Alinhamento Múltiplo de Sequências Biológicas

Pedro Jorge Oliveira Câmara - DRE: 120182069

Novembro 2024

Resumo

Alinhamento de sequências biológicas é uma forma de comparar e relacionar biomoléculas, possibilitando inferir características de suas estruturas, funções e relações evolutivas. No entanto, o processo de alinhamento acaba resultando em um problema de natureza combinatória: gostaríamos de testar todas as combinações possíveis e selecionar a que melhor nos convém. Como uma tentativa de resolução, são utilizadas diversas heurísticas para buscar a solução ótima ou, pelo menos, uma satisfatória. Com a finalidade de mensurar a qualidade dos algoritmos propostos pelos pesquisadores, existe o Benchmark Alignment dataBASE, que reúne dados de sequências biológicas reais já alinhadas, o que possibilita testes e verificações das soluções encontradas. Neste trabalho, é utilizado o Algoritmo Genético como uma tentativa de, dada uma sequência biológica, alinhá-las da melhor maneira possível; os resultados obtidos são satisfatórios para entradas com duas sequências, mas não apresentam bom desempenho para três ou mais.

1 Introdução

Biologia é o ramo da ciência que estuda a vida. Em particular, a biologia molecular é a área que busca explicar os fenômenos e eventos biológicos sob o ponto de vista das biomoléculas, atribuindo especial atenção às macromoléculas ácido desoxirribonucleico (DNA) e ácido ribonucleico (RNA), formados por nucleotídeos, e proteínas, formadas por aminoácidos.

Com os avanços computacionais do século XX e XXI, tornou-se possível a coleta e o registro de dados de significado biológico, provenientes das amostragens e medições empíricas, nos meios digitais, para posterior análise e intepretação, área do conhecimento conhecida atualmente por bioinformática ou biologia computacional.

1.1 Biologia Molecular

No campo da biologia molecular, é de interesse a análise de amostras de biomoléculas para constatar possíveis semelhanças funcionais, estruturais ou evolutivas entre elas. Essas análises se concentram em dois principais grupos: os ácidos nucleicos (DNA e RNA) e as proteínas.

O DNA é composto pelos nucleotídeos adenina, citosina, guanina e timina, enquanto que adenina, citosina, guanina e uracila compõe o RNA, conforme mostrado na Tabela 1.

Nucleotídeo	Sigla
Adenina	A
Citosina	C
Guanina	G
Timina	T
Uracila	U

Tabela 1: Nucleotídeos, blocos construtores do DNA (A, C, G, T) e do RNA (A, C, G, U)

As proteínas, por sua vez, são compostas por 20 aminoácidos, listados na Tabela 2.

Aminoácido	Sigla	Aminoácido	Sigla
Alanina	A	Leucina	L
Arginina	R	Lisina	K
Asparagina	N	Metionina	M
Ácido Aspártico	D	Fenilalanina	F
Cisteína	C	Prolina	P
Glutamina	Q	Serina	S
Ácido Glutâmico	E	Treonina	T
Glicina	G	Triptofano	W
Histidina	H	Tirosina	Y
Isoleucina	I	Valina	V

Tabela 2: Aminoácidos, blocos construtores das proteínas

A partir de uma combinação de seus respectivos blocos de construção, são manifestados o DNA, RNA e as proteínas que compõem um ser vivo, conforme ilustrado nas Figuras 1 e 2.

