

Variações do Problema de Distância de Rearranjos

Defesa de Doutorado

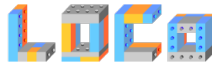
Candidato: **Alexsandro Oliveira Alexandrino**

Orientador: **Prof. Dr. Zanoni Dias**

Coorientador: **Prof. Dr. Ulisses Martins Dias**

27 de Março de 2024

Laboratório de Otimização Combinatória
Instituto de Computação
Universidade Estadual de Campinas



Agenda

1. Introdução
2. Conceitos e Definições
3. Nova 1.375-Aproximação para a Ordenação de Permutações por Transposições
4. Ordenação de Permutação por Transposições e Outros Rearranjos
5. Distância em Genomas Desbalanceados
6. Distância em Genomas Desbalanceados com Regiões Intergênicas
7. Conclusão

Introdução

- Distância evolucionária.
- Distância de rearranjos de genomas.
- Rearranjos de genomas: mutações de larga escala que afetam grandes trechos da sequência de DNA.
 - Reversões;
 - Transposições.

Distância de Rearranjos de Genomas

- A ocorrência de um rearranjo β está vinculada a um custo $w(\beta)$.
- O custo de uma sequência de rearranjos é o somatório dos custos dos rearranjos da sequência.
- Para dois genomas \mathcal{G}_o e \mathcal{G}_d , a distância de rearranjos é o menor custo possível para uma sequência de rearranjos transformar \mathcal{G}_o em \mathcal{G}_d .

Variações da Distância de Rearranjos

- Os problemas de distância de rearranjos variam de acordo com:
 - Representação do genoma:
 - Genes repetidos ou não repetidos;
 - O conjunto de genes de cada genoma é igual ou distinto;
 - Uso de regiões intergênicas ou apenas ordem dos genes.
 - Conjunto de rearranjos permitidos (modelo de rearranjo).
 - Função de custo:
 - Constante;
 - Por tipo de rearranjo;
 - Proporcional ao tamanho da região afetada.

Conceitos e Definições

Representação: Ordem Relativa dos Genes

- A maioria dos estudos usa apenas a ordem relativa dos genes de cada genoma, supondo que:
 - Não existem genes duplicados;
 - Os dois genomas comparados compartilham os mesmos genes.
- Assim, um genoma \mathcal{G} com sequência de genes $(g_1 g_2 \dots g_n)$ é representado por uma permutação $\pi = (\pi_1 \pi_2 \dots \pi_n)$, onde um número inteiro π_i corresponde a um gene g_i no genoma \mathcal{G} .
- Se a orientação dos genes é conhecida, cada elemento possui um sinal $+$ ou $-$ para indicar a orientação do gene correspondente.

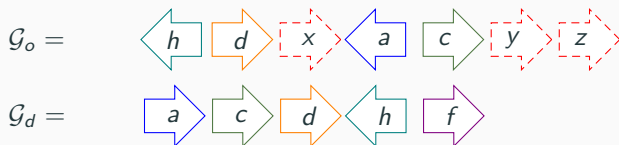
Representação: Ordem Relativa dos Genes

- Ao utilizar permutações, um dos genomas é representado como a permutação identidade $\iota^n = (1\ 2\ \dots\ n)$.
- O problema se torna equivalente ao problema da Ordenação de Permutações por Rearranjos.

Representação: Ordem Relativa dos Genes

- Quando uma das seguintes condições é atendida, a representação da ordem relativa dos genes é feita com strings:
 - Genes repetidos;
 - O conjunto de genes de cada genoma é distinto.
- Assim, um genoma $\mathcal{G} = (g_1 \ g_2 \ \dots \ g_n)$ é representado por uma string $A = (A_1 \ A_2 \ \dots \ A_n)$.
- Se a orientação dos genes é conhecida, cada elemento possui um sinal $+$ ou $-$ para indicar a orientação do gene correspondente.

Representação: Ordem Relativa dos Genes

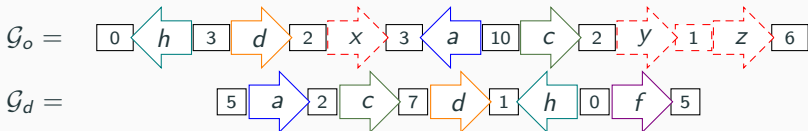


- Segmentos a serem removidos são mapeados com o valor α , já que são removidos independente do seu conteúdo.
- \mathcal{G}_d é mapeado em ι^n :
 - $\iota^n = (+1 +2 +3 +4 +5)$.
- \mathcal{G}_o é mapeado em A :
 - $A = (+4 +3 \alpha -1 +2 \alpha)$.
- Para facilitar a definição de conceitos como breakpoints ou grafo de ciclos, consideramos que uma string ou permutação σ possui os elementos $\sigma_0 = 0$ e $\sigma_{n+1} = n + 1$, que não são afetados por rearranjos por serem elementos artificiais.

Representação: Regiões Intergênicas

- Regiões intergênicas:
 - Nucleotídeos entre genes e nas extremidades do genoma.
 - Mais suscetíveis a mudanças do que genes.
 - Normalmente não possuem correspondência entre genomas e, portanto, apenas a quantidade de nucleotídeos (tamanho) de cada região é usada.
- Um genoma $\mathcal{G} = (\check{g}_1 g_1 \check{g}_2 g_2 \dots \check{g}_n g_n \check{g}_{n+1})$ é representado por uma string A e uma lista de regiões intergênicas \check{A} .
 - Cada gene g_i é representado pelo caractere A_i .
 - O tamanho da região intergênica \check{g}_i é representado pelo número \check{A}_i .

Representação: Regiões Intergênicas



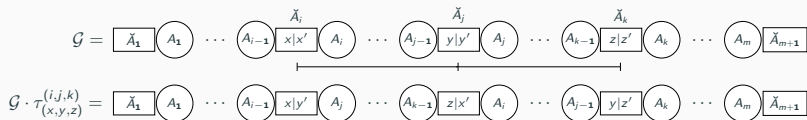
- \mathcal{G}_d é mapeado em $(\iota^n, \check{\iota})$:
 - $\iota^n = (+1 +2 +3 +4 +5)$;
 - $\check{\iota} = (5, 2, 7, 1, 0, 5)$.
- \mathcal{G}_o é mapeado em (A, \check{A}) :
 - $A = (+4 +3 \alpha -1 +2 \alpha)$;
 - $\check{A} = (0, 3, 2, 3, 10, 2, 6)$.

- Os rearranjos considerados nesta tese de doutorado são:
 - Reversão ρ ;
 - Transposição τ ;
 - Inserção ϕ e Deleção ψ (Indels);
 - Transposição Inversa $\rho\tau$;
 - Revreves $\rho\rho$;
 - Block Interchange bi .
- Exceto pelos indels, todos os rearranjos mencionados são conservativos (*i.e.*, não alteram a quantidade de material genético).
- A aplicação de um rearranjo β é denotada por $A \cdot \beta$ e $\check{A} \cdot \beta$.

- Uma reversão $\rho_{(x,y)}^{(i,j)}$ divide as regiões intergênicas \check{A}_i e \check{A}_{j+1} em $x|x'$ e $y|y'$, respectivamente, e inverte o segmento $(x' A_i \check{A}_{i+1} \dots A_j y)$.
- Em strings com sinais, os sinais dos elementos no segmento invertido são alterados.

$$\begin{aligned}
 \mathcal{G} &= \boxed{\check{A}_1} \circledast A_1 \cdots \circledast A_{i-1} \boxed{x|x'} \circledast A_i \cdots \circledast A_j \boxed{y|y'} \circledast A_{j+1} \cdots \circledast A_m \boxed{\check{A}_{m+1}} \\
 &\quad \underbrace{\hspace{10em}} \\
 \mathcal{G} \cdot \rho_{(x,y)}^{(i,j)} &= \boxed{\check{A}_1} \circledast A_1 \cdots \circledast A_{i-1} \boxed{x|y} \circledast (-A_j) \cdots \circledast (-A_i) \boxed{x'|y'} \circledast A_{j+1} \cdots \circledast A_m \boxed{\check{A}_{m+1}}
 \end{aligned}$$

- Uma transposição $\tau_{(x,y,z)}^{(i,j,k)}$ divide as regiões intergênicas \check{A}_i , \check{A}_j e \check{A}_k em $x|x'$, $y|y'$ e $z|z'$, respectivamente, e troca a posição dos segmentos $(x' A_i \check{A}_{i+1} \dots A_{j-1} y)$ e $(y' A_j \check{A}_{j+1} \dots A_{k-1} z)$.



