

Modelos Restritos e Intergênicos para a Ordenação por Reversões e Transposições

Defesa de Tese de Doutorado

Candidato: **Andre Rodrigues Oliveira**

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

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

9 de Dezembro de 2019



Laboratório de Otimização Combinatória
Instituto de Computação
Universidade Estadual de Campinas

1. Introdução
2. Conceitos Preliminares
3. Complexidade dos Problemas de Ordenação por Reversões e Transposições
4. Ordenação de Permutações Circulares com Sinais por Operações Super Curtas
5. Modelos Intergênicos
6. Conclusões

Introdução

- Distância evolucionária.
- Mudanças ocorridas no material genético.
- Mutações Pontuais:

	Inserção	Deleção	Substituição
Sequência Original	TGGCAG	TGGCAG	TGGCAG
Sequência Mutada	TGGCTATAG	TGAG	TGGTAG

- Rearranjos de Genoma:
 - Reversões.
 - Transposições.

Transformação por Rearranjos de Genoma

B. oleracea
(Repolho)



B. campestris
(Nabo)

Figura 1: Transformando repolho em nabo: três reversões são suficientes para transformar o DNA mitocondrial do repolho no do nabo [14].

Transformação por Rearranjos de Genoma

B. oleracea
(Repolho)



B. campestris
(Nabo)

Figura 1: Transformando repolho em nabo: três reversões são suficientes para transformar o DNA mitocondrial do repolho no do nabo [14].

Transformação por Rearranjos de Genoma

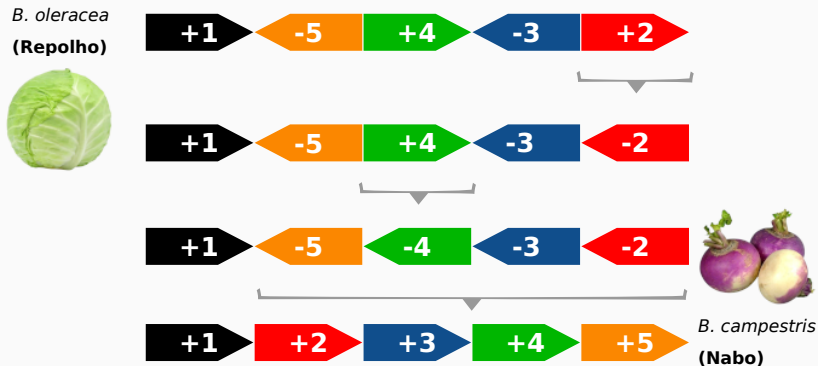


Figura 1: Transformando repolho em nabo: três reversões são suficientes para transformar o DNA mitocondrial do repolho no do nabo [14].

Conceitos Preliminares

- Supomos que os genomas a serem comparados compartilham os mesmos genes, e que não existem genes repetidos.
- Representados por meio de permutações (π) com ou sem sinais, onde cada elemento representa um gene ou um bloco de genes.
- Um dos genomas é representado pela permutação identidade, denotada por $\iota = (+1 \dots +n)$.

- Uma *reversão* $\rho(i, j)$, com $1 \leq i \leq j \leq n$, quando aplicada em uma permutação π , reverte a ordem e a orientação do bloco $[\pi_i.. \pi_j]$, ou seja:

- Uma *reversão* $\rho(i, j)$, com $1 \leq i \leq j \leq n$, quando aplicada em uma permutação π , reverte a ordem e a orientação do bloco $[\pi_i.. \pi_j]$, ou seja:

$$\pi = (\pi_1 \quad \dots \quad \pi_{i-1} \quad \pi_i \quad \pi_{i+1} \quad \dots \quad \pi_{j-1} \quad \pi_j \quad \pi_{j+1} \quad \dots \quad \pi_n)$$

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

- Uma *transposição* $\tau(i, j, k)$, com $1 \leq i < j < k \leq n + 1$, quando aplicada em uma permutação π , troca os blocos $[\pi_i.. \pi_{j-1}]$ e $[\pi_j.. \pi_{k-1}]$ de lugar, ou seja:

- Uma *transposição* $\tau(i, j, k)$, com $1 \leq i < j < k \leq n + 1$, quando aplicada em uma permutação π , troca os blocos $[\pi_i.. \pi_{j-1}]$ e $[\pi_j.. \pi_{k-1}]$ de lugar, ou seja:

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

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

- Temos uma permutação π que queremos transformar em ι .
- Definimos um modelo de rearranjo com um ou mais rearranjos e eventuais restrições, denotado por \mathcal{M} :
 - $r, t, rt, \bar{r}, \bar{rt}$.
 - prefixos/sufixos, ponderados, curtos, super curtos.
- Objetivo: encontrar a distância de ordenação da permutação π no modelo \mathcal{M} , denotada por $d_{\mathcal{M}}(\pi)$.

Complexidade dos Problemas de Ordenação por Reversões e Transposições

- Breakpoints: elementos adjacentes em π mas não em ι .
 - Tipo 1: (π_i, π_{i+1}) é um breakpoint se $|\pi_{i+1} - \pi_i| \neq 1$.
 - Utilizado em permutações sem sinais quando o modelo permite reversões.
 - Tipo 2: (π_i, π_{i+1}) é um breakpoint se $\pi_{i+1} - \pi_i \neq 1$.
 - Utilizado em permutações com sinais ou quando o modelo permite apenas transposições.
 - O número de breakpoints de uma permutação π em um modelo \mathcal{M} é denotado por $b_{\mathcal{M}}(\pi)$.
- Uma reversão remove no máximo dois breakpoints.
- Uma transposição remove no máximo três breakpoints.

Ordenação de Permutações sem Sinais por Reversões (SBR)

Foi provado ser NP-Difícil em 1997 dada sua forte relação com o problema NP-Difícil da decomposição máxima de ciclos [9].

Ordenação de Permutações sem Sinais por Reversões que Removem 2 Breakpoints (SB2R)

Possui algoritmo polinomial exato [10].

Ordenação de Permutações com Sinais por Reversões (SBSR)

Possui algoritmo polinomial exato [14].

Ordenação de Permutações sem Sinais por Transposições (SBT)

Foi provado ser NP-Difícil em 2011 utilizando uma redução do problema SAT [7].

Ordenação de Permutações sem Sinais por Transposições que Removem 3 Breakpoints (SB3T)

Entrada: uma permutação π .

Pergunta: $d_t(\pi) = \frac{b_t(\pi)}{3}$?

