Heurísticas para o Problema de Ordenação de Permutações com Sinais por Reversões e Transposições

Klairton de Lima Brito

Orientador: Prof. Dr. Zanoni Dias

Universidade Estadual de Campinas

29 de Setembro de 2016

Roteiro

Motivação

- Evolução
- Comparação entre genomas
- Mutações pontuais
- Rearranjos de genomas

Conceitos

Representação de um genoma

Um genoma é representado por uma n-tupla cujos elementos representam um gene ou um bloco de genes. Supondo que não haja repetição de genes, esta n-tupla é uma permutação $\pi = (\pi_1\, \pi_2\, \pi_3\, \ldots\, \pi_n)$, onde $\pi_i \in \{-n,\ -(n-1),\ \ldots,\ -2,\ -1,\ +1,\ +2,\ \ldots,\ +(n-1),\ +n\}$ tal que $|\pi_i| \neq |\pi_j| \leftrightarrow i \neq j$.

Permutações com e sem sinais

- $(+1\ +2\ +3\ -4\ -5)$
- $(1\ 2\ 3\ 4\ 5)$

Conceitos

Modelo de Rearranjo

- Define quais eventos podem ser aplicados para transformar uma permutação em outra.
- A distância entre as permutações π e σ no modelo M é denotada por $d_M(\pi, \sigma) = t$.

Alguns eventos estudados na literatura

- Fusão
- Fissão
- Translocação
- Reversão
- Transposição

Reversões

Definição

Uma reversão $\rho(i,j)$, onde $1 \leq i \leq j \leq n$, é um evento que inverte a posição e a orientação dos genes em um bloco do genoma.

$$\begin{aligned} \pi &= (+\pi_1 \ +\pi_2 \ +\pi_3 \ \ldots \ +\pi_{i-1} \ \underline{+\pi_i \ \ldots \ +\pi_j} \ +\pi_{j+1} \ \ldots \ +\pi_n) \\ \pi \cdot \rho(i,j) &= (+\pi_1 \ +\pi_2 \ +\pi_3 \ \ldots \ +\pi_{i-1} \ \underline{-\pi_j \ \ldots \ -\pi_i} \ +\pi_{j+1} \ \ldots \ +\pi_n) \end{aligned}$$

Exemplo

$$(+2 \ -6 \ \underline{-5 \ +4 \ -1 \ -7} \ +3) \cdot \rho(3,6) = (+2 \ -6 \ \underline{+7 \ +1 \ -4 \ +5} \ +3)$$

$$(2 \ 6 \ \underline{5 \ 4 \ 1 \ 7} \ 3) \cdot \rho(3,6) = (2 \ 6 \ \underline{7 \ 1 \ 4 \ 5} \ 3)$$

Definição

Uma transposição $\tau(i,j,k)$, onde $1 \leq i < j < k \leq n+1$, é um evento que troca de posição dois blocos adjacentes de um mesmo cromossomo, mas sem afetar a orientação e a posição dos genes dentro dos blocos.

$$\pi = (\pi_1\ \pi_2\ \pi_3\ \ldots\ \pi_{i-1}\ \underline{\pi_i\ \ldots\ \pi_{j-1}}\ \underline{\pi_j\ \ldots\ \pi_{k-1}}\ \pi_k\ \ldots\ \pi_n)$$
$$\pi \cdot \tau(i,j,k) = (\pi_1\ \pi_2\ \pi_3\ \ldots\ \pi_{i-1}\ \underline{\pi_j\ \ldots\ \pi_{k-1}}\ \underline{\pi_i\ \ldots\ \pi_{j-1}}\ \pi_k\ \ldots\ \pi_n)$$

Exemplo

$$(2\ \underline{6\ 5\ 4}\ \underline{1\ 7}\ 3) \cdot \tau(2,5,7) = (2\ \underline{1\ 7}\ \underline{6\ 5\ 4}\ 3)$$

Ordenação de uma permutação

Permutação identidade

- $\iota_n = (1\ 2\ 3\ \cdots\ n)$
- $\bar{\iota}_n = (+1\ +2\ +3\ \cdots\ +n)$

Problema de Ordenação de Permutações

- Encontrar a menor sequência de eventos definidos no modelo de rearranjo M que sejam capazes de transformar a permutação α na permutação identidade.
- $d_M(\alpha, \iota_n) = d_M(\alpha)$.