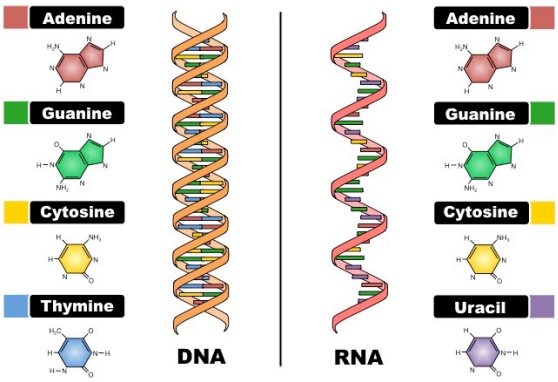

Figura 1: Representações das moléculas DNA e RNA

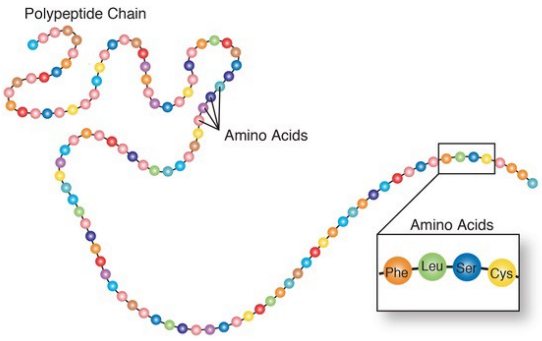

Figura 2: Representação da molécula de uma proteína

1.2 Sequências biológicas

A partir de uma amostragem e medição de uma biomolécula, chamamos de sequenciamento o processo de construir uma uma cadeia de caracteres que representa a sequência de nucleotídeos ou de aminoácidos que formam a sua estrutura.

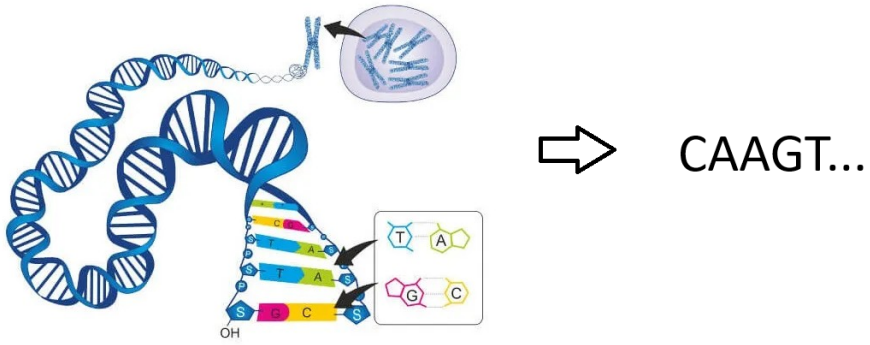

Figura 3: Exemplo de sequenciamento de ácido nucleico: a partir de uma amostra, são identificados os nucleotídeos que a formam e é construída uma sequência de caracteres que a representa. No caso do DNA, apenas uma das hélices é usada; nesta ilustração, consideramos a hélice da direita, composta por citosina, adenina, adenina, guanina e timina, formando a sequência CAAGT.

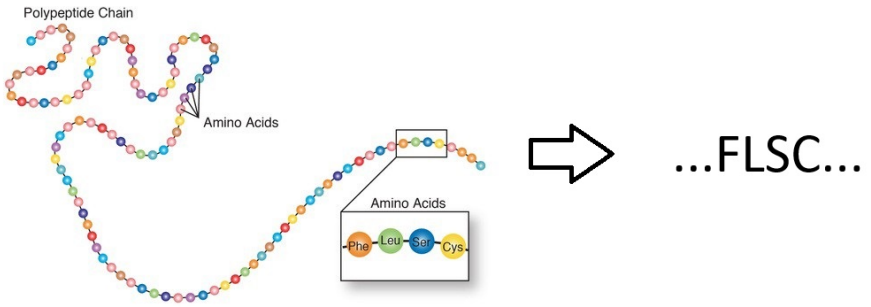

Figura 4: Exemplo de sequenciamento de proteína: a partir de uma amostra, são identificados os aminoácidos que a formam e construída uma sequência de caracteres que a representa. Nesta ilustração, consideramos os quatro aminoácidos destacados na imagem: fenilalanina, leucina, serina e cisteína, formando a sequência FLSC.