Também provado ser NP-Difícil [7].

Ordenação por Reversões e Transposições - Problemas

- Abordagem não ponderada:

Ordenação de Permutações com Sinais por Reversões e Transposições (SBSRT)

Ordenação de Permutações sem Sinais por Reversões e Transposições (SBRT)

- Abordagem ponderada por tipo:

Ordenação Ponderada de Permutações com Sinais por Reversões e Transposições (SBWSRT)

Ordenação Ponderada de Permutações sem Sinais por Reversões e Transposições (SBWRT)

- Apesar dos diversos algoritmos desenvolvidos nos últimos 20 anos, a complexidade dos quatro problemas era desconhecida.

SBWRT

Entrada: uma permutação sem sinais π , pesos w_ρ e w_τ tais que

$\frac{w_\tau}{w_\rho} \leq 1.5$, e um número real k .

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

- É possível obter uma redução do problema **SB3T**.
- Dada uma instância (π') para **SB3T** tal que $\pi' = (\pi'_1 \dots \pi'_n)$, seja:
 - $\pi = (\pi_1 \dots \pi_{2n})$ tal que $\pi_{2i-1} = 2\pi'_i - 1$ e $\pi_{2i} = 2\pi'_i$, para $i \in [1..n]$.
 - $k = w_\tau \cdot \frac{b_{rt}(\pi)}{3} = w_\tau \cdot \frac{b_t(\pi')}{3}$.
- (π') satisfaz **SB3T** se, e somente se, (π, w_τ, w_ρ, k) satisfaz **SBWRT**.

SBWSRT

Entrada: uma permutação com sinais π , pesos w_ρ e w_τ tais que

$\frac{w_\tau}{w_\rho} \leq 1.5$, e um número real k .

Pergunta: $d_{\overline{\pi}}^W(\pi) \leq k$?

- É possível obter uma redução do problema **SB3T**.
- Dada uma instância (π') para **SB3T** tal que $\pi' = (\pi'_1 \dots \pi'_n)$, seja:
 - $\pi = (\pi_1 \dots \pi_n)$ tal que $\pi_i = +\pi'_i$ para $i \in [1..n]$.
 - $k = w_\tau \cdot \frac{b_{\overline{\pi}}(\pi)}{3} = w_\tau \cdot \frac{b_t(\pi')}{3}$.
- (π') satisfaz **SB3T** se, e somente se, (π, w_τ, w_ρ, k) satisfaz **SBWSRT**.

SBRT

Entrada: uma permutação sem sinais π , e um inteiro k .

Pergunta: $d_{rt}(\pi) \leq k$?

SBSRT

Entrada: uma permutação com sinais π , e um inteiro k .

Pergunta: $d_{\bar{r}\bar{t}}(\pi) \leq k$?

- **SBRT** e **SBSRT** são subcasos das versões ponderadas **SBWRT** e **SBWSRT**, respectivamente, onde $w_\tau = w_\rho = 1$.

Ordenação de Permutações
Circulares com Sinais por
Operações Super Curtas

- Genomas podem ser lineares ou circulares [26]:
 - Permutações com ou sem sinais.
 - Permutações lineares ou circulares.
- Existem cenários evolutivos onde os rearranjos tendem afetar porções pequenas do genoma [11, 18, 25]:
 - Operações curtas e super curtas.
 - Operações ponderadas pelo tamanho.
- Cinco resultados já conhecidos de problemas envolvendo operações super curtas [12, 13, 15, 16], exceto:

Ordenação de Permutações Circulares com Sinais por Reversões e Transposições Super Curtas (SBSSSRT)

Encontrar o menor número de reversões/transposições super curtas que ordenam uma permutação circular com sinais π .

- Permutações lineares e operações super curtas cíclicas.
 - 1-reversões e swaps (2-reversões ou 2-transposições).
- Vetor Deslocamento e Vetor Deslocamento Válido (VDV).
- Qualquer sequência de ordenação \mathcal{S} para uma permutação π está associada a um VDV X , e vice versa.
- Dado um VDV X , temos o *valor cruzamento*, denotado por $c_{ij}(X)$, e o *número de cruzamentos*, denotado por $cn(X)$.
- Transformações $X' = T_{i,j}(X)$ entre VDV's.
 - contração ou contração estrita.

$$\begin{aligned}\pi &= (+4 \ +5 \ +3 \ +1 \ -2 \ -6) \\ X^1 &= (3, 3, 0, -3, -3, 0)\end{aligned}$$

Figura 2: Construção do grafo de permutação cíclica $G_{\pi}^{X^1}$ para $\pi = (+4 \ +5 \ +3 \ +1 \ -2 \ -6)$ e $X^1 = (3, 3, 0, -3, -3, 0)$.

Grafo de Permutação Cíclica G_{π}^X

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

$$X^1 = (3, 3, 0, -3, -3, 0)$$

$c_{ij}(X^1)$, com $i \neq j$

$i \downarrow$	1	2	3	4	5	6 $\leftarrow j$
1		0	-1	-1	-1	0
2	0		-1	-1	-1	0
3	+1	+1		-1	-1	0
4	+1	+1	+1		0	0
5	+1	+1	+1	0		0
6	0	0	0	0	0	

$$cn(X^1) = 8$$

Figura 2: Construção do grafo de permutação cíclica $G_{\pi}^{X^1}$ para $\pi = (+4 \ +5 \ +3 \ +1 \ -2 \ -6)$ e $X^1 = (3, 3, 0, -3, -3, 0)$.

Grafo de Permutação Cíclica G_{π}^X

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

$$X^1 = (3, 3, 0, -3, -3, 0)$$

$c_{ij}(X^1)$, com $i \neq j$

$i \downarrow$	1	2	3	4	5	6 $\leftarrow j$
1		0	-1	-1	-1	0
2	0		-1	-1	-1	0
3	+1	+1		-1	-1	0
4	+1	+1	+1		0	0
5	+1	+1	+1	0		0
6	0	0	0	0	0	

$$cn(X^1) = 8$$

$$G_{\pi}^{X^1} = (V, E)$$

$$V = (\pi_1, \dots, \pi_n)$$

$$e = (\pi_i, \pi_j) \in E \text{ se } c_{ij}(X^1) \neq 0$$

$$w(e) = |c_{ij}(X^1)| = |c_{ji}(X^1)|$$

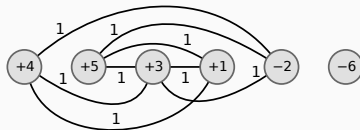
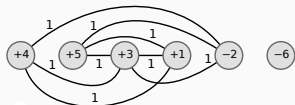


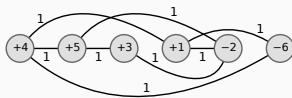
Figura 2: Construção do grafo de permutação cíclica $G_{\pi}^{X^1}$ para $\pi = (+4 \ +5 \ +3 \ +1 \ -2 \ -6)$ e $X^1 = (3, 3, 0, -3, -3, 0)$.

