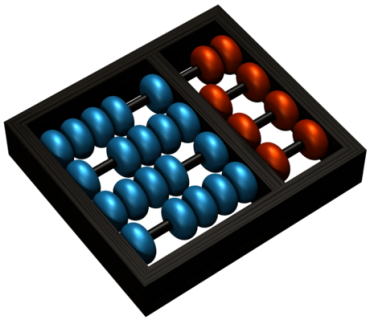
Mestrado em Ciência da Computação
Instituto de Computação
Universidade Estadual de Campinas



Proposta de Dissertação de Mestrado

Problema da Distância de Reversão Quase-Simétrica

Thiago da Silva Arruda - Orientando
Prof. Dr. Zanoni Dias - Orientador

Sumário

Resumo

Em Biologia Computacional, problemas da área de Rearranjo de Genomas buscam investigar o parentesco entre organismos por meio do cálculo do número mínimo de operações de rearranjo necessárias para transformar um genoma em outro. Reversão de genomas é um dos principais problemas desta área. Uma operação de reversão consiste em inverter a ordem de genes de uma subsequência do genoma.

Recentemente, em estudos realizados em genomas de grupos de bactérias, foi identificado um padrão que sugere que os eventos de reversão ocorridos nos genomas destes grupos, ao longo da evolução, possuem alto grau de simetria. O modelo de reversões quase-simétricas mostra-se adequado para ser usado em genomas que apresentam este padrão.

Devido o estudo do problema de Distância de Reversão Quase-Simétrica ter iniciado recentemente, ainda há poucas propriedades conhecidas e não há algoritmo aproximado disponível.

Este trabalho tem como objetivos aprofundar o estudo do problema, desenvolver novos algoritmos e melhorar os limitantes, tanto para o problema como um todo quanto para classes de permutações específicas.

1 Introdução

Dados os genomas de duas espécies, um problema de rearranjo de genomas busca identificar uma sequência de eventos de mutação que explica as diferenças genéticas entre as espécies. Esta sequência de eventos de mutação pode ser utilizada para estimar o grau de parentesco entre as espécies. Os eventos considerados afetam a ordem de grandes porções dos genomas, o que torna esta abordagem mais adequada para comparação de genomas completos, diferentemente do método de alinhamento de genomas, que considera somente mutações pontuais.

Um problema de rearranjo de genomas consiste em comparar genomas de dois indivíduos, representados por sequências de genes ou blocos conservados, para encontrar a menor sequência de eventos de rearranjo que transforma um genoma em outro. O tamanho desta sequência é denominado distância evolucionária entre indivíduos, considerando o principio da máxima parcimônia [1].

[1] Método estatístico baseado na noção filosófica de William Ockham: a melhor hipótese para explicar um processo é aquela que requer o menor número de suposições.

Em termos biológicos, o genoma de um indivíduo é constituído de cromossomos, os quais são conjuntos ordenados de genes. Um evento de rearranjo ocorre quando um cromossomo quebra e os segmentos resultantes são unidos, de forma que o conjunto inicial de genes permanece, mas a ordem é alterada. Para modelagem computacional, um genoma com n genes é representado como uma permutação de tamanho n.

Reversão é um tipo de rearranjo de genomas que consiste em inverter a ordem de um segmento da permutação [2]. Pesquisas recentes mostraram que o alinhamento de genomas de indivíduos de algumas espécies bacterianas apresenta um padrão em "X" [8]. Este padrão implica que os eventos de reversão que ocorrem nestes genomas possuem alto grau de simetria.

Para genomas que apresentam este padrão, foi constatado que utilizar o modelo de reversão genérico para calcular a distância evolucionária produz resultados incorretos [7].

Utilizar um modelo que permite apenas operações de reversão totalmente simétricas é inviável, pois não é possível calcular distância evolucionária entre todos os pares de permutações com mesmo tamanho [7].

O modelo de Reversão Quase-Simétrica, proposto por Dias e coautores [7], permite operações de reversão com, no máximo, um pequeno nível de assimetria, o que torna possível calcular distância evolucionária entre qualquer par de permutações com mesmo tamanho. Devido a isto, este modelo mostra-se adequado para genomas qua apresentam padrão em "X".

1.1 Objetivos

Devido o estudo do problema de Reversão Quase-Simétrica ter iniciado recentemente, ainda há poucas propriedades conhecidas e não há algoritmo aproximado disponível.

Este trabalho tem como objetivo aprofundar o estudo do problema, buscando identificar propriedades teóricas e elaborar novas formulações sobre o problema. A partir destes estudos objetiva-se melhorar o estado da arte do problema, o que consiste em desenvolver novos algoritmos e melhorar os limitantes, tanto para o problema como um todo quanto para classes de permutações específicas.

1.2 Organizacão da Proposta

O restante desta proposta está organizada da seguinte forma: a Seção 2 contém detalhes a respeito dos conceitos básicos de rearranjo de genomas. A Seção 3 apresenta formalmente o problema de Reversão Quase-Simétrica. A Seção 4 apresenta o plano de trabalho e o cronograma. A Seção 5 apresenta a metodologia que será utilizada para o cumprimento de tal plano. Por fim, a Seção 6 mostra como os resultados obtidos serão analisados.