1.3 Alinhamento de sequências

O alinhamento de sequências é o ato de, dada duas ou mais sequências, associar cada elemento de uma a um único elemento da outra. Uma característica dos ácidos nucleicos e das proteínas é que, em geral, sequências semelhantes apresentam características semelhantes nos seres vivos. Dessa forma, é de extremo interesse que seja possível realizar um alinhamento entre elas, de maneira a buscar suas similaridades. Em particular, se as características de uma das sequências já é conhecida, ao alinhá-la com uma amostragem desconhecida, podemos inferir informações estruturais, funcionais e evolutivas sobre ela.
Por exemplo, suponha duas sequências: ABC e $BCDE$; repare que elas possuem tamanhos diferentes. A ideia é alinhá-las uma sobre a outra, de forma a montar uma espécie de "matriz de caracteres", em que as colunas possuam o máximo de caracteres iguais possíveis ("matches"). Para o alinhamento, ambas devem possuir o mesmo tamanho, e isso é obtido inserindo espaçamentos ("gaps", representados por '-') nas sequências:

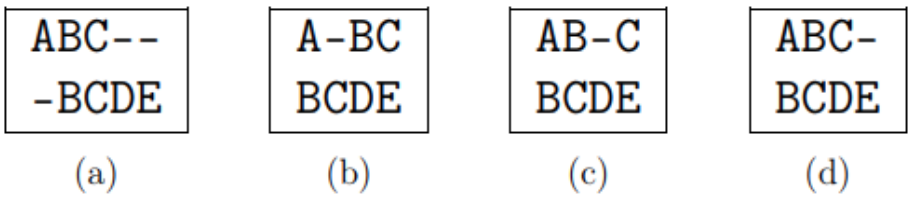

Figura 5: Exemplo de alinhamento de sequências. Visualmente, podemos notar a similaridade nos caracteres BC, presentes em ambas. Inserindo gaps, queremos encontrar um alinhamento em que eles estejam na mesma coluna. O alinhamento ótimo é encontrado no caso (a).

Em geral, estamos interessados na análise de múltiplas sequências, buscando inferir as características comuns a todas elas. Devido às suas origens experimentais de medição, normalmente as sequências não possuem o mesmo tamanho e não é possível simplesmente emparelhá-las.

1.4 Gaps, matches e mismatches

O processo de alinhamento consiste, portanto, na inserção dos espaços, chamados de gaps, de maneira que as colunas tenham o maior número de matches – caracteres iguais – e o menor número de mismatches – caracteres diferentes – possível.

Progressivamente inserimos os gaps, de maneira que, ao final, tenhamos todas as sequências com o mesmo tamanho:

A	B	C	-	-
D	B	C	D	E

Tabela 3: Exemplo de alinhamento. Na coluna 1 temos um mismatch, nas colunas 2 e 3 temos um match e nas colunas 3 e 4 temos um gap

Dessa maneira, queremos elaborar uma forma de mensurar a qualidade de um alinhamento, para que possamos encontrar um que seja satisfatório.

2 Modelagem matemática e computacional

A partir do entendimento das bases conceituais, podemos definir o problema. Para este trabalho, consideraremos apenas sequências de proteínas, formadas pelos 20 aminoácidos. Sendo assim, um alinhamento de múltiplas sequências biológicas é definido como:

> Dado o conjunto de sequências $\mathcal{S} = \{S_1, S_2, \ldots, S_n\}$ e uma função $f(\mathcal{A})$ que mensure a similaridade de um alinhamento $\mathcal{A}(\mathcal{S}) = \{S'_1, S'_2, \ldots, S'_n\}$, encontre o alinhamento ótimo $\mathcal{A}^*$ que maximize f.