Equivalência entre Distância e Ordenação

$$d_M(\sigma^{-1} \cdot \pi, \sigma^{-1} \cdot \sigma) = d_M(\alpha, \iota_n) = d_M(\alpha)$$

Permutações

- $\pi \quad = (2\ 3\ 4\ 1\ 5)$
- $\sigma \quad = (3\ 4\ 5\ 1\ 2)$
- $\sigma^{-1} = (4\ 5\ 1\ 2\ 3)$
- $\alpha \quad = (5\ 1\ 2\ 4\ 3) = \sigma^{-1} \cdot \pi$

Exemplo

$$\begin{aligned} \alpha \ &= (5\ 1\ 2\ 4\ 3) \\ \alpha^1 &= (\underline{5}\ \underline{1\ 2}\ 4\ 3) \cdot \tau(1,2,4) \\ \alpha^2 &= (1\ 2\ \underline{5\ 4\ 3}) \cdot \rho(3,5) \\ \alpha^2 &= (1\ 2\ \underline{3\ 4\ 5}) = \iota_5 \end{aligned}$$

$$\begin{aligned} \pi \ &= (2\ 3\ 4\ 1\ 5) \\ \pi^1 &= (\underline{2}\ \underline{3\ 4}\ 1\ 5) \cdot \tau(1,2,4) \\ \pi^2 &= (3\ 4\ \underline{2\ 1\ 5}) \cdot \rho(3,5) \\ \pi^2 &= (3\ 4\ \underline{5\ 1\ 2}) = \sigma \end{aligned}$$

Problema de Ordenação de Permutações sem Sinais por Reversões

- NP-Difícil [Caprara, 1999].
- Algoritmo 1.375-aproximado [Berman et al., 2002].

Exemplo

$$
\begin{aligned}
\pi\ &= (6\ 3\ 5\ 2\ 4\ 1)\\
\pi^1 &= (6\ 3\ 5\ \underline{2\ 4}\ 1) \cdot \rho(4,5)\\
\pi^2 &= (6\ 3\ \underline{5\ 4}\ 2\ 1) \cdot \rho(3,4)\\
\pi^3 &= (6\ \underline{3\ 4\ 5}\ 2\ 1) \cdot \rho(2,4)\\
\pi^4 &= (\underline{6\ 5\ 4\ 3\ 2\ 1}) \cdot \rho(1,6)\\
\pi^4 &= (\underline{1\ 2\ 3\ 4\ 5\ 6}) = \iota_6\\
d_r(\pi) &= 4
\end{aligned}
$$

Problema de Ordenação de Permutações com Sinais por Reversões

- Polinomial [Hannenhalli e Pevzner, 1999].

Exemplo

$$
\begin{aligned}
\pi &= (+6\ +3\ +5\ +2\ +4\ +1)\\
\pi^1 &= (\underline{+6\ +3\ +5\ +2\ +4}\ +1) \cdot \rho(1,5)\\
\pi^2 &= (-4\ -2\ -5\ -3\ \underline{-6\ +1}) \cdot \rho(5,6)\\
\pi^3 &= (-4\ -2\ \underline{-5\ -3\ -1}\ +6) \cdot \rho(3,5)\\
\pi^4 &= (-4\ \underline{-2\ +1}\ +3\ +5\ +6) \cdot \rho(2,3)\\
\pi^5 &= (\underline{-4\ -1\ +2\ +3}\ +5\ +6) \cdot \rho(1,4)\\
\pi^6 &= (\underline{-3\ -2\ +1}\ +4\ +5\ +6) \cdot \rho(1,3)\\
\pi^7 &= (\underline{-1}\ +2\ +3\ +4\ +5\ +6) \cdot \rho(1,1)\\
\pi^7 &= (\underline{+1}\ +2\ +3\ +4\ +5\ +6) = \bar{\iota}_6
\end{aligned}
$$

