Problemas de Comparação de Genomas

Ulisses Martins Dias
Tese de Doutorado
Universidade Estadual de Campinas

Orientador: Prof. Dr. Zanoni Dias

Agenda

Comparação de Genomas

- Parte I: Eventos de Transposição.
- Parte II: Eventos de Reversão
- Parte III: Distância entre genomas.

Parte I

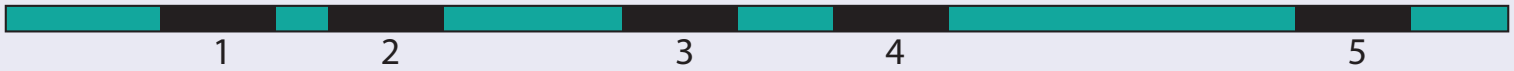
Eventos de Transposições

Heurísticas para o Problema da Distância de Transposição

Representação de genomas

- Listagem de blocos conservados.
- Sinal negativo para blocos com orientação contrária.
- Sinal positivo para blocos com mesma orientação

Genoma 1: $\iota = (1\ \ 2\ \ 3\ \ 4\ \ 5)$



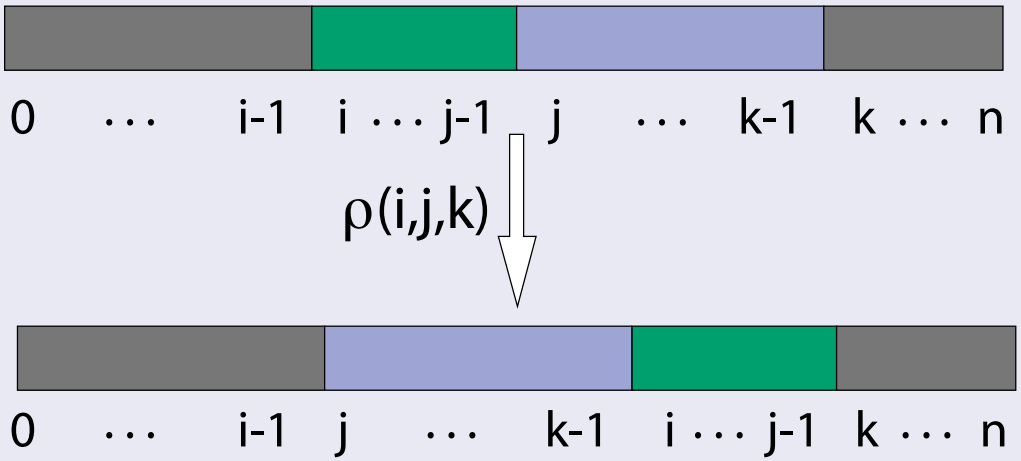
Genoma 2: $\pi = (-5\ \ 2\ \ -3\ \ 1\ \ 4)$

Mesma Orientação

Orientação contrária

Transposição

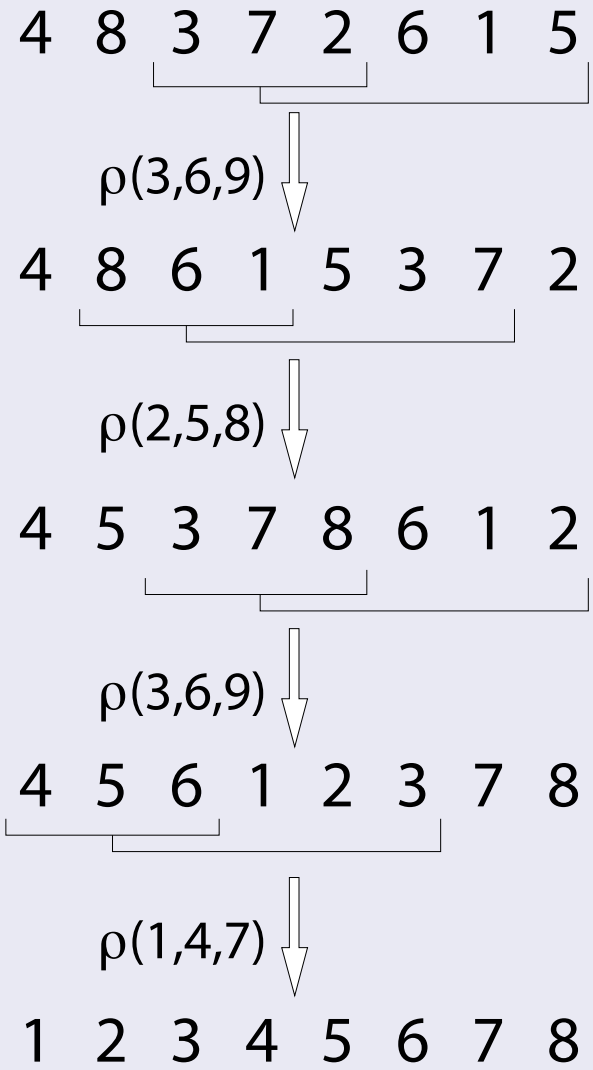
- Um segmento é destacado do genoma e inserido em outra posição com a mesma orientação.



Transposição

- Quando apenas os eventos de transposição são permitidos, assume-se que todos os blocos conservados nos genomas possuem a mesma orientação, não sendo mais necessária a utilização de sinais negativos.
- Formalmente, representa-se um genoma arbitrário como uma permutação π.
  - $\pi = (\pi_1\ \pi_2\ \ldots\ \pi_n)$
  - $\pi_i \in \mathbb{N}$
  - $0 < \pi_i \leq n$
  - $i \neq j \leftrightarrow \pi_i \neq \pi_j$.
- $\pi\rho(i,j,k) = (\pi_1\ \ldots\ \pi_{i-1}\ \pi_j\ \ldots \pi_{k-1}\ \pi_i \ldots \pi_{j-1}\ \pi_k \ldots \pi_n)$

Ordenação por Transposições



Motivação

- A maioria dos algoritmos foram criados com o propósito de provar algum fator de aproximação, ou apresentar uma nova estrutura de dados.
- Dessa forma, os resultados em uma análise prática foram colocados em segundo plano.
- A análise mostra que a resposta fornecida por esses algoritmos se afasta da distância real em uma grande quantidade de casos.

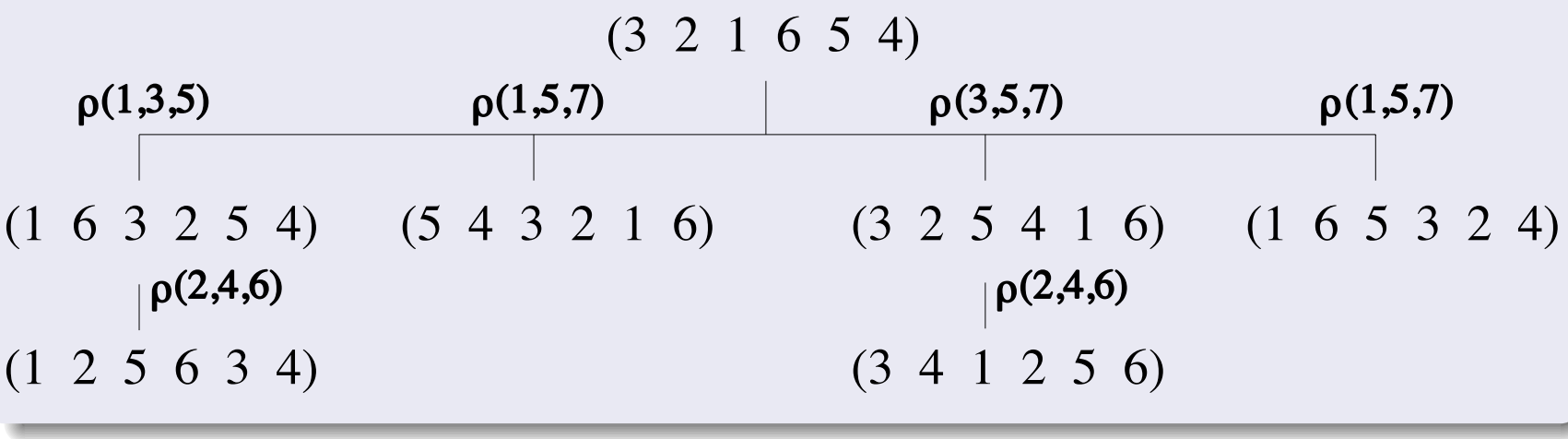
Heurísticas

Objetivos

- Apresentar um algoritmo de aproximação 1.375.
- O algoritmo utiliza a estrutura de grafos de Bafna e Pevzner, 1998.
- O algoritmo utiliza também idéias de Elias e Hartman, 2006.
- Foram implementadas 7 heurísticas capazes de fornecer melhores resultados na prática.

Look-ahead

- Uma das heurísticas utiliza uma estratégia de *look-ahead*.
- Esse passo foi gerado porque, em muitos casos, é difícil prever os efeitos de uma transposição.
- As transposições são classificadas de acordo com as configurações obtidas após um certo número de passos.



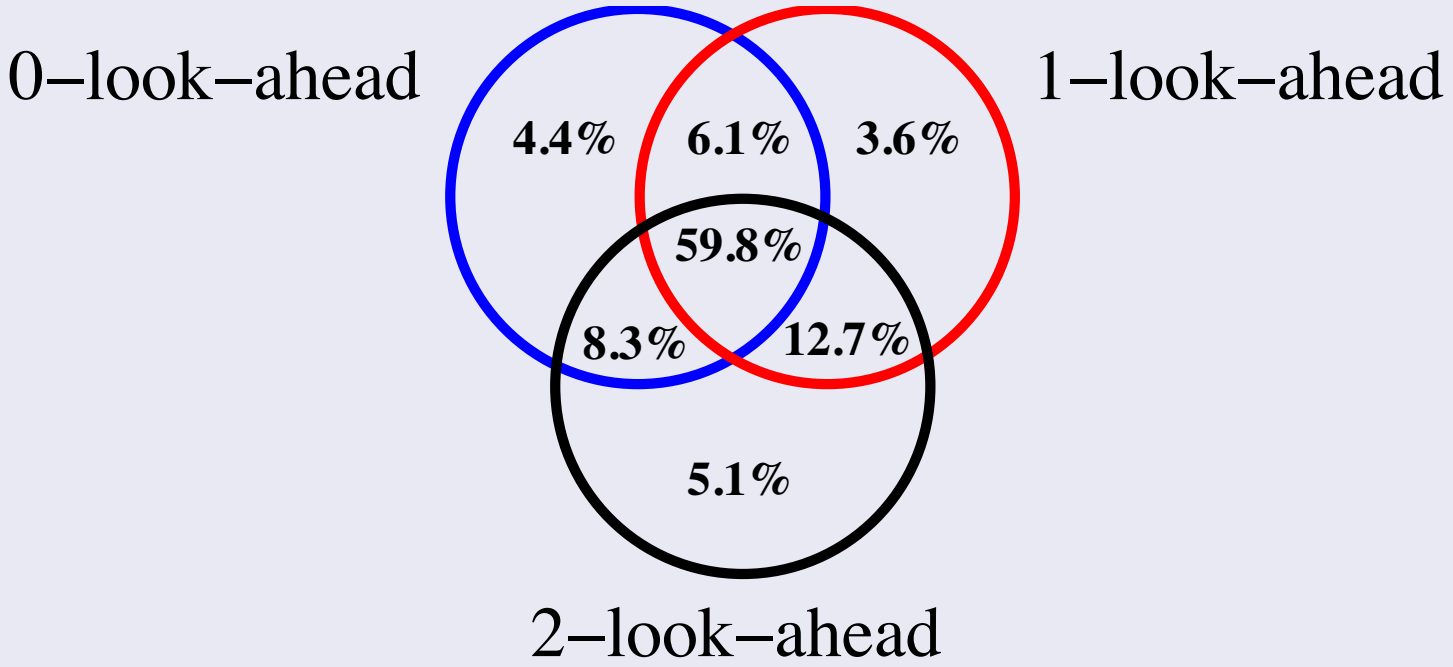
- O *look-ahead* é oneroso computacionalmente. Por isso, foi feita uma análise do tempo de resposta.

Tempo de Execução do *Look-ahead*

Tamanho	0-LA	1-LA	2-LA
10	0,00	0,00	0,01
20	0,00	0,01	0,01
30	0,00	0,09	0,08
40	0,00	0,57	0,60
50	0,00	1,78	1,86
60	0,00	5,24	5,58
70	0,01	17,12	17,37
80	0,01	40,79	53,88
90	0,02	103,50	115,26
100	0,04	244,80	234,19

Média do tempo (em minutos) consumido para cada permutação.

Performance em Permutações Longas



- O look-ahead melhora o resultado em 21,4% dos casos.
- Em 59,8% dos casos, os algoritmos forneceram a mesma resposta.

Performance em Permutações Curtas

Tamanho	0-LA	1-LA	2-LA
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0
5	0	0	0
6	0	0	0
7	0	0	0
8	18	0	0
9	1.300	0	0
10	25.938	1.206	1.055

Quantidade de respostas não ótimas fornecidas por cada versão do algoritmo ao variar o valor do *look-ahead*.

Outros Algoritmos

Tamanho	WDM	C/WCO	HS/H	BP/W	M	EH/DD
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	6	0	2	0	0	0
7	72	0	108	0	0	273
8	1.167	40	1.517	34	124	2.866
9	14.327	1.182	25.425	1.007	2.977	37.393
10	-	-	-	-	69.297	449.243

Resultados para outros algoritmos encontrados na literatura.

Publicação

- U. Dias and Z. Dias. Extending Bafna-Pevzner algorithm. In *Proceedings of the 5th International Conference of the Brazilian Association for Bioinformatics and Computational Biology (X-Meeting'2009)*, pages 5–5, Angra dos Reis, Brasil, 2009.
- U. Dias and Z. Dias. Extending Bafna-Pevzner algorithm. In *Proceedings of the 1st International Symposium on Biocomputing (ISB'2010)*, pages 1–8, Calicut, India, 2010.
- U. Dias and Z. Dias. An improved 1.375-approximation algorithm for the transposition distance problem. In *Proceeding of the 1st ACM International Conference on Bioinformatics and Computational Biology (ACM-BCB'2010)*, pages 334–337, Niagara Falls, EUA, 2010. ACM.

Distância de Transposição usando Programação em Lógica com Restrições

Objetivos

- Fornecer a solução exata para a distância de transposição.
- Criar uma nova abordagem para a distância de transposição ao fazer uso de uma área já bem consolidada em ciência da computação.

Modelos usando Programação em Lógica com Restrições

- Um modelo de Programação em Lógica com Restrições é composto de:
  - Variáveis: Representação das Permutações e Transposições
  - Restrições: Limitantes e Definição do Problema

Limitantes para o problema de transposição

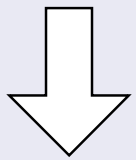
- Limitantes conhecidos para o problema.

- Limitantes Inferiores
  - Limitante de Breakpoints (**br**).
  - Limitante do Grafo de Ciclos de Bafna e Pevzner (**cg**).
- Limitantes Superiores
  - Limitante de Breakpoints (**br**).
  - Limitante do Grafo de Cicos de Bafna e Pevzner (**cg**).
  - Limitante do Grafo Gamma de Labarre (**gg**).

Permutações Reduzidas

- Permutações reduzidas diminuem o espaço de busca do problema.

$$6 \bullet 3 \quad 4 \bullet 7 \bullet 1 \quad 2 \bullet 5 \bullet 8$$



$$4 \bullet 2 \bullet 5 \bullet 1 \bullet 3$$

Problema de Satisfação de Restrição (CSP)

- O número de variáveis é desconhecido.
- Cria-se um modelo com um candidato T para a distância.
- Se esse candidato falhar, seleciona-se $T + 1$ como candidato.

Problema de Otimização de Restrição (COP)

- Define-se um conjunto de variáveis booleanas $B_1, \ldots, B_{Upper\_Bound}$ para indicar se uma transposição ocorreu.
- A função objetivo é a soma dessas variáveis booleanas.

Tempo médio de resposta (em segundos)

n	CSP					COP		
	def	**br**	**cg**	**cg_mdf**	**cg_red**	**def**	**cg**	**gg**
4	0.006	0.010	0.005	0.000	0.005	0.134	0.019	0.039
5	0.069	0.087	0.009	0.000	0.006	2.530	0.061	0.149
6	1.887	2.367	0.022	0.004	0.013	–	1.842	4.421
7	51.69	30.707	0.045	0.003	0.024	–	3.797	39.024
8	–	–	0.233	0.016	0.104	–	–	–
9	–	–	0.946	0.008	0.313	–	–	–
10	–	–	6.816	0.198	2.016	–	–	–
11	–	–	20.603	0.034	4.212	–	–	–

Média do tempo necessário para obter a distância. Para cada tamanho de permutação, foram usadas até 1000 instâncias de teste. O símbolo "–" é mostrado caso a bateria de testes não seja finalizada em 15 horas.