2.1 Função objetivo

Queremos elaborar uma função que meça o quão similares as sequências de um conjunto $\mathcal{A}$ são entre si.
Seja $\Lambda = \{A, R, N, D, C, E, Q, G, H, I, L, K, M, F, P, S, T, W, Y, V\}$ o conjunto que representa os aminoácidos. Os caracteres s de uma sequência $S' \in \mathcal{A}$ assumem valores $s \in \Lambda \cup \{-\}$.
Dados dois caracteres $s_{i,k} \in S'_i$ e $s_{j,k} \in S'_j$, definidos uma mensuração M que mede a similaridade (ou a "aceitação"para os gaps) do emparelhamento entre eles da seguinte forma:

$$M(s_{i,k}, s_{j,k}) = \begin{cases} 1 & \text{se } s_{i,k} = - \text{ e } s_{j,k} = - \\ 2 & \text{se } s_{i,k} \in \Lambda \text{ e } s_{j,k} = - \\ 3 & \text{se } s_{i,k} = - \text{ e } s_{j,k} \in \Lambda \\ PAM250[s_{i,k}, s_{j,k}] & \text{se } p \in \Lambda \text{ e } s_{j,k} \in \Lambda \end{cases}$$

Onde $PAM250$ (Point Accepted Mutation) é uma matriz simétrica que quantifica a relação de similaridade entre dois aminoácidos na estrutura de alinhamento de proteínas:

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	2																			
R	-2	6																		
N	0	0	2																	
D	0	-1	2	4																
C	-2	-4	-4	-5	4															
Q	0	1	1	2	-5	4														
E	0	-1	1	3	-5	2	4													
G	1	-3	0	1	-3	-1	0	5												
H	-1	2	2	1	-3	3	1	-2	6											
I	-1	-2	-2	-2	-2	-2	-2	-3	-2	5										
L	-2	-3	-3	-4	-6	-2	-3	-4	-2	2	6									
K	-1	3	1	0	-5	1	0	-2	0	-2	-3	5								
M	-1	0	-2	-3	-5	-1	-2	-3	-2	2	4	0	6							
F	-4	-4	-4	-6	-4	-5	-5	-5	-2	1	2	-5	0	9						
P	1	0	-1	-1	-3	0	-1	-1	0	-2	-3	-1	-2	-5	6					
S	1	0	1	0	0	-1	0	1	-1	-1	-3	0	-2	-3	1	3				
T	1	-1	0	0	-2	-1	0	0	-1	0	-2	0	-1	-2	0	1	3			
W	-6	2	-4	-7	-8	-5	-7	-7	-3	-5	-2	-3	-4	0	-6	-2	-5	17		
Y	-3	-4	-2	-4	0	-4	-4	-5	0	-1	-1	-4	-2	7	-5	-3	-3	0	10	
V	0	-2	-2	-2	-2	-2	-2	-1	-2	4	2	-2	2	-1	-1	-1	0	-6	-2	4

Figura 6: Matriz Point Accepted Mutation. Para valores negativos, a intepretação é que esse alinhamento não é favorável. Repare que a diagonal quantifica o quanto um determinado aminoácido contribui para a similaridade geral das sequências.

Dado um alinhamento $\mathcal{A}(\mathcal{S}) = \{S'_1, S'_2, ..., S'_n\}$, a função objetivo é construída pela soma das similaridades no alinhamento par a par:

$$f(\mathcal{A}) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \sum_{k=i}^{m} M(S'_{i,k}, S'_{j,k})$$

Onde n é o número de sequências sendo alinhadas e m é o tamanho das sequências $S'_1, S'_2, ..., S'_n$.

Queremos, portanto, encontrar um alinhamento $\mathcal{A}^*$ que maximize essa função, ou seja, que tenha o maior somatório possível de similaridades. Repare que, para a convenção de minimização, basta minimizarmos $-f$.

2.2 Metodologia

Evidentemente, a complexidade do problema é de natureza combinatória, NP-Hard: a força bruta testaria todas as combinações possíveis de sequências e selecionaria a de maior similaridade. Dentre as várias propostas de resolução desse problema, existe a aplicação do Algoritmo Genético (GA - Genetic Algorithm) para buscar um bom alinhamento.

A modelagem do GA neste trabalho se baseia no artigo An enhanced algorithm for multiple sequence alignment of protein sequences using genetic algorithm, publicado na revista EXCLI Journal, e na tese de mestrado de um aluno da UNICAMP Alinhamento múltiplo de proteínas utilizando algoritmos genéticos.