- Uma deleção $\psi_{(x,y)}^{(i,j)}$ divide as regiões intergênicas \check{A}_i e \check{A}_j em $x|x'$ e $y|y'$, respectivamente, e remove o segmento $(x' A_i \dots A_{j-1} y)$.

$$\begin{aligned}
 \mathcal{G} = & \boxed{\check{A}_1} \circledast \dots \circledast \overset{\check{A}_i}{\boxed{x|x'}} \circledast \dots \circledast \overset{\check{A}_j}{\boxed{y|y'}} \circledast \dots \circledast \boxed{\check{A}_{m+1}} \\
 & \underbrace{\hspace{10em}} \\
 \mathcal{G} \cdot \psi_{(x,y)}^{(i,j)} = & \boxed{\check{A}_1} \circledast \dots \circledast \boxed{x|y'} \circledast \dots \circledast \boxed{\check{A}_{m+1}}
 \end{aligned}$$

Outros Rearranjos (Permutações)

- Transposição inversa:

$$\pi = (\pi_1 \dots \pi_{i-1} \underline{\pi_i \dots \pi_{j-1}} \underline{\pi_j \dots \pi_{k-1}} \pi_k \dots \pi_n)$$

$$\pi \cdot \rho\tau_1(i, j, k) = (\pi_1 \dots \pi_{i-1} \underline{\pi_j \dots \pi_{k-1}} \underline{-\pi_{j-1} \dots -\pi_i} \pi_k \dots \pi_n)$$

$$\pi \cdot \rho\tau_2(i, j, k) = (\pi_1 \dots \pi_{i-1} \underline{-\pi_{k-1} \dots -\pi_j} \underline{\pi_i \dots \pi_{j-1}} \pi_k \dots \pi_n)$$

- Revrev:

$$\pi = (\pi_1 \dots \pi_{i-1} \underline{\pi_i \dots \pi_{j-1}} \underline{\pi_j \dots \pi_{k-1}} \pi_k \dots \pi_n)$$

$$\pi \cdot \rho\rho(i, j, k) = (\pi_1 \dots \pi_{i-1} \underline{-\pi_{j-1} \dots -\pi_i} \underline{-\pi_{k-1} \dots -\pi_j} \pi_k \dots \pi_n)$$

- Block interchange:

$$\pi = (\pi_1 \dots \pi_{i-1} \underline{\pi_i \dots \pi_j} \dots \pi_{x-1} \underline{\pi_x \dots \pi_y} \dots \pi_n)$$

$$\pi \cdot bi(i, j, x, y) = (\pi_1 \dots \pi_{i-1} \underline{\pi_x \dots \pi_y} \dots \pi_{x-1} \underline{\pi_i \dots \pi_j} \dots \pi_n)$$

- Modelo de rearranjo \mathcal{M} : define o conjunto de rearranjos permitidos no problema.
- Função de custo $w : \mathcal{M} \rightarrow \mathbb{R}$: associa um custo à ocorrência de um rearranjo pertencente a \mathcal{M} .

Distância de Rearranjos

Entrada: Um modelo de rearranjos \mathcal{M} , uma função de custo w , e um par de genomas $(\mathcal{G}_o, \mathcal{G}_d)$.

Objetivo: Encontrar uma sequência $S = (\beta_1, \beta_2, \dots, \beta_{|S|})$ de custo mínimo que transforma \mathcal{G}_o em \mathcal{G}_d e é formada apenas de rearranjos pertencentes a \mathcal{M} . O custo dessa sequência é chamada de distância $d_{\mathcal{M}}(\mathcal{G}_o, \mathcal{G}_d)$.

- Quando temos $w(\beta) = 1$ para todo $\beta \in \mathcal{M}$, queremos encontrar uma sequência S de tamanho mínimo.
- Quando o modelo \mathcal{M} e a função de custo w estão implícitos, podemos representar uma instância da seguinte maneira:
 - Instância clássica balanceada: $\mathcal{I} = (\pi)$;
 - Instância clássica desbalanceada: $\mathcal{I} = (A, \iota^n)$;
 - Instância intergênica: $\mathcal{I}^{ig} = (\mathcal{G}_o, \mathcal{G}_d)$, com $\mathcal{G}_o = (A, \check{A})$ e $\mathcal{G}_d = (\iota^n, \check{\iota}^n)$.

Nova 1.375-Aproximação para a Ordenação de Permutações por Transposições

Ordenação de Permutações por Transposições

- O algoritmo de 1.375-aproximação proposto por Elias e Hartman em 2006 [27] é um dos resultados mais importantes da área.
- Recentemente, Silva e coautores [39] mostraram que o algoritmo não garante o fator de 1.375 para todas as instâncias.
- Silva e coautores [39] apresentaram um novo algoritmo que garante o fator de aproximação de 1.375 para todas as permutações usando uma abordagem algébrica.
- Esse algoritmo possui complexidade de tempo de $O(n^6)$.
- Desenvolvemos um novo algoritmo que garante o fator de aproximação de 1.375 para todas as permutações e possui complexidade de tempo de $O(n^5)$.

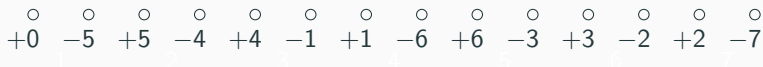
$$G(\pi) = (V, E_o \cup E_d)$$

$$V = \{+\pi_0, -\pi_1, +\pi_1, -\pi_2, +\pi_2, \dots, -\pi_n, +\pi_n, -\pi_{n+1}\}$$

$$E_o = \{(-\pi_i, +\pi_{i-1}) \mid 1 \leq i \leq n+1\}$$

$$E_d = \{(-\iota_i^n, +\iota_{i-1}^n) \mid 1 \leq i \leq n+1\}$$

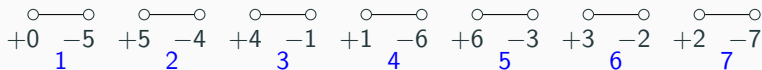
O grafo de ciclos $G(\pi) = (V, E_o \cup E_d)$ para $\pi = (5\ 4\ 1\ 6\ 3\ 2)$. Este grafo possui os ciclos $C_1 = (3, 1)$, $C_2 = (6, 2, 4)$ e $C_3 = (7, 5)$. A transposição $\tau(2, 4, 6)$ quebra o ciclo C_2 em três ciclos.



$$V = \{+\pi_0, -\pi_1, +\pi_1, -\pi_2, +\pi_2, \dots, -\pi_n, +\pi_n, -\pi_{n+1}\}$$

O grafo de ciclos $G(\pi) = (V, E_o \cup E_d)$ para $\pi = (5\ 4\ 1\ 6\ 3\ 2)$. Este grafo possui os ciclos $C_1 = (3, 1)$, $C_2 = (6, 2, 4)$ e $C_3 = (7, 5)$. A transposição $\tau(2, 4, 6)$ quebra o ciclo C_2 em três ciclos.

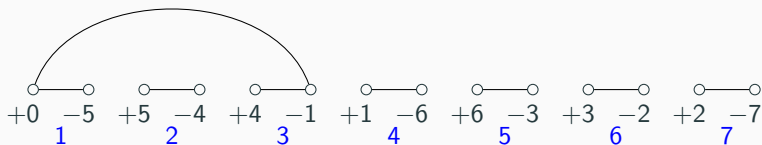
Grafo de Ciclos



$$E_o = \{(-\pi_i, +\pi_{i-1}) \mid 1 \leq i \leq n+1\}$$

O grafo de ciclos $G(\pi) = (V, E_o \cup E_d)$ para $\pi = (5 \ 4 \ 1 \ 6 \ 3 \ 2)$. Este grafo possui os ciclos $C_1 = (3, 1)$, $C_2 = (6, 2, 4)$ e $C_3 = (7, 5)$. A transposição $\tau(2, 4, 6)$ quebra o ciclo C_2 em três ciclos.