Grafo de Permutação Cíclica G_{π}^X



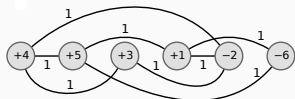
$G_{\pi}^{X^1}$, com $X^1 = (3, 3, 0, -3, -3, 0)$

(a)



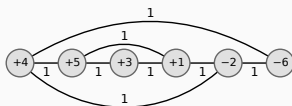
$G_{\pi}^{X^4}$, com $X^4 = (-3, 3, 0, 3, -3, 0)$

(d)



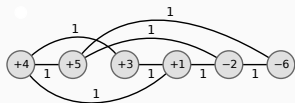
$G_{\pi}^{X^2}$, com $X^2 = (3, -3, 0, 3, -3, 0)$

(b)



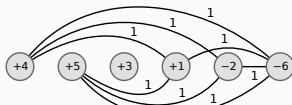
$G_{\pi}^{X^5}$, com $X^5 = (-3, 3, 0, -3, 3, 0)$

(e)



$G_{\pi}^{X^3}$, com $X^3 = (3, -3, 0, -3, 3, 0)$

(c)



$G_{\pi}^{X^6}$, com $X^6 = (-3, -3, 0, 3, 3, 0)$

(f)

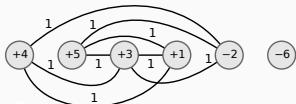
Figura 2: Grafos de permutação cíclica para $\pi = (+4 +5 +3 +1 -2 -6)$ e seis VDV's diferentes onde o número de cruzamentos é 8.

Componentes Ímpares e Operações Super Curtas

- Componentes $cc(G_\pi^X)$ e componentes ímpares $cc^-(G_\pi^X)$.
- Swaps induzidos não diminuem $cc^-(G_\pi^X)$.
- Sempre existe um swap induzido que não aumenta o valor de $cc^-(G_\pi^X)$.
- 1-reversões diminuem ou aumentam o valor de $cc^-(G_\pi^X)$ em uma unidade.
- Dado um VDV X para π , temos que $d_{\overline{sr}}(\pi, X) = cn(X) + cc^-(G_\pi^X)$.
- Ideia geral: encontrar um VDV X para uma permutação π que minimiza a soma $cn(X) + cc^-(G_\pi^X)$.

Grafo de Permutação Cíclica - Componentes e Distância

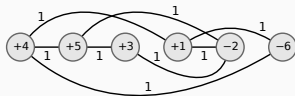
$$cn(X^1) = 8$$



$$G_\pi^{X^1}, \text{ com } X^1 = (3, 3, 0, -3, -3, 0)$$

(a)

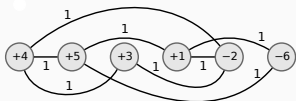
$$cn(X^4) = 8$$



$$G_\pi^{X^4}, \text{ com } X^4 = (-3, 3, 0, 3, -3, 0)$$

(d)

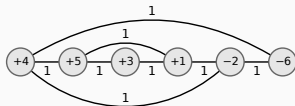
$$cn(X^2) = 8$$



$$G_\pi^{X^2}, \text{ com } X^2 = (3, -3, 0, 3, -3, 0)$$

(b)

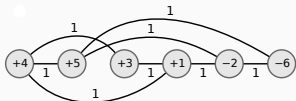
$$cn(X^5) = 8$$



$$G_\pi^{X^5}, \text{ com } X^5 = (-3, 3, 0, -3, 3, 0)$$

(e)

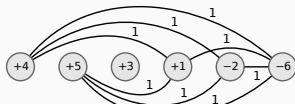
$$cn(X^3) = 8$$



$$G_\pi^{X^3}, \text{ com } X^3 = (3, -3, 0, -3, 3, 0)$$

(c)

$$cn(X^6) = 8$$



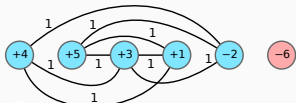
$$G_\pi^{X^6}, \text{ com } X^6 = (-3, -3, 0, 3, 3, 0)$$

(f)

Figura 3: Grafos de permutação cíclica e número de componentes para $\pi = (+4 +5 +3 +1 -2 -6)$ e seis VDV's diferentes onde o número de cruzamentos é 8.

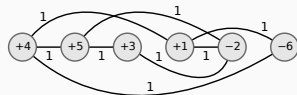
Grafo de Permutação Cíclica - Componentes e Distância

$$\begin{aligned}
 cn(X^1) &= 8 \\
 cc(G_\pi^{X^1}) &= 2 \\
 cc^-(G_\pi^{X^1}) &= 2 \\
 d_{\text{srst}}(\pi, X^1) &= 10
 \end{aligned}$$



$$G_\pi^{X^1}, \text{ com } X^1 = (3, 3, 0, -3, -3, 0)$$

(a)

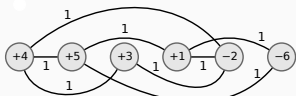


$$G_\pi^{X^4}, \text{ com } X^4 = (-3, 3, 0, 3, -3, 0)$$

(d)

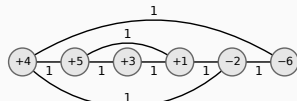
$$cn(X^4) = 8$$

$$cn(X^2) = 8$$



$$G_\pi^{X^2}, \text{ com } X^2 = (3, -3, 0, 3, -3, 0)$$

(b)

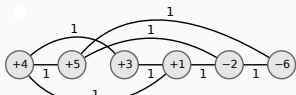


$$G_\pi^{X^5}, \text{ com } X^5 = (-3, 3, 0, -3, 3, 0)$$

(e)

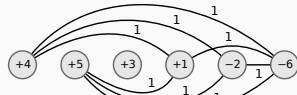
$$cn(X^5) = 8$$

$$cn(X^3) = 8$$



$$G_\pi^{X^3}, \text{ com } X^3 = (3, -3, 0, -3, 3, 0)$$

(c)



$$G_\pi^{X^6}, \text{ com } X^6 = (-3, -3, 0, 3, 3, 0)$$

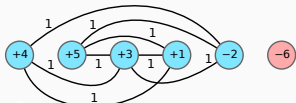
(f)

$$cn(X^6) = 8$$

Figura 3: Grafos de permutação cíclica e número de componentes para $\pi = (+4 +5 +3 +1 -2 -6)$ e seis VDV's diferentes onde o número de cruzamentos é 8.