O domínio do problema é discreto: o "espaço de busca"é dado pela entrada, um conjunto $\mathcal{S} = \{S_1, S_2, ..., S_n\}$ de sequências. Cada indivíduo da população é um alinhamento $\mathcal{A} =$

$\{S'_1, S'_2, ..., S'_n\}$ e o objetivo é que a população venha a convergir para um alinhamento $\mathcal{A}^*$ que minimize $-f$.

3 Algoritmo Genético

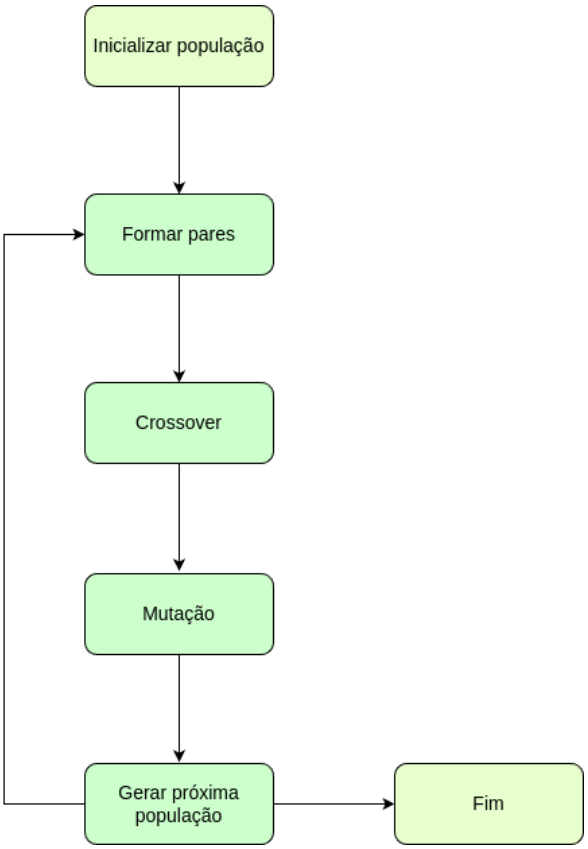

Figura 7: Fluxograma do Algoritmo Genético. Implementado em Python.

3.1 Inicialização da população

A população é inicializada aleatoriamente: a partir do tamanho da maior sequência, são criados 100 indivíduos $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_{100}$, cada um com gaps preenchidos em posições arbitrárias.

3.2 Formação de pares

São formados 50 pares de indivíduos que irão se reproduzir. A cada iteração k, atratividade de cada indivíduo $\mathcal{A}_i^{(k)}$ é calculada de forma inversamente proporcional à imagem:

$$\tau_i^{(k)} = \frac{1}{f(\mathcal{A}_i^{(k)}) - f_{min} + 1}$$

Cada indivíduo tem recebe um valor de probabilidade de ser escolhido:

$$P_i^{(k)} = \frac{\tau_i^{(k)}}{\sum_{i=1}^{100} \tau_i^{(k)}}$$

Então, geramos um valor aleatório $\alpha \in (0, 1)$ e atribuimos $p = P_1^k, i = 1$. Enquanto $\alpha > p$, fazemos $p \leftarrow p + P_i^k + 1$ e $i \leftarrow i + 1$, selecionando o indivíduo i para reproduzir.

3.3 Crossover

Uma vez selecionado um par, a operação de crossover consiste em cortar uma posição aleatória das sequências e misturar suas partes:

Posição	1	2	3	4	5
Sequência 1	K	L	M	D	E
Sequência 2	K	-	L	M	-

Posição	1	2	3	4	5
Sequência 1	A	B	C	W	Y
Sequência 2	A	-	C	W	V

Tabela 4: Exemplo de duas sequências que formaram um par

Suponha que o corte será feito verticalmente entre a posição 3 e 4:

Posição	1	2	3	4	5
Sequência 1	K	L	M	W	Y
Sequência 2	K	-	L	W	V

Posição	1	2	3	4	5
Sequência 1	A	B	C	D	E
Sequência 2	A	-	C	M	-

Tabela 5: Exemplo de crossover e a geração de dois filhos. As colunas modificadas foram as 4 e 5

Esse crossover também pode ocorrer de forma horizontal de maneira análoga.