2 Rearranjo de genomas

Para a formulação de algoritmos, um genoma com n genes é classicamente representado como uma *n-tupla* de números inteiros, onde cada número representa um gene. Considerando que esta *n-tupla* não possui repetição, então pode ser definida como uma permutação $\pi = (\pi_1\, \pi_2\, \pi_3 \ldots \pi_n)$ $1 \leq |\pi_i| \leq n$ $|\pi_i| \neq |\pi_j| \Leftrightarrow \pi_i \neq \pi_j$, em que cada elemento π_i possui um sinal, que indica a orientação do gene correspondente. Quando não se conhece a orientação dos genes, o sinal é omitido.

Um genoma circular pode ser representado como uma permutação. Para isto é necessário definir a origem de replicação, que corresponde ao ponto imediatamente anterior ao inicio da permutação, e o sentido de leitura do genoma. Neste trabalho a origem de replicação é representada pelo número "0" e o sentido de leitura é horário, de forma que π_1 é o primeiro elemento após a origem de replicação. A Figura 1 mostra um genoma circular e a respectiva permutação.

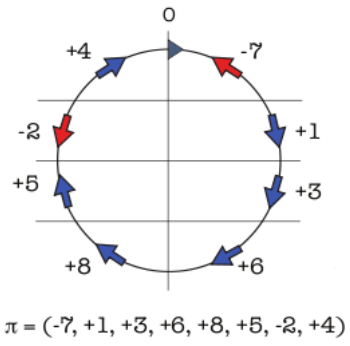


Figura 1: Representação de genomas circulares. A origem de replicação é representada pelo número "0" e o sentido de leitura é horário [6].

Um modelo de rearranjo é um conjunto de operações permitidas para transformar um genoma em outro. As principais operações propostas na literatura são: *reversão*, *transposicão*, *block-interchange*, *translocação*, *fissão* e a *fusão*.

Dado um modelo de rearranjo M, a distância de rearranjo entre duas permutações π e σ é o número mínimo t de operações ρ_i, pertencentes a M, necessário para transformar π em σ:

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

Esta distância pode ser denotada por $d_M(\pi, \sigma)$. A maior distância entre duas permutações de tamanho n, em relação ao modelo M, é denominada diâmetro, denotado por $D_M(n)$.

A permutação identidade ι é definida como $(1\,2\,3\,...\,n)$. A ordenação de uma permutação α consiste em calcular a distância de rarranjo entre α e a permutação identidade. A distância de ordenação de α, em relação a um modelo M, é denotada por $d_M(\alpha, \iota) = d_M(\alpha)$. Desta forma, a distância de rearranjo entre π e σ é equivalente a distância entre $\alpha = \pi\sigma^{-1}$ e ι, logo $d_M(\pi, \sigma) = d_M(\alpha) = d_M(\alpha, \iota)$.

A seguir é detalhado o Problema de Distância de Reversão.

2.1 Distância de Reversão

Uma operação de reversão em uma permutação π consiste em inverter a ordem dos elementos de um segmento. Dessa forma a reversão $\rho(i, j)$, onde $1 \leq i \leq j \leq n$, resulta em:

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

Quando não se conhece a orientação dos genes, tem-se o Problema de Reversão sem Sinal, que foi provado ser NP-Difícil por Caprara [4]. Bafna e Pevzner [2] fizeram os primeiros estudos sobre este problema, o que resultou em um algoritmo de aproximação de fator 1,75. Posteriormente, Christie [5] apresentou um algoritmo de aproximação de fator 1,5. Atualmente, o melhor algoritmo conhecido possui fator de aproximação 1,375, desenvolvido por Berman e coautores [3].

Para o caso em que se conhece a orientação dos genes, denominado Problema de Reversão com Sinal, existem algoritmos polinomiais exatos, sendo que o primeiro foi apresentado por Hannenhalli e Pevzner [12]. Atualmente o algoritmo mais eficiente disponível

na literatura possui complexidade sub-quadrática [16]. Para o caso em que se deseja saber apenas o valor distância de reversão, há um algoritmo linear [1].

Uma variação do problema, bastante estudada na literatura, consiste em um modelo de rearranjo em que as reversões sempre ocorrem no prefixo da permutação, conhecido como Problema de Ordenação de Panquecas [11, 13]. Atualmente o melhor algoritmo conhecido, desenvolvido por Fischer e Ginzinger [9], possui fator de aproximação 2.

Um resultado importante foi obtido por Medains, Water e Dias [14], que mostraram que toda teoria sobre genomas lineares pode ser facilmente adaptada para genomas circulares.

3 Problema de Reversão Quase-Simétrica

A análise comparativa de genomas de bactérias das famílias *Pseudomonadaceae* e dos grupos *Xanthomonas*, *Shewanella* e *Mycobacterium* sugere que os eventos de reversão que ocorrem nestes organismos são mais restritos do que permite o modelo convencional de reversão.