$$d_{\bar{r}}(\pi) = 7$$

Problema de Ordenação de Permutações sem sinais por Transposições

- NP-Difícil [Bulteau et al., 2012].
- Algoritmo 1.375-aproximado [Elias e Hartman, 2006].

Exemplo

$$\pi \;= (6\ 3\ 5\ 2\ 4\ 1)$$
$$\pi^1 = (6\ 3\ \underline{5\ 2}\ \underline{4}\ 1) \cdot \tau(3,5,6)$$
$$\pi^2 = (\underline{6}\ \underline{3\ 4\ 5}\ 2\ 1) \cdot \tau(1,2,5)$$
$$\pi^3 = (\underline{3\ 4\ 5\ 6}\ \underline{2\ 1}) \cdot \tau(1,5,7)$$
$$\pi^4 = (\underline{2}\ \underline{1}\ 3\ 4\ 5\ 6) \cdot \tau(1,2,3)$$
$$\pi^4 = (\underline{1}\ \underline{2}\ 3\ 4\ 5\ 6) = \iota_6$$
$$d_t(\pi) = 4$$

Problema de Ordenação de Permutações por Reversões e Transposições

Problema de Ordenação de Permutações sem Sinais por Reversões e Transposições

- Complexidade desconhecida.
- Algoritmo $2k$-aproximado [Rahman et al., 2008].

Problema de Ordenação de Permutações com Sinais por Reversões e Transposições

- Complexidade desconhecida.
- Algoritmo 2-aproximado [Walter et al., 1998].

Problema de Ordenação de Permutações sem Sinais por Reversões e Transposições

Fatos

- $d_{rt}(\pi) \leq d_r(\pi)$
- $d_{rt}(\pi) \leq d_t(\pi)$

Exemplo

$$
\begin{aligned}
\pi\ &= (6\ 3\ 5\ 2\ 4\ 1)\\
\pi^1 &= (6\ 3\ 5\ \underline{2\ 4}\ 1) \cdot \rho(4,5)\\
\pi^2 &= (6\ 3\ \underline{5\ 4}\ 2\ 1) \cdot \rho(3,4)\\
\pi^3 &= (6\ \underline{3\ 4\ 5}\ 2\ 1) \cdot \rho(2,4)\\
\pi^4 &= (\underline{6\ 5\ 4\ 3\ 2\ 1}) \cdot \rho(1,6)\\
\pi^4 &= (\underline{1\ 2\ 3\ 4\ 5\ 6}) = \iota_6\\
d_r(\pi) &= 4
\end{aligned}
$$

$$
\begin{aligned}
\pi\ &= (6\ 3\ 5\ 2\ 4\ 1)\\
\pi^1 &= (6\ 3\ \underline{5\ 2}\ \underline{4}\ 1) \cdot \tau(3,5,6)\\
\pi^2 &= (\underline{6}\ \underline{3\ 4\ 5}\ 2\ 1) \cdot \tau(1,2,5)\\
\pi^3 &= (\underline{3\ 4\ 5\ 6}\ \underline{2\ 1}) \cdot \tau(1,5,7)\\
\pi^4 &= (\underline{2}\ \underline{1}\ 3\ 4\ 5\ 6) \cdot \tau(1,2,3)\\
\pi^4 &= (\underline{1}\ \underline{2}\ 3\ 4\ 5\ 6) = \iota_6\\
d_t(\pi) &= 4
\end{aligned}
$$