Grafo de Permutação Cíclica - Componentes e Distância

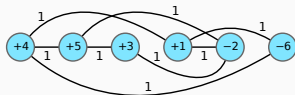
$$\begin{aligned} cn(X^1) &= 8 \\ cc(G_\pi^{X^1}) &= 2 \\ cc^-(G_\pi^{X^1}) &= 2 \\ d_{ssrl}(\pi, X^1) &= 10 \end{aligned}$$



$$G_\pi^{X^1}, \text{ com } X^1 = (3, 3, 0, -3, -3, 0)$$

(a)

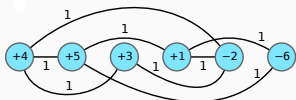
$$\begin{aligned} cn(X^4) &= 8 \\ cc(G_\pi^{X^4}) &= 1 \\ cc^-(G_\pi^{X^4}) &= 0 \\ d_{ssrl}(\pi, X^4) &= 8 \end{aligned}$$



$$G_\pi^{X^4}, \text{ com } X^4 = (-3, 3, 0, 3, -3, 0)$$

(d)

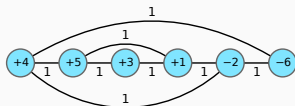
$$\begin{aligned} cn(X^2) &= 8 \\ cc(G_\pi^{X^2}) &= 1 \\ cc^-(G_\pi^{X^2}) &= 0 \\ d_{ssrl}(\pi, X^2) &= 8 \end{aligned}$$



$$G_\pi^{X^2}, \text{ com } X^2 = (3, -3, 0, 3, -3, 0)$$

(b)

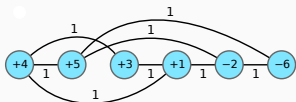
$$\begin{aligned} cn(X^5) &= 8 \\ cc(G_\pi^{X^5}) &= 1 \\ cc^-(G_\pi^{X^5}) &= 0 \\ d_{ssrl}(\pi, X^5) &= 8 \end{aligned}$$



$$G_\pi^{X^5}, \text{ com } X^5 = (-3, 3, 0, -3, 3, 0)$$

(e)

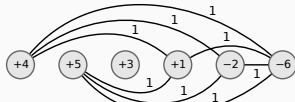
$$\begin{aligned} cn(X^3) &= 8 \\ cc(G_\pi^{X^3}) &= 1 \\ cc^-(G_\pi^{X^3}) &= 0 \\ d_{ssrl}(\pi, X^3) &= 8 \end{aligned}$$



$$G_\pi^{X^3}, \text{ com } X^3 = (3, -3, 0, -3, 3, 0)$$

(c)

$$cn(X^6) = 8$$



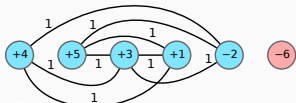
$$G_\pi^{X^6}, \text{ com } X^6 = (-3, -3, 0, 3, 3, 0)$$

(f)

Figura 3: Grafos de permutação cíclica e número de componentes para $\pi = (+4 +5 +3 +1 -2 -6)$ e seis VDV's diferentes onde o número de cruzamentos é 8.

Grafo de Permutação Cíclica - Componentes e Distância

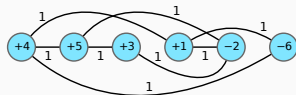
$$\begin{aligned} cn(X^1) &= 8 \\ cc(G_\pi^{X^1}) &= 2 \\ cc^-(G_\pi^{X^1}) &= 2 \\ d_{ssrt}(\pi, X^1) &= 10 \end{aligned}$$



$$G_\pi^{X^1}, \text{ com } X^1 = (3, 3, 0, -3, -3, 0)$$

(a)

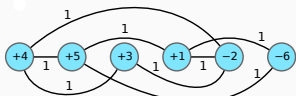
$$\begin{aligned} cn(X^4) &= 8 \\ cc(G_\pi^{X^4}) &= 1 \\ cc^-(G_\pi^{X^4}) &= 0 \\ d_{ssrt}(\pi, X^4) &= 8 \end{aligned}$$



$$G_\pi^{X^4}, \text{ com } X^4 = (-3, 3, 0, 3, -3, 0)$$

(d)

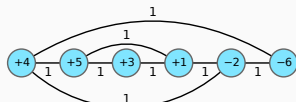
$$\begin{aligned} cn(X^2) &= 8 \\ cc(G_\pi^{X^2}) &= 1 \\ cc^-(G_\pi^{X^2}) &= 0 \\ d_{ssrt}(\pi, X^2) &= 8 \end{aligned}$$



$$G_\pi^{X^2}, \text{ com } X^2 = (3, -3, 0, 3, -3, 0)$$

(b)

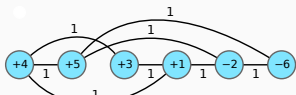
$$\begin{aligned} cn(X^5) &= 8 \\ cc(G_\pi^{X^5}) &= 1 \\ cc^-(G_\pi^{X^5}) &= 0 \\ d_{ssrt}(\pi, X^5) &= 8 \end{aligned}$$



$$G_\pi^{X^5}, \text{ com } X^5 = (-3, 3, 0, -3, 3, 0)$$

(e)

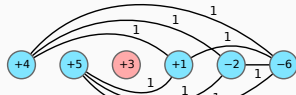
$$\begin{aligned} cn(X^3) &= 8 \\ cc(G_\pi^{X^3}) &= 1 \\ cc^-(G_\pi^{X^3}) &= 0 \\ d_{ssrt}(\pi, X^3) &= 8 \end{aligned}$$



$$G_\pi^{X^3}, \text{ com } X^3 = (3, -3, 0, -3, 3, 0)$$

(c)

$$\begin{aligned} cn(X^6) &= 8 \\ cc(G_\pi^{X^6}) &= 2 \\ cc^-(G_\pi^{X^6}) &= 0 \\ d_{ssrt}(\pi, X^6) &= 8 \end{aligned}$$



$$G_\pi^{X^6}, \text{ com } X^6 = (-3, -3, 0, 3, 3, 0)$$

(f)

Figura 3: Grafos de permutação cíclica e número de componentes para $\pi = (+4 +5 +3 +1 -2 -6)$ e seis VDV's diferentes onde o número de cruzamentos é 8.