Ao alinhar dois genomas destes grupos bacteriais em um plano cartesiano (*dot plot*) é observado um padrão em "X"; a Figura 2 exemplifica este tipo de alinhamento. Este padrão caracteriza que há predominância dos eventos de reversão que possuem simetria em relação à origem de replicação.

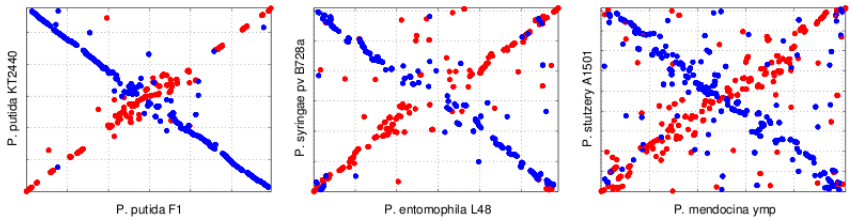
O padrão em "X" foi primeiramente observado na pesquisa genética conduzida por Eisen [8], ao comparar genomas de alguns organismos bacteriais. No mesmo trabalho há a conclusão de que este padrão não ocorre ao acaso, resultante da análise de um modelo estatístico.

Este padrão pode ser explicado pelo fato de, diferentemente dos seres vertebrados, bactérias possuírem DNA circular. Supõe-se que esta característica favoreça eventos de reversão simétricos [8].

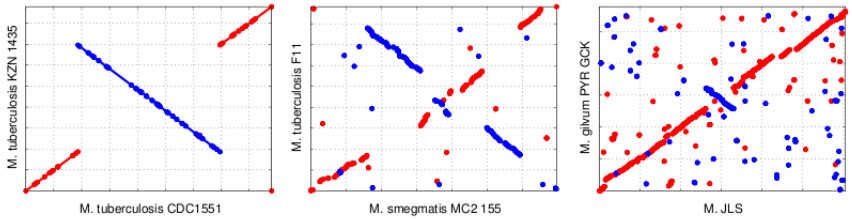
A partir da observação de que eventos de reversão fortemente assimétricos podem destruir um padrão de alinhamento em "X", conclui-se que reversões assimétricas não devem ser muito comuns em genomas bacteriais.

Nesta seção, três funções, aplicáveis a permutações π, são de interesse:

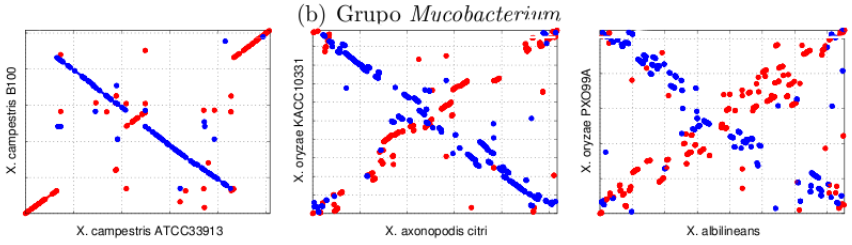
- **Position:** $p(\pi, i) = k \Leftrightarrow |\pi[k]| = i,\ p(\pi, i) \in \{1, 2, ..., n\}$.



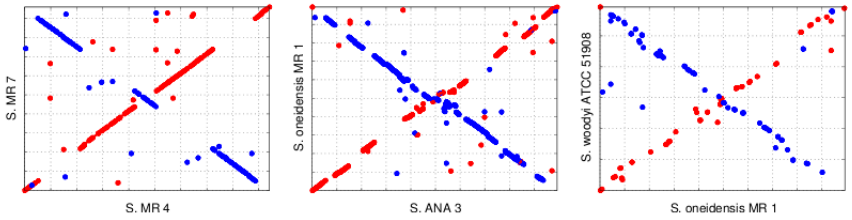
(a) Família *Pseudomonadaceae*



(b) Grupo *Mucobacterium*

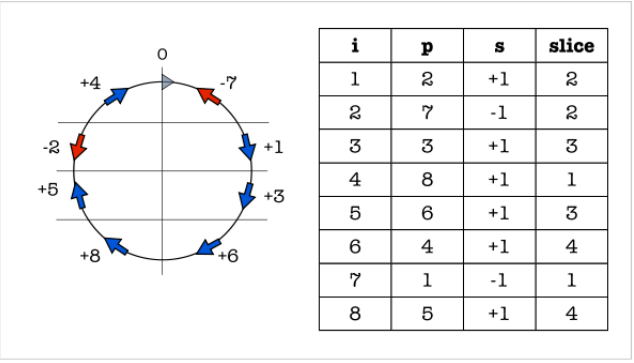


(c) Grupo *Xanthomonas*



(d) Grupo *Shewanella*

Figura 2: Alinhamento de genomas de algumas espécies de bactérias [6].



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 3: Exemplo das funções position(p), sign(s) e slice [6].

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

A Figura 3 mostra um exemplo da aplicação das funções position (p), sign(s) e slice aos elementos de uma permutação.

Ohlebusch [15] foi o primeiro a propor uma modelagem computacional para o problema caracterizado pelo padrão em "X" (predominância de reversões simétricas).

