Problema de Distância de Reversão Quase-Simétrica

Thiago da Silva Arruda
Zanoni Dias

Instituto de Computação - Universidade Estadual de Campinas

19 de Setembro de 2012

Agenda

Rearranjo de Genomas

- **Problema**: determinar a distância evolutiva entre indivíduos.
- **Metodologia**: identificar uma sequência de eventos de mutação que explica as diferenças genéticas entre indivíduos.
- Os eventos considerados afetam grandes partes de um genoma.

Rearranjo de Genomas

- **Problema**: determinar a distância evolutiva entre indivíduos.
- **Metodologia**: identificar uma sequência de eventos de mutação que explica as diferenças genéticas entre indivíduos.
- Os eventos considerados afetam grandes partes de um genoma.

Rearranjo de Genomas

- **Problema**: determinar a distância evolutiva entre indivíduos.
- **Metodologia**: identificar uma sequência de eventos de mutação que explica as diferenças genéticas entre indivíduos.
- Os eventos considerados afetam grandes partes de um genoma.

Representação de Genomas

- Um genoma com n genes é classicamente representado como uma **n-tupla** de números inteiros.
- Caso não haja repetição, então esta *n-tupla* pode ser representada como uma **permutação** $\pi = (\pi_1\, \pi_2\, \pi_3 \ldots \pi_n)$, $1 \leq |\pi_i| \leq n$, $|\pi_i| \neq |\pi_j| \leftrightarrow i \neq j$.
- Para alguns modelos de rearranjo, a **orientação** dos genes é considerada, o que é representado pela atribuição de **sinais** aos elementos da permutação.

Representação de Genomas

- Um genoma com n genes é classicamente representado como uma **n-tupla** de números inteiros.
- Caso não haja repetição, então esta *n-tupla* pode ser representada como uma **permutação** $\pi = (\pi_1\, \pi_2\, \pi_3 \ldots \pi_n)$, $1 \leq |\pi_i| \leq n$, $|\pi_i| \neq |\pi_j| \leftrightarrow i \neq j$.
- Para alguns modelos de rearranjo, a **orientação** dos genes é considerada, o que é representado pela atribuição de **sinais** aos elementos da permutação.

Representação de Genomas

- Um genoma com n genes é classicamente representado como uma **n-tupla** de números inteiros.
- Caso não haja repetição, então esta *n-tupla* pode ser representada como uma **permutação** $\pi = (\pi_1\, \pi_2\, \pi_3 \ldots \pi_n)$, $1 \leq |\pi_i| \leq n$, $|\pi_i| \neq |\pi_j| \leftrightarrow i \neq j$.
- Para alguns modelos de rearranjo, a **orientação** dos genes é considerada, o que é representado pela atribuição de **sinais** aos elementos da permutação.

Distância de Rearranjo

- Um modelo de rearanjo de genomas é composto por um conjunto de **operações**.
- A distância de rearranjo entre duas permutações π e σ é o número **mínimo** t de operações ρ_i necessário para transformar π em σ:

$$\rho_t(\ldots(\rho_2(\rho_1\pi))) = \sigma.$$

$$d(\pi, \sigma) = t.$$

Distância de Rearranjo

- Um modelo de rearanjo de genomas é composto por um conjunto de **operações**.
- A distância de rearranjo entre duas permutações π e σ é o número **mínimo** t de operações ρ_i necessário para transformar π em σ:

$$\rho_t\,(\,\ldots(\rho_2(\,\rho_1\pi))) = \sigma.$$

$$d(\pi, \sigma) = t.$$

Ordenação de Genomas

- A permutação identidade ι é definida como $(1\,2\,3\,\ldots\,n)$.
- A **ordenação** de uma permutação π consiste no processo de transformar π na permutação identidade.
- A **distância** de ordenação de π é denotada por $d(\pi, \iota) = d(\pi)$.

Ordenação de Genomas

- A permutação identidade ι é definida como $(1\,2\,3\,\ldots\,n)$.
- A **ordenação** de uma permutação π consiste no processo de transformar π na permutação identidade.
- A **distância** de ordenação de π é denotada por $d(\pi, \iota) = d(\pi)$.

Ordenação de Genomas

- A permutação identidade ι é definida como $(1\,2\,3\,\ldots\,n)$.
- A **ordenação** de uma permutação π consiste no processo de transformar π na permutação identidade.
- A **distância** de ordenação de π é denotada por $d(\pi, \iota) = d(\pi)$.

Reversão Genomas

- Reversão é uma operação que consiste em inverter a **ordem** e os **sinais** de elementos de um segmento da permutação.
- Estado da Arte:
  - **Com sinal:** Algoritmo exato de complexidade sub-quadrática [6].
  - **Sem sinal:** NP-Difícil [2], melhor algoritmo conhecido possui fator de aproximação 1,375 [1].

Reversão Genomas

- Reversão é uma operação que consiste em inverter a **ordem** e os **sinais** de elementos de um segmento da permutação.
- Estado da Arte:
  - **Com sinal:** Algoritmo exato de complexidade sub-quadrática [6].
  - **Sem sinal:** NP-Difícil [2], melhor algoritmo conhecido possui fator de aproximação 1,375 [1].

Reversão Genomas

- Reversão é uma operação que consiste em inverter a **ordem** e os **sinais** de elementos de um segmento da permutação.
- Estado da Arte:
  - **Com sinal:** Algoritmo exato de complexidade sub-quadrática [6].
  - **Sem sinal:** NP-Difícil [2], melhor algoritmo conhecido possui fator de aproximação 1,375 [1].

Reversão Genomas

- Reversão é uma operação que consiste em inverter a **ordem** e os **sinais** de elementos de um segmento da permutação.
- Estado da Arte:
  - **Com sinal:** Algoritmo exato de complexidade sub-quadrática [6].
  - **Sem sinal:** NP-Difícil [2], melhor algoritmo conhecido possui fator de aproximação 1,375 [1].

