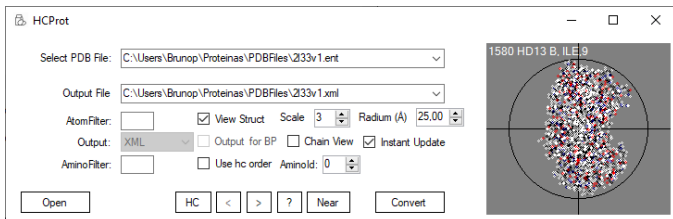
Não há modificações no passo de prune.

Início da seção 5

- 1 Motivação
- 2 Nosso problema
- 3 Objetivo
- 4 BP com Quatérnios
- 5 Simulações computacionais**
- 6 Referências

Ambiente desenvolvido

- HCProt¹: Software para pré-processamento de instâncias PDB, chamado HCProt. Versão com ferramentas visuais para facilitar a criação de ordenações manuais.



- HCProtCLI²: interface linha de comando para a automação do pré-processamento;
- MolecularConformation.jl³: Pacote Julia do algoritmo BP, desenvolvido em colaboração com Emerson Vitor Castelani.

¹<https://github.com/caomem/PDBReader>

²<https://github.com/caomem/HCProtCLI>

³<https://github.com/evcastelani/MolecularConformation.jl>

Resultados computacionais

Problema	LDE	RMSD	Tempo
pdb1ba5	4.257×10^{-21}	1.8249×10^{-10}	1.5480×10^{-1}
pdb1d1n	4.268×10^{-11}	1.1200×10^{-11}	4.3957×10^{-1}
pdb1dp3	2.048×10^{-10}	8.2794×10^{-11}	3.4111×10^{-1}
pdb1du1	2.921×10^{-21}	1.1862×10^{-12}	2.2095×10^{-2}
pdb1fcl	1.650×10^{-10}	2.5527×10^{-12}	5.0979×10^{-1}
pdb1fd6	3.248×10^{-20}	1.4000×10^{-11}	2.9719×10^{-1}
pdb1i2u	4.779×10^{-22}	9.1058×10^{-12}	9.5673×10^{-2}
pdb1i2v	3.179×10^{-22}	9.5660×10^{-13}	1.3251×10^{-1}
pdb1jlz	3.852×10^{-22}	8.6556×10^{-12}	2.9969×10^{-2}
pdb1k0v	8.541×10^{-23}	3.0991×10^{-13}	2.3770×10^{-1}

Tabela: Simulações computacionais sobre o algoritmo QuaternionBP em exemplares reais.

Início da seção 6

- 1 Motivação
- 2 Nosso problema
- 3 Objetivo
- 4 BP com Quatérnios
- 5 Simulações computacionais
- 6 Referências**

- [1] H.M. Berman, K. Henrick, and H. Nakamura.
Announcing the worldwide protein data bank, 2003.
- [2] 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.
- [3] James B Saxe.
Embeddability of weighted graphs in k-space is strongly np-hard.
In Proc. of 17th Allerton Conference in Communications, Control and Computing, Monticello, IL, pages 480–489, 1979.
- [4] 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.
- [5] H Bradford Thompson.
Calculation of cartesian coordinates and their derivatives from internal molecular coordinates.
The Journal of Chemical Physics, 47(9):3407–3410, 1967.

Obrigado!



Contato: g.philippi@grad.ufsc.br
UFSC - Blumenau