O modelo proposto permite somente operações de Reversão Simétrica. No mesmo trabalho foi apresentado um algoritmo exato para o problema, que possui complexidade linear.

Formalmente, uma reversão simétrica $\ddot{\rho}$ pode ser definida, através da definição da operação de reversão, da seguinte forma:

$$\ddot{\rho} = \rho(i + 1, n - i),\ \ 0 \leq i \leq \left\lceil \frac{n}{2} \right\rceil$$

A Figura 4 exemplifica uma operação de reversão simétrica.

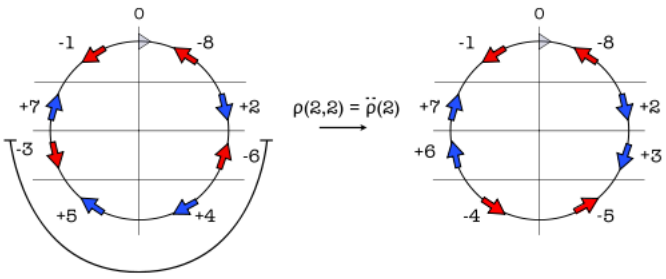


Figura 4: Exemplo de operação de reversão simétrica [6].

A partir da análise do modelo de reversão simétrica, é constatado que somente é

possivel encontrar uma sequência de reversões simétricas que transforma uma permutação π em outra permutação σ, ambas de tamanho n, se $slice(\pi, i) = slice(\sigma, i)$, para todo $1 \leq i \leq n$.

Esta constatação implica a redução do conjunto de permutações que podem ser ordenadas. O número de permutações com sinal e tamanho n é $2^n n!$, mas apenas $2^{\lceil \frac{n}{2} \rceil}$ permutações podem ser ordenadas por este modelo, isto impossibilita sua utilização para o cálculo da distância evolutiva entre indivíduos em geral.

No trabalho de Dias e coautores [7] foi apresentado o modelo de **Reversão Quase-Simétrica**, que permite reversões com assimetria de no máximo uma unidade. Este modelo é justificado pelo fato de que em genomas reais espera-se que nem todos os eventos de reversão sejam perfeitamente simétricos [8].

Com reversões quase-simétricas é possível estabelecer uma sequência de operações que transforma qualquer permutação π de tamanho n em qualquer outra permutação σ também de tamanho n, o que implica que é possível ordenar todas as permutações de tamanho n.

Formalmente, o problema de reversão quase-simétrica pode ser definido, a partir da definição da operação de reversão, da seguinte forma:

$$\overline{\rho}(i, j) = \rho(i + 1, n - j), \;\; 0 \leq i, j \leq n - 1, \;\; |i - j| \;\leq 1$$

Reversões quase-simétricas são um caso especial das reversões k-assimétricas:

$$\hat{\rho}(i, j) = \rho(i + 1, n - j), \;\; 0 \leq i, j \leq n - 1, \;\; |i - j| \leq k \;, \;\; k \geq 1$$

A dificuldade dos problemas de Reversão Quase-Simétrica e k-Assimétrica aparenta ser consideravelmente maior do que o Problema de Reversão Simétrica [6]. Ainda não se conhece a complexidade de ambos problemas.

A seguir são detalhados os limitantes conhecidos, os algoritmos já desenvolvidos e as classes e famílias de permutações já definidas [7].

3.1 Limitantes

A fim de obter limitantes para o problema de reversões quase-simétricas, é considerado o número mínimo de operações necessárias para posicionar corretamente um único elemento

(Definições 2 e 1), que corresponde ao número de slices percorridas para posicioná-lo corretamente [7].

Definição 1 *Uma reversão quase-simétrica $\overline{\rho}(i,j)$* ***age*** *em um elemento k se $i \leq p(\pi,k) \leq j$.*

Definição 2 *O número mínimo de operações de reversão quase-simétricas que devem agir em um elemento k para posiciona-lo corretamente na permutação π é definido por $d_\pi[k]$.*

Lema 1 *Se $slice(\pi, j) = k$, então $slice(\pi\overline{\rho}, j) \in \{k–1, k, k+1\}$, para toda reversão quase-simétrica $\overline{\rho}$.*

Lema 2 *Se $slice(\pi, i) = j$ e $slice(\iota, i) = k$, então $d_\pi[i] \geq |j-k|$ reversões quase-simétricas devem agir no elemento i para posicioná-lo adequadamente.*

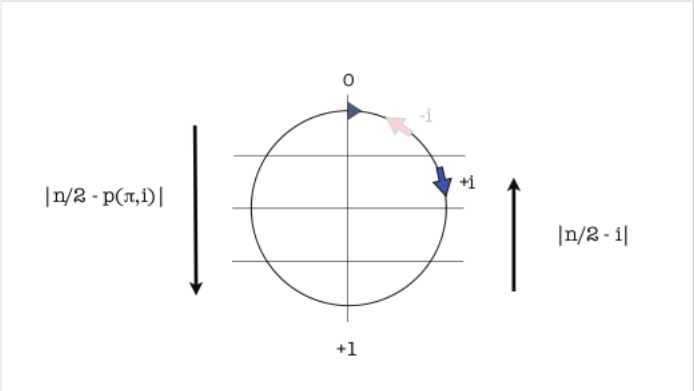
O Lema 2 é em um limite inferior para $d_\pi[i]$. Este lema não considera o sinal que um elemento terá ao ser corretamente posicionado. Quando um elemento está posicionado corretamente, mas com sinal diferente do desejado, é necessário movimentá-lo até a posição diametralmente oposta a origem de replicação e depois trazê-lo de volta, o que resulta em um número maior de operações.