$$
\begin{aligned}
\pi\ &= (6\ 3\ 5\ 2\ 4\ 1)\\
\pi^1 &= (6\ 3\ \underline{5\ 2}\ \underline{4}\ 1) \cdot \tau(3,5,6)\\
\pi^2 &= (6\ \underline{3\ 4\ 5}\ 2\ 1) \cdot \rho(2,4)\\
\pi^3 &= (\underline{6\ 5\ 4\ 3\ 2\ 1}) \cdot \rho(1,6)\\
\pi^3 &= (\underline{1\ 2\ 3\ 4\ 5\ 6}) = \iota_6\\
d_{rt}(\pi) &= 3
\end{aligned}
$$

Reversões Curtas e Super Curtas

k-reversão

Chamamos uma reversão $\rho(i, j)$ de k-reversão se $k = j - i + 1$.

Reversões Curtas

Uma k-reversão é dita curta se $k \leq 3$.
Exemplo: $\pi = (1\ \underline{2\ 3\ 4}\ 5) \cdot \rho(2, 4) = (1\ \underline{4\ 3\ 2}\ 5)$

Reversões Super Curtas

Uma k-reversão é dita super curta se $k \leq 2$.
Exemplo: $\pi = (1\ 2\ \underline{3\ 4}\ 5) \cdot \rho(3, 4) = (1\ 2\ \underline{4\ 3}\ 5)$

Reversões Curtas

Problema de Ordenação de Permutações sem Sinais por Reversões Curtas

- Complexidade desconhecida.
- Algoritmo 2-aproximado [Heath e Vergara, 2003].

Problema de Ordenação de Permutações com Sinais por Reversões Curtas

- Complexidade desconhecida.
- Algoritmo 5-aproximado [Galvão et al., 2015].

Reversões Super Curtas

Problema de Ordenação de Permutações sem Sinais por Reversões Super Curtas

- Algoritmo exato (Bubble Sort) [Knuth, 1973].
- Complexidade $\mathcal{O}(n^2)$.

Problema de Ordenação de Permutações com Sinais por Reversões Super Curtas

- Algoritmo exato [Galvão et al., 2015].
- Complexidade $\mathcal{O}(n^3)$.

Transposições Curtas e Super Curtas

(x, y)-transposição

Chamamos uma transposição $\tau(i, j, k)$ de (x, y)-transposição se $x = j - i$ e $y = k - j$.

Transposições Curtas

Uma (x, y)-transposição é dita curta se $x + y \leq 3$.
Exemplo: $\pi = (1\ \underline{2}\ \underline{3\ 4}\ 5) \cdot \tau(2, 3, 5) = (1\ \underline{3\ 4}\ \underline{2}\ 5)$

Transposições Super Curtas

Uma (x, y)-transposição é dita super curta se $x + y = 2$.
Exemplo: $\pi = (1\ 2\ \underline{3}\ \underline{4}\ 5) \cdot \tau(3, 4, 5) = (1\ 2\ \underline{4}\ \underline{3}\ 5)$

Transposições Curtas e Super Curtas

Problema de Ordenação de Permutações sem Sinais por Transposições Curtas

- Complexidade desconhecida.
- Algoritmo $\frac{5}{4}$-aproximado [Jiang et al., 2014].

Problema de Ordenação de Permutações sem Sinais por Transposições Super Curtas

- Algoritmo exato (Bubble Sort) [Knuth, 1973].
- Complexidade $\mathcal{O}(n^2)$.

Operações Curtas

Problema de Ordenação de Permutações sem sinais por Operações (Reversões e Transposições) Curtas

- Complexidade desconhecida.
- Algoritmo 2-aproximado [Vergara, 1998].

Problema de Ordenação de Permutações com sinais por Operações (Reversões e Transposições) Curtas

- Complexidade desconhecida.
- Algoritmo 3-aproximado [Galvão et al., 2015].