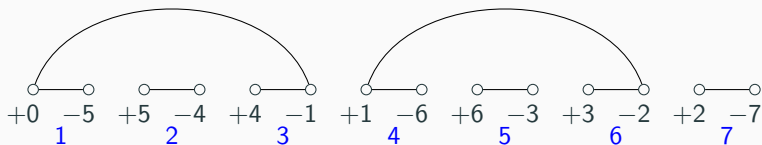
Grafo de Ciclos



$$E_d = \{(-l_i^n, +l_{i-1}^n) \mid 1 \leq i \leq n+1\}$$

O grafo de ciclos $G(\pi) = (V, E_o \cup E_d)$ para $\pi = (5\ 4\ 1\ 6\ 3\ 2)$. Este grafo possui os ciclos $C_1 = (3, 1)$, $C_2 = (6, 2, 4)$ e $C_3 = (7, 5)$. A transposição $\tau(2, 4, 6)$ quebra o ciclo C_2 em três ciclos.

Grafo de Ciclos



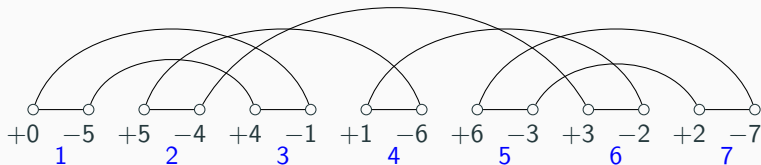
$$V = \{+\pi_0, -\pi_1, +\pi_1, -\pi_2, +\pi_2, \dots, -\pi_n, +\pi_n, -\pi_{n+1}\}$$

$$E_o = \{(-\pi_i, +\pi_{i-1}) \mid 1 \leq i \leq n+1\}$$

$$E_d = \{(-\iota_i^n, +\iota_{i-1}^n) \mid 1 \leq i \leq n+1\}$$

O grafo de ciclos $G(\pi) = (V, E_o \cup E_d)$ para $\pi = (5\ 4\ 1\ 6\ 3\ 2)$. Este grafo possui os ciclos $C_1 = (3, 1)$, $C_2 = (6, 2, 4)$ e $C_3 = (7, 5)$. A transposição $\tau(2, 4, 6)$ quebra o ciclo C_2 em três ciclos.

Grafo de Ciclos



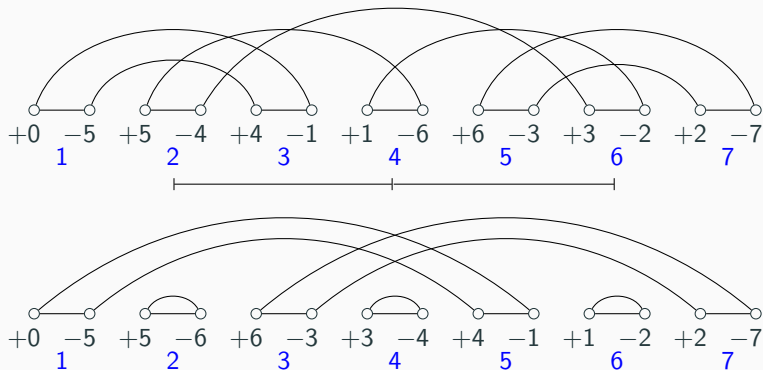
$$V = \{+\pi_0, -\pi_1, +\pi_1, -\pi_2, +\pi_2, \dots, -\pi_n, +\pi_n, -\pi_{n+1}\}$$

$$E_o = \{(-\pi_i, +\pi_{i-1}) \mid 1 \leq i \leq n+1\}$$

$$E_d = \{(-\pi_i^n, +\pi_{i-1}^n) \mid 1 \leq i \leq n+1\}$$

O grafo de ciclos $G(\pi) = (V, E_o \cup E_d)$ para $\pi = (5\ 4\ 1\ 6\ 3\ 2)$. Este grafo possui os ciclos $C_1 = (3, 1)$, $C_2 = (6, 2, 4)$ e $C_3 = (7, 5)$. A transposição $\tau(2, 4, 6)$ quebra o ciclo C_2 em três ciclos.

Grafo de Ciclos



O grafo de ciclos $G(\pi) = (V, E_o \cup E_d)$ para $\pi = (5\ 4\ 1\ 6\ 3\ 2)$. Este grafo possui os ciclos $C_1 = (3, 1)$, $C_2 = (6, 2, 4)$ e $C_3 = (7, 5)$. A transposição $\tau(2, 4, 6)$ quebra o ciclo C_2 em três ciclos.

- Um ciclo com k arestas de origem e k arestas de destino é um k -ciclo.
- Um k -ciclo é ímpar se k é ímpar.
- Chamamos um 1-ciclo de unitário ou trivial.
- Uma permutação π' é simples se todo k -ciclo em $G(\pi')$ tem no máximo 3 arestas de origem ($k \leq 3$).
- Se $\pi = \iota^n$, então o grafo possui exatamente $n + 1$ ciclos e todos são unitários.

- $c(\pi)$: número de ciclos.
- $c_{\text{odd}}(\pi)$: número de ciclos ímpares.
- $\Delta c(\pi, \tau) = c(G(\pi \cdot \tau)) - c(G(\pi)) \in \{-2, 0, 2\}$.
- $\Delta c_{\text{odd}}(\pi, \tau) = c_{\text{odd}}(G(\pi \cdot \tau)) - c_{\text{odd}}(G(\pi)) \in \{-2, 0, 2\}$.
- Uma transposição τ é um 2-move se aumenta o número de ciclos ímpares em 2 (i.e., $\Delta c_{\text{odd}}(\pi, \tau) = 2$).

- O primeiro passo do algoritmo consiste num processo de transformação da permutação de entrada π em uma permutação simples π' que mantém o mesmo limitante inferior para a distância do problema de Ordenação por Transposições.
- O segundo passo é aplicar dois 2-moves em $G(\pi')$, se tal sequência existir.
 - Se $G(\pi')$ não admite uma sequência de dois 2-moves, então podemos atualizar o limitante inferior.

- No entanto, Silva e coautores [39] mostraram que o processo de simplificação de π pode fazer com que a atualização do limitante inferior não seja válida.
- Os autores mostraram que existem casos onde $G(\pi)$ admite uma sequência de dois 2-moves, mas a permutação simplificada não admite tal sequência.
- Portanto, a busca pela sequência de dois 2-moves deve ser feita antes da simplificação da permutação.
- Buscar tal sequência de forma exaustiva leva tempo $O(n^6)$.

Busca Aprimorada por um 2-Move

- Provamos que se existe 2-move para $G(\pi)$, então conseguimos achar outro 2-move τ que atende as seguintes propriedades:
 - τ afeta três arestas de origem $o_{i'}$, $o_{j'}$, e $o_{k'}$ do mesmo ciclo; e
 - $i' \in \{1, 2\}$, ou $j' = i' + 1$, ou $k' = j' + 1$.
- Essa prova consiste na listagem de todos os possíveis casos para um 2-move.
- Com isso, conseguimos desenvolver uma busca por um 2-move que possui complexidade de tempo quadrática, já que sempre podemos fixar um dos parâmetros (i, j, k) em relação a outro.
- Modificamos o algoritmo original para usar essa busca quadrática, o que resultou numa complexidade final de $O(n^5)$.