Publicação

- U. Dias and Z. Dias. Constraint programming models for transposition distance problem. In *Proceedings of the 4th Brazilian Symposium on Bioinformatics (BSB'2009)*, volume 5676 of *Lecture Notes in Computer Science*, pages 13–23, Porto Alegre, Brasil, 2009.

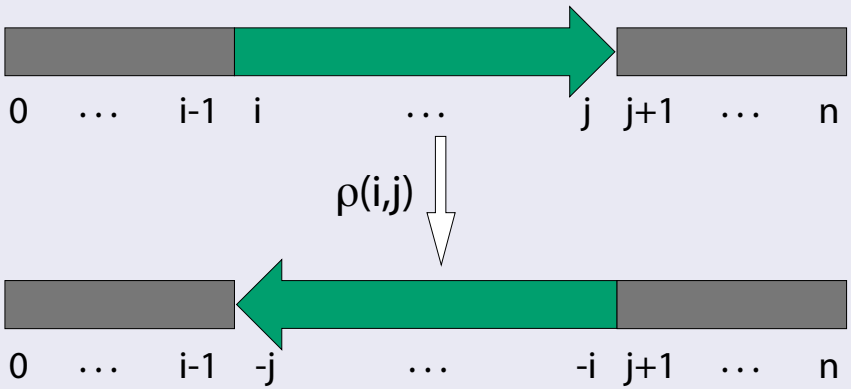
Parte II

Eventos de Reversão

Uma Ferramenta de Simulação para o Estudo de Reversões Simétricas em Genomas Bacteriais

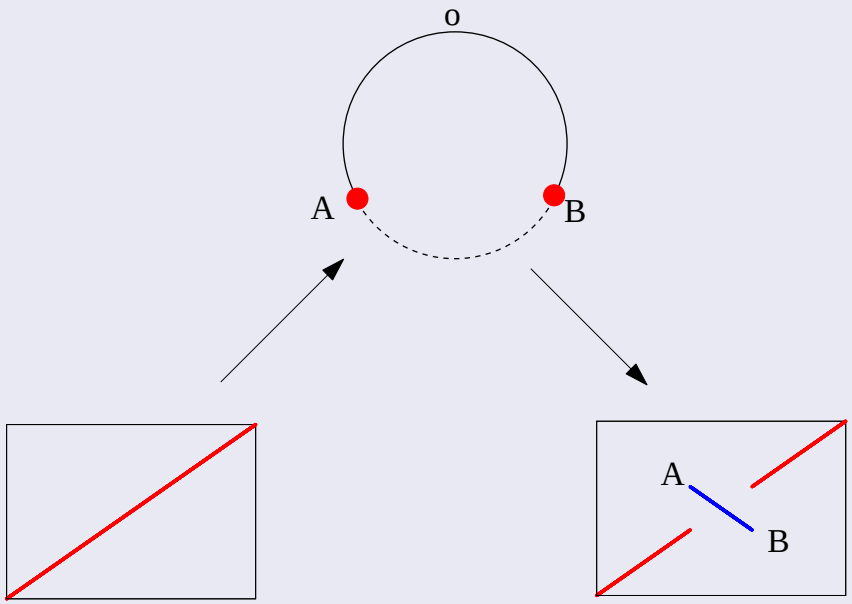
Reversões

- Eventos de reversão ocorrem quando um bloco é destacado do genoma e inserido no mesmo lugar, mas com a ordem e a orientação invertidas.



Reversões Simétricas

- Uma reversão é dita simétrica se os *breakpoints* estão igualmente distantes da origem de replicação.



Reversões Simétricas

Motivação

- Reversões simétricas são o principal mecanismo que explica o padrão em "X"
- Frequente em alinhamentos de organismos bacteriais relacionados.

Reversões Simétricas

Objetivos

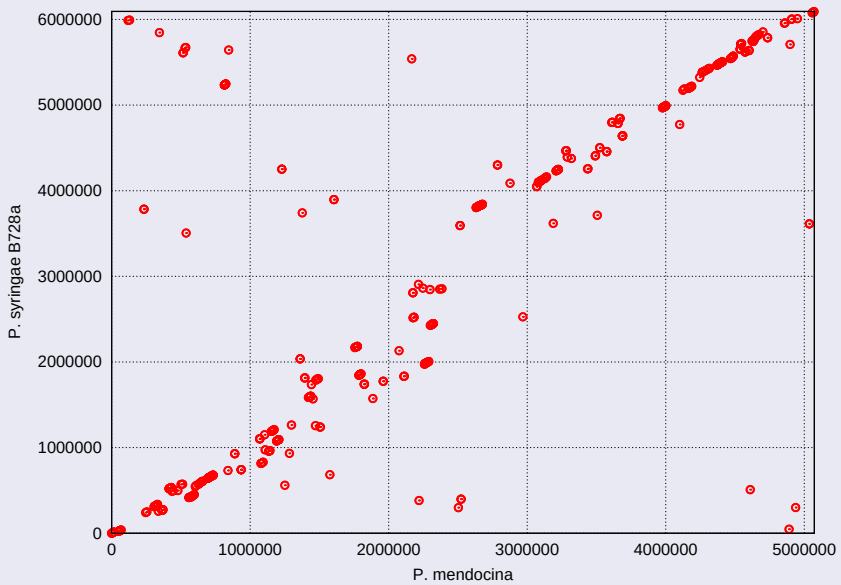
- Simular a evolução de genomas bacteriais em termos de reversões simétricas.
- O foco está na obtenção de *dotplots* que reproduzam a variedade de padrões em "X" que são observados.

Simulation Tool

- We used *Pseudomonadaceae* family to model SIB main features. Some reasons for this choice are:
  - The 'X' pattern is clearly observed in this family.
  - The family presents an interesting diversity of 'X' patterns.
  - There are 18 fully sequenced *Pseudomonadaceae* genomes.

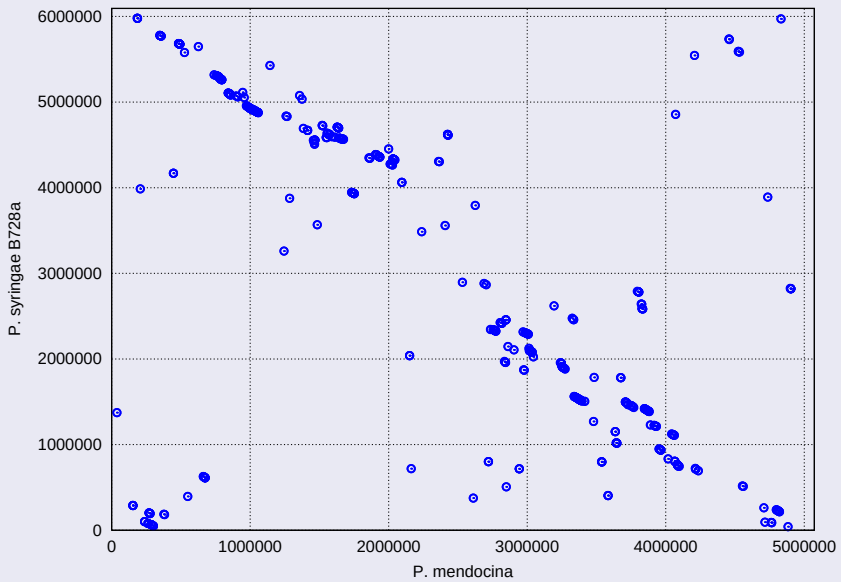
Recuperando informações dos *dotplots*

- A maioria dos blocos conservados entre dois genomas que possuem a mesma orientação (pontos vermelhos) ocorrem ao redor da reta r, que está próxima da diagonal principal.



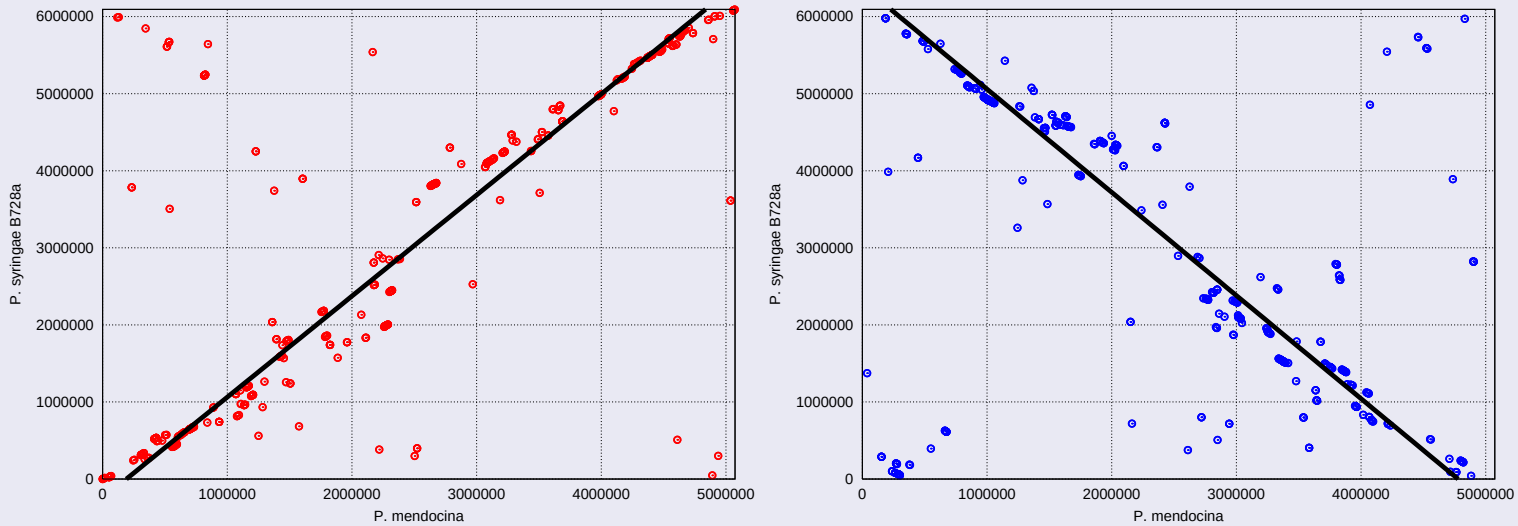
Recuperando informações dos *dotplots*

- A maioria dos blocos conservados entre dois genomas que possuem orientação contrária (pontos azuis) ocorrem ao redor da reta s, que está próxima da diagonal secundária.



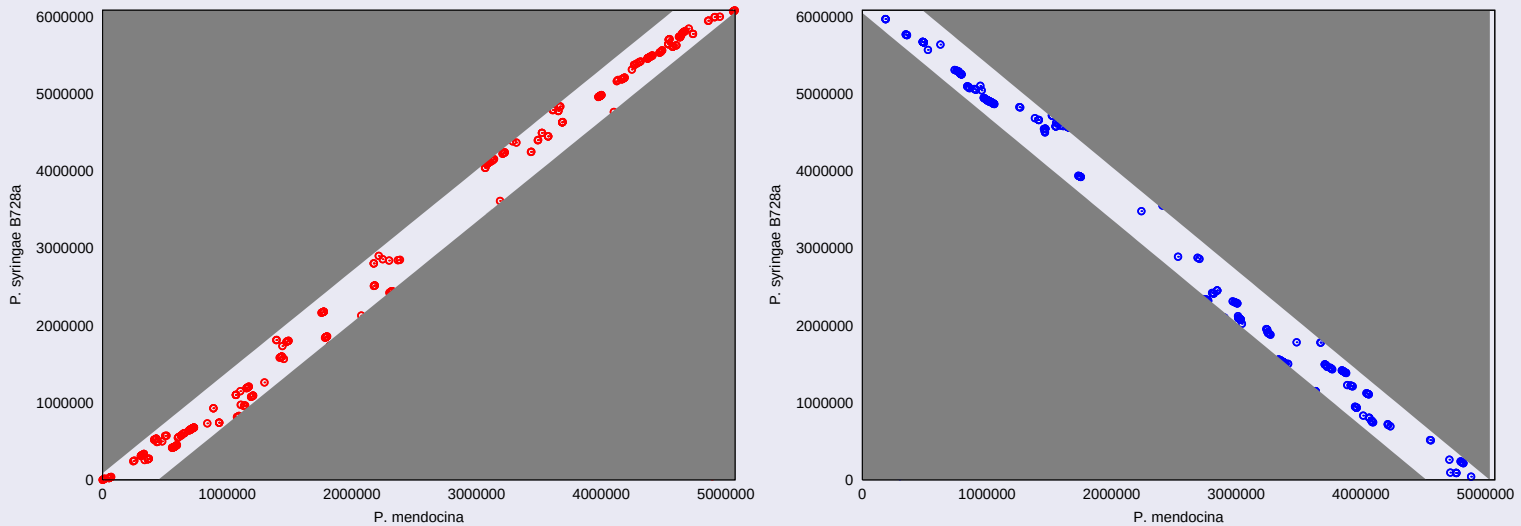
Recuperando informações dos *dotplots*

- As retas r e s são obtidas da seguinte maneira:
  - Calcular a subsequência (de)crescente mais longa dos pontos vermelhos (azuis).
  - Calcular as respectivas retas usando o método dos mínimos quadrados.



Recuperando informações dos *dotplots*

- Chamamos de R_1 a região ao r e R_2 a região ao redor de s. Tanto R_1 quanto R_2 possuem 10% da área total do dotplot.



Recuperando informações dos *dotplots*

- Define-se d_1, d_2 e d_3 como:
  - d_1: número de pontos vermelhos em R_1 dividido pelo total de pontos no *dotplot*.
  - d_2: número de pontos azuis em R_2 dividido pelo total de pontos no *dotplot*.
  - d_3: número de pontos que não foram contabilizados nem em d_1 e nem em d_2 dividido pelo total de pontos no *dotplot*.

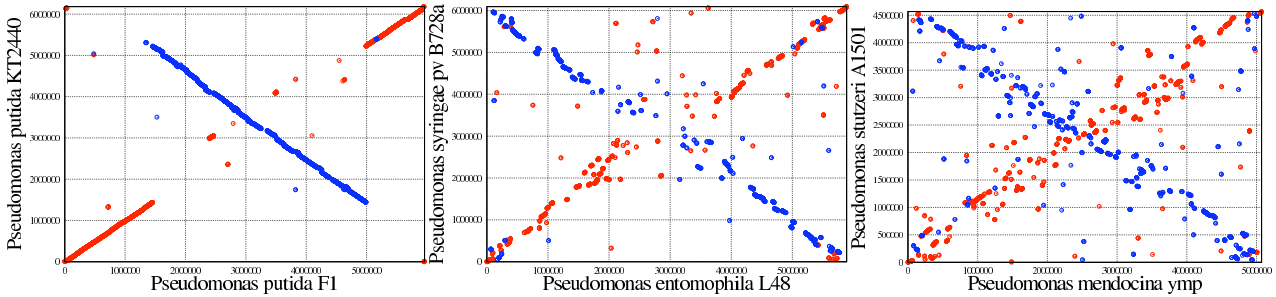
Comparação entre *dotplots*

- A distância euclidiana entre as densidades d_1, d_2 e d_3 é usada para comparar *dotplots*.
  - Sejam A e B dois *dotplots*.
  - Sejam $d_A = \{d_{A1}, d_{A2}, d_{A3}\}$ e $d_B = \{d_{B1}, d_{B2}, d_{B3}\}$ as densidades associadas a A e B, respectivamente.
  - A distância entre A e B é dada por :

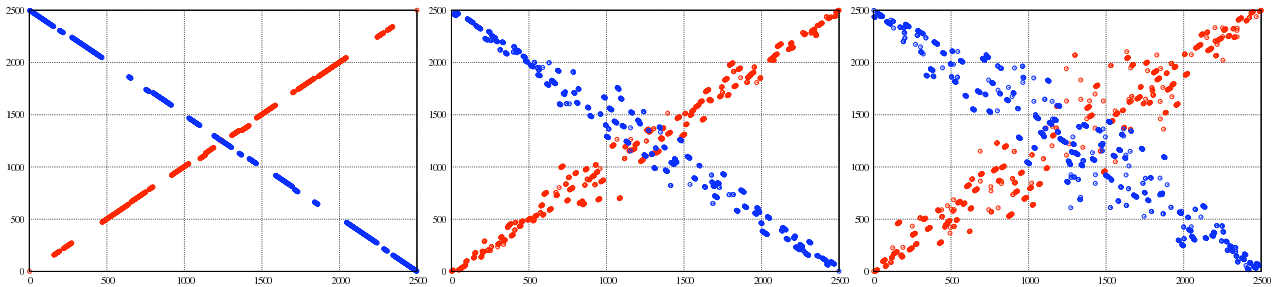
$$dist(A, B) = \sqrt{(d_{A1} - d_{B1})^2 + (d_{A2} - d_{B2})^2 + (d_{A3} - d_{B3})^2}$$

Comparação entre *dotplots*

- Dado um *dotplot* real, a distância euclidiana possibilita encontrar o *dotplot* simulado mais próximo.
  - *Dotplots* reais