Operações Super Curtas

Problema de Ordenação de Permutações sem sinais por Operações (Reversões e Transposições) Super Curtas

- Algoritmo exato (Bubble Sort) [Knuth, 1973].
- Complexidade $\mathcal{O}(n^2)$.

Problema de Ordenação de Permutações com sinais por Operações (Reversões e Transposições) Super Curtas

- Algoritmo exato [Galvão et al., 2015].
- Complexidade $\mathcal{O}(n^3)$.

Estado da arte considerando permutações lineares

Modelo de Rearranjo	Sinais	Complexidade	Melhor Solução Teórica
Reversão	com	Polinomial	Algoritmo exato $\mathcal{O}(n^{\frac{3}{2}})$
	sem	NP-Difícil	Algoritmo 1.375-aproximado
Transposição	sem	NP-Difícil	Algoritmo 1.375-aproximado
Reversão e Transposição	com	Desconhecida	Algoritmo 2-aproximado
	sem	Desconhecida	Algoritmo 2k-aproximado
Reversão Curta	com	Desconhecida	Algoritmo 5-aproximado
	sem	Desconhecida	Algoritmo 2-aproximado
Transposição Curta	sem	Desconhecida	Algoritmo 5/4-aproximado
Reversão e Transposição Curta	com	Desconhecida	Algoritmo 3-aproximado
	sem	Desconhecida	Algoritmo 2-aproximado
Reversão Super Curta	com	Polinomial	Algoritmo exato $\mathcal{O}(n^3)$
	sem	Polinomial	Algoritmo exato $\mathcal{O}(n^2)$
Transposição Super Curta	sem	Polinomial	Algoritmo exato $\mathcal{O}(n^2)$
Reversão e Transposição Super Curta	com	Polinomial	Algoritmo exato $\mathcal{O}(n^3)$
	sem	Polinomial	Algoritmo exato $\mathcal{O}(n^2)$

Tabela: Estado da arte para diferentes tipos de modelos de rearranjo.

Breakpoints

Uma *permutação estendida* pode ser obtida a partir de π inserindo-se dois novos elementos: $\pi_0 = 0$ e $\pi_{n+1} = n + 1$.

Definição

Um breakpoint ocorre entre um par de elementos π_i e π_{i+1} de π, com $0 \leq i \leq n$, se $\pi_{i+1} - \pi_i \neq 1$. Exemplo:

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

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

Strips

Os breakpoints dividem uma permutação em strips, que são intervalos maximais sem breakpoints.

Não leva em consideração os elementos π_0 e π_{n+1}, que são inseridos na permutação estendida.

Exemplo

A permutação sem sinais $\pi = (0\ 1\ 2\ \cdot\ 5\ \cdot\ 4\ \cdot\ 3\ \cdot\ 6)$ apresenta as seguintes strips (1 2), (5), (4) e (3).

Permutação Reduzida

Uma permutação π pode ser transformada em uma permutação reduzida π_{red}, tal que $d_t(\pi) \leq d_t(\pi_{red})$.

Etapas

- Remover a primeira *strip*, se ela iniciar com 1.
- Remover a última *strip*, se ela terminar com n.
- Substituir cada *strip* pelo menor elemento contido nela.
- Renomear a sequência final de modo a obter uma permutação válida.

Permutação Reduzida

Exemplo

$\pi = (+1\ +2\ \cdot\ -9\ -8\ \cdot\ +5\ +6\ +7\ \cdot\ +3\ +4)$
$\pi' = (-9\ -8\ \cdot\ +5\ +6\ +7\ \cdot\ +3\ +4)$
$\pi'' = (-9\ \cdot\ +5\ \cdot\ +3)$
$\pi_{red} = (-3\ \cdot\ +2\ \cdot\ +1)$

$\pi_{red} \cdot \rho(2,2) \cdot \rho(3,3) \cdot \rho(1,3) = \bar{\iota}_3$
$d_{\overline{rt}}(\pi_{red}) = 3$
$\pi \cdot \rho(5,7) \cdot \rho(3,9) \cdot \rho(3,4) = \bar{\iota}_9$
$d_{\overline{rt}}(\pi) = 3$

Objetivos

Reversões e/ou Transposições Curtas

- Algoritmos aproximados.
- Classes de permutações.