Tabela 1: Comparação entre os resultados experimentais do novo algoritmo (**NEW**) e os resultados dos algoritmos propostos por Elias e Hartman [27] (**EH**) e Silva e coautores [39] (**SKRW**), em todas permutações de tamanho $n \leq 12$.

n	APROX MAX			APROX MED			DIST MED			% SOLUÇÕES ÓTIMAS		
	EH	SKRW	NEW	EH	SKRW	NEW	EH	SKRW	NEW	EH	SKRW	NEW
2	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	100.00	100.00	100.00
3	1.00	1.00	1.00	1.00	1.00	1.00	1.20	1.20	1.20	100.00	100.00	100.00
4	1.00	1.00	1.00	1.00	1.00	1.00	1.6086	1.6086	1.6086	100.00	100.00	100.00
5	1.00	1.00	1.00	1.00	1.00	1.00	2.0924	2.0924	2.0924	100.00	100.00	100.00
6	1.33	1.00	1.00	1.0004	1.00	1.00	2.6063	2.6050	2.6050	99.86	100.00	100.00
7	1.33	1.25	1.25	1.0129	1.0113	1.0014	3.1762	3.1704	3.1311	94.90	95.47	99.40
8	1.50	1.25	1.25	1.0210	1.0183	1.0042	3.7178	3.7076	3.6512	91.64	92.65	98.29
9	1.50	1.25	1.25	1.0301	1.0256	1.0085	4.2796	4.2603	4.1846	86.62	88.54	96.10
10	1.50	1.25	1.25	1.0341	1.0282	1.0125	4.8051	4.7772	4.7032	83.80	86.53	93.94
11	1.50	1.33	1.33	1.0392	1.0321	1.0170	5.3526	5.3157	5.2367	79.40	82.98	90.88
12	1.50	1.33	1.33	1.0415	1.0336	1.0206	5.8694	5.8248	5.7514	76.67	80.91	88.27

Ordenação de Permutação por Transposições e Outros Rearranjos

Ordenação de Permutação por Reversões

- Possui algoritmo polinomial para permutações com sinais [29].
- NP-difícil para permutações sem sinais [26].

Ordenação de Permutação por Transposições

- Foi provado ser NP-difícil em 2012 [25], apesar de existirem algoritmos de aproximação desde 1995 [15].

Ordenação de Permutação por Reversões e Transposições

- Foi provado ser NP-difícil apenas em 2019 [36], mas existem algoritmos de aproximação desde 1995 [45].

Problemas com Transposições e Outros Rearranjos

- Apesar de existirem algoritmos para vários modelos que envolvem transposições e a combinação de reversões, transposições inversas e revrevs, a complexidade desses problemas ainda estava em aberto.
- 2-aproximação para a Ordenação de Permutações com Sinais por Reversões, Transposições e Transposições Inversas [28] (1999).
- 2.25-aproximação para a Ordenação de Permutações sem Sinais por Reversões, Transposições e Transposições Inversas [31] (2008).
- 1.75-aproximação para Ordenação de Permutações com Sinais por Reversões, Transposições, Transposições Inversas e Revrevs [30] (2001).
- 1.5-aproximação para a Ordenação de Permutações com Sinais por Reversões, Transposições e Transposições Inversas Ponderadas quando $1 \leq w_\tau/w_\rho \leq 2$ [14] (2008).
 - w_ρ é o peso de uma reversão.
 - w_τ é o peso de uma transposição, transposições inversa ou revrev.

Consideramos os seguintes modelos:

- $\mathcal{M}_1 = \{\tau, \rho\tau\}$: Transposições e Transposições Inversas;
- $\mathcal{M}_2 = \{\rho, \tau, \rho\tau\}$: Reversões, Transposições e Transposições Inversas;
- $\mathcal{M}_3 = \{\tau, \rho\rho\}$: Transposições e Revrevs;
- $\mathcal{M}_4 = \{\rho, \tau, \rho\rho\}$: Reversões, Transposições e Revrevs;
- $\mathcal{M}_5 = \{\tau, \rho\tau, \rho\rho\}$: Transposições, Transposições Inversas e Revrevs;
- $\mathcal{M}_6 = \{\rho, \tau, \rho\tau, \rho\rho\}$: Reversões, Transposições, Transposições Inversas e Revrevs.

Conceitos Preliminares - Breakpoints

Breakpoint de reversões sem sinais

Um *breakpoint de reversões sem sinais* existe entre um par de elementos consecutivos (π_i, π_{i+1}) se $|\pi_{i+1} - \pi_i| \neq 1$, para $0 \leq i \leq n$.

$$\pi = (0 \circ 4 \ 3 \circ 5 \circ 1 \ 2 \circ 6 \ 7)$$

Breakpoint de transposições / breakpoint de reversões com sinais

Um *breakpoint de transposições* (ou *breakpoint de reversões com sinais*) existe entre um par de elementos consecutivos (π_i, π_{i+1}) se $\pi_{i+1} - \pi_i \neq 1$, para $0 \leq i \leq n$.

$$\pi = (0 \circ 4 \circ 3 \circ 5 \circ 1 \ 2 \circ 6 \ 7)$$

$$\pi = (+0 \circ -4 \ -3 \circ +5 \circ +1 \ +2 \circ -6 \circ +7)$$

- As provas de dificuldade apresentadas são baseadas em reduções do problema NP-difícil da Ordenação de Permutações por Transposições Ótimas **B3T** [25].
- Esse problema foi utilizado em provas de complexidade para a Ordenação de Permutações por Transposições [25] e a Ordenação de Permutações por Reversões e Transposições [36].

Ordenação de Permutações por Transposições Ótimas (B3T)

Entrada: Uma permutação sem sinais π .

Pergunta: $d_{\tau}(\pi) = b_{\tau}(\pi)/3$?

Obs: Uma transposição remove no máximo 3 breakpoints.

Ordenação de Permutações por Rearranjos Ponderados (SbWR)

Entrada: Uma permutação com sinais π , um inteiro k e pesos $\{w_\rho, w_\tau\}$, tal que $w_\tau/w_\rho \leq 1.5$.

Pergunta: $d_{\mathcal{M}}^w(\pi) \leq k$?

- É possível obter uma redução do problema **B3T**.
- Dada uma instância (π) para **B3T**, seja:
 - $\pi' = (+\pi_1 +\pi_2 \dots +\pi_n)$;
 - $k = w_\tau b_\tau(\pi)/3$.
- (π) satisfaz **B3T** se, e somente se, $(\pi', k, w_\rho, w_\tau)$ satisfaz **SbWR**.

Ordenação de Permutações por Rearranjos Ponderados (SbWR)

Entrada: Uma permutação sem sinais π , um inteiro k e pesos $\{w_\rho, w_\tau\}$, tal que $w_\tau/w_\rho \leq 1.5$.

Pergunta: $d_{\mathcal{M}}^w(\pi) \leq k$?

- É possível obter uma redução do problema **B3T**.
- Dada uma instância (π) para **B3T**, seja:
 - $\pi' = (\pi'_1 \dots \pi'_{2n})$ tal que $\pi'_{2i-1} = 2\pi_i - 1$ e $\pi'_{2i} = 2\pi_i$, com $1 \leq i \leq n$;
 - $k = w_\tau b_\tau(\pi)/3$.
- (π) satisfaz **B3T** se, e somente se, $(\pi', k, w_\rho, w_\tau)$ satisfaz **SbWR**.

Distância em Genomas Desbalanceados

Modelos Considerados

- $\mathcal{M}_{\rho}^{\phi,\psi} = \{\rho, \psi, \phi\}$: reversões e *indels* em strings sem sinais;
- $\mathcal{M}_{\tau}^{\phi,\psi} = \{\tau, \psi, \phi\}$: transposições e *indels* em strings sem sinais;
- $\mathcal{M}_{\rho,\tau}^{\phi,\psi} = \{\rho, \tau, \psi, \phi\}$: reversões, transposições, e *indels* em strings com ou sem sinais.

Complexidade dos Problemas

O problema de Distância de Rearranjos é NP-difícil para os modelos $\mathcal{M}_{\rho}^{\phi,\psi}$ e $\mathcal{M}_{\tau}^{\phi,\psi}$, considerando strings sem sinais, e para o modelo $\mathcal{M}_{\rho,\tau}^{\phi,\psi}$, considerando strings com ou sem sinais.

- Adaptamos o conceito de breakpoints para genomas desbalanceados.
- Sejam π^A e π^L strings simplificadas, com tamanho n' , criadas como cópias de A e L^n , respectivamente, mas removendo os elementos que não são comuns.