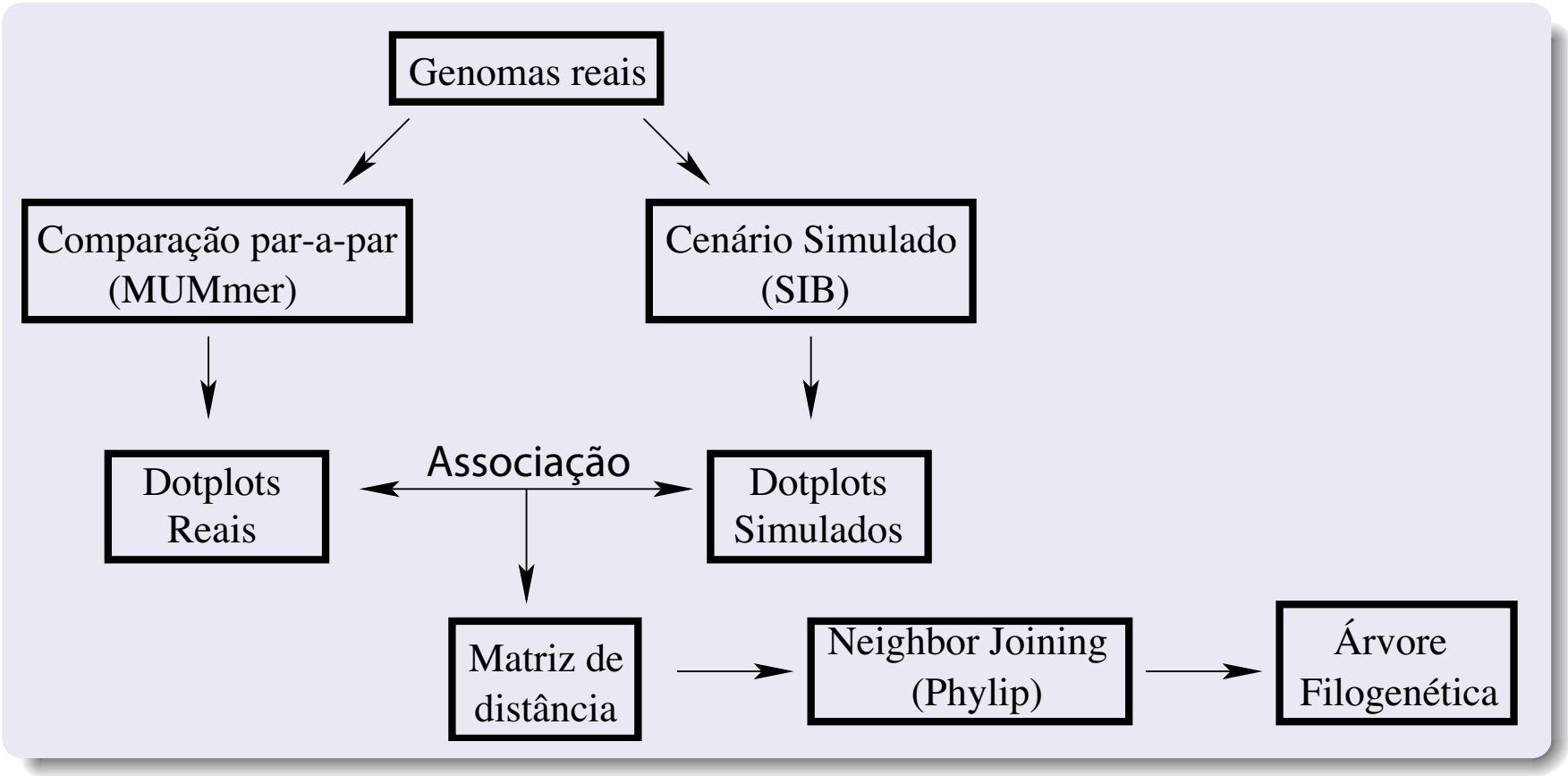


  - *Dotplots* simulados

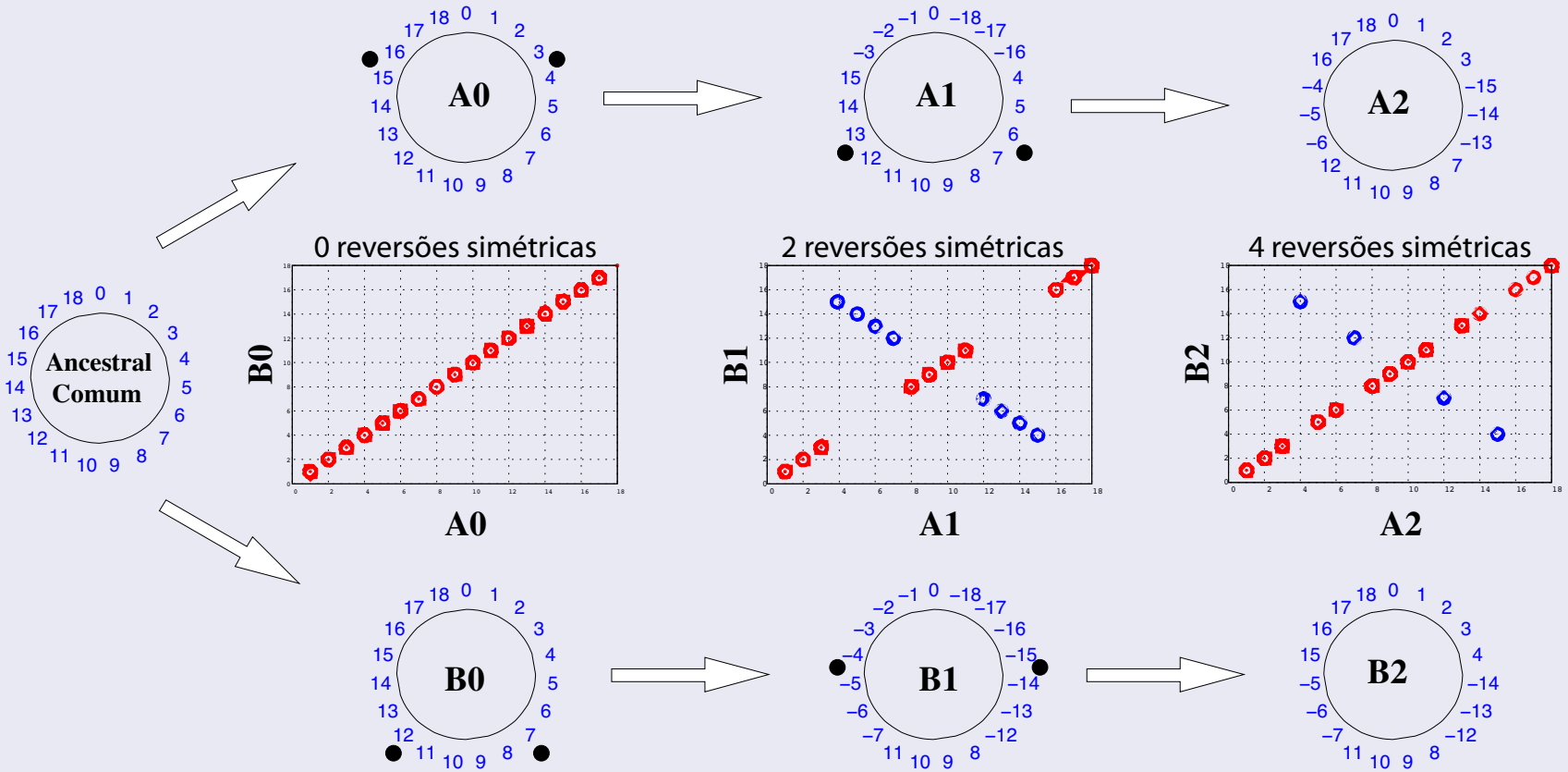


Validação do SIB

- Pipeline para geração de árvores filogenéticas

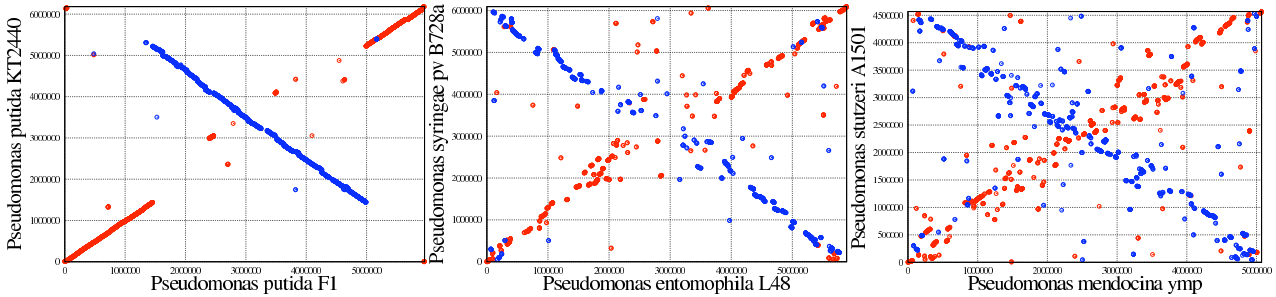


Cenário Simulado

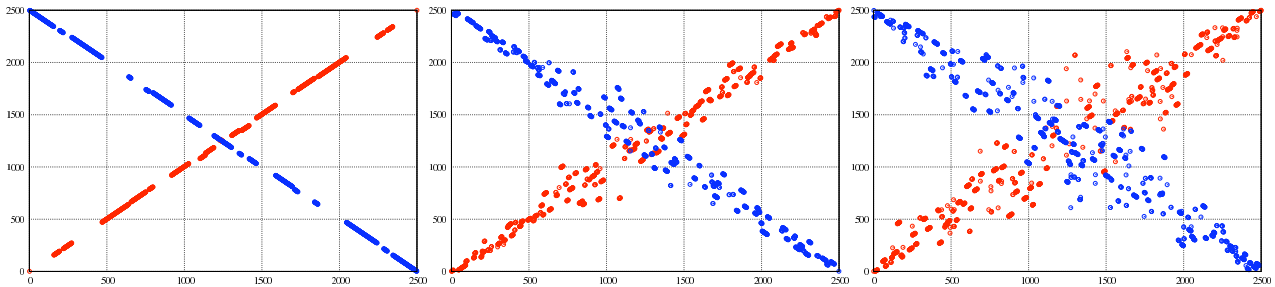


Validação do SIB

- Cada *dotplots* real é associado com o *dotplot* simulado mais próximo.
  - *Dotplots* reais



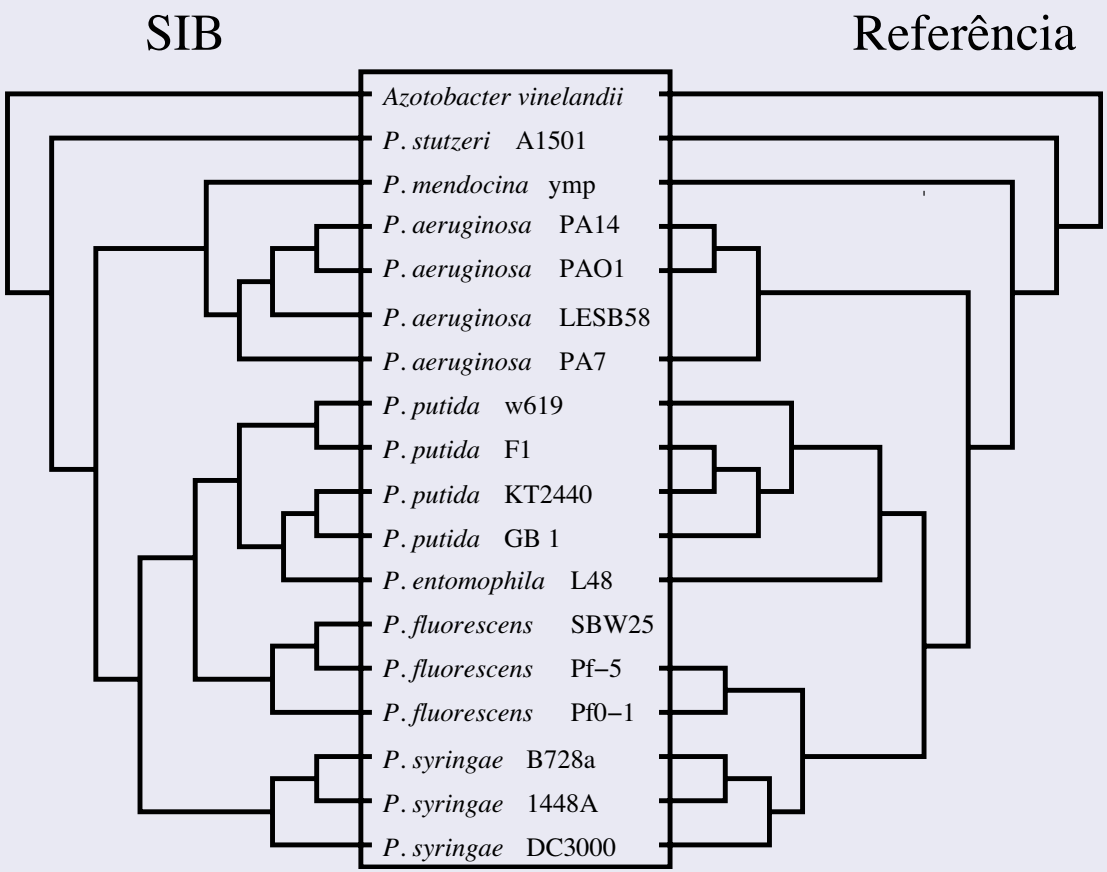
  - *Dotplots* simulados



Resultados

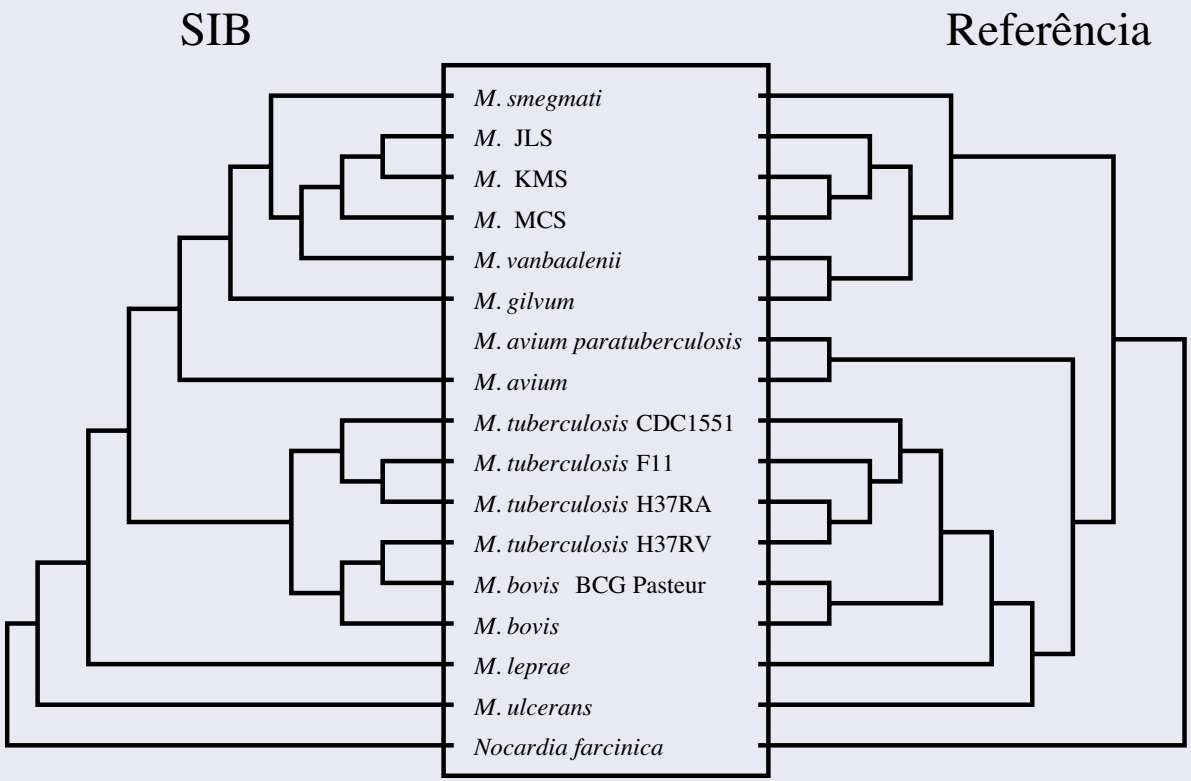
- As árvores geradas com o SIB foram comparadas com árvores de referência propostas na literatura.
- Foram usados organismos da família *Pseudomonadaceae* e dos gêneros *Mycobacterium*, *Shewanella*, *Xanthomonas*.
- O `p-value` para a comparação com a referência é dado abaixo.
  - *Pseudomonadaceae*: $7{,}0 \times 10^{-6}$
  - *Shewanella*: $1{,}3 \times 10^{-2}$
  - *Xanthomonas*: $1{,}3 \times 10^{-2}$
  - *Mycobacterium*: $7{,}5 \times 10^{-5}$
- O `p-value` está de acordo com uma inspeção visual pelas figuras a seguir.