3.4 Mutação

Com probabilidade de 1%, um indivíduo da população sofre uma mutação. Ela consiste em um "swap"entre o primeiro gap e o aminoácido mais próximo à direita na sequência:

A	-	B	C

Tabela 6: Indivíduo que sofrerá uma mutação

A	B	-	C

Tabela 7: Indivíduo após sofrer uma mutação

3.5 Nova geração

Para a população da próxima geração, todos os indivíduos, pais e filhos, são ordenados de acordo com seu valor de fitness, do melhor para o pior, e são selecionados os 100 primeiros.

3.6 Critério de parada

O critério de parada adotado foi o de número máximo de iterações: 50 gerações.

4 Resultados

Uma vez implementado o algoritmo, sua execução consiste em dar como entrada um conjunto de sequências $\mathcal{S} = \{S_1, S_2, ..., S_n\}$ e obter o melhor alinhamento $\mathcal{A}^*$ encontrado. Para conjuntos simples com duas sequências, os resultados atingem o ótimo; para conjuntos mais complexos, o algoritmo tem dificuldade para encontrar um alinhamento satisfatório.

4.1 BAliBASE

Benchmark Alignment dataBASE (BAliBASE) é um dataset de alinhamento múltiplo de sequências que contém as soluções ótimas.
Cada arquivo possui um conjunto de sequências alinhadas. Para fins de teste de algum algoritmo, basta remover os gaps do alinhamento:

Nome	**Sequência**
hmgl_trybr	`gpeeRKVYEEMAEKDKERYKREM--------`
hmgt_mouse	`speeKQAYIQLAKDDRIRYDNEMksweeqmae`
hmgb_chite	`--kdKSEWEAKAATAKQNYIRALqeyerngg-`
hmgl_wheat	`seseKAPYVAKANKLKGEYNKAIaaynkgesa`

Tabela 8: Exemplo de um alinhamento registrado no BAliBASE. As letras maiúsculas representam a subsequência que gera a similaridade.

4.2 Experimentos

O primeiro experimento foi feito a partir de uma sequência simples e hipotética:

Sequência S_1	G	C	P	F	S	-	S	P	N	V	E	A
Sequência S_2	G	C	P	Y	G	C	S	P	E	A	D	

Tabela 9: Sequência hipotética de aminoácidos

Para encontrar o alinhamento ótimo $\mathcal{A}^*$, basta acrescentar um gap no final de S_2. Se calcularmos a função f, teremos uma fitness de $f(\mathcal{A}^*) = -48$.
Em geral, o GA apresenta bom comportamento para esse exemplo, convergindo na maioria das vezes para o ótimo. Esporadicamente, a população converge para um mínimo local, com valor de $f(\mathcal{A}^*) = -41$, resultando nos seguintes alinhamentos.

Sequência S_1	G	C	P	F	S	-	S	P	N	V	E	A
Sequência S_2	G	C	P	Y	G	C	S	P	E	A	D	-

Tabela 10: Alinhamento ótimo, $f(\mathcal{A}^*) = -48$

Sequência S_1	G	C	P	F	S	-	S	P	N	V	E	A
Sequência S_2	G	C	P	Y	G	C	S	P	-	E	A	D

Tabela 11: Alinhamento ótimo local, $f(\mathcal{A}) = -41$

4.3 Experimentos em dados reais do BAliBASE

Os experimentos foram feitos utilizando os dados do BAliBASE 2.0 na referência 1, no teste 1, nos dados 1aab, que armazenam dados de uma família de proteínas chamadas High Mobility Group, do mesmo exemplo da Tabela 8.