Para obter-se o valor exato de $d_\pi[i]$ é necessário considerar o sinal que o elemento terá na permutação resultante. Os Lemas 3 e 4 mostram como calcular $d_\pi[i]$, respectivamente, para os casos em que a permutação possui tamanho par e ímpar, o que é ilustrado na Figura 5.

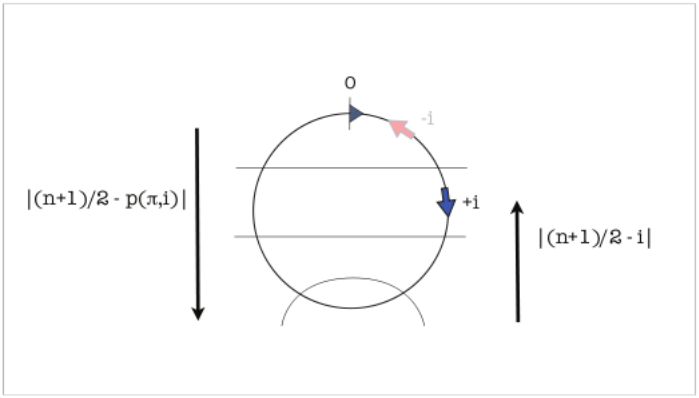
Os Lemas 3 e 4 calculam o número de operações necessárias para posicionar um único elemento, sem considerar o posicionamento dos demais elementos. Contudo, na ordenação da permutação, uma operação de reversão quase-simétrica necessária para posicionar um elemento pode interferir no posicionamento de outros elementos. O Lema 5 estabelece um limite inferior para o número de operações que agem em um elemento que já está corretamente posicionado.

Lema 3 *Seja π um genoma com um número par de genes ($n = 2k$), então $d_n[i] = |\frac{n}{2} - p(\pi,i)| + |\frac{n}{2} - i| + 1$ se um dos itens a seguir for verdade:*

- $i \leq \frac{n}{2}$, $p(\pi,i) \leq \frac{n}{2}$ *e* $s(\pi,i) = -1$



(a) Permutações de tamanho par



(b) Permutações de tamanho ímpar

Figura 5: Número de reversões quase-simétricas necessárias para posicionar um elemento quando o posicionamento direto resultaria em um sinal oposto ao desejado [6].

- $i \leq \frac{n}{2}$, $p(\pi, i) > \frac{n}{2}$ *e* $s(\pi, i) = +1$
- $i > \frac{n}{2}$, $p(\pi, i) \leq \frac{n}{2}$ *e* $s(\pi, i) = +1$
- $i > \frac{n}{2}$, $p(\pi, i) > \frac{n}{2}$ *e* $s(\pi, i) = -1$

Lema 4 *Seja π um genoma com um número impar de genes ($n = 2k + 1$), então $d_n[i] = |\frac{n+1}{2} - p(\pi, i)| + |\frac{n+1}{2} - i| + 1$ se um dos itens a seguir for verdade:*

- $i \leq \frac{n+1}{2}$, $p(\pi, i) \leq \frac{n+1}{2}$ *e* $s(\pi, i) = -1$
- $i \leq \frac{n+1}{2}$, $p(\pi, i) > \frac{n+1}{2}$ *e* $s(\pi, i) = +1$
- $i > \frac{n+1}{2}$, $p(\pi, i) \leq \frac{n+1}{2}$ *e* $s(\pi, i) = +1$
- $i > \frac{n+1}{2}$, $p(\pi, i) > \frac{n+1}{2}$ *e* $s(\pi, i) = -1$

Lema 5 *Seja i um elemento na posição correta ($\pi_i = i$). Se existe um elemento j tal que $slice(\pi, j) < slice(\pi, i)$ e $d_\pi[j] > 0$, então no mínimo duas reversões quase simétricas devem agir em i para ordenar π, o que significa que $d_\pi[i] \geq 2$.*

As funções *Lower* e *Upper*, apresentadas nas Definições 3 e 4, usam a distância $d_\pi[i]$. A função *Lower* é um limitante inferior para o problema de Reversão Quase-Simétrica. Por outro lado, não se pode afirmar que a função *Upper* é um limitante superior, mas uma busca exaustiva realizada para todas as permutações π, de tamanho até 9, permite conjecturar que *Upper* é um limitante superior [7].

Definição 3 *$Lower(\pi) = max_{1 \leq i \leq n} d_\pi[i]$.*

Definição 4 *$Upper(\pi) = \sum_{1 \leq i \leq n} d_\pi[i]$.*

A Tabela 1 mostra as distâncias exatas, valores de *Lower*, *Upper* e diâmetro para todas as permutações de tamanho até 10. A Figura 6 exemplifica a aplicação das funções *Lower* e *Upper*.