Resultados



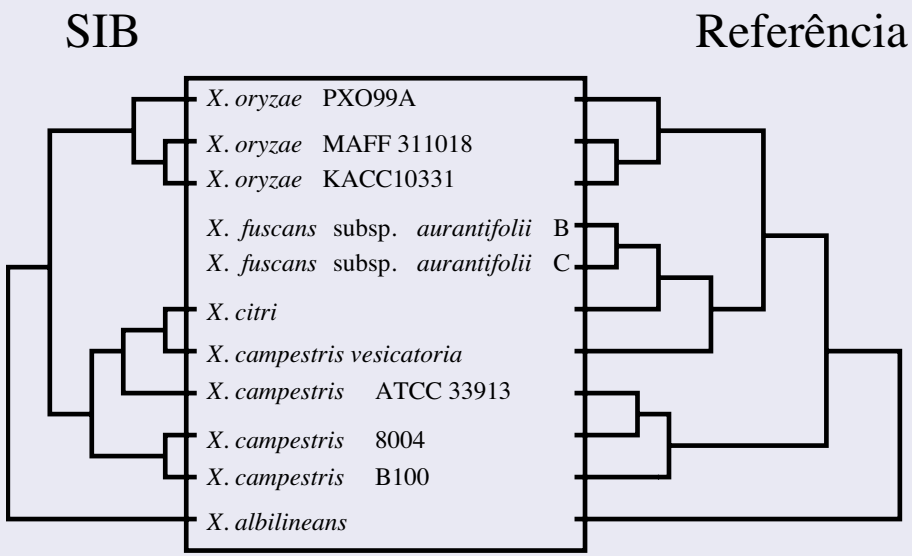
Comparação entre a árvore da família *Pseudomonadaceae* produzida pelo SIB e a árvore proposta por Setubal et al, 2009.

Resultados

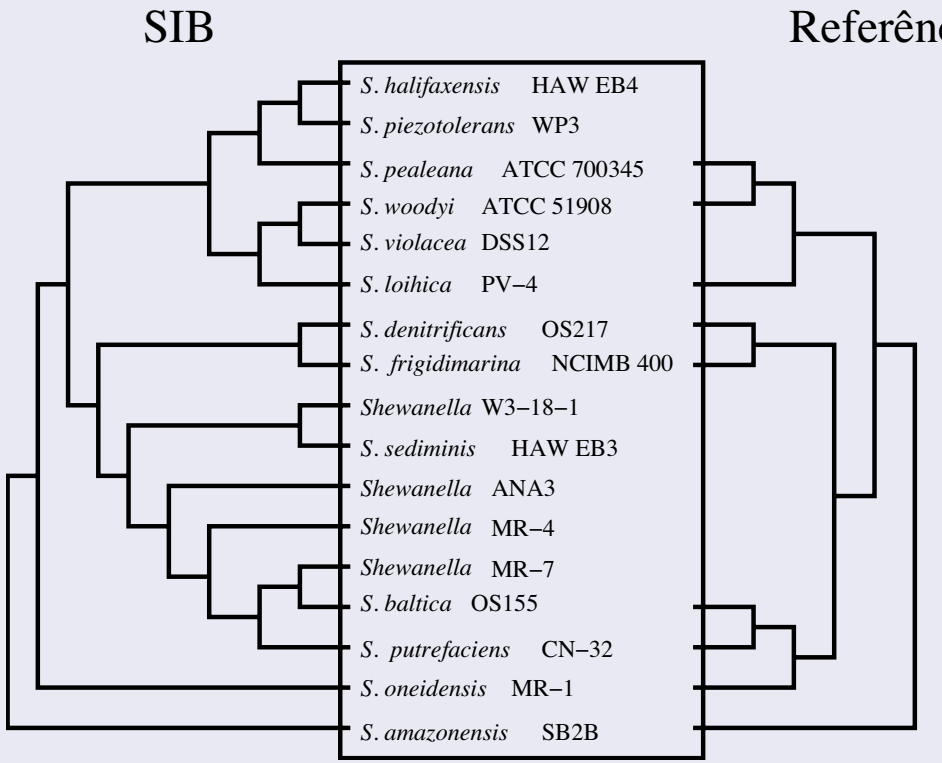


Comparação entre a árvore do grupo *Mycobacterium* produzida pelo SIB e a árvore proposta por Alam et al, 2010.

Resultados



Comparação entre a árvore do grupo *Xanthomonas* produzida pelo SIB e a árvore proposta por Moreira et al, 2010.



Comparação entre a árvore do grupo *Shewanella* produzida pelo SIB e a árvore proposta por Williams et al, 2010.

Publicação

- U. Dias, Z. Dias, and J. C. Setubal. A simulation tool for the study of symmetric inversions in bacterial genomes. In *Proceedings of the 18th Annual RECOMB Satellite Workshop on Comparative Genomics (RECOMB-CG'2010)*, pages 240 – 251, Ottawa, Canada, 2010.

Distância de Reversão Quase-Simétrica

Distância de Reversão Quase-Simétrica

Motivação

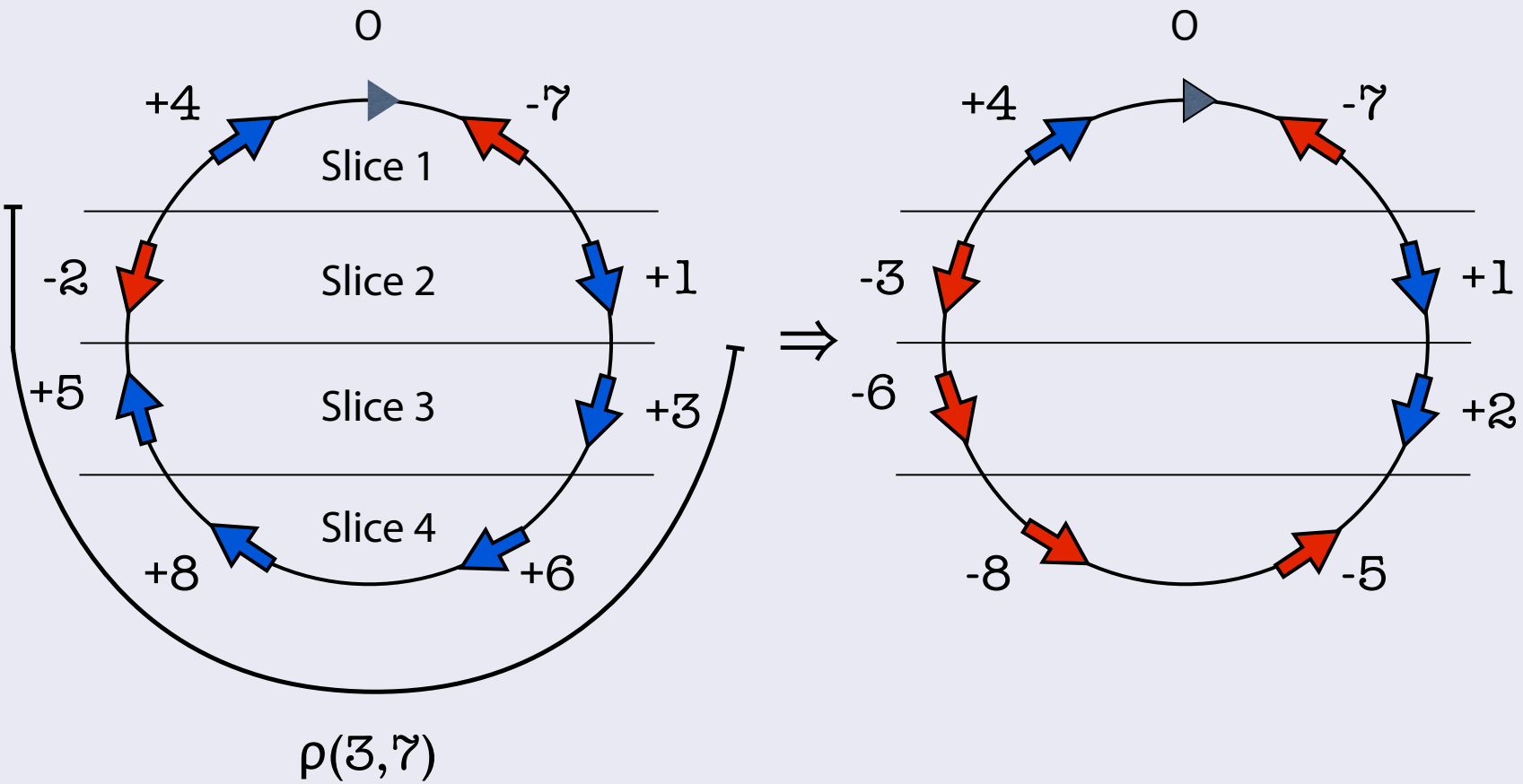
- A ocorrência de simetria nas reversões reforça a necessidade por soluções algorítmicas para o problema.
- Algoritmos de reversão produzem resultados incorretos quando usados para calcular o genoma ancestral nos cenários em que o evento de reversão é predominantemente simétrico.

Objetivos

- Apresentar o problema de ordenação por Reversões Quase-Simétricas.
- Fornecer a base para trabalhos futuros no contexto de algoritmos de aproximação e heurísticas.

Distância de Reversão Quase-Simétrica

- Problema das Reversões Quase-Simétricas.

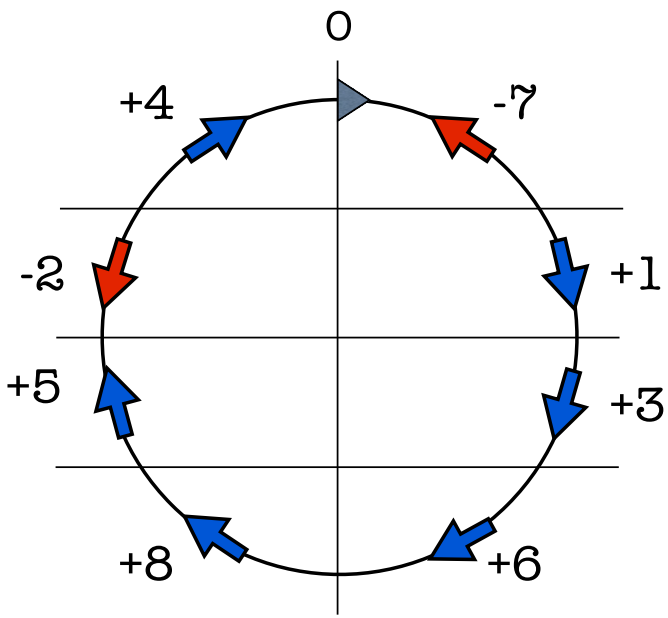


Distância de Reversão Quase-Simétrica

Position, Sign e Slice

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

Position, Sign e Slice



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

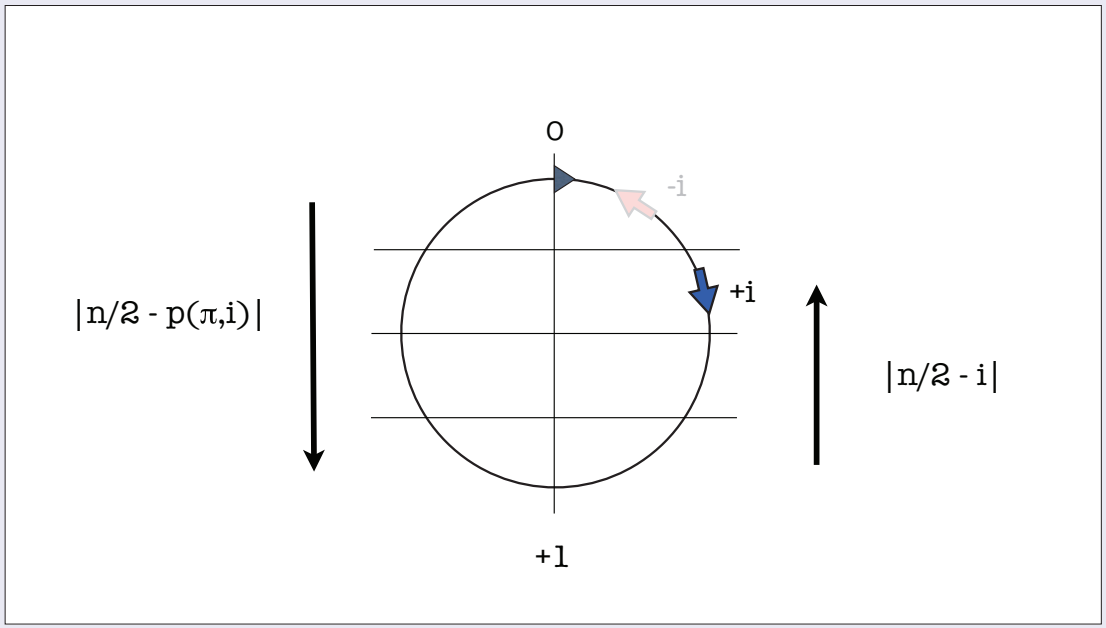
Posicionar um único elemento

- Define-se $d_\pi[i]$ como o número mínimo de reversões quase-simétricas que devem agir no elemento i para ordenar a permutação π.
- Seja $\overline{\rho}$ uma reversão quase-simétrica. Se $slice(\pi, j) = k$, então $slice(\pi\overline{\rho}, j) \in \{k-1, k, k+1\}$.
- Se $slice(\pi, i) = j$ e $slice(\iota, i) = k$, então um total de $d_\pi[i] \geq |j - k|$ reversões quase-simétricas devem agir no elemento i para posicioná-lo adequadamente.

Distância de Reversão Quase-Simétrica

- Em certos casos, o caminho direto resultaria em um posicionamento com sinal negativo.
- É preciso mover o elemento até a região diametralmente oposta à origem de replicação e depois mover ao local desejado.

Permutações de tamanho par

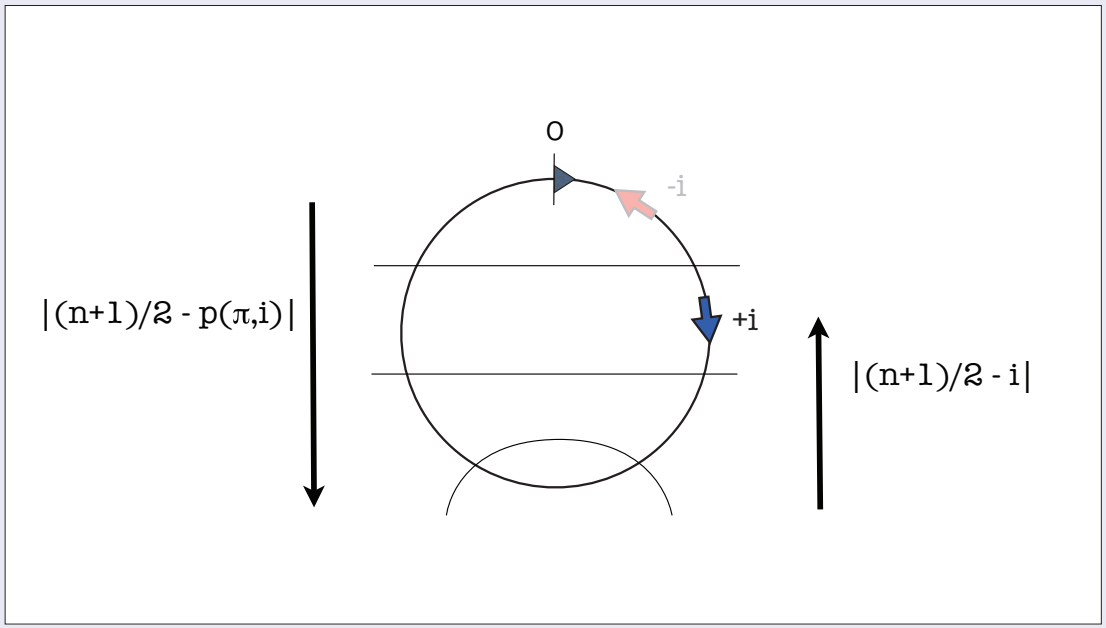


- $d_\pi[i] = |\frac{n}{2} - p(\pi, i)| + |\frac{n}{2} - i| + 1$

Distância de Reversão Quase-Simétrica

- Em certos casos, o caminho direto resultaria em um posicionamento com sinal negativo.
- É preciso mover o elemento até a região diametralmente oposta à origem de replicação e depois mover ao local desejado.