Reversões e Transposições

- Permutações com sinais.
- *Look ahead* [Dias e Dias, 2010].
- *Sliding window* [Dias et al., 2014].

Objetivos

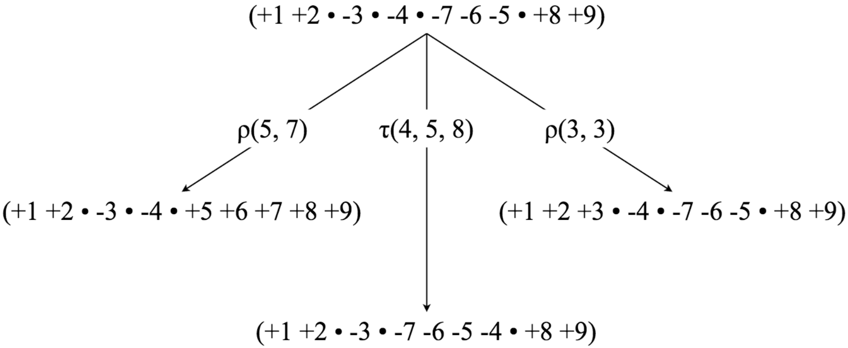

Figura: Heurística *look ahead*.

Objetivos

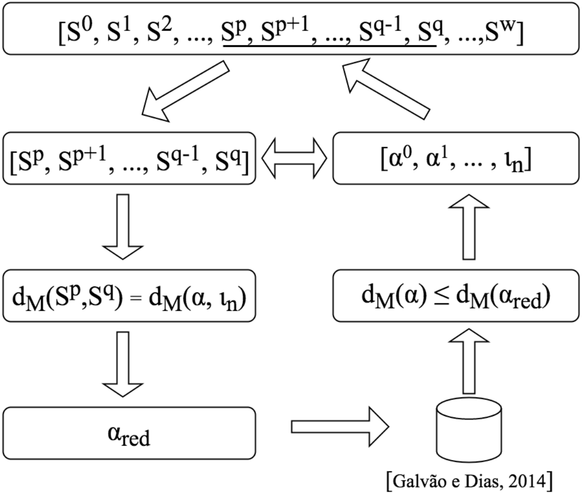

Figura: Heurística *sliding window*.

Metodologia

Reversões e/ou Transposições Curtas

- Foco teórico e prático.
- Estudo de teoremas e provas de trabalhos existentes na literatura.
- Propor um algoritmo aproximado.

Reversões e Transposições

- Foco prático.
- *Sliding window* em permutações com sinais.
- Aplicação do *look ahead* nos algoritmos aproximados existentes na literatura juntamente com o *sliding window*.

Cronograma

	2016										2017												2018	
	M	A	M	J	J	A	S	O	N	D	J	F	M	A	M	J	J	A	S	O	N	D	J	F
1	*	*	*	*		*	*	*	*															
2	*	*	*	*	*	*				*			*			*			*					
3					*	*																		
4								*																
5													*	*	*	*								
6										*	*	*												
7													*	*	*									
8																*	*	*						
9																			*	*	*			
10											*	*		*	*		*	*		*	*	*		
11																							*	
12																								*

1. Obtenção dos créditos obrigatórios em disciplinas do programa de mestrado.
2. Revisão da literatura.
3. Escrita da proposta de mestrado.
4. Exame de Qualificação de Mestrado (EQM).
5. Participação no Programa de Estágio Docente (PED).
6. Desenvolvimento da heurística *sliding window* e execução do experimento.