Em um primeiro momento, consideramos apenas duas sequências, hmgl_trybr e hmgt_mouse:

Nome	**Sequência**
hmgl_trybr	`gpeeRKVYEEMAEKDKERYKREM--------`
hmgt_mouse	`speeKQAYIQLAKDDRIRYDNEMksweeqmae`

Tabela 12: Alinhamento ótimo $\mathcal{A}^*$ das sequências, conforme registrado no BAliBASE. Em nossa métrica, $f(\mathcal{A}^*) = -93$

Removemos os gaps e usamos essa estrutura como entrada para o algoritmo:

Nome	**Sequência**
hmgl_trybr	`gpeeRKVYEEMAEKDKERYKREM`
hmgt_mouse	`speeKQAYIQLAKDDRIRYDNEMksweeqmae`

Tabela 13: Conjunto $\mathcal{S}$ de entrada para o algoritmo genético

Os alinhamentos encontrados são mais variados, convergindo para alguns mínimos locais. Em geral, é notável que o algoritmo percebe que deveria encaixar todos os gaps no final da sequência hmgl_trybr, visto que ele converge para um alinhamento com valor $f(\mathcal{A}) = -76$ com frequência:

Nome	**Sequência**
hmgl_trybr	`gpeeRKVYEEMAEKDKERYKRE--------M`
hmgt_mouse	`speeKQAYIQLAKDDRIRYDNEMksweeqmae`

Tabela 14: Alinhamento ótimo local $\mathcal{A}$ das sequências encontrado pelo GA, com $f(\mathcal{A}) = -76$

No entanto, em diversas rodadas também são obtidos resultados discrepantes do esperado, com $f = -58$ e um alinhamento insatisfatório:

Nome	Sequência
hmgl_trybr	SPEEKQAYIQLAKDDRIRYDNEMKSWEEQMAE
hmgt_mouse	GPEERKVYEEMAEK--DK--E-RYK-R-EM--

Tabela 15: Alinhamento ótimo local $\mathcal{A}$ insatisfatório das sequências encontrado pelo GA, com $f(\mathcal{A}) = -58$

Se aumentarmos a população para 500 indivíduos, em geral há convergência para o ótimo global.

Apesar do algoritmo conseguir, em certas condições, convergir para o ótimo global ou para um ótimo local satisfatório quando são apenas duas sequências, esse cenário se torna mais desafiador quando incrementamos o tamanho da entrada. Inserindo mais uma sequência do dataset e removendo os gaps, obtemos:

Nome	Sequência
hmgl_trybr	gpeeRKVYEEMAEKDKERYKREM
hmgt_mouse	speeKQAYIQLAKDDRIRYDNEMksweeqmae
hmgb_chite	kdKSEWEAKAATAKQNYIRALqeyerngg

Tabela 16: Sequências biológicas do BAliBASE

Onde o alinhamento ótimo é dado por:

Nome	Sequência
hmgl_trybr	gpeeRKVYEEMAEKDKERYKREM--------
hmgt_mouse	speeKQAYIQLAKDDRIRYDNEMksweeqmae
hmgb_chite	--kdKSEWEAKAATAKQNYIRALqeyerngg-

Tabela 17: Alinhamento ótimo das três sequências, $f(A^*) = -167$

Para esse caso, o GA não consegue, em nenhum momento, encontrar o alinhamento ótimo. É necessário aumentar o tamanho da população, pois com 100 indivíduos, os valores de f não se aproximam sequer de -100. Aumentando a população para 1000, encontramos como solução mais baixa o valor de $f = -125$.

5 Conclusão

Para um problema de natureza complexa e levando em consideração a implementação relativamente "ingênua"do Algoritmo Genético, os resultados preliminares são interessantes. Para duas sequências, a implementação não parece divergir com frequência e demonstra bom comportamento. Apesar disso, são necessárias maiores sofisticações para para três sequências e, principalmente, para alinhamento múltiplo com os dados reais da BAliBASE, que apresentam não só um grande número de sequências como também um comprimento maior da cadeia de caracteres.