Permutações de tamanho ímpar



- $d_\pi[i] = |\frac{(n+1)}{2} - p(\pi, i)| + |\frac{(n+1)}{2} - i|$

Distância de Reversão Quase-Simétrica

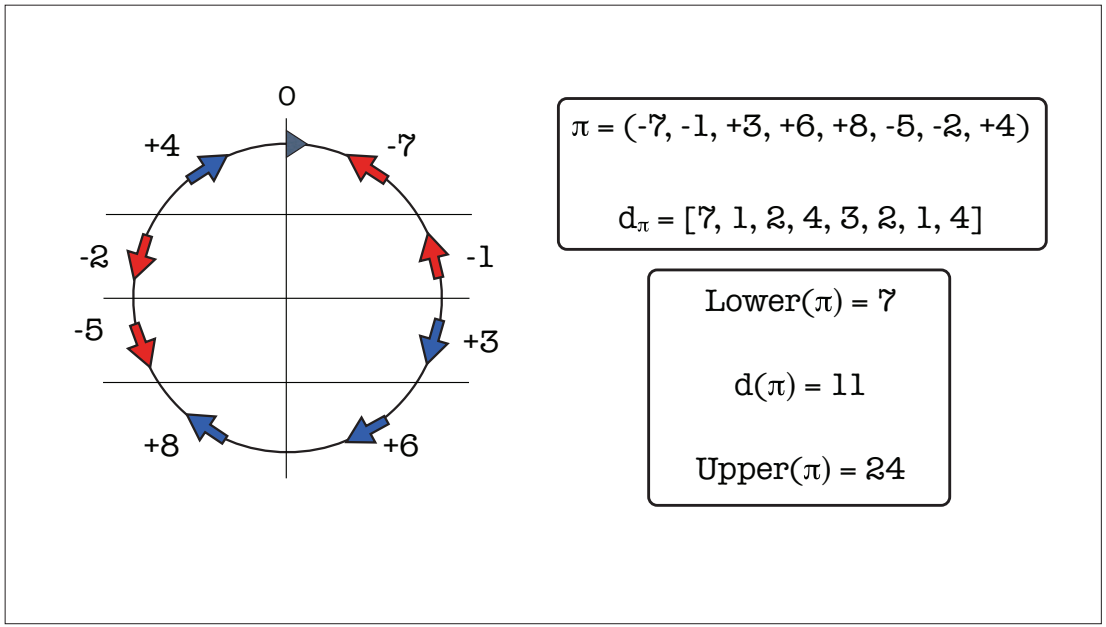
- Seja π um genoma com número par de genes ($n = 2k$), $d_\pi[i] = |\frac{n}{2} - p(\pi, i)| + |\frac{n}{2} - i| + 1$ se um dos itens a seguir for verdade.
  - $i \leq \frac{n}{2}$, $p(\pi, i) \leq \frac{n}{2}$ e $s(\pi, i) = -1$
  - $i \leq \frac{n}{2}$, $p(\pi, i) > \frac{n}{2}$ e $s(\pi, i) = +1$
  - $i > \frac{n}{2}$, $p(\pi, i) \leq \frac{n}{2}$ e $s(\pi, i) = +1$
  - $i > \frac{n}{2}$, $p(\pi, i) > \frac{n}{2}$ e $s(\pi, i) = -1$

Distância de Reversão Quase-Simétrica

- Seja π um genoma com um número ímpar de genes $(n = 2k + 1)$, $d_\pi[i] = |\frac{(n+1)}{2} - p(\pi, i)| + |\frac{(n+1)}{2} - i|$ 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 $s(\pi, i) = -1$

Distância de Reversão Quase-Simétrica

- Seja i um elemento com sinal positivo em sua 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 $d_\pi[i] \geq 2$.
- $Lower(\pi) = max_{1 \leq i \leq n} d_\pi[i]$.
- $Upper(\pi) = \sum_{1 \leq i \leq n} d_\pi[i]$. (conjectura)



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

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 dentre (menor número de reversões):

- 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

Algoritmo 3: Algoritmo Guloso Básico

Data: π
$k \leftarrow 0$
while $\pi \neq \iota$ **do**
    $\overline{\rho} \leftarrow \overline{\rho}'$ que minimiza $\min\{d[i], d[n-i-1]\} + d[i] + d[n-i+1]$, para $1 \leq i \leq \lceil \frac{n}{2} \rceil$, em $\overline{\rho}'\pi$ para todo $\overline{\rho}'$ aplicável a π
    $\pi \leftarrow \overline{\rho}\pi$
    $k \leftarrow k + 1$
end
return k

Resultados

	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%

Resultados

	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%

Resultados

	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%

- 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 $n \geq 5$.

Resultados

N	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	D
1	1	1	0	1								**1**
2	1	1	2	2	3							**3**
3	3	3	2	4	3			5				**5**
4	3	3	4	4	5	5		6	5			**6**
5	5	5	4	6	5	7	6	7	7	8	7	**8**
6	5	5	6	6	7	7	8	8	8	9	10	**10**
7	7	7	6	8	7	9	8	9	10	11	10	**11**
8	7	7	8	8	9	9	10	10	11	12	13	**13**
9	9	9	8	10	9	11	10	11	13	14	13	**15**
10	9	9	10	10	11	11	12	12	14	14	16	**16**

Lista de distâncias fornecidas pelas conjecturas sobre as famílias F_1 a F_{10}. Todos os valores conjecturados na tabela coincidem com os valores exatos de distância para $N \leq 10$. A última coluna mostra o valor do diâmetro para cada valor de N.

Publicação

- Z. Dias, U. Dias, J. C. Setubal, and L. S. Heath. Sorting genomes using almost-symmetric inversions. In *Proceedings of the 27th Symposium On Applied Computing (SAC'2012)*, pages 1–7, Riva del Garda, Italy, 2012.

Geração de Scaffolds usando Assinaturas de Reversão

Scaffolds usando Assinaturas de Reversão

Motivação

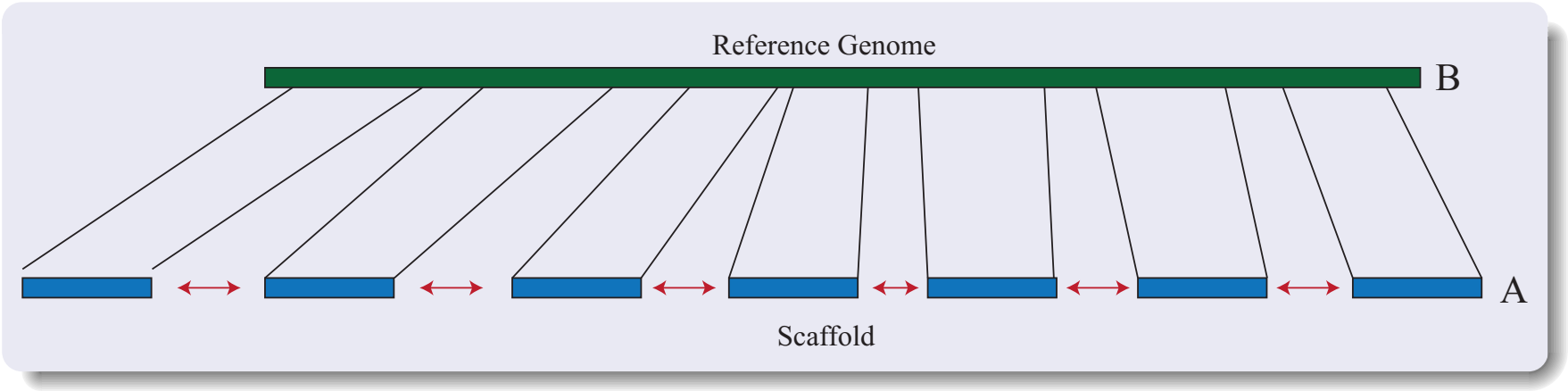
- Sequenciar genomas até o estágio *draft* diminui os custos.
- Nesse estágio, o resultado é um grupo de subsequências chamadas de contigs.
- Construir um *scaffold* consiste em determinar a ordem a orientação desses contigs.

Objetivo

- Desenvolver um algoritmo para construção de scaffolds usando genoma de referência.
- O algoritmo deve levar em consideração a ocorrência de eventos de reversões entre o genoma *draft* e o genoma de referência.

Construção de *scaffolds* usando genoma de referência

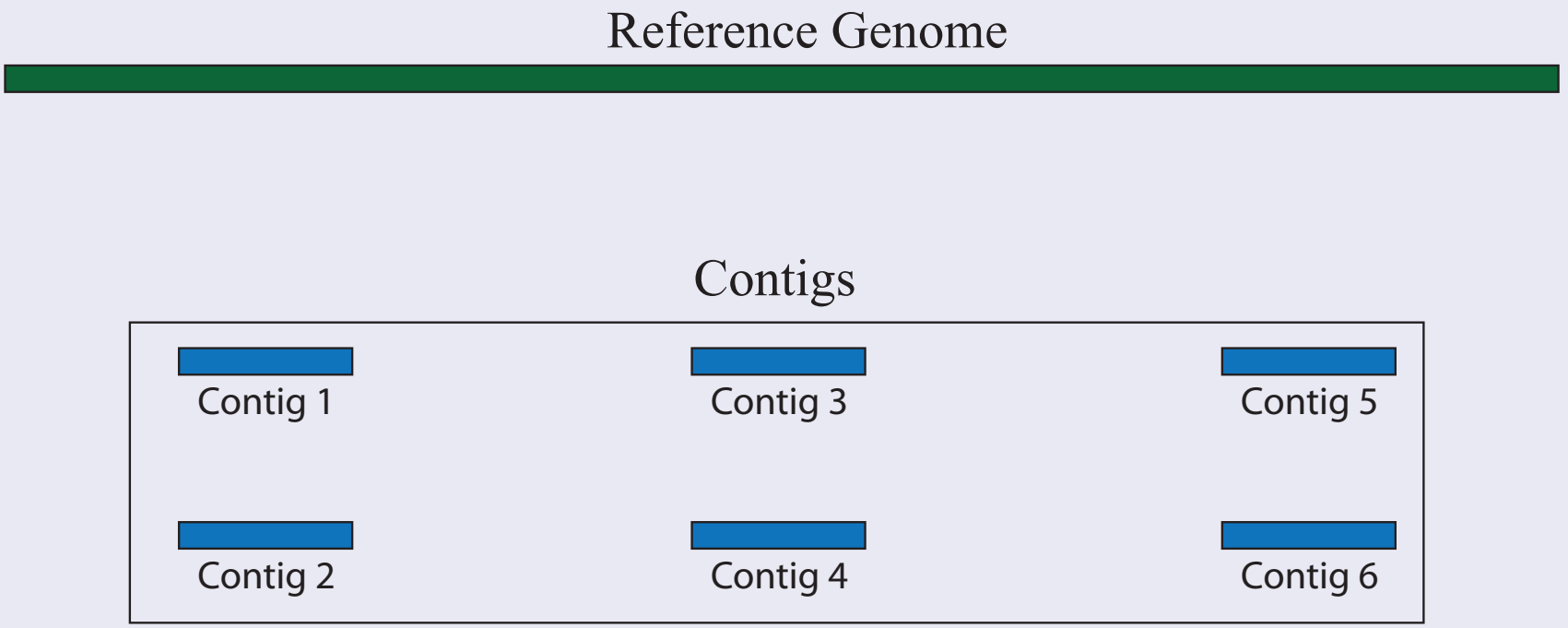
- Dois organismos A e B são próximos na árvore evolutiva.
  - O genoma de A está incompleto.
  - O genoma de B já foi completamente sequenciado.
- O genoma de B pode guiar a construção do *scaffold* de A.
  - A abordagem comum consiste em mapear os contigs no genoma de referência, mas apenas funciona bem com genomas colineares.



Construção de *scaffolds* usando genoma de referência

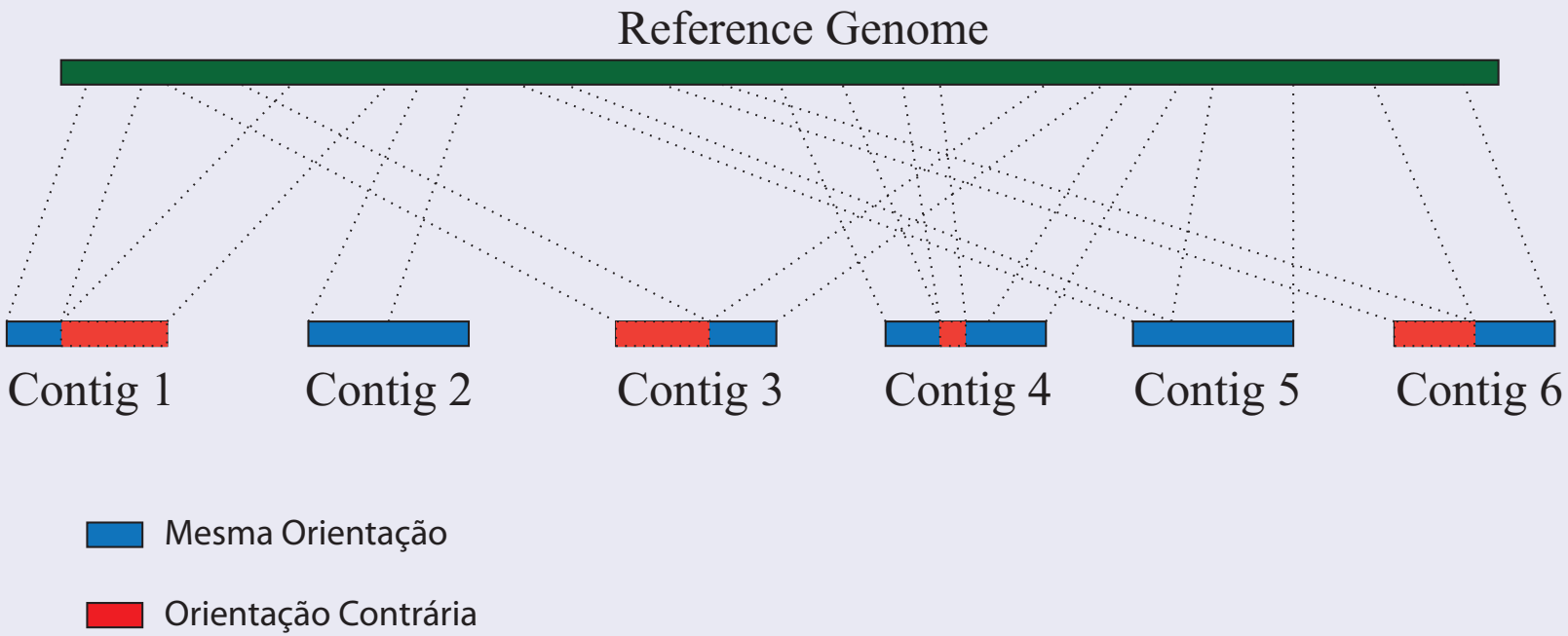
- Neste trabalho, apresentamos um algoritmo para obtenção de *scaffolds* que leva em consideração a presença de reversões.
- O algoritmo recebe como entrada dois genomas, sendo um incompleto e outro completo.

Dados de Entrada



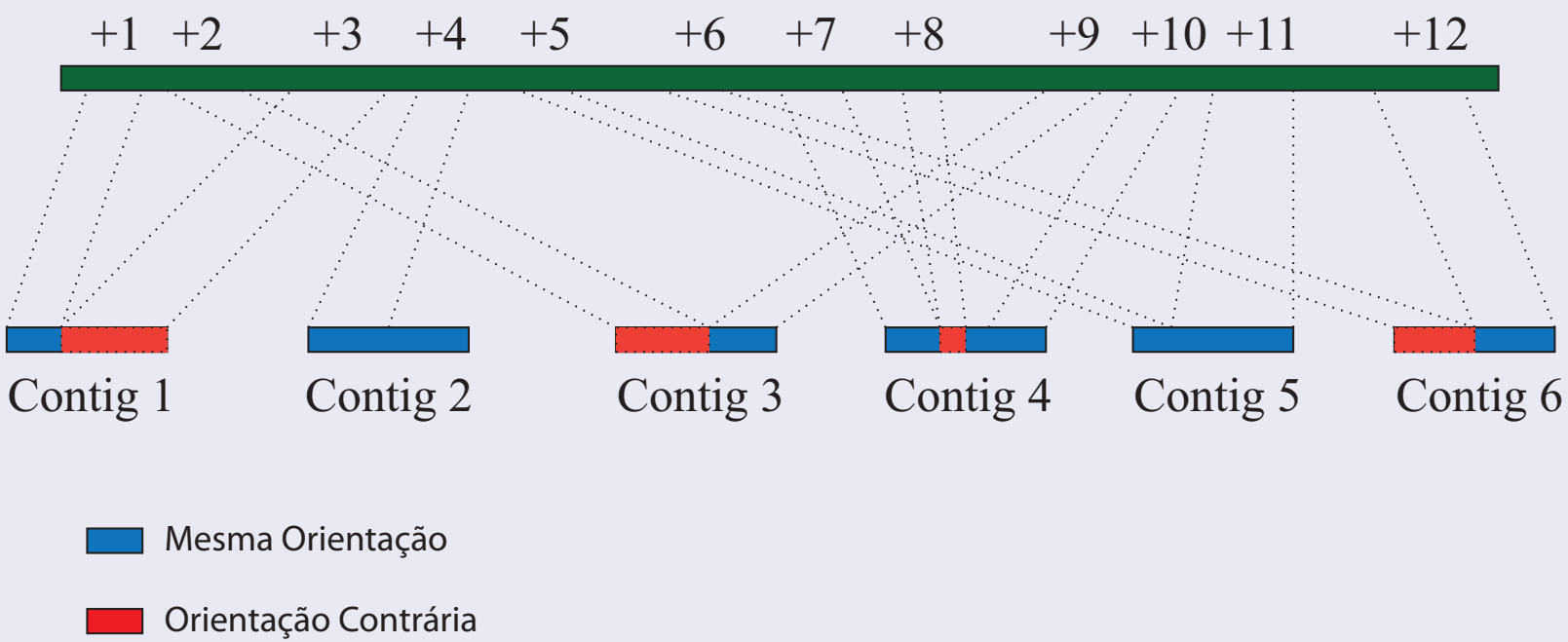
1 - Detecção de blocos conservados

- MUMmer, Mauve ou Blast podem ser usados.