Cronograma

	2016										2017												2018	
	M	A	M	J	J	A	S	O	N	D	J	F	M	A	M	J	J	A	S	O	N	D	J	F
1	*	*	*	*		*	*	*	*															
2	*	*	*	*	*	*				*			*			*			*					
3					*	*																		
4								*																
5													*	*	*	*								
6										*	*	*												
7													*	*	*									
8																*	*	*						
9																			*	*	*			
10											*	*		*	*		*	*		*	*	*		
11																							*	
12																								*

7. Desenvolvimento da heurística *look ahead* e execução do experimento.
8. Investigação de problemas de ordenação de permutações sem sinais utilizando operações curtas.
9. Investigação de problemas de ordenação de permutações com sinais utilizando operações curtas.
10. Escrita da dissertação.
11. Revisão da dissertação.
12. Defesa da dissertação.

Referências

Alberto Caprara (1999)
Sorting Permutations by Reversals and Eulerian Cycle Decompositions
SIAM Journal on Discrete Mathematics 12(1), 91 – 110.

Sridhar Hannenhalli and Pavel A. Pevzner (1999)
Transforming Cabbage into Turnip: Polynomial Algorithm for Sorting Signed Permutations by Reversals
Journal of the ACM 46(1), 1 – 27.

Laurent Bulteau and Guillaume Fertin and Irena Rusu (2012)
Sorting by Transpositions is Difficult
SIAM Journal on Computing 3(26), 1148 – 1180.

Maria E. M. T. Walter and Zanoni Dias and João Meidanis (1998)
Reversal and Transposition Distance of Linear Chromosomes
IEEE Computer Society, 96 – 102.

Referências

Isaac Elias and Tzvika Hartman (2006)
A 1.375-Approximation Algorithm for Sorting by Transpositions
IEEE/ACM Transactions on Computational Biology and Bioinformatics 3(4), 369 – 379.

Piotr Berman and Sridhar Hannenhalli and Marek Karpinski (2002)
1.375-Approximation Algorithm for Sorting by Reversals
10th Annual European Symposium on Algorithms (ESA'2002), 200 – 210.

Gustavo R. Galvão and Orlando Lee and Zanoni Dias (2015)
Sorting Signed Permutations by Short Operations
Algorithms for Molecular Biology (10)1, 1 – 17.

Lenwood S. Heath and John Paul C. Vergara (2003)
Sorting by Short Swaps
Journal of Computational Biology (10)5, 775 – 789.

Referências

John Paul C. Vergara (1998)
Sorting by Bounded Permutations
Virginia Polytechnic Institute and State University, 1998.

Mark R. Jerrum (1985)
The Complexity of Finding Minimum-length Generator Sequences
Theoretical Computer Science (36)2-3, 265 – 289.

Ulisses Dias and Zanoni Dias (2010)
Extending Bafna-Pevzner Algorithm
ISB'2010, 1 – 8.

Ulisses Dias and Gustavo R. Galvão and Carla N. Lintzmayer and Zanoni Dias (2014)
A General Heuristic for Genome Rearrangement Problems
Journal of Bioinformatics and Computational Biology (12)3, 26.

Referências

Atif Rahman and Swakkhar Shatabda and Masud Hasan (2008)
An Approximation Algorithm for Sorting by Reversals and Transpositions
Journal of Discrete Algorithms 6(3), 449 – 457.

Haitao Jiang and Haodi Feng and Daming Zhu (2014)
An 5/4-Approximation Algorithm for Sorting Permutations by Short Block Moves
ISAAC'2014, 491 – 503.

Donald E. Knuth (1973)
Fundamental Algorithms: The Art of Computer Programming
1973.

Gustavo R. Galvão and Zanoni Dias (2014)
An Audit Tool for Genome Rearrangement Algorithms
Journal of Experimental Algorithmics (19), 1–34.

Heurísticas para o Problema de Ordenação de Permutações com Sinais por Reversões e Transposições

Klairton de Lima Brito

Orientador: Prof. Dr. Zanoni Dias

Universidade Estadual de Campinas

29 de Setembro de 2016