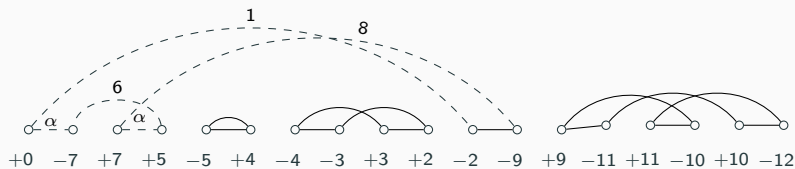
Strings Simplificadas

Considerando a instância $\mathcal{I} = (A, L^n)$, com $A = (4\ 3\ \alpha\ 1\ 2\ 6\ 7)$ e $L^n = (1\ 2\ 3\ 4\ 5\ 6\ 7\ 8)$, temos $\pi^A = (4\ 3\ 1\ 2\ 6\ 7)$ e $\pi^L = (1\ 2\ 3\ 4\ 6\ 7)$, com $n' = 6$.

- Definimos breakpoints de acordo com elementos adjacentes em π^L .

- Mostramos limitantes inferiores para a distância de rearranjos baseados no número de breakpoints e no tamanho de $\Sigma_{\iota^n} \setminus \Sigma_A$.
- $b_{\mathcal{M}}(\mathcal{I}) + |\Sigma_{\iota^n} \setminus \Sigma_A| = 0$ se, e somente se, $A = \iota^n$.
- Além disso, provamos que sempre existe sequência de reversões ou transposições que diminui o valor de $b_{\mathcal{M}}(\mathcal{I}) + |\Sigma_{\iota^n} \setminus \Sigma_A|$.
- Com isso, desenvolvemos algoritmos gulosos com os seguintes fatores de aproximação:
 - 2 para $\mathcal{M}_{\rho}^{\phi, \psi}$;
 - 3 para $\mathcal{M}_{\tau}^{\phi, \psi}$;
 - 3 para $\mathcal{M}_{\rho, \tau}^{\phi, \psi}$.

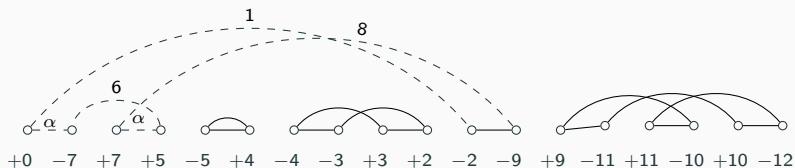
Grafo de Ciclos Rotulado



Grafo de ciclos rotulado $G(\mathcal{I}) = G(A, \iota^n)$ para as strings ι^n , com $n = 11$, e $A = (\alpha \ +7 \ \alpha \ -5 \ -4 \ +3 \ -2 \ +9 \ +11 \ +10)$.

- $V = \{+\pi_0^A, -\pi_1^A, +\pi_1^A, -\pi_2^A, +\pi_2^A, \dots, -\pi_{n'}^A, +\pi_{n'}^A, -\pi_{n'+1}^A\}$.
- $E_o = \{e_i = (+\pi_{i-1}^A, -\pi_i^A) : 1 \leq i \leq n' + 1\}$.
- $E_d = \{e'_i = (+\pi_{i-1}^{\iota}, -\pi_i^{\iota}) : 1 \leq i \leq n' + 1\}$.
- $\ell : (E_o \cup E_d) \rightarrow (\Sigma_{\iota^n} \setminus \Sigma_A) \cup \{\alpha, \emptyset\}$ é uma função que atribui um rótulo a cada aresta do grafo.

Grafo de Ciclos Rotulado

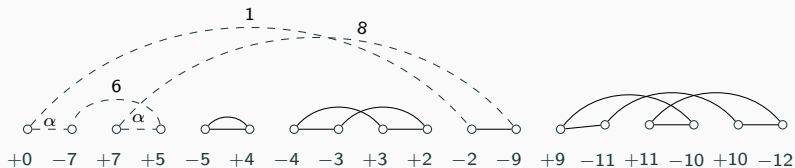


Grafo de ciclos rotulado $G(\mathcal{I}) = G(A, \iota^n)$ para as strings ι^n , com $n = 11$, e $A = (\alpha \ +7 \ \alpha \ -5 \ -4 \ +3 \ -2 \ +9 \ +11 \ +10)$.

- O rótulo $\ell(e_i) = \emptyset$ se π_{i-1}^A e π_i^A são consecutivos em A . Caso contrário, temos que $\ell(e_i) = \alpha$.
- O rótulo $\ell(e'_i) = \emptyset$ se π_{i-1}^{ι} e π_i^{ι} são consecutivos em ι^n . Caso contrário, temos o rótulo $\ell(e'_i) = \pi_{i-1}^{\iota} + 1$.
- Um ciclo é *rotulado* se possui aresta de origem rotulada. Caso contrário, o ciclo é *limpo*.
- O 3-ciclo $C_1 = (6, 1, 2)$ é um ciclo rotulado.

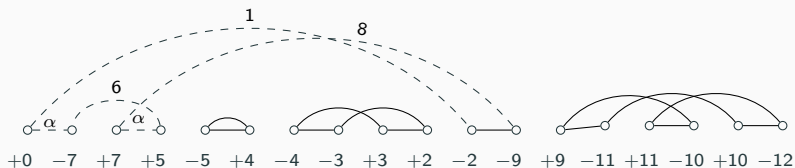
- O número de ciclos limpos é denotado por $c_{\text{clean}}(A, \iota^n)$.
- $A = \iota^n$ se, e somente se, o grafo $G(A, \iota^n)$ possui $n + 1$ ciclos unitários limpos.

Runs e Potencial de Indels



- Um *run de inserção* é um caminho maximal que inicia e termina com arestas de destino rotuladas e toda aresta de origem nesse caminho é uma aresta limpa. Ex: $(+0, -2, -9, +7)$.
- Um *run de deleção* é um caminho maximal que inicia e termina com arestas de origem rotuladas e toda aresta de destino nesse caminho é uma aresta limpa. Ex: $(+5, +7)$.
- O número de *runs* (inserção ou deleção) de um ciclo C é denotado por $\Lambda(C)$.

Runs e Potencial de Indels



- Dado um ciclo C , o *potencial de indels* de C é igual a:

$$\lambda(C) = \begin{cases} \left\lceil \frac{\Lambda(C)+1}{2} \right\rceil, & \text{se } \Lambda(C) > 0 \\ 0, & \text{caso contrário.} \end{cases}$$

- *Potencial de indels* do grafo: $\lambda(A, \iota^n) = \sum_{C \in \mathcal{G}(A, \iota^n)} \lambda(C)$.
- O ciclo possui $C_1 = (6, 1, 2)$ possui $\Lambda(C_1) = 4$ e $\lambda(C_1) = 3$.
- O grafo $G(A, \iota^n)$ possui $\lambda(A, \iota^n) = 3$.
- $A = \iota^n$ se, e somente se, $n + 1 - c(A, \iota^n) + \lambda(A, \iota^n) = 0$.

- Consideramos os seguintes modelos:
 - Block interchanges em strings sem sinais;
 - Block interchanges e reversões em strings com sinais;
 - Transposições em strings sem sinais;
 - Transposições e reversões em strings com sinais.
- Apresentamos limitantes para esses modelos usando o número de ciclos do grafo e o potencial de indel.
- Provamos que sempre é possível achar uma sequência de operações que aumenta o número de ciclos ou diminui o potencial de indels do grafo.
- Com isso, desenvolvemos algoritmos de 2-aproximação para todos os modelos.

Distância em Genomas Desbalanceados com Regiões Intergênicas

Modelos Considerados

- $\mathcal{M}_{\rho}^{\phi,\psi} = \{\rho, \psi, \phi\}$: reversões e *indels* em strings sem sinais;
- $\mathcal{M}_{\tau}^{\phi,\psi} = \{\tau, \psi, \phi\}$: transposições e *indels* em strings sem sinais;
- $\mathcal{M}_{\rho,\tau}^{\phi,\psi} = \{\rho, \tau, \psi, \phi\}$: reversões, transposições, e *indels* em strings com ou sem sinais.