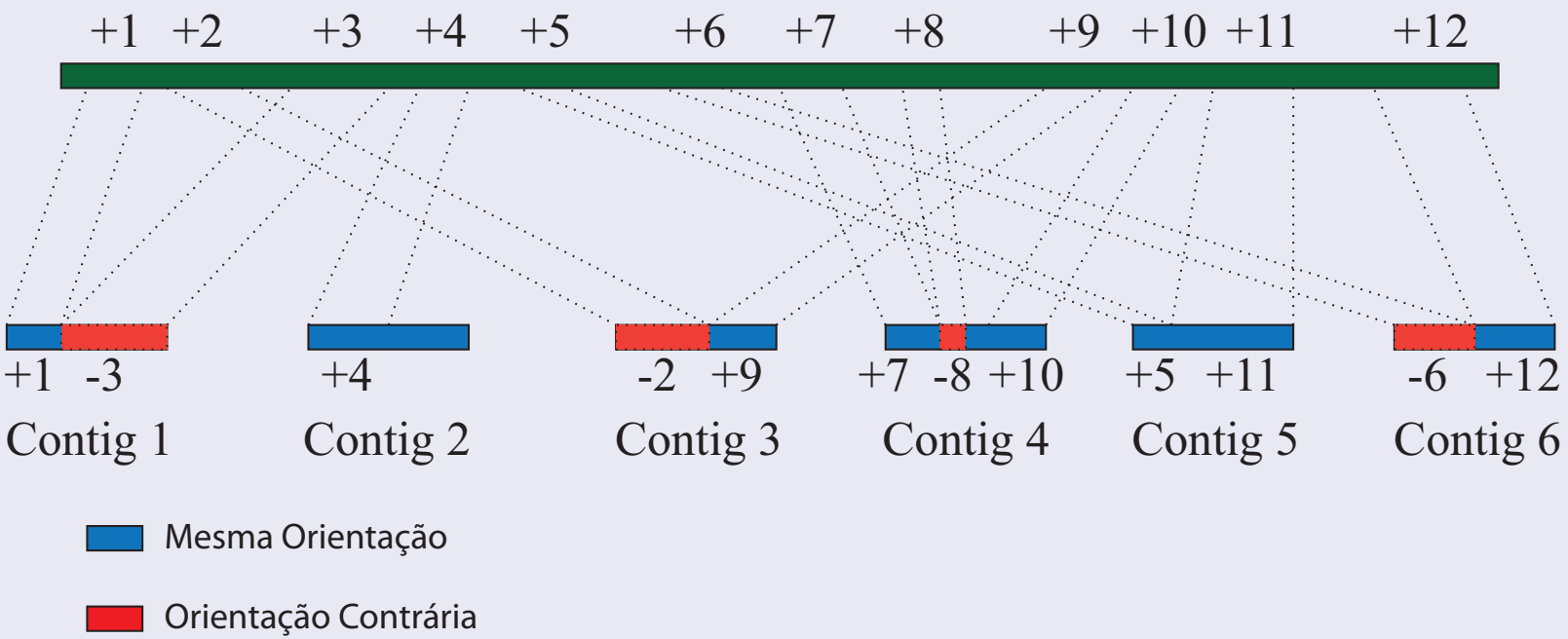
2 - Rotular os blocos conservados do genoma de referência.

- São rotulados como uma permutação identidade.



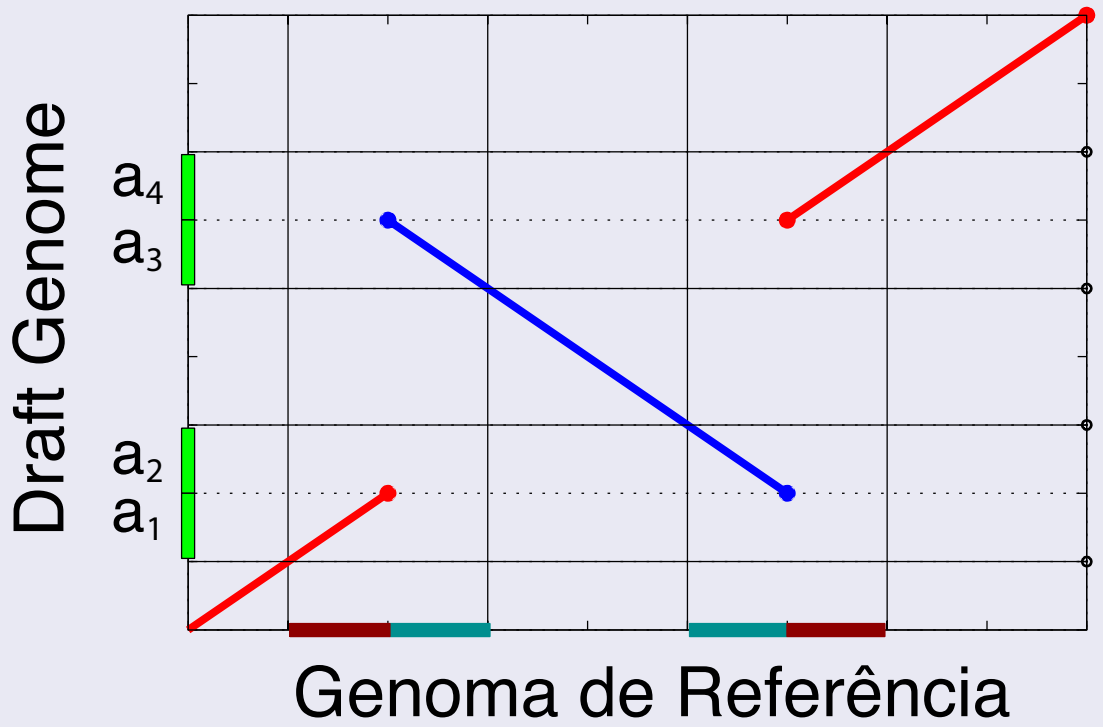
3 - Rotular os blocos conservados do genoma incompleto.

- São rotulados de acordo com a referência.



Assinaturas de Reversão

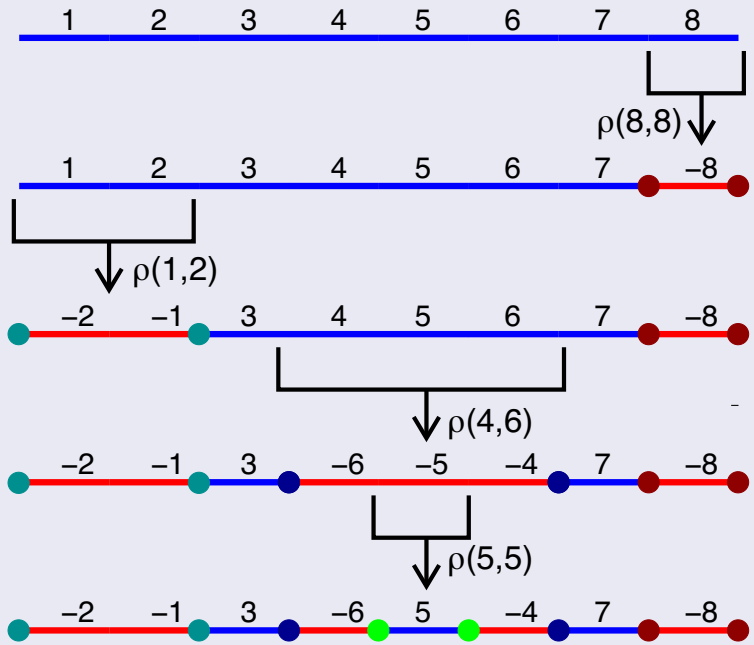
- Uma assinatura de reversão (IS) é um breakpoint (π_i, π_{i+1}) tal que π_i e π_{i+1} possuem sinais distintos.
- Um par de assinaturas (π_i, π_{i+1}) e (π_j, π_{j+1}) formam um IS-Pair se $\pi_i = -\pi_j + 1$ e $\pi_{i+1} = -\pi_{j+1} + 1$,

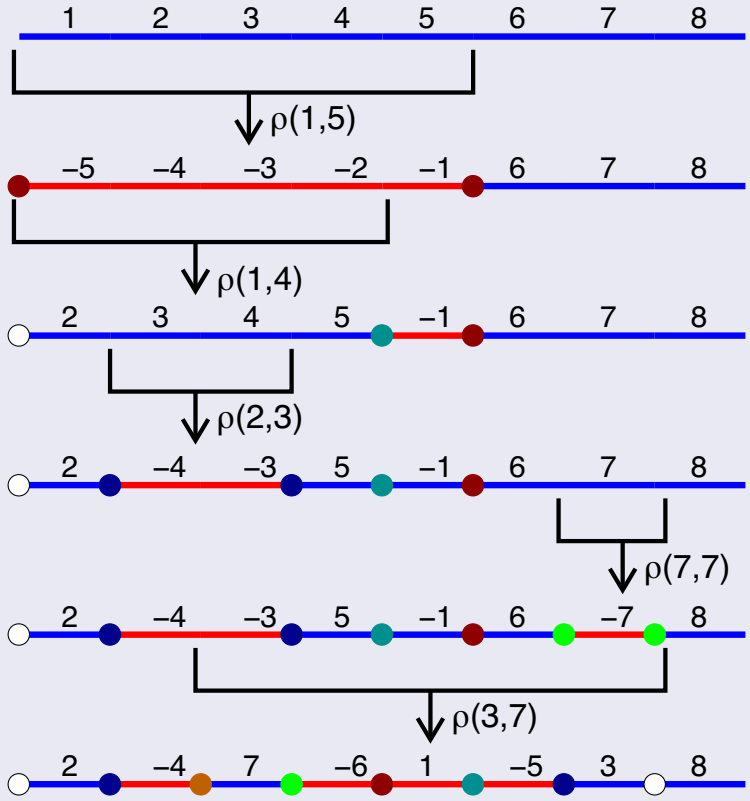


Assinaturas de Reversão

- As assinaturas são observáveis dependendo das inversões que podem ocorrer.
  - Simétricas
  - Aninhadas
  - Seguras
  - Genéricas

Reversões Seguras X Reversões Genéricas

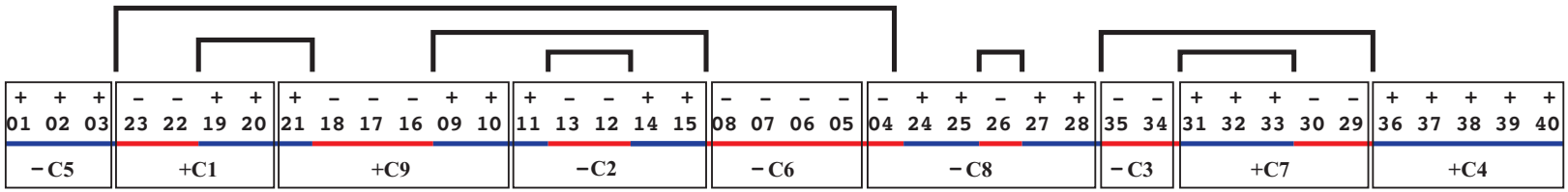




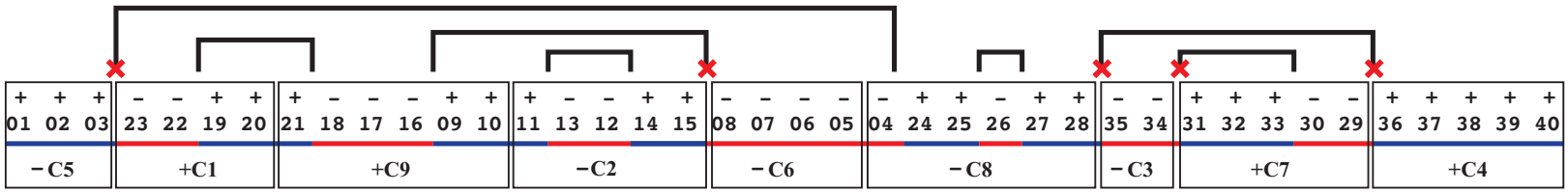
Algoritmo

- Iniciar com o contig mais à esquerda (contig com o bloco 1).
- Adicionar novos contigs, usando as assinaturas quando necessário.
- Quando reversões genéricas ocorrem, pode ser necessário prosseguir com o algoritmo escolhendo um contig que ainda não foi posicionado.

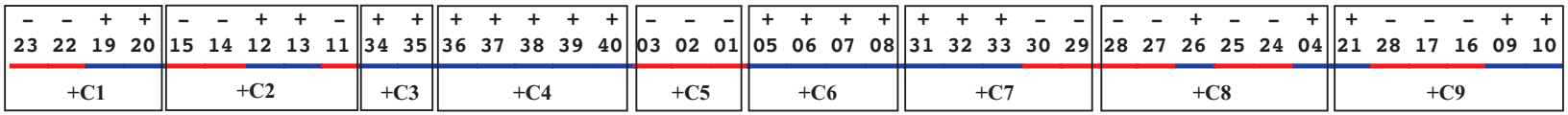
Exemplo (Histórico de reversões)



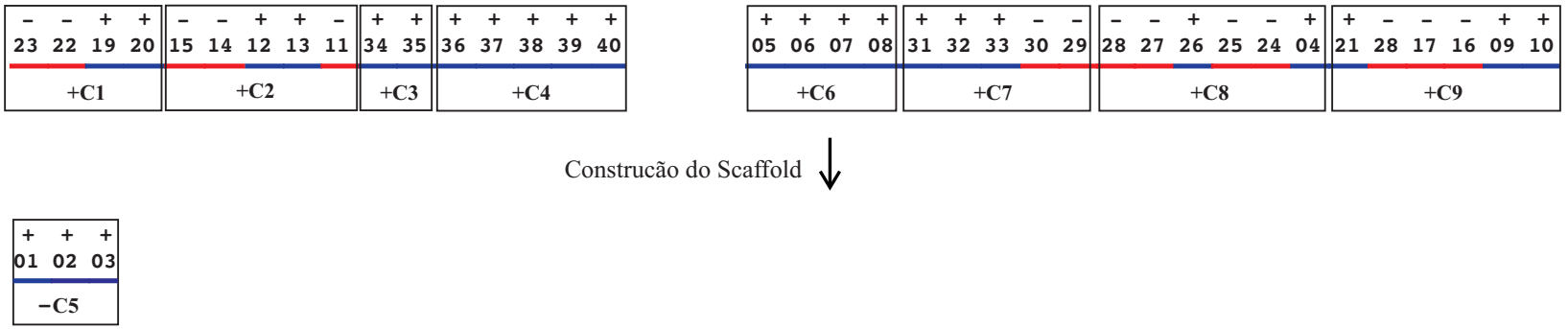
Exemplo (Assinaturas não identificadas)



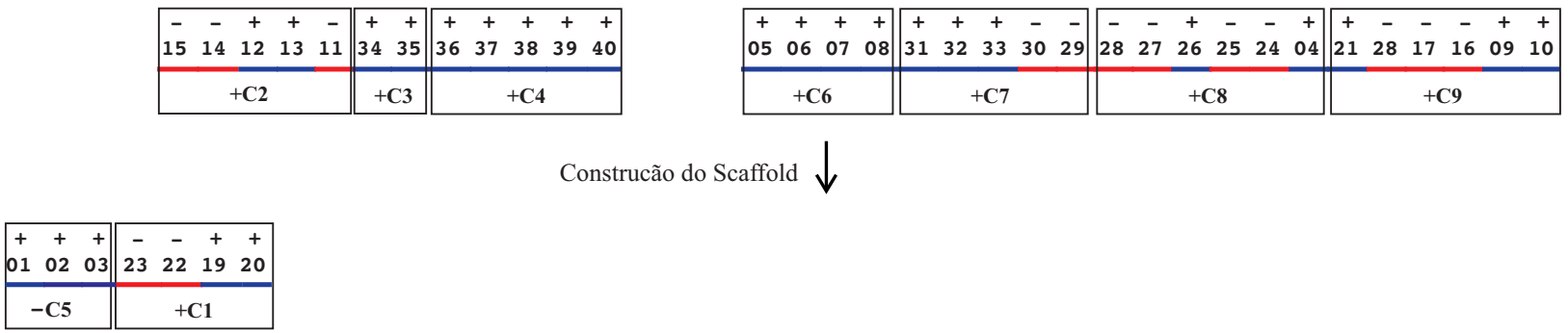
Exemplo (Configuração Inicial)