Reversão Genomas

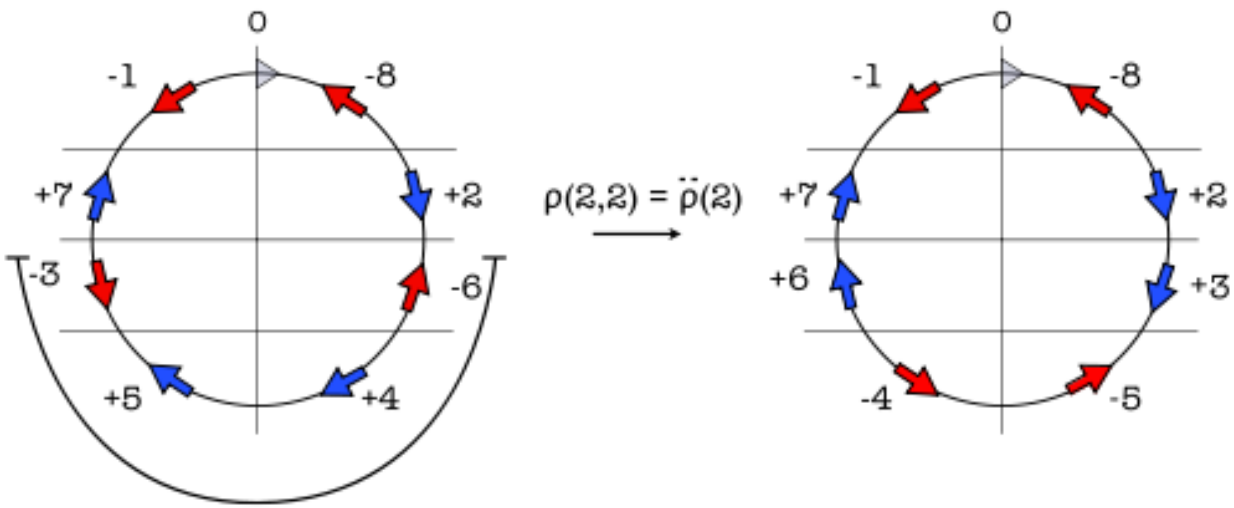


Figura: Operação de reversão [3]

Reversão Simétrica

- Estudos sugerem que, para certos grupos de organismos, os eventos de reversão ocorridos possuem alto grau de **simetria** [4].

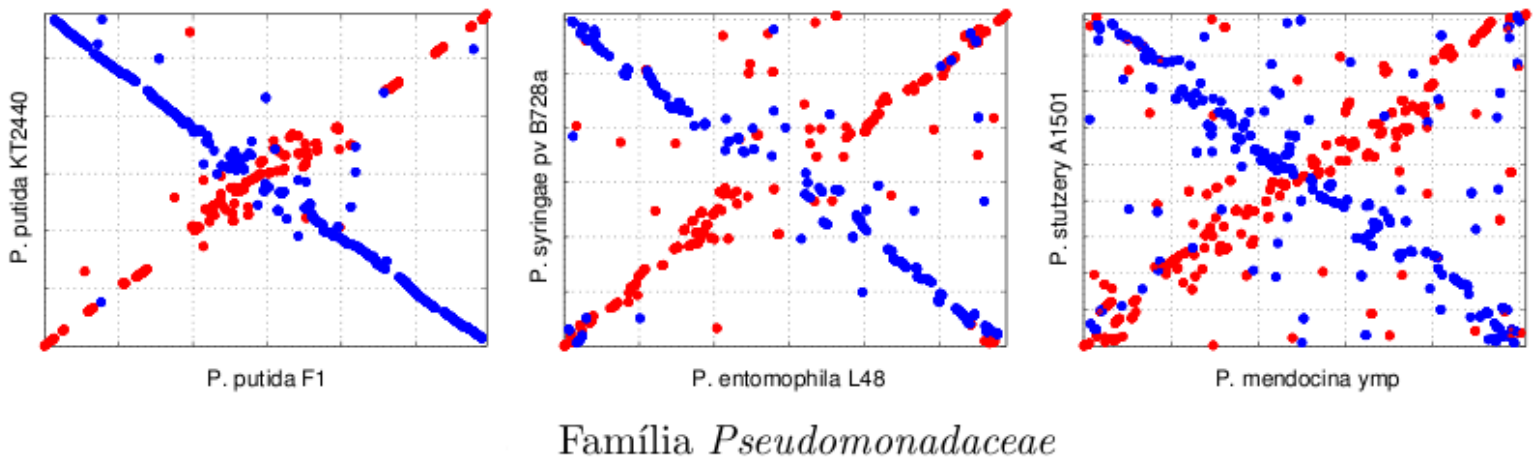


Figura: Alinhamento de genomas de algumas espécies de bactérias [3] .

Propriedades de Reversão Simétrica

- **Position:** $p(\pi, i) = k \leftrightarrow |\pi[k]| = i,\ p(\pi, i) \in \{1, 2, ..., n\}$.
- **Sign:** $s(\pi, i) = k \leftrightarrow \pi[p(\pi, i)] = ki,\ s(\pi, i) \in \{1, -1\}$.
- **Slice:** $slice(\pi, i) = min\{p(\pi, i), n - p(\pi, i) + 1\}$, $slice(\pi, i) \in \{1, 2, ..., \frac{n}{2}\}$.