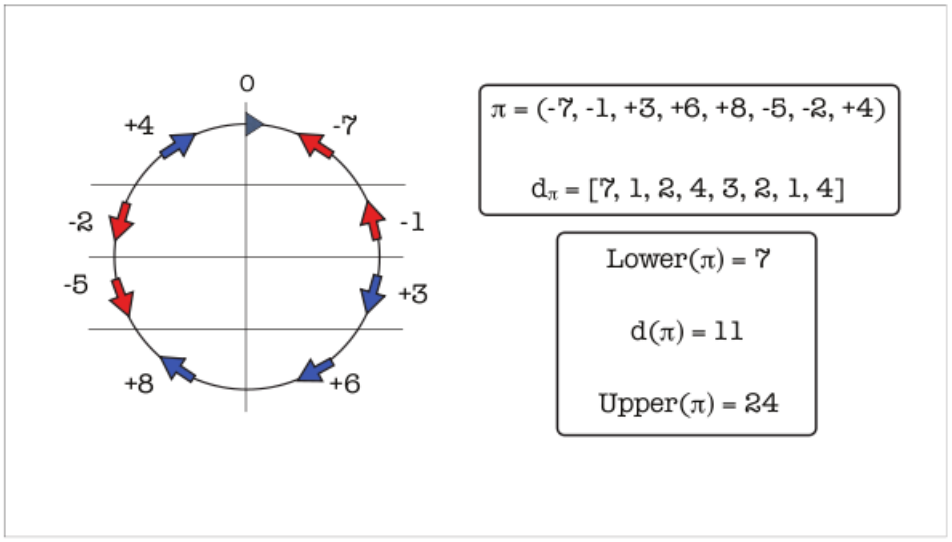


Figura 6: Aplicação das Funções *Lower* e *Upper* [6].

3.2 Algoritmos de ordenação

No trabalho de Dias e coautores [7] foram apresentados os primeiros algoritmos para o problema de ordenação por Reversões Quase-Simétricas. Estes algoritmos são heurísticas que não possuem provas de fator de aproximação.

			Média		
N	**Permutações**	**Diam**	**Lower**	**Dist**	**Upper**
2	8	3	1,25	1,50	2,00
3	48	5	2,13	2,96	4,46
4	384	6	2,90	4,26	7,48
5	3.840	8	3,71	5,68	11,09
6	46.080	10	4,52	7,02	15,33
7	645.120	11	5,36	8,41	20,21
8	10.321.920	13	6,20	9,76	25,75
9	185.794.560	15	7,06	11,15	31,94
10	3.715.891.200	16	–	12,51	–

Tabela 1: Valores médios de $Lower(\pi)$, $d(\pi)$ e $Upper(\pi)$ para todas as permutações de tamanho menor ou igual a 10. **Diam** corresponde ao diâmetro para um dado tamanho de permutação [6].

O algoritmo básico (Algoritmo 1) baseia-se em posicionar primeiro os elementos que possuem menor slice na permutação identidade. Se $slice(\iota, i) < slice(\iota, j)$ então i será posicionado antes que j. Desta forma um elemento já posicionado não será afetado pelos subsequentes posicionamentos de elementos.

Algoritmo 1 Algoritmo básico

Data: π
for $i \leftarrow 1$ **to** $\lceil \frac{n}{2} \rceil$ **do**
  Coloque o elemento i em sua posição correta
  Coloque o elemento $n - i + 1$ em sua posição correta
end for

É fácil verificar que os elementos i e $n - i + 1$ possuem o mesmo slice. Por convenção, o algoritmo básico posiciona o elemento i antes do elemento $n - i + 1$. Uma otimização local consiste em, a cada iteração, escolher a sequência de posicionamentos que exucuta menos operações. O Algoritmo 2 implementa esta otimização.

Algoritmo 2 Algoritmo básico com otimização de slice

Data: π
for $i \leftarrow 1$ **to** $\lceil \frac{n}{2} \rceil$ **do**
  Escolha a melhor opção entre:
  - Coloque o elemento i em sua posição correta
    + Coloque o elemento $n - i + 1$ em sua posição correta
  - Coloque o elemento $n - i + 1$ em sua posição correta
    + Coloque o elemento i em sua posição correta

end for

Os Algoritmos 1 e 2 possuem complexidade $O(n^3)$, pois é necessário tempo $O(n)$ para

posicionar um elemento, e ao todo são necessárias $O(n^2)$ reversões simétricas, de acordo com o limitante superior (Definição 4).

Em Dias e coautores [7] também foi apresentado um algoritmo guloso para o problema de Reversões Quase-Simétricas (Algoritmo 3). Este algoritmo busca minimizar a função objetivo $h(n) = min\{d[i], d[n-i-1] + d[i] + d[i-i+1]\}$ para todo $\overline{\rho}'$ aplicável a π.