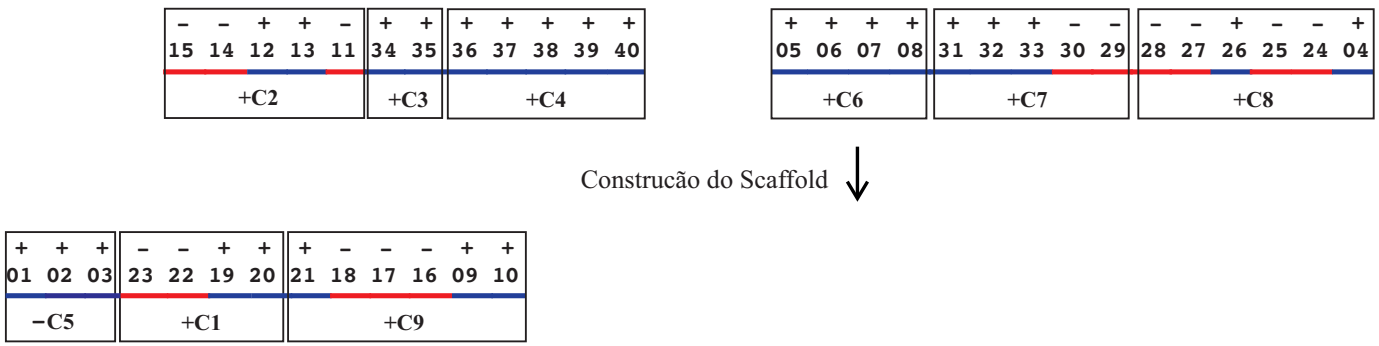
Adicionando o bloco (+1)



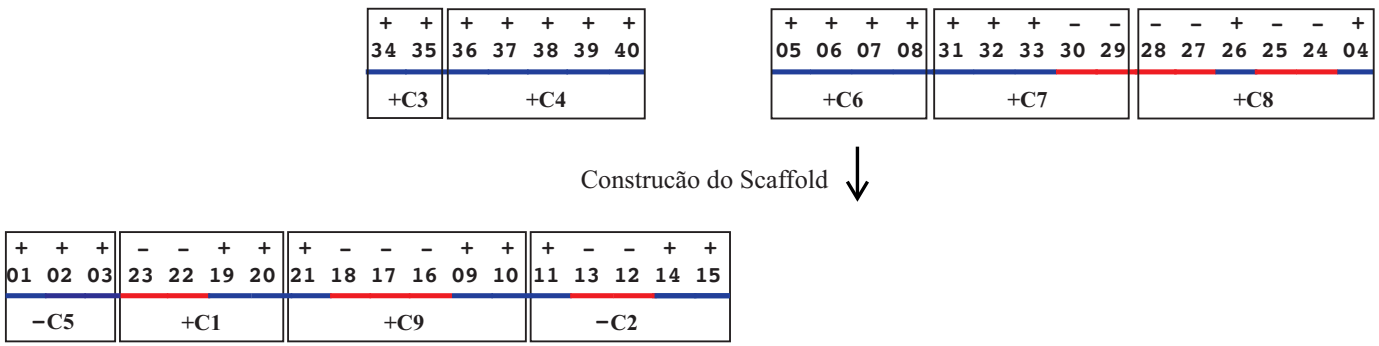
Usando o IS $[-04, +24]$



Adicionando o próximo bloco na sequência (Nenhum IS detectado)

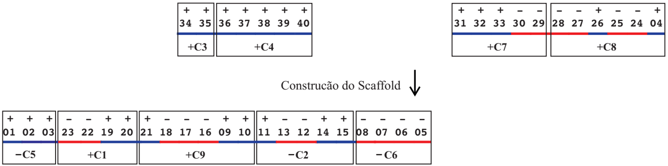


Adicionando o próximo bloco na sequência (Nenhum IS detectado)



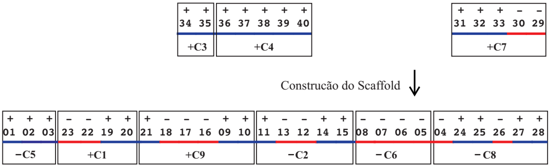
Algoritmo

Usando o IS $[-16, +09]$



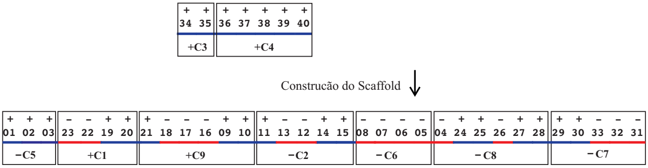
Algoritmo

Adicionando o próximo bloco na sequência (Nenhum IS detectado)

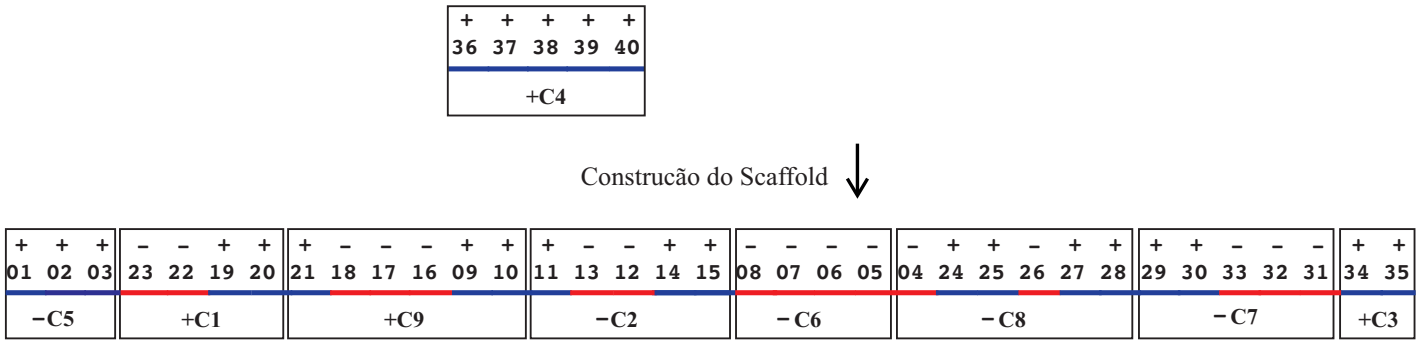


Algoritmo

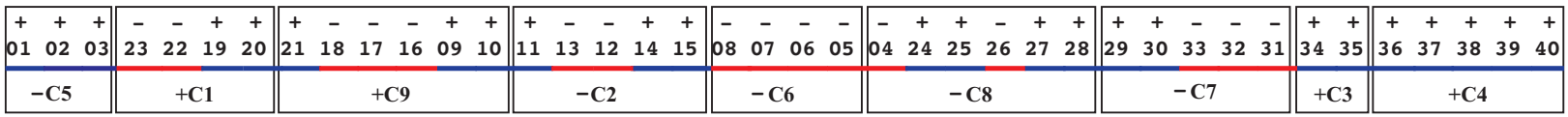
Nenhum IS detectado: o par $([+28,-35],[-29,+36])$ está ausente



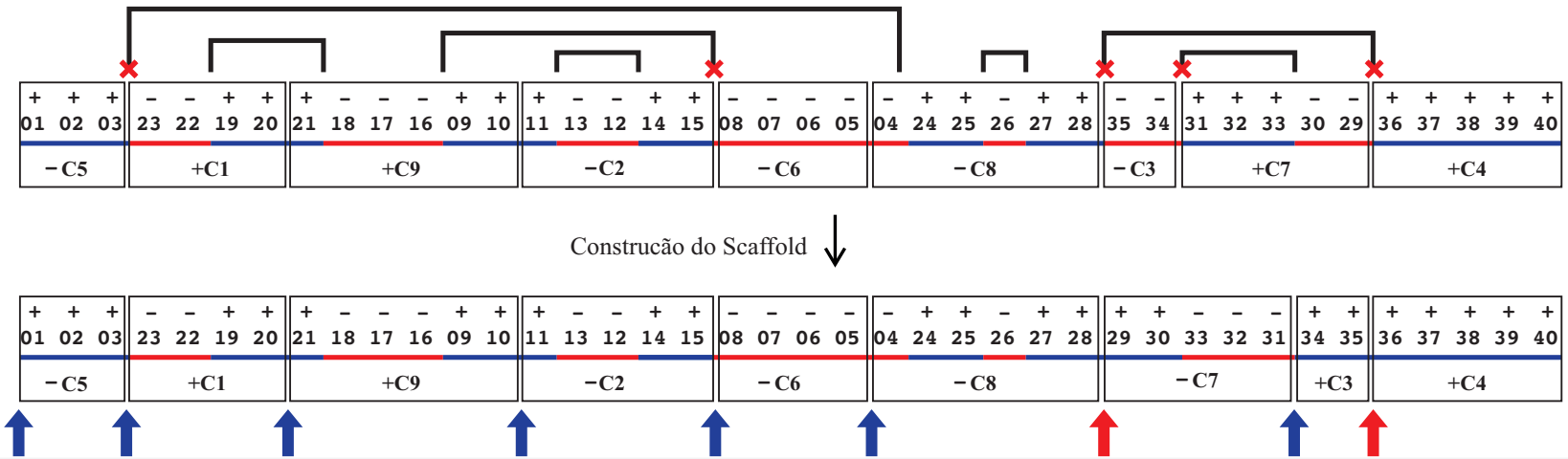
Usando o IS $[+33, -30]$



Nenhum IS detectado: o par ([+28, −35],[−29, +36]) está ausente



Exemplo (Avaliação do resultado)



Softwares usados nos testes

- SIS - Implementação do algoritmo apresentado.
  - nucmer e promer foram usados para detectar os blocos.
- Programas usados
  - ABACAS (nucmer)
  - CONTIGuator (BLAST+)
  - fillScaffold (nucmer e promer)
  - Mauve Aligner (Mauve)
  - OSLay (nucmer)
  - Projector 2 (blat)
  - r2cat (realiza a sua própria detecção de blocos)

Dados de entrada

- Foram usados contigs reais obtidos de montagens preliminares de 18 genomas (21 cromossomos) disponíveis no NCBI.
- Esses genomas se encontram atualmente completos. Dessa forma, é possível saber a ordem correta em todos os casos.

Casos de Testes

- Para cada genoma incompleto, obtém-se uma lista dos 20 genomas completos mais próximos.
  - NUCMi (visto a seguir) foi usado para gerar essa lista.
- Todos os programas de construção de scaffolds foram executados com cada um dos 21 cromossomos.
  - Os 20 genomas de referência mais próximos foram usados.
- **Top 1**: média dos resultados quando o genoma mais próximo é usado como referência.
- **Top 10**: média dos resultados quando os 10 genomas mais próximos são usados como referência.
- **Top 20**: média dos resultados quando os 20 genomas mais próximos são usados como referência.

Adjacências - Média

Programas	Top 1	Top 10	Top 20
ABACAS	33,80%	23,95%	13,20%
CONTIGuator	42,02%	30,35%	20,01%
fillScaffolds (nucmer)	48,47%	34,10%	21,32%
fillScaffolds (promer)	44,19%	32,94%	21,63%
Mauve	58,65%	46,31%	32,17%
OSLay	46,83%	34,03%	21,28%
Projector 2	36,89%	25,35%	15,51%
r2cat	59,72%	44,19%	30,30%
SIS (nucmer)	56,39%	43,95%	28,55%
SIS (promer)	**60,96%**	**50,01%**	**37,27%**

Performance de cada programa usando a média das porcentagem de adjacências corretas.

Adjacências - Melhores Programas

Programas	Top 1	Top 10	Top 20
ABACAS	17,39%	11,74%	7,61%
CONTIGuator	13,04%	9,57%	6,30%
fillScaffolds (nucmer)	21,74%	11,30%	8,70%
fillScaffolds (promer)	0,00%	6,96%	8,04%
Mauve	26,09%	25,65%	23,26%
OSLay	21,74%	15,65%	11,30%
Projector 2	8,70%	6,09%	4,57%
r2cat	13,04%	16,09%	15,65%
SIS (nucmer)	39,13%	27,83%	20,87%
SIS (promer)	**56,52%**	**59,13%**	**64,57%**

Porcentagem das instâncias em que cada programa forneceu os melhores resultados em termos de adjacências.

Cobertura - Média

Programas	Top 1	Top 10	Top 20
ABACAS	27,82%	19,91%	11,27%
CONTIGuator	41,96%	29,34%	19,27%
fillScaffolds (nucmer)	42,94%	29,87%	18,99%
fillScaffolds (promer)	40,15%	29,23%	19,51%
Mauve	55,90%	41,85%	29,46%
OSLay	42,50%	29,03%	18,24%
Projector 2	34,29%	22,54%	13,93%
r2cat	56,81%	40,36%	27,51%
SIS (nucmer)	52,37%	39,45%	25,63%
SIS (promer)	**58,57%**	**46,01%**	**34,62%**

Performance de cada programa nos testes. Nesta tabela, o critério utilizado foi a média das coberturas.

Cobertura - Melhores Programas

Programas	Top 1	Top 10	Top 20
ABACAS	13,04%	11,74%	7,83%
CONTIGuator	13,04%	10,00%	6,52%
fillScaffolds (nucmer)	17,39%	11,30%	7,39%
fillScaffolds (promer)	0,00%	6,96%	7,39%
Mauve	34,78%	25,65%	20,22%
OSLay	17,39%	12,17%	8,91%
Projector 2	8,70%	6,09%	4,57%
r2cat	13,04%	17,39%	14,35%
SIS (nucmer)	39,13%	24,35%	16,95%
SIS (promer)	**47,83%**	**51,74%**	**56,52%**

Porcentagem das instâncias em que cada programa forneceu os melhores resultados usando como critério a cobertura.

Publicação

- Z. Dias, U. Dias, and J. C. Setubal. Using inversion signatures to generate draft genome sequence scaffolds. In *Proceeding of the 2nd ACM International Conference on Bioinformatics and Computational Biology (BCB'2011)*, pages 39–48, Chicago, EUA, 2011.

Parte III

Distâncias entre Genomas

Distâncias entre Genomas

Distâncias entre Genomas

Motivação

- A metodologia clássica para obter a distância evolutiva entre duas espécies se baseia na comparação de pequenas regiões muito conservadas durante o processo evolutivo.
- Atualmente, uma grande quantidade de genomas completos estão disponíveis.
- Surge a necessidade de novas metodologias de classificação de genomas baseadas na comparação do genoma completo.