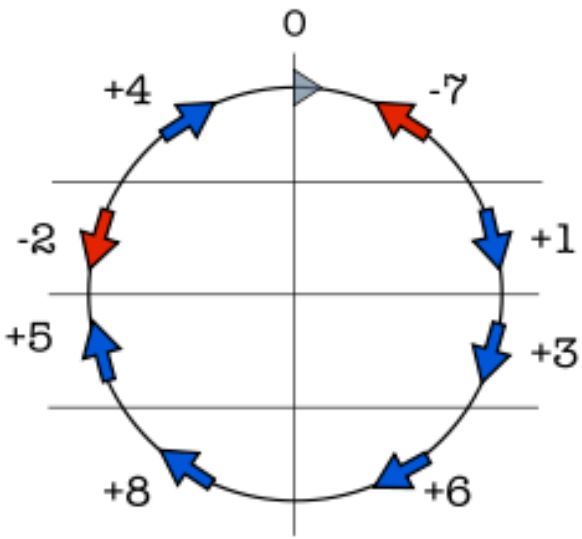
Propriedades de Reversão Simétrica

- **Position:** $p(\pi, i) = k \leftrightarrow |\pi[k]| = i,\ p(\pi, i) \in \{1, 2, ..., n\}$.
- **Sign:** $s(\pi, i) = k \leftrightarrow \pi[p(\pi, i)] = ki,\ s(\pi, i) \in \{1, -1\}$.
- **Slice:** $slice(\pi, i) = min\{p(\pi, i), n - p(\pi, i) + 1\}$, $slice(\pi, i) \in \{1, 2, ..., \frac{n}{2}\}$.

Propriedades de Reversão Simétrica

- **Position:** $p(\pi, i) = k \leftrightarrow |\pi[k]| = i,\ \ p(\pi, i) \in \{1, 2, ..., n\}$.
- **Sign:** $s(\pi, i) = k \leftrightarrow \pi[p(\pi, i)] = ki,\ \ s(\pi, i) \in \{1, -1\}$.
- **Slice:** $slice(\pi, i) = min\{p(\pi, i), n - p(\pi, i) + 1\}$, $slice(\pi, i) \in \{1, 2, ..., \frac{n}{2}\}$.

Propriedades de Reversão Simétrica



i	p	s	slice
1	2	+1	2
2	7	-1	2
3	3	+1	3
4	8	+1	1
5	6	+1	3
6	4	+1	4
7	1	-1	1
8	5	+1	4

Figura: Exemplo das funções position(p), sign(s) e slice [3] .

Modelo para Reversão Simétrica

- Utilizar o modelo de Reversão para calcular distância de rearranjo conduz a resultados incorretos [3].
- Ohlebusch [5] propos o modelo que permite somente operações de Reversão Simétrica.
- Definição formal:
  $\ddot{\rho}(i) = \rho(i+1, n-i), \;\; 0 \leq i \leq \left\lceil \frac{n}{2} \right\rceil$.
- Para transformar uma permutação π em outra permutação σ é necessário que $slice(\pi, i) = slice(\sigma, i)$, para todo $1 \leq i \leq n$.
- Apenas $2^{\lceil \frac{n}{2} \rceil}$ das $2^n n!$ permutações com sinal de tamanho n podem ser ordenadas.

Modelo para Reversão Simétrica

- Utilizar o modelo de Reversão para calcular distância de rearranjo conduz a resultados incorretos [3].
- Ohlebusch [5] propos o modelo que permite somente operações de Reversão Simétrica.
- Definição formal:
  $\ddot{\rho}(i) = \rho(i+1, n-i), \;\; 0 \leq i \leq \lceil \frac{n}{2} \rceil$.
- Para transformar uma permutação π em outra permutação σ é necessário que $slice(\pi, i) = slice(\sigma, i)$, para todo $1 \leq i \leq n$.
- Apenas $2^{\lceil \frac{n}{2} \rceil}$ das $2^n n!$ permutações com sinal de tamanho n podem ser ordenadas.

Modelo para Reversão Simétrica

- Utilizar o modelo de Reversão para calcular distância de rearranjo conduz a resultados incorretos [3].
- Ohlebusch [5] propos o modelo que permite somente operações de Reversão Simétrica.
- Definição formal:
  $\ddot{\rho}(i) = \rho(i+1, n-i), \;\; 0 \leq i \leq \lceil \frac{n}{2} \rceil$.
- Para transformar uma permutação π em outra permutação σ é necessário que $slice(\pi, i) = slice(\sigma, i)$, para todo $1 \leq i \leq n$.
- Apenas $2^{\lceil \frac{n}{2} \rceil}$ das $2^n n!$ permutações com sinal de tamanho n podem ser ordenadas.

Modelo para Reversão Simétrica

- Utilizar o modelo de Reversão para calcular distância de rearranjo conduz a resultados incorretos [3].
- Ohlebusch [5] propos o modelo que permite somente operações de Reversão Simétrica.
- Definição formal:
  $\ddot{\rho}(i) = \rho(i+1, n-i), \;\; 0 \leq i \leq \lceil \frac{n}{2} \rceil$.
- Para transformar uma permutação π em outra permutação σ é necessário que $\mathit{slice}(\pi, i) = \mathit{slice}(\sigma, i)$, para todo $1 \leq i \leq n$.
- Apenas $2^{\lceil \frac{n}{2} \rceil}$ das $2^n n!$ permutações com sinal de tamanho n podem ser ordenadas.

Modelo para Reversão Simétrica

- Utilizar o modelo de Reversão para calcular distância de rearranjo conduz a resultados incorretos [3].
- Ohlebusch [5] propos o modelo que permite somente operações de Reversão Simétrica.
- Definição formal:
  $\ddot{\rho}(i) = \rho(i+1, n-i), \;\; 0 \leq i \leq \lceil \frac{n}{2} \rceil$.
- Para transformar uma permutação π em outra permutação σ é necessário que $slice(\pi, i) = slice(\sigma, i)$, para todo $1 \leq i \leq n$.
- Apenas $2^{\lceil \frac{n}{2} \rceil}$ das $2^n n!$ permutações com sinal de tamanho n podem ser ordenadas.