Solução Ótima \times Número de Cruzamentos Mínimo

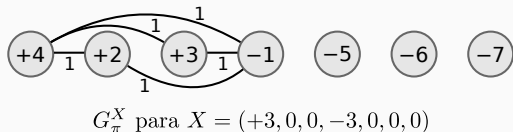


Figura 4: Exemplo de um VDV X cujo número de cruzamentos é mínimo, porém existe um VDV X' tal que $cn(X') > cn(X)$ cuja distância é mínima.

Solução Ótima \times Número de Cruzamentos Mínimo

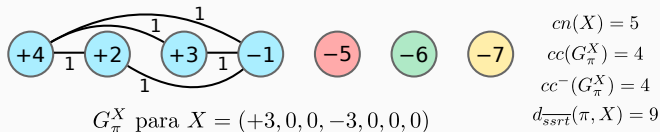


Figura 4: Exemplo de um VDV X cujo número de cruzamentos é mínimo, porém existe um VDV X' tal que $cn(X') > cn(X)$ cuja distância é mínima.

Solução Ótima \times Número de Cruzamentos Mínimo

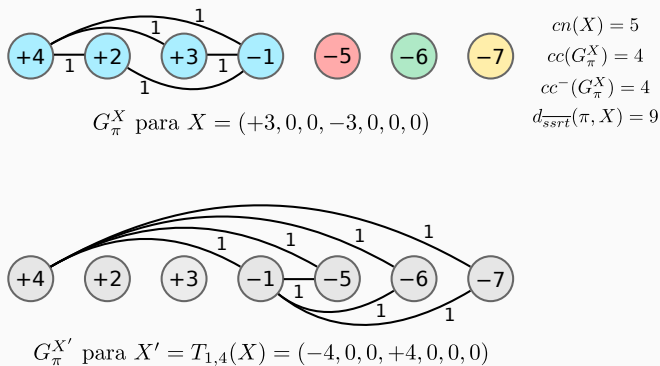


Figura 4: Exemplo de um VDV X cujo número de cruzamentos é mínimo, porém existe um VDV X' tal que $cn(X') > cn(X)$ cuja distância é mínima.

Solução Ótima \times Número de Cruzamentos Mínimo

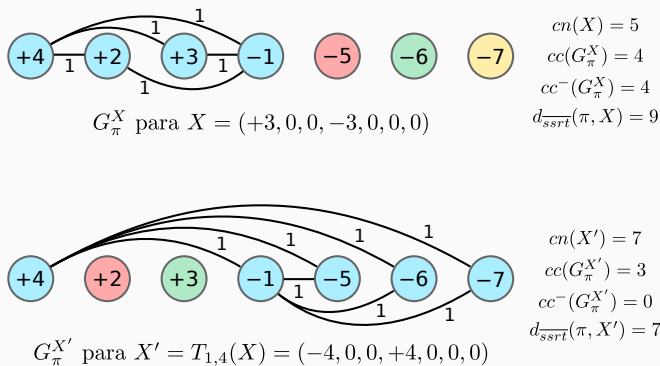


Figura 4: Exemplo de um VDV X cujo número de cruzamentos é mínimo, porém existe um VDV X' tal que $cn(X') > cn(X)$ cuja distância é mínima.

- Denotamos por S o conjunto de todos os VDV's X que possuem valor mínimo de número de cruzamentos.
- Denotamos por S' o conjunto de todos os VDV's X' que não estão em S tais que $X' = T_{i,j}(X)$ para algum $X \in S$.
- Algoritmo desenvolvido:
 - mostrar que a solução ótima $d_{ssrt}(\pi, X^*) = d_{ssrt}(\pi)$ é tal que $X^* \in S \cup S'$.
 - mostrar que é possível encontrar $X^* \in S \cup S'$ em tempo polinomial.

Modelos Intergênicos

- Informações presentes no genoma são perdidas.
- Regiões entre os genes, chamadas de regiões intergênicas [1,17].
- Melhora nos estimadores de distância [1,2].
- Cenários inferidos por algoritmos que consideram regiões intergênicas ficaram mais próximos dos simulados que algoritmos que não as consideram [8].
- Investigamos o modelo intergênico que permite reversões e transposições, considerando operações super curtas e sem restrições de tamanho.

Rearranjos Intergênicos

- Assumimos que os rearranjos quebram regiões intergênicas, e não genes.
 - Reversão Intergênica $\rho_{(x,y)}^{(i,j)}$.
 - Transposição Intergênica $\rho_{(x,y,z)}^{(i,j,k)}$.

Rearranjos Intergênicos

- Assumimos que os rearranjos quebram regiões intergênicas, e não genes.
 - Reversão Intergênica $\rho_{(x,y)}^{(i,j)}$.
 - Transposição Intergênica $\rho_{(x,y,z)}^{(i,j,k)}$.

G Gene 3 TTCAGAT Gene 2 Gene 1 CTCT Gene 7 CT Gene 6 AGTGACA Gene 4 ATCC Gene 8 Gene 5 TGGCTCTGA

(a)

Figura 5: Exemplos de Rearranjos Intergênicos.

Rearranjos Intergênicos

- Assumimos que os rearranjos quebram regiões intergênicas, e não genes.
 - Reversão Intergênica $\rho_{(x,y)}^{(i,j)}$.
 - Transposição Intergênica $\rho_{(x,y,z)}^{(i,j,k)}$.

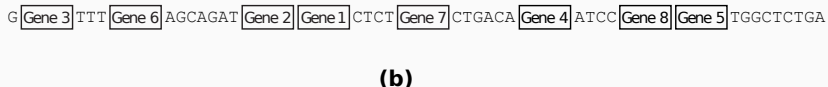
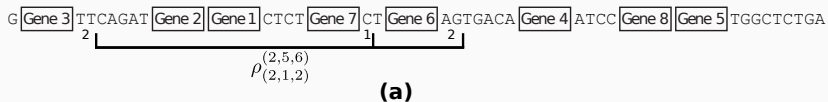


Figura 5: Exemplos de Rearranjos Intergênicos.

Rearranjos Intergênicos

- Assumimos que os rearranjos quebram regiões intergênicas, e não genes.
 - Reversão Intergênica $\rho_{(x,y)}^{(i,j)}$.
 - Transposição Intergênica $\rho_{(x,y,z)}^{(i,j,k)}$.