Algoritmo 3 Algoritmo Guloso

```
Data: π
k ← 1
while π ≠ ι do
  ρ̄ ← ρ̄' que minimiza a expressão min{d[i], d[n − i − 1] + d[i] + d[i − i + 1]},
      para 1 ≤ i ≤ n/2, em ρ̄'π para todo ρ̄' aplicável a π
  π ← ρ̄π
  k←k + 1
end while
return k
```

O algoritmo guloso possuí uma complexidade $O(n^5)$, pois é necessário tempo $O(n)$ para calcular a função a ser minimizada para todo $1 \leq i \leq \frac{n}{2}$, o que resulta em tempo $O(n^3)$ para cada reversão (ao todo são realizadas $O(n^2)$ reversões).

As Tabelas 2, 3 e 4 mostram os resultados experimentais para os algoritmos básico, básico com otimização e guloso, quando executados com todas as permutações de tamanho até 9 e mostra comparações destes resultados com os exatos. Para compar os resultados, utilizou-se a razão $\frac{d_{alg}(\pi)}{d(\pi)}$, onde $d_{alg}(\pi)$ representa a distância fornecida pelo algoritmo e $d(\pi)$ representa a distância exata.

O algoritmo básico apresenta resultados gradativamente mais distantes do ótimo, em relação ao tamanho das permutações (Tabela 2). A otimização presente no Algoritmo 2 apresenta melhoras nos resultados, particularmente na diminuição da razão em relação a distância ótima. O algoritmo guloso apresenta melhores resultados, com razão média de aproximação menor que 2 em todos os testes, contudo em alguns casos a distância ao ótimo é muito maior, como evidenciado pela razão máxima.

Com base nestes resultados, acredita-se que o problema de Reversão Quase-Simétrica apresenta muitas oportunidades de evolução [7]. Tendo em vista que muitos dos problemas de Rearranjo de Genomas já há mais tempo explorados (particularmente o problema de Reversão) possuem atualmente estado da arte mais evoluído, com algoritmos polinomiais aproximativos ou exatos conhecidos.

	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 2: Performance do Algoritmo 1. A tabela mostra o valor máximo (**Operações - MAX**) e o médio (**Operações - AVG**) das distâncias fornecidas pelo algoritmo, bem como o valor máximo (**Razão - MAX**) e o médio (**Razão -AVG**) da razão de aproximação destes valores em relação a distância exata. A ultima coluna fornece a porcentagem de soluções que o algoritmo retorna a solução exata [6].

3.3 Famílias e classes de permutações

Para estudar o problema de Reversão Quase-Simétrica, no trabalho de Dias e coatores [7] foram definidas famílias e classes de permutações, com o intuito de elaborar algoritmos específicos.

Uma família é um grupo de permutações que compartilham uma mesma regra de definição. Para uma família, existe uma única permutação de tamanho n, para todo número natural n.

Classes são grupos mais abrangentes, que podem conter mais de uma permutação do mesmo tamanho n. Classes podem conter famílias de permutações.

As classes e famílias selecionadas compreendem permutações difíceis de serem ordenadas pelos algoritmos genéricos (Algoritmos 1, 2 e 3) previamente apresentados.

3.3.1 Famílias

Foram definidas dez famílias de permutações, e para cada uma foi conjecturado a distância (Conjectura 1). Estas conjecturas são verdadeiras para todas as permutações π, para $|\pi| \leq 10$.

Para cada família foi elaborado um algoritmo específico de ordenação, estes algoritmos apresentam melhores resultados do que os algoritmos genéricos apresentados anteriormente para o Problema de Reversão Quase-Simétrica.

	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 3: Performance do Algoritmo 2. A tabela mostra o valor máximo (**Operações - MAX**) e o médio (**Operações - AVG**) das distâncias fornecidas pelo algoritmo, bem como o valor máximo (**Razão - MAX**) e o médio (**Razão -AVG**) da razão de aproximação destes valores em relação a distância exata. A ultima coluna fornece a porcentagem de soluções que o algoritmo retorna a solução exata [6].

Conjectura 1 *Para cada familia $F_k(n)$ conjectura-se que a distância $d(F_k(n))$ corresponde a 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* $n \geq 5$.

3.3.2 Classes

Na Conjectura 2 são apresentados os resultados sobre classes do problema de Reversão Quase-Simétrica obitidos por Dias e coautores [7].

Conjectura 2 *Sejam*

- $C_1(n, k) = \rho(1, k) \cdot \iota$, *para* $1 \leq k \leq n$.
- $C_2(n, i, j) = \rho(i, j) \cdot \iota$, *para* $1 \leq i \leq j \leq n$.

	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 4: Performance do Algoritmo 3. A tabela mostra o valor máximo (**Operações - MAX**) e o médio (**Operações - AVG**) das distâncias fornecidas pelo algoritmo, bem como o valor máximo (**Razão - MAX**) e o médio (**Razão -AVG**) da razão de aproximação destes valores em relação a distância exata. A ultima coluna fornece a porcentagem de soluções que o algoritmo retorna a solução exata [6].