Reversão Quase-Simétrica

- No trabalho de Dias e coautores [3] foi apresentado o modelo de **Reversão Quase-Simétrica**.
- Permite reversões com assimetria de no máximo uma unidade.
- Com este modelo é possível transformar uma permutação π de tamanho n em qualquer outra permutação σ, também de tamanho n.
- Definição formal:
  $\overline{\rho}(i,j) = \rho(i+1, n-j), \;\; 0 \leq i, \;\; j \leq n-1, \;\; |i-j| \;\; \leq 1$

Reversão Quase-Simétrica

- No trabalho de Dias e coautores [3] foi apresentado o modelo de **Reversão Quase-Simétrica**.
- Permite reversões com assimetria de no máximo uma unidade.
- Com este modelo é possível transformar uma permutação π de tamanho n em qualquer outra permutação σ, também de tamanho n.
- Definição formal:
  $\overline{\rho}(i,j) = \rho(i+1, n-j), \;\; 0 \leq i, \;\; j \leq n-1, \;\; |i-j| \;\; \leq 1$

Reversão Quase-Simétrica

- No trabalho de Dias e coautores [3] foi apresentado o modelo de **Reversão Quase-Simétrica**.
- Permite reversões com assimetria de no máximo uma unidade.
- Com este modelo é possível transformar uma permutação π de tamanho n em qualquer outra permutação σ, também de tamanho n.
- Definição formal:
  $\overline{\rho}(i,j) = \rho(i+1, n-j), \;\; 0 \leq i, \;\; j \leq n-1, \;\; |i-j| \;\; \leq 1$

Reversão Quase-Simétrica

- No trabalho de Dias e coautores [3] foi apresentado o modelo de **Reversão Quase-Simétrica**.
- Permite reversões com assimetria de no máximo uma unidade.
- Com este modelo é possível transformar uma permutação π de tamanho n em qualquer outra permutação σ, também de tamanho n.
- Definição formal:
  $\overline{\rho}(i,j) = \rho(i+1, n-j), \;\; 0 \leq i, \;\; j \leq n-1, \;\; |i-j| \;\; \leq 1$

Estado da Arte

- Ainda não se conhece a complexidade do problema.
- O estado da arte compreende:
  - Limitantes
  - Algoritmos
  - Classes e Famílias de permutações

Estado da Arte

- Ainda não se conhece a complexidade do problema.
- O estado da arte compreende:
  - Limitantes
  - Algoritmos
  - Classes e Famílias de permutações

Estado da Arte

- Ainda não se conhece a complexidade do problema.
- O estado da arte compreende:
  - Limitantes
  - Algoritmos
  - Classes e Famílias de permutações

Estado da Arte

- Ainda não se conhece a complexidade do problema.
- O estado da arte compreende:
  - Limitantes
  - Algoritmos
  - Classes e Famílias de permutações

Limitantes

Definição

O número mínimo de operações de reversão quase-simétricas que devem **agir** *em um elemento i para posicioná-lo adequadamente na permutação π é definido por $d_\pi[i]$.*

Lema

Se $slice(\pi, i) = j$ e $slice(\iota, i) = k$, então o total de $d_\pi[i] \geq |j - k|$ reversões quase-simétricas devem agir no elemento i para posicioná-lo adequadamente.

Limitantes

Definição

O número mínimo de operações de reversão quase-simétricas que devem **agir** *em um elemento i para posicioná-lo adequadamente na permutação π é definido por $d_\pi[i]$.*

Lema

Se $slice(\pi, i) = j$ e $slice(\iota, i) = k$, então o total de $d_\pi[i] \geq |j - k|$ reversões quase-simétricas devem agir no elemento i para posicioná-lo adequadamente.

Limitantes

Lema

Quando π possui tamanho **par**:
$d_\pi[i] = |\frac{n}{2} - p(\pi, i)| + |\frac{n}{2} - i| + 1.$