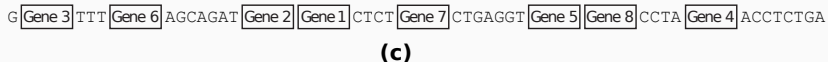
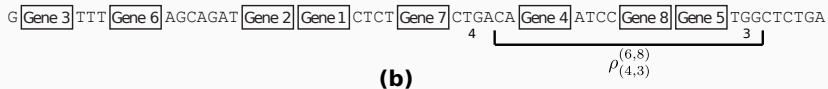
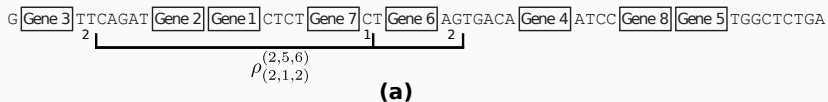


Figura 5: Exemplos de Rearranjos Intergênicos.

Trocas Intergênicas e Transposições Genéricas

- Troca Intergênica $\rho_{(x,y,z)}^{(i,j,k)}$ onde $i = j$ ou $j = k$.

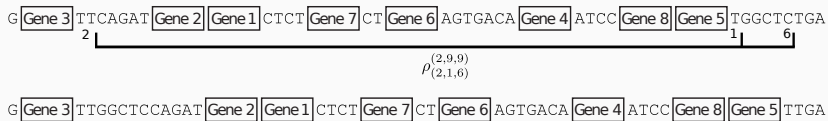


Figura 6: Exemplo de uma Troca Intergênica.

- Transposições Genéricas:
 - Transposição Intergênica.
 - Troca Intergênica.

Representação de genomas

- Um genoma \mathcal{G} é representado por uma sequência de genes $(\pi_1, \pi_2, \dots, \pi_n)$ alternada com uma sequência de regiões intergênicas $(\check{\pi}_1, \check{\pi}_2, \dots, \check{\pi}_{n+1})$:

$$\mathcal{G} = (\pi, \check{\pi}) = (\check{\pi}_1, \pi_1, \check{\pi}_2, \pi_2, \dots, \check{\pi}_n, \pi_n, \check{\pi}_{n+1})$$

Representação de genomas

- Um genoma \mathcal{G} é representado por uma sequência de genes $(\pi_1, \pi_2, \dots, \pi_n)$ alternada com uma sequência de regiões intergênicas $(\check{\pi}_1, \check{\pi}_2, \dots, \check{\pi}_{n+1})$:

$$\mathcal{G} = (\pi, \check{\pi}) = (\check{\pi}_1, \pi_1, \check{\pi}_2, \pi_2, \dots, \check{\pi}_n, \pi_n, \check{\pi}_{n+1})$$

$\mathcal{G}_1(\pi)$: G Gene3 TTCAGAT Gene2 Gene1 CTCT Gene7 CT Gene6 AGTGACA Gene4 ATCC Gene8 Gene5 TGGCTCTGA

$\mathcal{G}_2(\iota)$: CAT Gene1 AACCG Gene2 T Gene3 CTGTA Gene4 ACTCAC Gene5 GGC Gene6 TGTCTG Gene7 GA Gene8 CT

(a)

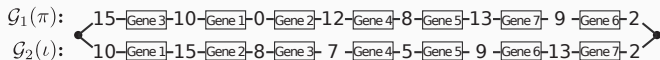
$\mathcal{G}_1(\pi)$: ● 1-Gene3-7-Gene2-0-Gene1-4-Gene7-2-Gene6-7-Gene4-4-Gene8-0-Gene5-9 ●

$\mathcal{G}_2(\iota)$: ● 3-Gene1-5-Gene2-1-Gene3-5-Gene4-6-Gene5-3-Gene6-7-Gene7-2-Gene8-2 ●

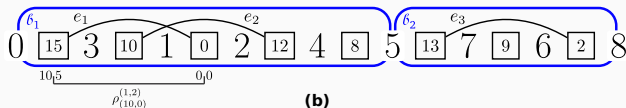
(b)

Figura 7: Representação dos genomas no modelo intergênico.

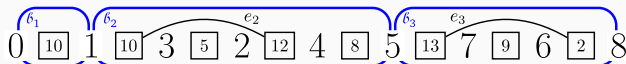
Operações Intergênicas Super Curtas - Grafo Intergênico



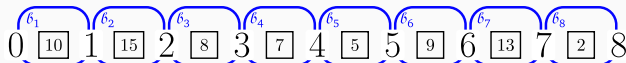
(a)



(b)



(c)



(d)

Figura 8: Dada a instância intergênica em (a), temos em (b) o grafo intergênico $I(\pi, \tilde{\pi}, \iota)$, em (c) o grafo intergênico $I(\pi', \tilde{\pi}', \iota)$ tal que $(\pi', \tilde{\pi}') = (\pi, \tilde{\pi}) \cdot \rho_{(10,0)}^{(1,2)}$ e em (d) o grafo intergênico $I(\iota, \tilde{\iota}, \iota)$.

Ordenação de Permutações sem Sinais por Reversões Intergênicas Super Curtas (SBSSIR)

Ordenação de Permutações sem Sinais por Transposições Intergênicas Super Curtas (SBSSIT)

Ordenação de Permutações sem Sinais por Reversões e Transposições Intergênicas Super Curtas (SBSSIRIT)

- Um algoritmo de 3-aproximação.
- Um algoritmo de $(1 + \frac{1}{\ell})$ -aproximação quando as permutações possuem pelo menos n inversões, onde $\ell = \frac{\text{inv}(\pi)}{n}$.
- Os resultados experimentais mostram que **SBSSIRIT** retornou resultados mais próximos do limitante inferior.

Ordenação de Permutações com Sinais por Reversões Intergênicas Super Curtas (SBSSSIR)

Ordenação de Permutações com Sinais por Reversões e Transposições Intergênicas Super Curtas (SBSSSIRIT)

- Um algoritmo de 5-aproximação.
- Um algoritmo de $(1 + \frac{2}{\ell})$ -aproximação quando as permutações possuem pelo menos n inversões, onde $\ell = \frac{inv(\pi)}{n}$.
- Os resultados experimentais mostram que **SBSSSIRIT** retornou resultados mais próximos do limitante inferior.