- $C_3(n,i,j) = \rho(1,k) \cdot \rho(k+1,n) \cdot \rho(1,n) \cdot \iota$, *para* $1 \leq k \leq n$.
- $C_4(n) = \{\rho(i_1,i_1) \cdot \rho(i_2,i_2)) \cdot ... \cdot \rho(i_r,i_r) \cdot \iota$, *para* $1 \leq i_k \leq n$, $1 \leq k \leq r$, $i_p \neq i_q, 1 \leq p \leq q \leq r$, $1 \leq r \leq n\}$
- $C_5(n) = \{\rho(i_1,j_1) \cdot \rho(i_2,j_2)) \cdot ... \cdot \rho(i_r,j_r) \cdot \iota$, *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 permutação. Então:

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

Todas as conjecturas relacionadas a classes são verdadeiras para todas as permutações π, com $|\pi| \leq 10$. Para cada classe foi desenvolvido um algoritmo, os quais satisfazem as conjecturas em todas as permutações testadas.

4 Cronograma e Plano de trabalho

O cronograma das atividades é apresentado na Tabela 5 e compreende o período de Março de 2012 até Fevereiro de 2014, totalizando 24 meses. Para que o objetivo deste trabalho seja alcançado, as tarefas listadas abaixo devem ser cumpridas.

	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 5: Cronograma de atividades

1. Obtenção de créditos.
2. Levantamento bibliográfico.
3. Exame de Qualificação de Mestrado (EQM).
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.
7. Escrita da dissertação.
8. Revisão final do texto da dissertação.
9. Defesa da dissertação.

5 Metodologia

Até o momento, o problema de distância de reversão quase simétrica foi pouco explorado; resultados iniciais para o problema estão no trabalho de Dias e coautores [7]. Não se conhece a qual classe de complexidade o problema pertence, não há algoritmo aproximativo conhecido e as heurísticas existentes apresentam resultados significativamente distantes do ótimo (o que foi detalhado na Seção 3).

Este trabalho irá aprofundar o estudo do problema, objetivando desenvolver algoritmos e descobrir novas propriedades. Para isto, serão considerados resultados de outros problemas de rearranjo, principalmente o problema de distância de reversão, que já é bastante explorado na literatura e possui inclusive algoritmo exato de complexidade polinomial.

Ao longo do trabalho, serão implementados algoritmos para obter resultados experimentais e testar hipóteses sobre o problema. Caso necessário, também serão realizadas provas de conjecturas para validar propriedades do problema.

6 Análise de resultados

Para todos os algoritmos produzidos será analisada a complexidade computacional. Para algoritmos de aproximação provas de corretude serão fornecidas.

Todos os algoritmos produzidos também serão avaliados quantitativamente, utilizando a ferramenta *Rearrangement Distance Database*[1] [10], que possui uma base de dados de distâncias exatas para os vários problemas de rearranjo de genomas (inclusive para Reversão Quase-Simétrica), para todas as permutações de tamanho menor ou igual a 13.

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

Referências

[1] David A. Bader, Bernard M. E. Moret, and Mi Yan. A Linear-Time Algorithm for Computing Inversion Distance between Signed Permutations with an Experimental Study. *Journal of Computational Biology*, 8:483–491, 2001.

[2] Vineet Bafna and Pavel A. Pevzner. Genome Rearrangements and Sorting by Reversals. *SIAM Journal of Computing*, 25(2):272–289, February 1996.

[3] 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.

[4] 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.

[5] David A. Christie. A 3/2-approximation algorithm for sorting by reversals. In *Proceedings of the ninth annual ACM-SIAM symposium on Discrete algorithms*, SODA'1998, pages 244–252, Philadelphia, PA, USA, 1998. Society for Industrial and Applied Mathematics.

[6] Ulisses Martins Dias. *Problemas de Comparação de Genomas*. PhD thesis, UNICAMP, March 2012.

[7] 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.

[8] 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.

[9] Johannes Fischer and Simon W. Ginzinger. A 2-approximation algorithm for sorting by prefix reversals. In *Proceedings of the 13th annual European conference on Algorithms*, ESA'2005, pages 415–425, Berlin, Heidelberg, 2005. Springer-Verlag.

[10] Gustavo Rodrigues Galvão and Zanoni Dias. Computing rearrangement distance of every permutation in the symmetric group. In *Proceedings of the 2011 ACM Symposium on Applied Computing*, SAC'2011, pages 106–107, New York, NY, USA, 2011. ACM.

[11] William H. Gates. Bounds for sorting by prefix reversal. *Discrete Mathematics*, pages 47–57, 1979.

[12] Sridhar S Hannenhalli and Pavel A. Pevzner. Transforming men into mice (polynomial algorithm for genomic distance problem). In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, FOCS'1995, pages 581–592, Washington, DC, USA, 1995. IEEE Computer Society.

[13] Mohammad H. Heydari and I.Hal Sudborough. On the Diameter of the Pancake Network. *Journal of Algorithms*, 25(1):67–94, 1997.

[14] J. Meidanis, M. E. M. T. Walter, and Z. Dias. Reversal Distance of Signed Circular Chromosomes. Technical Report IC-00-23, University of Campinas, 2000.

[15] 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.

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