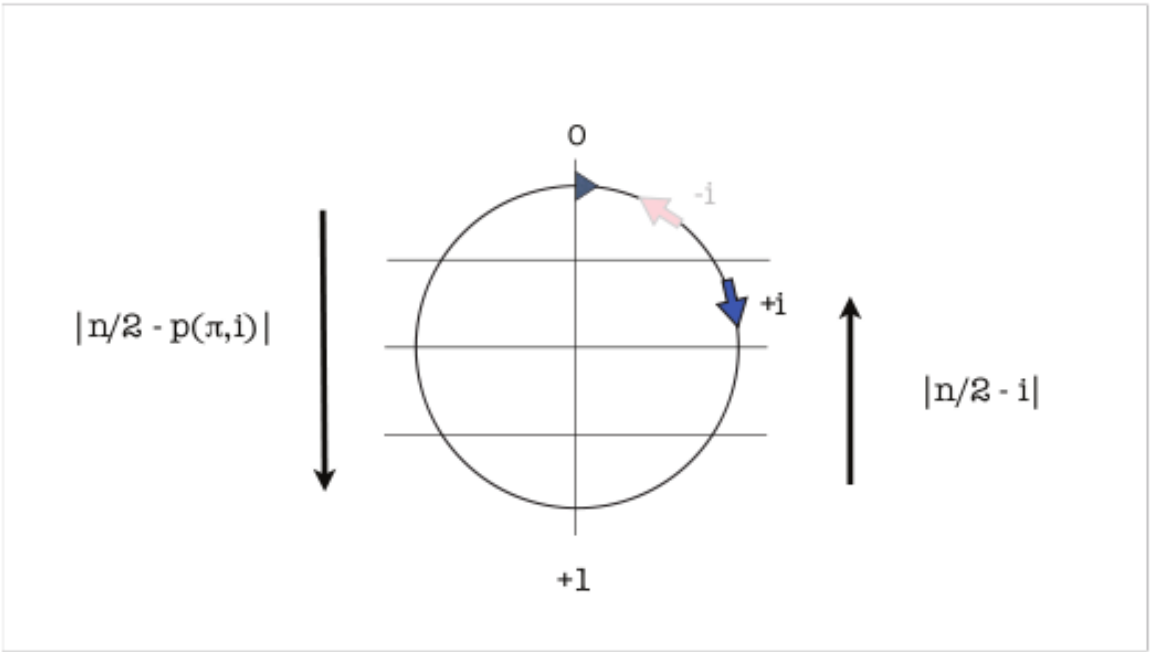


Figura: Posicionamento de um elemento, caso par[3] .

Limitantes

Lema

Quando π *possui tamanho* **impar**:
$d_\pi[i] = |\frac{n+1}{2} - p(\pi, i)| + |\frac{n+1}{2} - i|$.

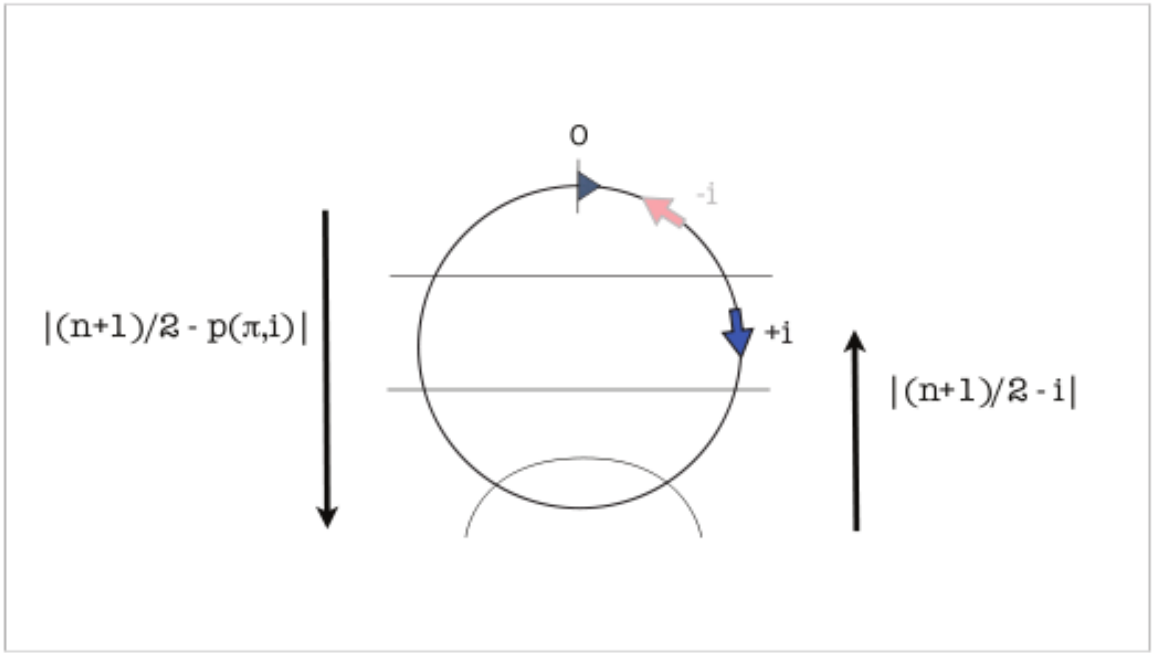


Figura: Posicionamento de um elemento, caso impar [3] .

Limitantes

Definição

$Lower(\pi) = max_{1 \leq i \leq n} d_\pi[i]$

Definição

$Upper(\pi) = \sum_{1 \leq i \leq n} d_\pi[i]$

- A função *Lower* é um limitante inferior para o problema de Reversão Quase-Simétrica.
- Não é possível afirmar que a função *Upper* é um limitante superior.

Limitantes

Definição

$Lower(\pi) = max_{1 \leq i \leq n} d_{\pi}[i]$

Definição

$Upper(\pi) = \sum_{1 \leq i \leq n} d_{\pi}[i]$

- A função *Lower* é um limitante inferior para o problema de Reversão Quase-Simétrica.
- Não é possível afirmar que a função *Upper* é um limitante superior.

Limitantes

Definição

$Lower(\pi) = max_{1 \leq i \leq n} d_\pi[i]$

Definição

$Upper(\pi) = \sum_{1 \leq i \leq n} d_\pi[i]$

- A função *Lower* é um limitante inferior para o problema de Reversão Quase-Simétrica.
- Não é possível afirmar que a função *Upper* é um limitante superior.

Limitantes

Definição

$Lower(\pi) = max_{1 \leq i \leq n} d_\pi[i]$

Definição

$Upper(\pi) = \sum_{1 \leq i \leq n} d_\pi[i]$

- A função *Lower* é um limitante inferior para o problema de Reversão Quase-Simétrica.
- Não é possível afirmar que a função *Upper* é um limitante superior.

Limitantes

Definição

$Lower(\pi) = max_{1 \leq i \leq n} d_\pi[i]$

Definição

$Upper(\pi) = \sum_{1 \leq i \leq n} d_\pi[i]$

- A função *Lower* é um limitante inferior para o problema de Reversão Quase-Simétrica.
- Não é possível afirmar que a função *Upper* é um limitante superior.

Algoritmos

- Foram apresentados alguns algoritmos iniciais para o problema [3].
- Estes algoritmos são heurísticas que não possuem prova de fator de aproximação.
- Estes algoritmos foram avaliados experimentalmente com todas as permutações de tamanho até 9.