O Grafo de Breakpoints Ponderado

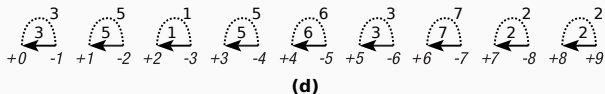
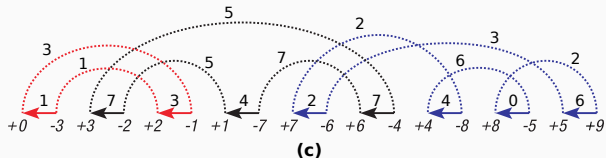
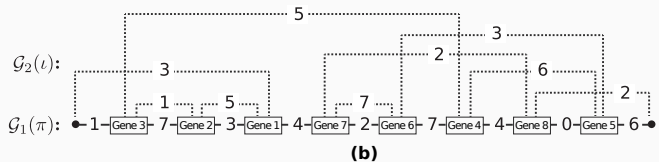
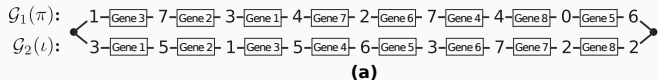


Figura 9: Exemplo da representação de uma instância $(\pi, \tilde{\pi}, \iota)$ no grafo de breakpoints ponderado.

Ordenação de Permutações com Sinais por Reversões Intergênicas (SB SIR)

- Uma redução do problema NP-Difícil **3-PARTIÇÃO**.
- Um algoritmo de 2-aproximação.
- Testes experimentais mostraram resultados muito bons, com aproximação experimental abaixo de 1.1 em todos os conjuntos.
- O modelo que ignora regiões intergênicas começa a subestimar mais cedo o número de operações aplicadas.

Ordenação de Permutações sem Sinais por Transposições Intergênicas (SBIT)

- É uma generalização do problema NP-Difícil **SBT**.
- Um algoritmo de 3.5-aproximação.

Ordenação de Permutações sem Sinais por Transposições Genéricas (SBGT)

- É uma generalização do problema NP-Difícil **SBT**.
- Um algoritmo de 2.5-aproximação.
- Testes experimentais mostram que os estimadores de **SBGT** são melhores que os de **SBIT**, mesmo em conjunto de dados onde apenas transposições intergênicas são aplicadas.

Ordenação de Permutações com Sinais por Reversões e Transposições Intergênicas (SBSIRIT)

- É uma generalização do problema NP-Difícil **SBSRT**.
- Um algoritmo de 3-aproximação.

Ordenação de Permutações com Sinais por Reversões Intergênicas e Transposições Genéricas (SBSIRGT)

- É uma generalização do problema NP-Difícil **SBSRT**.
- Um algoritmo de 2.5-aproximação.
- Testes experimentais mostram que os estimadores de **SBSIRGT** são melhores que os de **SBSIRIT**, mesmo em conjunto de dados onde apenas transposições intergênicas são aplicadas.

Conclusões

Principais Contribuições

- Mostramos que duas versões ponderadas dos problemas de ordenação envolvendo reversões e transposições são NP-Difíceis, quando $\frac{w_r}{w_p} \leq 1.5$.
 - Como consequência, as versões não ponderadas também são NP-Difíceis.
- Um algoritmo polinomial exato para ordenar permutações circulares com sinais por reversões e transposições super curtas.
- Incorporamos as regiões intergênicas:
 - no modelo que permite operações super curtas, criando algoritmos de aproximação que foram testados em conjuntos de dados simulados.
 - no modelo que permite reversões e transposições, mostrando as complexidades e criando algoritmos de aproximação que foram testados em conjuntos de dados simulados.

- Investigar a complexidade dos modelos ponderados quando $\frac{w_\tau}{w_\rho} > 1.5$.
- Investigar algoritmos com fatores de aproximação menores para todas as versões intergênicas.
- Verificar a complexidade do modelo intergênico onde as operações são super curtas.
- Incorporar eventos não conservativos (indels) nos modelos intergênicos, para generalizar sua aplicação.

- “**Sorting signed circular permutations by super short operations**”, publicado em 2018 para a revista *Algorithms for Molecular Biology* [22].
- “**On the Complexity of Sorting by Reversals and Transpositions Problems**”, publicado em 2019 na revista *Journal of Computational Biology* [19].
- “**Super Short Operations on Both Gene Order and Intergenic Sizes**”, publicado em 2019 na revista *Algorithms for Molecular Biology* [24].
 - Versão preliminar apresentada no *Brazilian Symposium on Bioinformatics (BSB)* em 2018 [23].

- **“Sorting Signed Permutations by Intergenic Reversals”**, submetido no mês de Maio de 2019 na revista *IEEE/ACM Transactions on Computational Biology and Bioinformatics*.
- **“A 3.5-approximation Algorithm for Sorting Permutations by Intergenic Transpositions”**, submetido no mês de Dezembro de 2019 para a *International Conference on Algorithms for Computational Biology (AlCoB)*.

- “**Sorting by Weighted Reversals and Transpositions**”, publicado em 2019 na revista *Journal of Computational Biology* [21].
 - Versão preliminar apresentada no *Brazilian Symposium on Bioinformatics (BSB)* em 2018 [20].
- “**Heuristics for the Reversal and Transposition Distance Problem**”, aceito para publicação em 2019 na revista *IEEE/ACM Transactions on Computational Biology and Bioinformatics* [6].
 - Versão preliminar apresentada na *International Conference on Algorithms for Computational Biology (AlCoB)* em 2018 [5].
- “**Sorting by Genome Rearrangements on both Gene Order and Intergenic Sizes**”, aceito para publicação em 2019 na revista *Journal of Computational Biology* [4].
 - Versão preliminar apresentada no *International Symposium on Bioinformatics Research and Applications (ISBRA)* em 2019 [3].

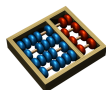
- [1] P. Biller, L. Guéguen, C. Knibbe, and E. Tannier.
Breaking Good: Accounting for Fragility of Genomic Regions in Rearrangement Distance Estimation.
Genome Biology and Evolution, 8(5):1427–1439, 2016.
- [2] P. Biller, C. Knibbe, G. Beslon, and E. Tannier.
Comparative Genomics on Artificial Life.
In *Proceedings of the 12th Conference on Computability in Europe (CiE'2016)*, volume 9709 of *Lecture Notes in Computer Science*, pages 35–44. Springer International Publishing, 2016.
- [3] K. L. Brito, G. Jean, G. Fertin, A. R. Oliveira, U. Dias, and Z. Dias.
Sorting by reversals, transpositions, and indels on both gene order and intergenic sizes.
In *Proceedings of the 15th International Symposium on Bioinformatics Research and Applications (ISBRA'2019)*, volume 11490 of *Lecture Notes in Computer Science*, pages 28–39. Springer International Publishing, 2019.
- [4] K. L. Brito, G. Jean, G. Fertin, A. R. Oliveira, U. Dias, and Z. Dias.
Sorting by Genome Rearrangements on both Gene Order and Intergenic Sizes.
Journal of Computational Biology, pages 1–18, Accepted.