5.1 Trabalhos futuros

A mesclagem de computação e biologia me interessou bastante. Ao longo do desenvolvimento do trabalho, tive algumas ideias que me deixaram empolgado para pôr em prática:

5.1.1 Reformular a modelagem

Algo que me impressionou é a liberdade na construção da função f, o que certamente pode impactar no desempenho dos algoritmos e nas soluções obtidas. Experimentar outras formulações da função de fitness e outras matrizes de relação de aminoácidos (pois existem outras) é do meu interesse.

5.1.2 Exploração da base de dados BAliBASE e do problema de alinhamento para DNA e RNA

A BAliBASE possui dados robustos e diversos algoritmos já foram propostos para alinhamento. Nos artigos que pesquisei, sempre são mostradas tabelas de comparações das soluções propostas até o momento e as vantagens e desvantagens entre elas. Uma característica que me chamou a atenção é o tempo grande de processamento mesmo para sequências "pequenas".

5.1.3 Migração para o NumPy e C++

Em termos de programação, devido à natureza de strings e caracteres, em um primeiro momento não foi possível utilizar o NumPy, visto que ele não trabalha bem com dados não numéricos. Apesar disso, durante o desenvolvimento do trabalho, feito em Python, notei que poderia construir um dicionário que mapeie cada aminoácido a um número entre 0 e 19, o gap '-' para -1 e utilizar o NumPy, em particular após perceber que não são feitas operações aritméticas e, portanto, apenas a representação seria diferente. Da mesma maneira, seria possível construir um "dicionário de volta", que mapeie os números inteiros de -1 a 19 para os aminoácidos ou para o gap, de maneira a recuperar as strings ao término do algoritmo.
No contexto deste trabalho, as experimentações não foram grandes o suficiente para causar problemas de performance mas, ao utilizar de fato dados mais robustos do BAliBASE, certamente o uso do NumPy seria mais eficiente.
Em um dos artigos que me baseei, o autor construiu seu GA em C, o que me fez questionar se a diferença de performance seria muito discrepante se fosse feita uma comparação entre as implementações em Python e C++, então eu construiria o algoritmo nessa linguagem.

5.1.4 Aplicação do Ant Colony e do Bee Colony

Durante minhas pesquisas, encontrei artigos que utilizam o Ant Colony Optimization para esse problema de alinhamento. Por ter encontrado materiais mais detalhados sobre o Algoritmo Genético, decidi usá-lo, mas gostaria de implementar a Colônia de Formigas e compará-la ao GA neste cenário. Não pesquisei, tampouco "esbarrei"em materiais que citassem a aplicação do Bee Colony, o que me fez querer saber sobre o seu desempenho

nos alinhamentos. Essas são as principais heurísticas que me chamaram a atenção tanto na disciplina quanto para o problema mas, a depender do contexto, eu estaria disposto a experimentar outras abordagens, tanto por outras heurísticas quanto por soluções exatas (como, por exemplo, programação dinâmica, uma das técnicas usadas nessa área).

5.1.5 Paralelização

Notei que algumas partes do problema são paralelizáveis, em particular o cálculo da função f. Como ela é construída par a par, seria possível paralelizar essas somas e aumentar a performance para problemas com um número grande de sequências e de tamanho da cadeia de caracteres.

5.2 Publicação

Por fim, respondendo à pergunta na observação da proposta do trabalho: sim, eu estou motivado e teria tempo para aprimorar esse trabalho, publicá-lo em um evento acadêmico e, certamente, fazer um TCC e um mestrado na área!

6 Referências

Manish Kumar - An enhanced algorithm for multiple sequence alignment of protein sequences using genetic algorithm
Sergio Jeferson Rafael Ordine - Alinhamento múltiplo de proteínas utilizando algoritmos genéticos
Opportunities for Combinatorial Optimization in Computational Biology
A review on multiple sequence alignment from the perspective of genetic algorithm
The PAM250 Scoring Matrix
BAliBASE 2.0