Algoritmos

- Foram apresentados alguns algoritmos iniciais para o problema [3].
- Estes algoritmos são heurísticas que não possuem prova de fator de aproximação.
- Estes algoritmos foram avaliados experimentalmente com todas as permutações de tamanho até 9.

Algoritmos

- Foram apresentados alguns algoritmos iniciais para o problema [3].
- Estes algoritmos são heurísticas que não possuem prova de fator de aproximação.
- Estes algoritmos foram avaliados experimentalmente com todas as permutações de tamanho até 9.

Algoritmos

	Algoritmo Básico				
	Operações		Razão		Exatos
N	MAX	AVG	MAX	AVG	
2	3	1,50	1,00	1,00	100,00%
3	7	3,50	2,50	1,18	66,67%
4	10	5,33	3,00	1,25	48,96%
5	17	8,50	4,50	1,50	16,98%
6	21	11,23	5,00	1,60	8,20%
7	31	15,50	6,50	1,84	1,87%
8	36	19,16	7,00	1,96	0,64%
9	49	24,50	8,50	2,20	0,12%

Tabela: Performance do Algoritmo 1 [3] .

Algoritmos

	Algoritmo Básico com Otimização de Slice				
	Operações		Razão		Exatos
N	**MAX**	**AVG**	**MAX**	**AVG**	
2	3	1,50	1,00	1,00	100,00%
3	7	3,33	2,00	1,13	75,00%
4	10	5,08	2,33	1,19	56,77%
5	17	7,86	2,83	1,38	23,36%
6	21	10,55	3,50	1,50	11,78%
7	31	14,22	4,50	1,69	3,12%
8	36	17,87	5,00	1,83	1,13%
9	49	22,42	6,25	2,01	0,23%

Tabela: Performance do Algoritmo 2 [3] .

Algoritmos

	Algoritmo Guloso				
	Operações		Razão		Exatos
N	MAX	AVG	MAX	AVG	
2	3	1,50	1,00	1,00	100,00%
3	5	3,08	1,25	1,04	87,50%
4	8	4,59	1,75	1,08	76,30%
5	12	6,85	2,75	1,21	41,43%
6	15	8,89	2,75	1,27	25,88%
7	21	11,70	3,80	1,39	10,26%
8	25	14,10	3,60	1,44	5,18%
9	32	17,41	4,75	1,57	1,62%

Tabela: Performance do Algoritmo 3 [3].

Famílias e Classes de Permutações

- Ainda não existe algoritmo de ordenação exato para o Problema de Distância de Reversão Quase-Simétrica.
- Para estudar o problema, pode-se definir famílias e classes de permutações [3].
- Para cada família ou classe, pode se **conjecturar** a distância exata correspondente.
- Um algoritmo de ordenação especifico pode ser elaborado para cada família ou classe.

Famílias e Classes de Permutações

- Ainda não existe algoritmo de ordenação exato para o Problema de Distância de Reversão Quase-Simétrica.
- Para estudar o problema, pode-se definir famílias e classes de permutações [3].
- Para cada família ou classe, pode se **conjecturar** a distância exata correspondente.
- Um algoritmo de ordenação especifico pode ser elaborado para cada família ou classe.

Famílias e Classes de Permutações

- Ainda não existe algoritmo de ordenação exato para o Problema de Distância de Reversão Quase-Simétrica.
- Para estudar o problema, pode-se definir famílias e classes de permutações [3].
- Para cada família ou classe, pode se **conjecturar** a distância exata correspondente.
- Um algoritmo de ordenação especifico pode ser elaborado para cada família ou classe.

Famílias e Classes de Permutações

- Ainda não existe algoritmo de ordenação exato para o Problema de Distância de Reversão Quase-Simétrica.
- Para estudar o problema, pode-se definir famílias e classes de permutações [3].
- Para cada família ou classe, pode se **conjecturar** a distância exata correspondente.
- Um algoritmo de ordenação especifico pode ser elaborado para cada família ou classe.

Famílias de Permutações

- Uma família é um grupo de permutações que compartilham uma mesma regra de definição.
- Para uma família, existe uma permutação de tamanho n, para todo número natural n.
- Para cada familia $F_k(n)$ foi conjecturado que $d(F_k(n))$ corresponde à distância exata [3].
- As conjecturas foram validadas para todas permutações π, tais que $|\pi| \leq 10$.

Famílias de Permutações

- Uma família é um grupo de permutações que compartilham uma mesma regra de definição.
- Para uma família, existe uma permutação de tamanho n, para todo número natural n.
- Para cada familia $F_k(n)$ foi conjecturado que $d(F_k(n))$ corresponde à distância exata [3].
- As conjecturas foram validadas para todas permutações π, tais que $|\pi| \leq 10$.

Famílias de Permutações

- Uma família é um grupo de permutações que compartilham uma mesma regra de definição.
- Para uma família, existe uma permutação de tamanho n, para todo número natural n.
- Para cada familia $F_k(n)$ foi conjecturado que $d(F_k(n))$ corresponde à distância exata [3].
- As conjecturas foram validadas para todas permutações π, tais que $|\pi| \leq 10$.

Famílias de Permutações

- Uma família é um grupo de permutações que compartilham uma mesma regra de definição.
- Para uma família, existe uma permutação de tamanho n, para todo número natural n.
- Para cada familia $F_k(n)$ foi conjecturado que $d(F_k(n))$ corresponde à distância exata [3].
- As conjecturas foram validadas para todas permutações π, tais que $|\pi| \leq 10$.

Famílias de Permutações

- Para cada família $F_k(n)$, conjectura-se que a distância $d(F_k(n))$ corresponde à distância exata.