Complexidade dos Problemas

O problema de Distância de Rearranjos Intergênicos é NP-difícil para os modelos $\mathcal{M}_{\rho}^{\phi,\psi}$ e $\mathcal{M}_{\tau}^{\phi,\psi}$, considerando strings sem sinais, e para o modelo $\mathcal{M}_{\rho,\tau}^{\phi,\psi}$, considerando strings com ou sem sinais.

Definição

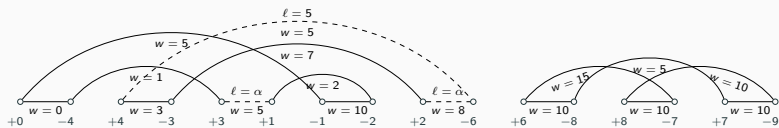
Dizemos que (A_i, A_{i+1}) é um *breakpoint intergênico*, para $0 \leq i \leq |A|$, se uma dessas duas condições é verdadeira:

- (A_i, A_{i+1}) é um *breakpoint* usando a mesma definição do modelo não intergênico;
- (A_i, A_{i+1}) é um par adjacente em ι^n , mas a região intergênica $\check{A}_{i+1} \neq \check{\iota}_j^n$, onde $j = \max(A_i, A_{i+1})$.

O número de breakpoints intergênicos é denotado por $bi_{\mathcal{M}}(\mathcal{G}_o, \mathcal{G}_d)$.

- Mostramos limitantes inferiores para a distância de rearranjos baseados no número de breakpoints intergênicos e no tamanho de $\Sigma_{\iota^n} \setminus \Sigma_A$.
- $bi_{\mathcal{M}}(\mathcal{G}_o, \mathcal{G}_d) + |\Sigma_{\iota^n} \setminus \Sigma_A| = 0$ se, e somente se, $\mathcal{G}_o = \mathcal{G}_d$.
- Além disso, provamos que sempre existe sequência de reversões ou transposições que diminui o valor de $bi_{\mathcal{M}}(\mathcal{G}_o, \mathcal{G}_d) + |\Sigma_{\iota^n} \setminus \Sigma_A|$.
- Com isso, desenvolvemos algoritmos com os seguintes fatores de aproximação:
 - 4 para $\mathcal{M}_{\rho}^{\phi, \psi}$;
 - 4.5 para $\mathcal{M}_{\tau}^{\phi, \psi}$;
 - 6 para $\mathcal{M}_{\rho, \tau}^{\phi, \psi}$.

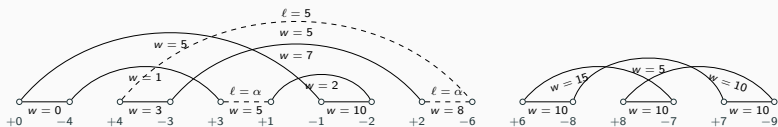
Grafo de Ciclos Rotulado e Ponderado



Grafo de ciclos rotulado e ponderado para $\mathcal{I}^g = (\mathcal{G}_o, \mathcal{G}_d)$, com $\mathcal{G}_o = (A, \check{A})$ e $\mathcal{G}_d = (\iota^n, \check{\iota}^n)$, onde:

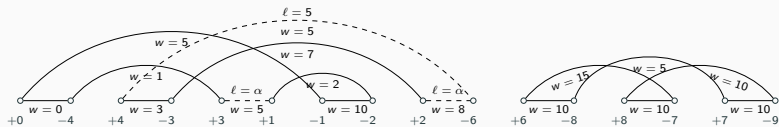
- $A = (+0 \ +4 \ +3 \ \alpha \ -1 \ +2 \ \alpha \ +6 \ +8 \ +7 \ +9)$,
- $\check{A} = (0, 3, 2, 3, 10, 2, 6, 10, 10, 10)$,
- $\iota^n = (+0 \ +1 \ +2 \ +3 \ +4 \ +5 \ +6 \ +7 \ +8 \ +9)$,
- $\check{\iota}^n = (5, 2, 7, 1, 2, 3, 15, 5, 10)$.

Grafo de Ciclos Rotulado e Ponderado



- Os vértices, arestas e rótulos são similares ao Grafo de Ciclos Rotulado.
- Além disso, o grafo $G(\mathcal{G}_o, \mathcal{G}_d)$ possui uma função de custo $w : (E_o \cup E_d) \rightarrow \mathbb{Z}^*$, tal que:
 - $w(e'_i)$ é igual à soma dos tamanhos das regiões intergênicas entre os elementos π_{i-1}^t e π_i^t em \mathcal{G}_d ;
 - $w(e_i)$ é igual à soma dos tamanhos das regiões intergênicas entre os elementos π_{i-1}^A e π_i^A em \mathcal{G}_o .
- Um ciclo $C = (o_1, d_1, o_2, d_2, \dots, o_m, d_m)$ é *balanceado* se a soma dos custos das arestas de destino é igual à soma dos custos das arestas de origem.

Grafo de Ciclos Rotulado e Ponderado



- Um ciclo C é *limpo* se todas as suas arestas de origem e todas as suas arestas de destino são limpas, caso contrário, dizemos que C é *rotulado*.
- Um ciclo C é *bom* se ele é balanceado e limpo.
- Neste exemplo temos os ciclos $C_1 = (4, 1, 3)$, $C_2 = (5, 2)$, $C_3 = (8, 6, 7)$. O ciclo C_1 é um ciclo rotulado e desbalanceado (a soma dos custos das arestas de origem é igual a 15, enquanto a soma dos custos das arestas de destino é igual a 8). O ciclo C_2 é um ciclo rotulado e desbalanceado. O ciclo C_3 é um ciclo bom.

Grafo de Ciclos Rotulado e Ponderado

- O número de ciclos bons em $G(\mathcal{G}_o, \mathcal{G}_d)$ é denotado por $c_g(\mathcal{G}_o, \mathcal{G}_d)$.
- $G(\mathcal{G}_o, \mathcal{G}_d)$ possui apenas ciclos unitários bons se, e somente se, $\mathcal{G}_o = \mathcal{G}_d$.

- Consideramos os seguintes modelos:
 - Reversões em strings com sinais;
 - Transposições em strings sem sinais;
 - Transposições e reversões em strings com sinais.
- Apresentamos limitantes para esses modelos usando o número de ciclos bons no grafo.
- Provamos que sempre é possível achar uma sequência de operações que aumenta o número de ciclos bons no grafo.
- Com isso, desenvolvemos algoritmos com os seguintes fatores de aproximação:
 - 2.5 para $\mathcal{M}_{\rho}^{\phi, \psi}$;
 - 4 para $\mathcal{M}_{\tau}^{\phi, \psi}$;
 - 4 para $\mathcal{M}_{\rho, \tau}^{\phi, \psi}$.

Conclusão

Ordenação de Permutações por Transposições

- Novo algoritmo de 1.375-aproximação.
- Correção de um problema do algoritmo original.
- Possui melhor complexidade de tempo.

Ordenação de Permutações por Transposições e Outros Rearranjos

- Mostramos que o problema é NP-difícil para modelos que contém transposições e a combinação de reversões, transposições inversas e revrevs, quando $w_\tau/w_\rho \leq 1.5$.
- Consequentemente, as versões não ponderadas também são NP-difíceis.
- Consideramos permutações com ou sem sinais.

Distância em Genomas Desbalanceados com ou sem Regiões Intergênicas

- Incorporação de inserções e deleções nos modelos.
- Análise de complexidade de diversos problemas para strings com e sem sinais.
- Definição do Grafo de Ciclos Rotulado e do Grafo de Ciclos Rotulado e Ponderado.
- Algoritmos de aproximação baseados em breakpoints e em grafo de ciclos.
- Experimentos computacionais em dados sintéticos e dados reais.

- Investigação da complexidade dos problemas de Ordenação de Permutações por Transposições e Outros Rearranjos quando $w_{\tau}/w_{\rho} > 1.5$.
- Considerar a abordagem ponderada para os problemas de Distância em Genomas Desbalanceados com ou sem Regiões Intergênicas.