- [5] K. L. Brito, A. R. Oliveira, U. Dias, and Z. Dias.
Heuristics for the sorting signed permutations by reversals and transpositions problem.
In *Proceedings of the 5th International Conference on Algorithms for Computational Biology (AlCoB'2018)*, volume 10849 of *Lecture Notes in Computer Science*, pages 65–75, Cham, 2018. Springer International Publishing.
- [6] K. L. Brito, A. R. Oliveira, U. Dias, and Z. Dias.
Heuristics for the reversal and transposition distance problem.
IEEE/ACM Transactions on Computational Biology and Bioinformatics, PP:1–13, 2019.
- [7] L. Bulteau, G. Fertin, and I. Rusu.
Sorting by Transpositions is Difficult.
SIAM Journal on Discrete Mathematics, 26(3):1148–1180, 2012.
- [8] L. Bulteau, G. Fertin, and E. Tannier.
Genome rearrangements with indels in intergenes restrict the scenario space.
BMC Bioinformatics, 17(S14):225–231, 2016.
- [9] A. Caprara.
Sorting Permutations by Reversals and Eulerian Cycle Decompositions.
SIAM Journal on Discrete Mathematics, 12(1):91–110, 1999.
- [10] D. A. Christie.
Genome Rearrangement Problems.
PhD thesis, Department of Computing Science, University of Glasgow, 1998.

- [11] D. A. Dalevi, N. Eriksen, K. Eriksson, and S. G. E. Andersson.
Measuring genome divergence in bacteria: A case study using chlamydian data.
Journal of Molecular Evolution, 55(1):24–36, 2002.
- [12] G. R. Galvão, C. Baudet, and Z. Dias.
Sorting Circular Permutations by Super Short Reversals.
IEEE/ACM Trans. Comput. Biol. Bioinform., 14(3):620–633, 2017.
- [13] G. R. Galvão, O. Lee, and Z. Dias.
Sorting Signed Permutations by Short Operations.
Algorithms for Molecular Biology, 10(1):1–17, 2015.
- [14] S. Hannenhalli and P. A. Pevzner.
Transforming Cabbage into Turnip: Polynomial Algorithm for Sorting Signed Permutations by Reversals.
Journal of the ACM, 46(1):1–27, 1999.
- [15] M. R. Jerrum.
The Complexity of Finding Minimum-length Generator Sequences.
Theoretical Computer Science, 36(2-3):265–289, 1985.
- [16] D. E. Knuth.
The art of Computer Programming: Fundamental Algorithms.
Addison-Wesley, Reading, Massachusetts, 1973.

- [17] C. Lemaitre, L. Zaghoul, M.-F. Sagot, C. Gautier, A. Arneodo, E. Tannier, and B. Audit.
Analysis of fine-scale mammalian evolutionary breakpoints provides new insight into their relation to genome organisation.
BMC Genomics, 10(1):1–12, 2009.
- [18] A. McLysaght, C. Seoighe, and K. H. Wolfe.
High frequency of inversions during eukaryote gene order evolution.
In *Comparative Genomics: Empirical and Analytical Approaches to Gene Order Dynamics, Map Alignment and the Evolution of Gene Families*, pages 47–58, Dordrecht, 2000. Springer Netherlands.
- [19] A. R. Oliveira, K. L. Brito, U. Dias, and Z. Dias.
On the Complexity of Sorting by Reversals and Transpositions Problems.
Journal of Computational Biology, 26(11):1223–1229, 2019.
- [20] A. R. Oliveira, K. L. Brito, Z. Dias, and U. Dias.
Sorting by weighted reversals and transpositions.
In *Proceedings of the 11th Brazilian Symposium on Bioinformatics (BSB'2018)*, volume 11228 of *Lecture Notes in Computer Science*, pages 38–49. Springer International Publishing, 2018.
- [21] A. R. Oliveira, K. L. Brito, Z. Dias, and U. Dias.
Sorting by Weighted Reversals and Transpositions.
Journal of Computational Biology, 26(5):420–431, 2019.

- [22] A. R. Oliveira, G. Fertin, U. Dias, and Z. Dias.
Sorting signed circular permutations by super short operations.
Algorithms for Molecular Biology, 13(13):1–16, 2018.
- [23] A. R. Oliveira, G. Jean, G. Fertin, U. Dias, and Z. Dias.
Super Short Reversals on Both Gene Order and Intergenic Sizes.
In *Proceedings of the 11th Brazilian Symposium on Bioinformatics (BSB'2018)*, volume 11228 of *Lecture Notes in Computer Science*, pages 14–25. Springer International Publishing, 2018.
- [24] A. R. Oliveira, G. Jean, G. Fertin, U. Dias, and Z. Dias.
Super Short Operations on Both Gene Order and Intergenic Sizes.
Algorithms for Molecular Biology, 14(21):1–17, 2019.
- [25] C. Seoighe, N. Federspiel, T. Jones, N. Hansen, V. Bivolarovic, R. Surzycki, R. Tamse, C. Komp, L. Huizar, R. W. Davis, S. Scherer, E. Tait, D. J. Shaw, D. Harris, L. Murphy, K. Oliver, K. Taylor, M.-A. Rajandream, B. G. Barrell, and K. H. Wolfe.
Prevalence of small inversions in yeast gene order evolution.
Proceedings of the National Academy of Sciences, 97(26):14433–14437, 2000.
- [26] M. Trucksis, J. Michalski, Y. K. Deng, and J. B. Kaper.
The vibrio cholerae genome contains two unique circular chromosomes.
Proceedings of the National Academy of Sciences, 95(24):14464–14469, 1998.



Obrigado!

Defesa de Tese de Doutorado

Candidato: **Andre Rodrigues Oliveira**

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

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

9 de Dezembro de 2019



Laboratório de Otimização Combinatória
Instituto de Computação
Universidade Estadual de Campinas