Família de Permutação	**Distância**
$F_1(n) = [-1, +2, +3, \ldots, +n]$	$d(F_1(n)) = 2\lfloor \frac{n-1}{2} \rfloor + 1$, para $n \geq 1$.
$F_2(n) = [+(n-1), \ldots, +2, +1, -n]$	$d(F_2(n)) = 2\lfloor \frac{n-1}{2} \rfloor + 1$, para $n \geq 2$.
$F_3(n) = [+n, -1, -2, \ldots, -(n-1)]$	$d(F_3(n)) = 2\lceil \frac{n-1}{2} \rceil$, para $n \geq 3$.
$F_4(n) = [-1, -2, \ldots, -(n-1), -n]$	$d(F_4(n)) = 2\lceil \frac{n}{2} \rceil$, para $n \geq 2$.
$F_5(n) = [+n, +(n-1), \ldots, +2, +1]$	$d(F_5(n)) = 2\lfloor \frac{n}{2} \rfloor + 1$, para $n \geq 2$.
$F_6(n) = [+1, -2, \ldots, -(n-1), -n]$	$d(F_6(n)) = 2\lceil \frac{n}{2} \rceil + 1$, para $n \geq 5$.
$F_7(n) = [+n, +(n-1), \ldots, +2, -1]$	$d(F_7(n)) = 2\lfloor \frac{n}{2} \rfloor + 2$, para $n \geq 5$.
$F_8(n) = [-1, +2, \ldots, +(n-1), -n]$	$d(F_8(n)) = n + 2$, para $n \geq 5$.
$F_9(n) = [+n, -1, +(n-2), -3, \ldots, +2, -(n-1)]$ (n par)	$d(F_9(n)) = \lceil \frac{3n}{2} \rceil - 1$, para $n \geq 4$.
$F_9(n) = [+n, -2, +(n-2), -4, \ldots, -(n-1), +1]$ (n ímpar)	$d(F_9(n)) = \lceil \frac{3n}{2} \rceil - 1$, para $n \geq 4$.
$F_{10}(n) = [-1, +2, -3, +4, \ldots, (-1)^n n]$	$d(F_{10}(n)) = \lceil \frac{3n}{2} \rceil$, para $n \geq 5$.

Classes de Permutações

- Classes são grupos que podem conter mais de uma permutação do mesmo tamanho n.
- Classes podem conter famílias de permutações.
- Foram definidas classes de permutações C_k [3].
- Para cada classe $C_k(n)$ foi conjecturado uma distância $d(C_k(n))$ [3].
- Estas conjecturas foram validadas, usando o **algoritmo** correspondente, para todas permutações π, tais que $|\pi| \leq 10$.

Classes de Permutações

- Classes são grupos que podem conter mais de uma permutação do mesmo tamanho n.
- Classes podem conter famílias de permutações.
- Foram definidas classes de permutações C_k [3].
- Para cada classe $C_k(n)$ foi conjecturado uma distância $d(C_k(n))$ [3].
- Estas conjecturas foram validadas, usando o **algoritmo** correspondente, para todas permutações π, tais que $|\pi| \leq 10$.

Classes de Permutações

- Classes são grupos que podem conter mais de uma permutação do mesmo tamanho n.
- Classes podem conter famílias de permutações.
- Foram definidas classes de permutações C_k [3].
- Para cada classe $C_k(n)$ foi conjecturado uma distância $d(C_k(n))$ [3].
- Estas conjecturas foram validadas, usando o **algoritmo** correspondente, para todas permutações π, tais que $|\pi| \leq 10$.

Classes de Permutações

- Classes são grupos que podem conter mais de uma permutação do mesmo tamanho n.
- Classes podem conter famílias de permutações.
- Foram definidas classes de permutações C_k [3].
- Para cada classe $C_k(n)$ foi conjecturado uma distância $d(C_k(n))$ [3].
- Estas conjecturas foram validadas, usando o **algoritmo** correspondente, para todas permutações π, tais que $|\pi| \leq 10$.

Classes de Permutações

- Classes são grupos que podem conter mais de uma permutação do mesmo tamanho n.
- Classes podem conter famílias de permutações.
- Foram definidas classes de permutações C_k [3].
- Para cada classe $C_k(n)$ foi conjecturado uma distância $d(C_k(n))$ [3].
- Estas conjecturas foram validadas, usando o **algoritmo** correspondente, para todas permutações π, tais que $|\pi| \leq 10$.

Classes de Permutações

Classes de permutações que foram definidas em Dias e coautores [3].

- $C_1(n,k) = \rho(1,k) \cdot \iota$, *para* $1 \leq k \leq n$.
- $C_2(n,i,j) = \rho(i,j) \cdot \iota$, *para* $1 \leq i \leq j \leq n$.
- $C_3(n,i,j) = \{\rho(1,k) \cdot \rho(k+1,n) \cdot \rho(1,n) \cdot \iota$, *para* $1 \leq k \leq n\}$.
- $C_4(n) = \{\rho(i_1,i_1) \cdot \rho(i_2,i_2)) \cdot \ldots \cdot \rho(i_r,i_r) \cdot \iota$, *para* $1 \leq i_k \leq n$, $1 \leq k \leq r,\ i_p \neq i_q, 1 \leq p \leq q \leq r,\ 1 \leq r \leq n\}$
- $C_5(n) = \{\rho(i_1,j_1) \cdot \rho(i_2,j_2)) \cdot \ldots \cdot \rho(i_r,j_r) \cdot \iota$, *para* $1 \leq i_k \leq j_k \leq n, 1 \leq k \leq r,\ [i_p,j_p] \cap [i_q,j_q] = \emptyset,$ $1 \leq p \leq q \leq r, 1 \leq r \leq n\}$

Classes de Permutações

Conjectura

Distância para as classes de permutação.