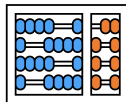
1. “A 1.375-Approximation Algorithm for Sorting by Transpositions with Faster Running Time”, apresentado na conferência *Brazilian Symposium on Bioinformatics* (BSB) em 2022 [2] (Capítulo 3, Seção 3.1);
2. “On the Complexity of Some Variations of Sorting by Transpositions”, publicado na revista *Journal of Universal Computer Science* em 2020 [7] (Capítulo 3, Seção 3.2);
3. “Genome Rearrangement Distance with Reversals, Transpositions, and Indels”, publicado na revista *Journal of Computational Biology* em 2021 [8] (Capítulo 4, Seção 4.2);
4. “Labeled Cycle Graph for Transposition and Indel Distance”, publicado na revista *Journal of Computational Biology* em 2022 [10] (Capítulo 4, Seção 4.3);

5. “Block Interchange and Reversal Distance on Unbalanced Genomes”, apresentado na conferência *Brazilian Symposium on Bioinformatics* (BSB) em 2023 [13] (Capítulo 4, Seção 4.3);
6. “Incorporating Intergenic Regions into Reversal and Transposition Distances with Indels”, apresentado na conferência RECOMB *Comparative Genomics* em 2021. Uma versão estendida foi publicada na revista *Journal of Bioinformatics and Computational Biology* em 2021 [9] (Capítulo 5, Seção 5.2);
7. “Reversal Distance on Genomes with Different Gene Content and Intergenic Regions Information”, apresentado na conferência *Algorithms for Computational Biology* (AlCoB) em 2021 [1] (Capítulo 5, Seção 5.3.1);

8. “Reversal and Indel Distance with Intergenic Region Information”, publicado na revista *IEEE/ACM Transactions on Computational Biology and Bioinformatics* em 2023 [3] (Capítulo 5, Seção 5.3.1);
9. “Transposition Distance Considering Intergenic Regions for Unbalanced Genomes”, apresentado na conferência *International Symposium on Bioinformatics Research and Applications (ISBRA)* em 2022 [11] (Capítulo 5, Seção 5.3.2);
10. “Reversal and Transposition Distance on Unbalanced Genomes Using Intergenic Information”, publicado na revista *Journal of Computational Biology* em 2023 [12] (Capítulo 5, Seção 5.3.2);

- Além dos artigos diretamente relacionados a esta tese previamente mencionados, outras contribuições foram feitas em problemas relacionados.
 - 1 artigo apresentado em conferência nacional [38].
 - 11 artigos apresentados em conferências internacionais [5, 16, 18, 20, 22, 23, 33, 37, 40, 43, 44].
 - 11 artigos publicados em revistas internacionais [4, 6, 17, 19, 21, 24, 32, 34, 35, 41, 42].
- “Rearrangement Distance Problems: An updated survey”, aceito para publicação na revista *ACM Computing Surveys* em 2024.
 - Revisão dos principais resultados da literatura de rearranjo de genomas dos últimos anos, o que inclui todos os resultados publicados nesta tese.

Obrigado!



Variações do Problema de Distância de Rearranjos

Defesa de Doutorado

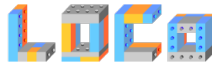
Candidato: **Alexsandro Oliveira Alexandrino**

Orientador: **Prof. Dr. Zanoni Dias**

Coorientador: **Prof. Dr. Ulisses Martins Dias**

27 de Março de 2024

Laboratório de Otimização Combinatória
Instituto de Computação
Universidade Estadual de Campinas



- [1] Alexandrino, A. O., Brito, K. L., Oliveira, A. R., Dias, U., and Dias, Z. (2021a). Reversal Distance on Genomes with Different Gene Content and Intergenic Regions Information. In *Proceedings of the 8th International Conference on Algorithms for Computational Biology (AICoB'2021)*, volume 12715, pages 121–133. Springer International Publishing.
- [2] Alexandrino, A. O., Brito, K. L., Oliveira, A. R., Dias, U., and Dias, Z. (2022a). A 1.375-Approximation Algorithm for Sorting by Transpositions with Faster Running Time. In *Proceedings of the 15th Brazilian Symposium on Bioinformatics (BSB'2022)*, volume 13523 of *Lecture Notes in Computer Science*, pages 147–157. Springer Nature Switzerland.

- [3] Alexandrino, A. O., Brito, K. L., Oliveira, A. R., Dias, U., and Dias, Z. (2023a). Reversal and Indel Distance with Intergenic Region Information. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 20(3):1628–1640.
- [4] Alexandrino, A. O., Lintzmayer, C. N., and Dias, Z. (2020a). Sorting Permutations by Fragmentation-Weighted Operations. *Journal of Bioinformatics and Computational Biology*, 18(2):2050006.1–2050006.31.
- [5] Alexandrino, A. O., Miranda, G. H. S., Lintzmayer, C. N., and Dias, Z. (2019). Approximation Algorithms for Sorting Permutations by Length-Weighted Short Rearrangements. *Electronic Notes in Theoretical Computer Science*, 346:29–40.

- [6] Alexandrino, A. O., Miranda, G. H. S., Lintzmayer, C. N., and Dias, Z. (2020b). Length-weighted λ -rearrangement Distance. *Journal of Combinatorial Optimization*, pages 1–24.
- [7] Alexandrino, A. O., Oliveira, A. R., Dias, U., and Dias, Z. (2020c). On the Complexity of Some Variations of Sorting by Transpositions. *Journal of Universal Computer Science*, 26(9):1076–1094.
- [8] Alexandrino, A. O., Oliveira, A. R., Dias, U., and Dias, Z. (2021b). Genome Rearrangement Distance with Reversals, Transpositions, and Indels. *Journal of Computational Biology*, 28(3):235–247.
- [9] Alexandrino, A. O., Oliveira, A. R., Dias, U., and Dias, Z. (2021c). Incorporating Intergenic Regions into Reversal and Transposition Distances with Indels. *Journal of Bioinformatics and Computational Biology*, 19(06):2140011.

- [10] Alexandrino, A. O., Oliveira, A. R., Dias, U., and Dias, Z. (2022b). Labeled Cycle Graph for Transposition and Indel Distance. *Journal of Computational Biology*, 29(03):243–256.
- [11] Alexandrino, A. O., Oliveira, A. R., Jean, G., Fertin, G., Dias, U., and Dias, Z. (2022c). Transposition Distance Considering Intergenic Regions for Unbalanced Genomes. In *Proceedings of the 18th International Symposium on Bioinformatics Research and Applications (ISBRA'2022)*, volume 13760, pages 100–113. Springer Nature Switzerland.
- [12] Alexandrino, A. O., Oliveira, A. R., Jean, G., Fertin, G., Dias, U., and Dias, Z. (2023b). Reversal and Transposition Distance on Unbalanced Genomes Using Intergenic Information. *Journal of Computational Biology*, 30(8):861–876.

- [13] Alexandrino, A. O., Siqueira, G., Brito, K. L., Oliveira, A. R., Dias, U., and Dias, Z. (2023c). Block Interchange and Reversal Distance on Unbalanced Genomes. In *Proceedings of the 16th Brazilian Symposium on Bioinformatics (BSB'2023)*, volume 13954 of *Lecture Notes in Computer Science*, pages 1–13. Springer Nature Switzerland.
- [14] Bader, M., Abouelhoda, M. I., and Ohlebusch, E. (2008). A Fast Algorithm for the Multiple Genome Rearrangement Problem with Weighted Reversals and Transpositions. *BMC Bioinformatics*, 9(1):1–13.
- [15] Bafna, V. and Pevzner, P. A. (1995). Sorting Permutations by Transpositions. In *Proceedings of the 6th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'1995)*, pages 614–623, Philadelphia, PA, USA. Society for Industrial and Applied Mathematics.

- [16] Brito, K. L., Alexandrino, A. O., Oliveira, A. R., Dias, U., and Dias, Z. (2020). Sorting by Reversals and Transpositions with Proportion Restriction. In *Proceedings of the 13th Brazilian Symposium on Bioinformatics (BSB'2020)*, pages 117–128. Springer International Publishing.
- [17] Brito, K. L., Alexandrino, A. O., Oliveira, A. R., Dias, U., and Dias, Z. (2021a). Reversals and Transpositions Distance with Proportion Restriction. *Journal of Bioinformatics and Computational Biology*, 19(04):2150013.
- [18] Brito, K. L., Alexandrino, A. O., Oliveira, A. R., Dias, U., and Dias, Z. (2021b). Reversals Distance Considering Flexible Intergenic Regions Sizes. In *Proceedings of the 8th International Conference on Algorithms for Computational Biology (AlCoB'2021)*, pages 134–145. Springer International Publishing.