- $d(C_1(n,k)) = 2\lfloor \frac{n-k}{2} \rfloor + 1$, *para* $1 \leq k \leq n$.
- $d(C_2(n,i,j)) = 2\lfloor \frac{|n-i-j+1|}{2} \rfloor + 1$, *para* $1 \leq i \leq j \leq n$.
- $d(C_3(n,i,j)) \leq 2\lfloor \frac{n-k}{2} \rfloor + 2\lfloor \frac{k}{2} \rfloor + 3$, *para* $1 \leq k \leq n$.
- $d(C_4(n)) \leq 2n$.
- $d(C_5(n)) \leq 3n$.

Cronograma e Plano de Trabalho

	2012										2013												2014	
#	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	X	X	X	X		X	X	X	X															
2			X	X	X	X	X	X	X															
3								X																
4							X	X	X	X	X	X												
5												X	X	X	X	X	X							
6																	X	X	X	X	X	X		
7											X	X				X	X				X	X		
8																							X	
9																								X

Tabela: Cronograma de Atividades

1. Obtenção de créditos.
2. Levantamento bibliográfico.
3. Exame de Qualificação de Mestrado (EQM).

Cronograma e Plano de Trabalho

	2012										2013												2014	
#	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	X	X	X	X		X	X	X	X															
2			X	X	X	X	X	X	X															
3								X																
4							X	X	X	X	X	X												
5												X	X	X	X	X	X							
6																	X	X	X	X	X	X		
7											X	X				X	X				X	X		
8																							X	
9																								X

Tabela: Cronograma de Atividades

4. Estudo de novas classes de permutações que podem ser ordenadas em tempo polinomial.
5. Pesquisa de novos algoritmos heurísticos para o problema de Reversão Quase-Simétrica.
6. Pesquisa de melhoria do diâmetro e limitantes do problema de Reversão Quase-Simétrica.

Cronograma e Plano de Trabalho

	2012										2013												2014	
#	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	X	X	X	X		X	X	X	X															
2			X	X	X	X	X	X	X															
3								X																
4							X	X	X	X	X	X												
5												X	X	X	X	X	X							
6																	X	X	X	X	X	X		
7											X	X				X	X				X	X		
8																							X	
9																								X

Tabela: Cronograma de Atividades

7. Escrita da dissertação.
8. Revisão final do texto da dissertação.
9. Defesa da dissertação.

Metodologia

- Serão considerados resultados de outros problemas, particularmente o problema de Reversão.
- Ao longo do trabalho, serão implementados algoritmos, com intuito de obter-se resultados experimentais.
- Caso necessário, serão realizadas provas de propriedades e limitantes do problema.

Metodologia

- Serão considerados resultados de outros problemas, particularmente o problema de Reversão.
- Ao longo do trabalho, serão implementados algoritmos, com intuito de obter-se resultados experimentais.
- Caso necessário, serão realizadas provas de propriedades e limitantes do problema.

Metodologia

- Serão considerados resultados de outros problemas, particularmente o problema de Reversão.
- Ao longo do trabalho, serão implementados algoritmos, com intuito de obter-se resultados experimentais.
- Caso necessário, serão realizadas provas de propriedades e limitantes do problema.

Análise de Resultados

- Os algoritmos produzidos serão avaliados quanto a complexidade.
- Para algoritmos aproximados, provas de corretude serão fornecidas.
- Para avaliações quantitativas, será usada a ferramenta *Rearrangement Distance Database*[1].

[1] http://mirza.ic.unicamp.br:8080/bioinfo/index.jsf

Análise de Resultados

- Os algoritmos produzidos serão avaliados quanto a complexidade.
- Para algoritmos aproximados, provas de corretude serão fornecidas.
- Para avaliações quantitativas, será usada a ferramenta *Rearrangement Distance Database*[1].

[1] http://mirza.ic.unicamp.br:8080/bioinfo/index.jsf

Análise de Resultados

- Os algoritmos produzidos serão avaliados quanto a complexidade.
- Para algoritmos aproximados, provas de corretude serão fornecidas.
- Para avaliações quantitativas, será usada a ferramenta *Rearrangement Distance Database*[1].

[1] http://mirza.ic.unicamp.br:8080/bioinfo/index.jsf

Problema de Distância de Reversão Quase-Simétrica

Thiago da Silva Arruda
Zanoni Dias

Instituto de Computação - Universidade Estadual de Campinas

19 de Setembro de 2012

Obrigado.


Piotr Berman, Sridhar Hannenhalli, and Marek Karpinski.
1.375-Approximation Algorithm for Sorting by Reversals.
In *Proceedings of the 10th Annual European Symposium on Algorithms*, ESA'2002, pages 200–210, London, UK, 2002. Springer-Verlag.


Alberto Caprara.
Sorting by reversals is difficult.
In *Proceedings of the first annual international conference on Computational molecular biology*, RECOMB'1997, pages 75–83, New York, NY, USA, 1997. ACM.


Zanoni Dias, Ulisses Dias, Lenwood S. Heath, and João C. Setubal.
Sorting genomes using almost-symmetric inversions.
In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, SAC'2012, pages 1368–1374, New York, NY, USA, 2012. ACM.


Jonathan Eisen, John Heidelberg, Owen White, and Steven Salzberg.
Evidence for symmetric chromosomal inversions around the replication origin in bacteria.
Genome Biology, 1(6):research0011.1–research0011.9, 2000.


Enno Ohlebusch, Mohamed Ibrahim Abouelhoda, and Kathrin Hockel.
A linear time algorithm for the inversion median problem in circular bacterial genomes.
Jornal of Discrete Algorithms, 5(4):637–646, 2007.


Eric Tannier, Anne Bergeron, and Marie-France Sagot.
Advances on sorting by reversals.
Discrete Applied Mathematics, 155(6-7):881–888, April 2007.