- [19] Brito, K. L., Alexandrino, A. O., Oliveira, A. R., Dias, U., and Dias, Z. (2023a). Genome Rearrangement Distance with a Flexible Intergenic Regions Aspect. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 20(03):1641–1653.
- [20] Brito, K. L., Alexandrino, A. O., Siqueira, G., Oliveira, A. R., Dias, U., and Dias, Z. (2024). Maximum Alternating Balanced Cycle Decomposition and Applications in Sorting by Intergenic Operations Problems. In *Proceedings of the 21st Annual Satellite Conference of RECOMB on Comparative Genomics (RECOMB-CG 2024)*, volume 14616, pages 153–172. Springer International Publishing.
- [21] Brito, K. L., Oliveira, A. R., Alexandrino, A. O., Dias, U., and Dias, Z. (2021c). An Improved Approximation Algorithm for the Reversal and Transposition Distance Considering Gene Order and Intergenic Sizes. *Algorithms for Molecular Biology*, 16(1):1–21.

- [22] Brito, K. L., Oliveira, A. R., Alexandrino, A. O., Dias, U., and Dias, Z. (2021d). Reversal and Transposition Distance of Genomes Considering Flexible Intergenic Regions. In *Proceedings of the XI Latin and American Algorithms, Graphs and Optimization Symposium (LAGOS'2021)*, pages 21–29. Procedia Computer Science, Elsevier.
- [23] Brito, K. L., Oliveira, A. R., Alexandrino, A. O., Dias, U., and Dias, Z. (2022). A New Approach for the Reversal Distance with Indels and Moves in Intergenic Regions. In *Proceedings of 19th Annual Satellite Conference of RECOMB on Comparative Genomics (RECOMB-CG 2022)*, volume 13234, pages 205–220. Springer International Publishing.
- [24] Brito, K. L., Oliveira, A. R., Alexandrino, A. O., Dias, U., and Dias, Z. (2023b). Rearrangement Distance with Reversals, Indels, and Moves in Intergenic Regions on Signed and Unsigned Permutations. *Journal of Bioinformatics and Computational Biology*, 21(02):2350009.

- [25] Bulteau, L., Fertin, G., and Rusu, I. (2012). Sorting by Transpositions is Difficult. *SIAM Journal on Discrete Mathematics*, 26(3):1148–1180.
- [26] Caprara, A. (1999). Sorting Permutations by Reversals and Eulerian Cycle Decompositions. *SIAM Journal on Discrete Mathematics*, 12(1):91–110.
- [27] Elias, I. and Hartman, T. (2006). A 1.375-Approximation Algorithm for Sorting by Transpositions. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 3(4):369–379.
- [28] Gu, Q.-P., Peng, S., and Sudborough, I. H. (1999). A 2-Approximation Algorithm for Genome Rearrangements by Reversals and Transpositions. *Theoretical Computer Science*, 210(2):327–339.

- [29] Hannenhalli, S. and Pevzner, P. A. (1999). Transforming Cabbage into Turnip: Polynomial Algorithm for Sorting Signed Permutations by Reversals. *Journal of the ACM*, 46(1):1–27.
- [30] Lin, G.-H. and Xue, G. (2001). Signed Genome Rearrangement by Reversals and Transpositions: Models and Approximations. *Theoretical Computer Science*, 259(1-2):513–531.
- [31] Lou, X. and Zhu, D. (2008). A 2.25-Approximation Algorithm for Cut-and-Paste Sorting of Unsigned Circular Permutations. In *Computing and Combinatorics*, volume 5092, pages 331–341, Heidelberg, Germany. Springer International Publishing.
- [32] Miranda, G. H. S., Alexandrino, A. O., Lintzmayer, C. N., and Dias, Z. (2021). Approximation Algorithms for Sorting λ -Permutations by λ -Operations. *Algorithms*, 14(6):175.

- [33] Oliveira, A. R., Alexandrino, A. O., Jean, G., Fertin, G., Dias, U., and Dias, Z. (2022). Sorting by k -Cuts on Signed Permutations. In *Proceedings of 19th Annual Satellite Conference of RECOMB on Comparative Genomics (RECOMB-CG 2022)*, volume 13234, pages 189–204. Springer International Publishing.
- [34] Oliveira, A. R., Alexandrino, A. O., Jean, G., Fertin, G., Dias, U., and Dias, Z. (2023). Approximation Algorithms for Sorting by k -Cuts on Signed Permutations. *Journal of Combinatorial Optimization*, 45(1):6.
- [35] Oliveira, A. R., Brito, K. L., Alexandrino, A. O., Siqueira, G., Dias, U., and Dias, Z. (2024 (in press)). Rearrangement Distance Problems: An updated survey. *ACM Computing Surveys*.

- [36] Oliveira, A. R., Brito, K. L., Dias, U., and Dias, Z. (2019). On the Complexity of Sorting by Reversals and Transpositions Problems. *Journal of Computational Biology*, 26:1223–1229.
- [37] Pinheiro, P. O., Alexandrino, A. O., Oliveira, A. R., de Souza, C. C., and Dias, Z. (2020). Heuristics for Breakpoint Graph Decomposition with Applications in Genome Rearrangement Problems. In *Proceedings of the 13th Brazilian Symposium on Bioinformatics (BSB'2020)*, pages 129–140. Springer International Publishing.
- [38] Pinheiro, P. O., Alexandrino, A. O., Oliveira, A. R., de Souza, C. C., and Dias, Z. (2021). Algorithms for the Maximum Eulerian Cycle Decomposition Problem. In *Anais do LIII Simpósio Brasileiro de Pesquisa Operacional (SBPO'2021)*, volume 53, page 139228.

- [39] Silva, L. A. G., Kowada, L. A. B., Rocco, N. R., and Walter, M. E. M. T. (2022). A New 1.375-Approximation Algorithm for Sorting by Transpositions. *Algorithms for Molecular Biology*, 17(1):1–17.
- [40] Siqueira, G., Alexandrino, A. O., and Dias, Z. (2022). Signed Rearrangement Distances Considering Repeated Genes and Intergenic Regions. In *Proceedings of 14th International Conference on Bioinformatics and Computational Biology (BICoB'2022)*, volume 83, pages 31–42. EasyChair.
- [41] Siqueira, G., Alexandrino, A. O., and Dias, Z. (2023a). Signed Rearrangement Distances Considering Repeated Genes, Intergenic Regions, and Indels. *Journal of Combinatorial Optimization*, 46(2):16.

- [42] Siqueira, G., Alexandrino, A. O., Oliveira, A. R., and Dias, Z. (2021a). Approximation Algorithm for Rearrangement Distances Considering Repeated Genes and Intergenic Regions. *Algorithms for Molecular Biology*, 16(1):1–23.
- [43] Siqueira, G., Oliveira, A. R., Alexandrino, A. O., and Dias, Z. (2021b). Heuristics for Cycle Packing of Adjacency Graphs for Genomes with Repeated Genes. In *Proceedings of the 14th Brazilian Symposium on Bioinformatics (BSB'2021)*, pages 93–105. Springer International Publishing.
- [44] Siqueira, G., Oliveira Alexandrino, A., Rodrigues Oliveira, A., Jean, G., Fertin, G., and Dias, Z. (2023b). Approximating Rearrangement Distances with Replicas and Flexible Intergenic Regions. In *Proceedings of the 19th International Symposium on Bioinformatics Research and Applications (ISBRA'2023)*, volume 14248, pages 241–254. Springer.

- [45] Walter, M. E. M. T., Dias, Z., and Meidanis, J. (1998). Reversal and Transposition Distance of Linear Chromosomes. In *Proceedings of the 5th International Symposium on String Processing and Information Retrieval (SPIRE'1998)*, pages 96–102, Los Alamitos, CA, USA. IEEE Computer Society.