Objetivo

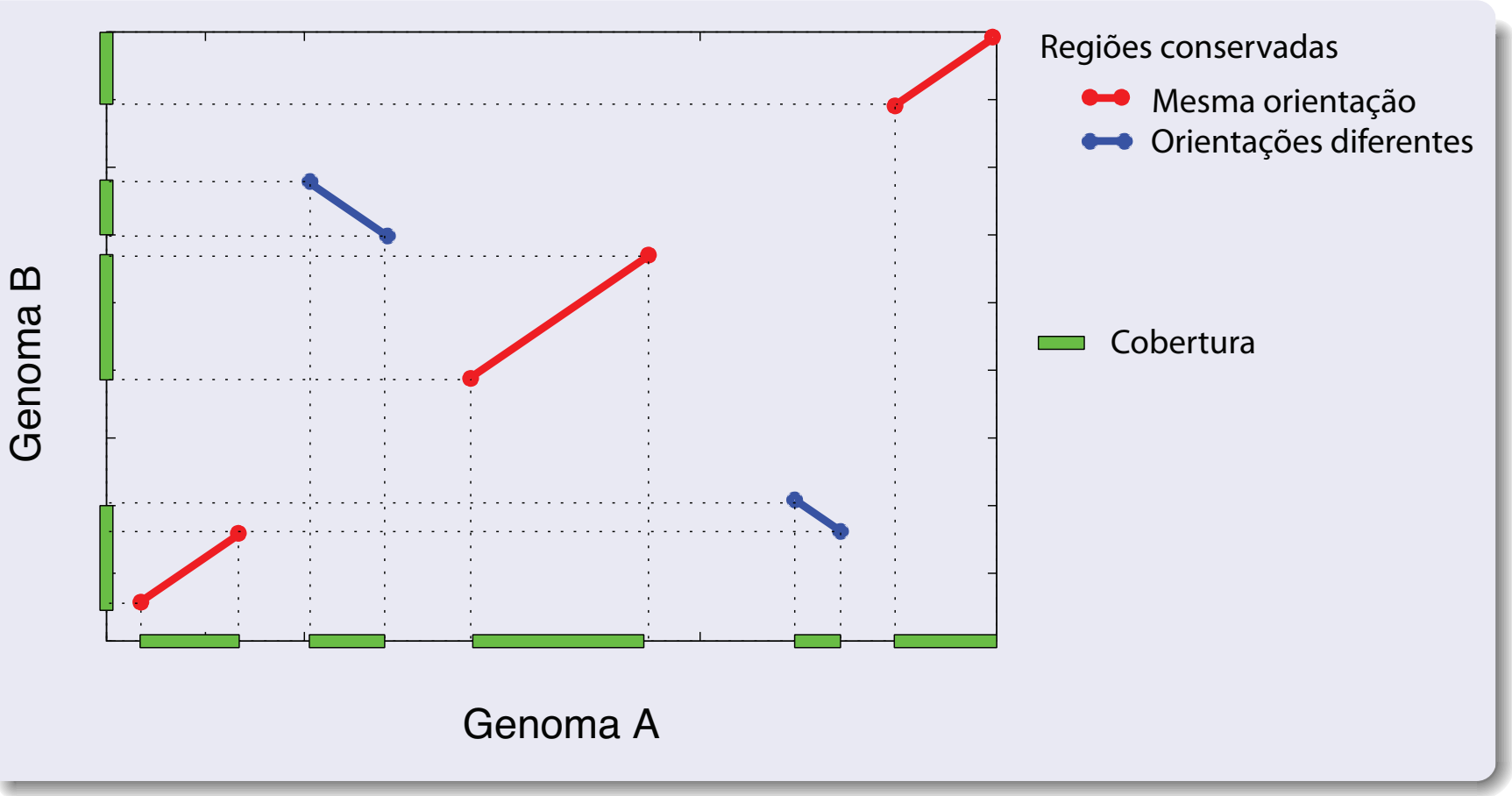
- Desenvolver medidas que podem ser computadas de maneira eficiente.
- Idealmente, as distância deve refletir as divergências evolucionárias entre ambos.

Medidas de distância

- NUCMi e PROMi diferem quanto ao método de identificar os blocos conservados entre os genomas.
  - NUCMi: usa o programa `nucmer`, que identifica blocos similares na cadeia de nucleotídeos.
  - PROMi: usa o programa `promer`, que realiza a comparação usando a tradução da sequência de DNA em proteínia.
- Como a sequência de aminoácidos tende a se divergir mais lentamente que a sequência de nucleotídeos, o `promer` possui uma maior sensibilidade, encontrando mais regiões conservadas que o `nucmer`.

Medidas de distância

- Os blocos conservados são projetados nas ordenadas e abscissas do gráfico cartesiano.



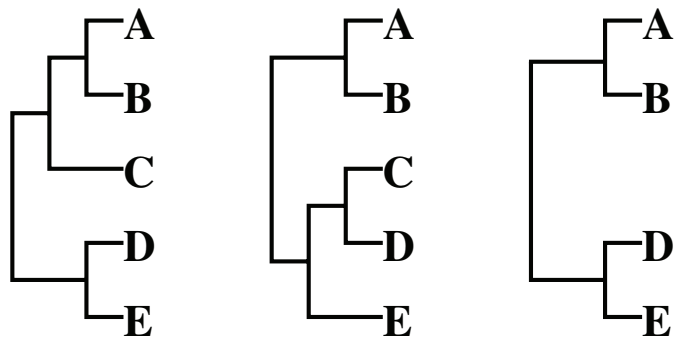
- Para obter NUCMi ou PROMi:

$$\text{Dist}(A, B) = 1 - \frac{Px + Py}{L_A + L_B}$$

Onde P_x é a soma dos tamanhos de todas as projeções no eixo x, P_y é o análogo para o eixo y e L_A e L_B são os tamanhos dos genomas A e B, respectivamente

Análise Comparativa das Medidas

- Usamos NUCMi, PROMi e MUMi para gerar matrizes de distância para os grupos *Pseudomonadaceae*, *Shewanella*, *Xanthomonas* e *Mycobacterium*.
- O programa `neighbor` gerou as árvores filogenéticas a partir das matrizes de distância.
- As árvores foram comparadas com árvores de referência usando uma sub-árvore consenso chamada MAST (do inglês, *maximum agreement subtree*).



Árvore 1 Árvore 2 MAST

Análise Comparativa das Medidas

Genomas Próximos

Grupo Bacterial	Organismos	NUCMi	PROMi	MUMi
Pseudomonadaceae	18	**15**	14	14
Mycobacterium	16	**15**	14	14
Xanthomona	9	**8**	6	**8**
Shewanella	9	**8**	7	**8**
Soma	50	46	41	44

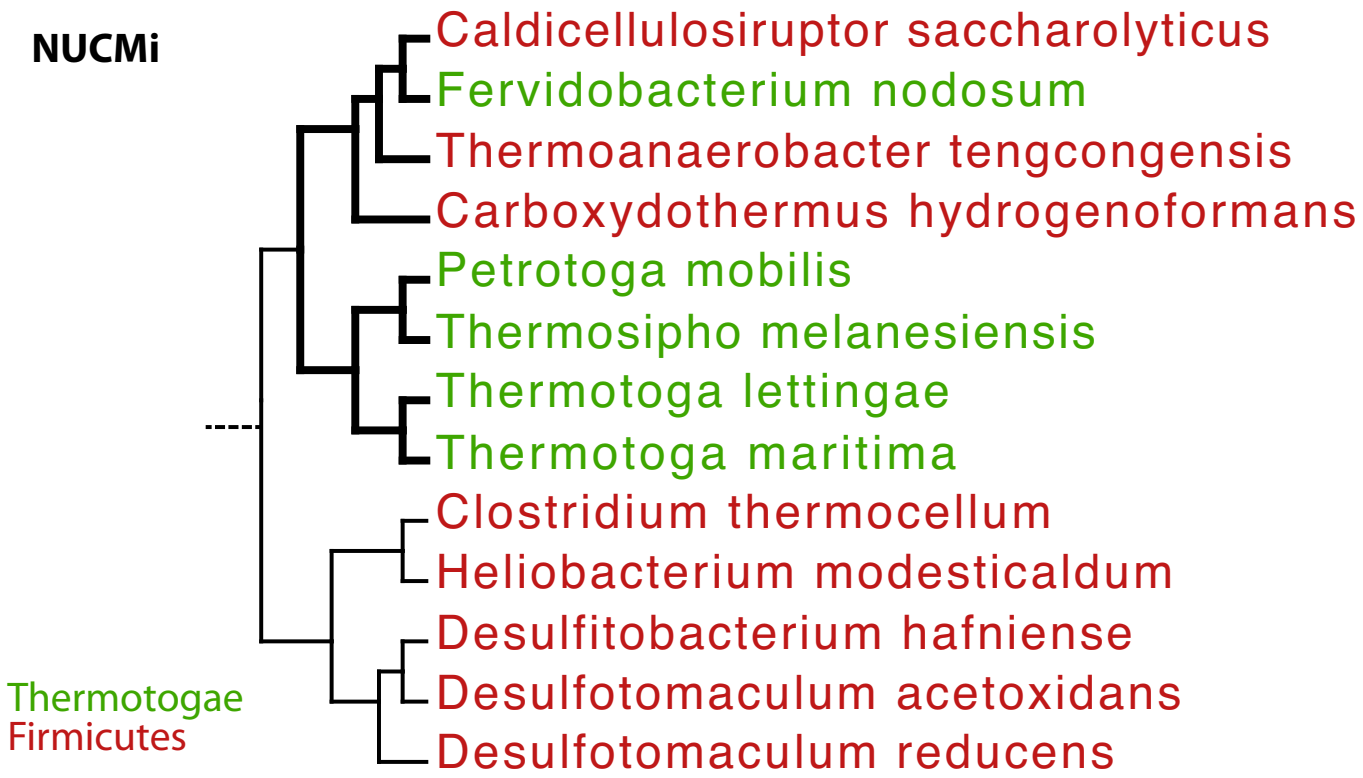
Genomas Distantes

Grupo Bacterial	Organismos	NUCMi	PROMi	MUMi
actinobacteria	52	23	**29**	16
alphaproteobacteria	43	19	**21**	11
bacteroidetes	15	5	**8**	**8**
betaproteobacteria	18	**12**	10	9
chlamydiae	7	5	**6**	**6**
chlorobia	5	**3**	**3**	**3**
chloroflexi	5	2	2	**3**
cyanobacteria	18	8	10	**11**
deinococcus e thermus	4	**3**	**3**	**3**
deltaproteobacteria	19	10	**14**	9
epsilonproteobacteria	12	**7**	**7**	5
firmicutes	57	26	25	**27**
fusobacteriales	4	**3**	**3**	**3**
gammaproteobacteria	45	**15**	13	14
spirochaetales	7	**4**	**4**	3
tenericutes	16	8	**13**	6
thermotogae	5	**3**	**3**	**3**
Soma	332	156	174	140

Fornecendo o conjunto completo aos programas, PROMi obteve 105 espécies na MAST, enquanto NUCMi obteve 94 e MUMi obteve 53

Análise Comparativa das Medidas

- Análise do agrupamento de classes.



- Thermotogae = 5 espécies
- Tamanho da menor árvore contendo todas = 8 espécies
- Agrupamento = $\frac{5}{8} = 0{,}625$

Agrupamento das Classes

Grupo Bacterial	NUCMi	PROMi	MUMi
actinobacteria	0.173	0.158	0.157
alphaproteobacteria	0,651	0,439	0,129
bacteroidetes	0,046	0,058	0,050
betaproteobacteria	0,134	1,000	0,054
chlamydiae	0,023	0,027	0,021
chlorobia	1,000	1,000	1,000
chloroflexi	0,017	0,015	0,015
cyanobacteria	1,000	1,000	0,061
deinococcus e thermus	1,000	1,000	1,000
deltaproteobacteria	0,063	0,074	0,057
epsilonproteobacteria	0,857	1,000	0,104
firmicutes	0,175	0,934	0,172
fusobacteriales	1,000	1,000	0,308
gammaproteobacteria	0,336	0,196	0,135
spirochaetales	0,024	0,027	0,055
tenericutes	1,000	1,000	0,222
thermotogae	0,625	1,000	0,041
Média	0,478	0,584	0,211

Análise Comparativa das Medidas

- Foi fornecida pelos autores da árvore de referência de genomas distantes, mediante requisição, uma árvore contendo a distância entre cada nó e o seu nó pai.
- Essa informação extra permite recriar uma matriz de distância com as distâncias obtidas nos ramos da árvore.
- Comparar a matriz de referência R com as matrizes obtidas por cada uma das medidas, denotadas pela letra A.
- Um erro é tabulado se
  - $A(X, Y) < A(X, Z)$ e $R(X, Y) > R(X, Z)$
  - $A(X, Y) > A(X, Z)$ e $R(X, Y) < R(X, Z)$
- Caso contrário, um acerto será tabulado. A pontuação da matriz A é a porcentagem de acertos.

Análise Comparativa das Medidas

Grupo Bacterial	Organismos	NUCMi (%)	PROMi (%)	MUMi (%)
actinobacteria	52	76,18	81,15	69,23
alphaproteobacteria	43	72,63	80,60	75,84
bacteroidetes	15	58,11	83,99	75,22
betaproteobacteria	18	65,45	76,14	64,84
chlamydiae	7	70,95	93,33	86,67
chlorobia	5	64,44	86,67	68,89
chloroflexi	5	55,56	86,67	82,22
cyanobacteria	18	62,42	78,60	72,73
deinococcus e thermus	4	80,00	93,33	73,33
deltaproteobacteria	19	71,65	76,98	60,70
epsilonproteobacteria	12	63,78	80,42	57,58
firmicutes	57	71,72	84,17	61,15
fusobacteriales	4	66,67	66,67	33,33
gammaproteobacteria	45	50,68	79,66	65,05
spirochaetales	7	67,14	87,62	61,43
tenericutes	16	80,59	75,42	62,69
thermotogae	5	51,11	95,56	64,44
Média	19,52	66,42	82,76	66,78

Publicação

- U. Dias, Z. Dias, and J. C. Setubal. Two new whole-genome distance measures. In *Proceedings of the 6th Brazilian Symposium on Bioinformatics (BSB'2011)*, pages 61 – 64, Brasília, Brasil, 2011.
- U. Dias, Z. Dias, and J. C. Setubal. Evaluation of genome distance measures using tree-derived distances. In *Proceedings of the 6th Brazilian Symposium on Bioinformatics (BSB'2011)*, pages 73 – 76, Brasília, Brasil, 2011.

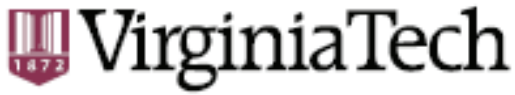
Contribuições

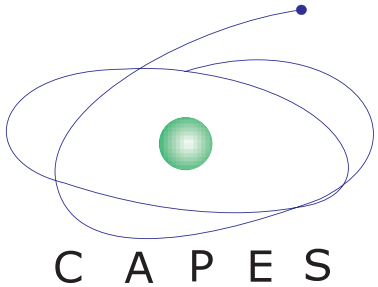
- Eventos de Transposição
  - Algoritmo de Aproximação
  - Modelos usando Programação em Lógica com Restrição
- Eventos de Reversão (sem simetria)
  - Algoritmo para Construção de *Scaffold*
- Eventos de Reversão (com simetria)
  - Ferramenta de Simulação
  - Estudo teórico para o problema
    - Definições, lemas e conjecutras
    - Limitantes
    - Algoritmos básicos
    - Algoritmo Guloso
    - Classes e famílias de permutações.
- Distância entre Genomas
  - Medidas NUCMi e PROMi

Produção Científica

	Congressos		
	Nacional	Internacional	Total
Trabalho Completo	1	4	5
Resumo Estendido	2	1	3
Total	3	5	8

Agradecimentos

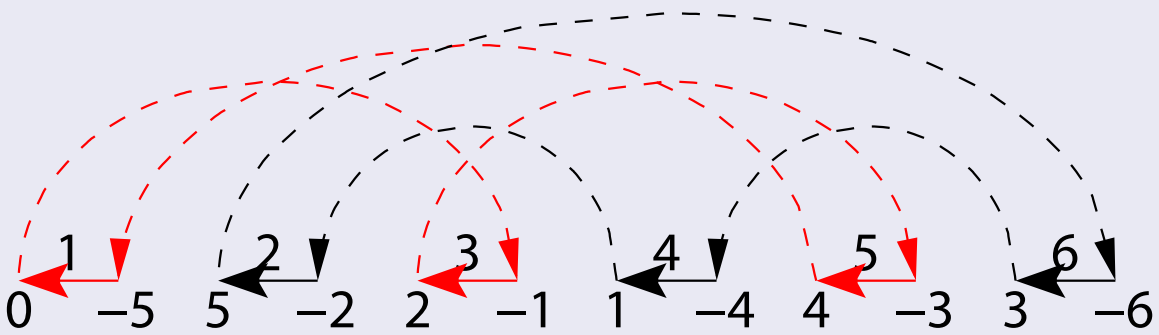


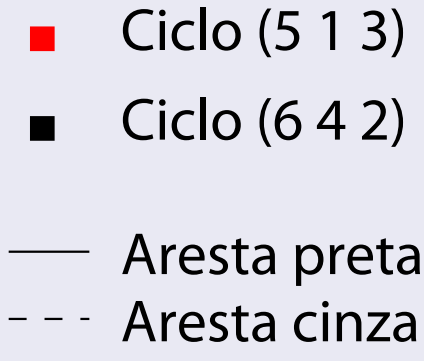


Grafo de Ciclos

- Vértices:
  - $\{-\pi_1, +\pi_1, -\pi_2, +\pi_2, \ldots, -\pi_n, +\pi_n\} \cup \{0, -(n+1)\}$
- Arestas:
  - Cinzas: $\{+(i-1), -i | 1 \leq i \leq n+1\}$
  - Pretas: $\{(-\pi_i, +\pi_{i-1}) | 1 \leq i \leq n+1\}$

$\pi = (5\ 2\ 1\ 4\ 3)$



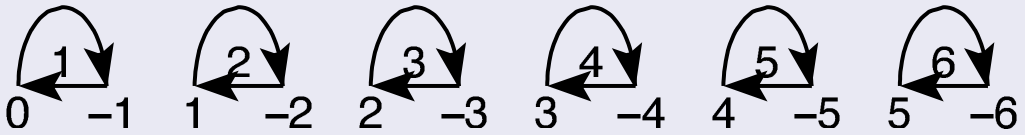


- Decomposição em ciclos

Grafo de Ciclos

- Um ciclo é par se possuir um número par de arestas pretas e impar, caso contrário.
- A identidade é a única permutação com apenas ciclos tamanho 1 (ímpares).

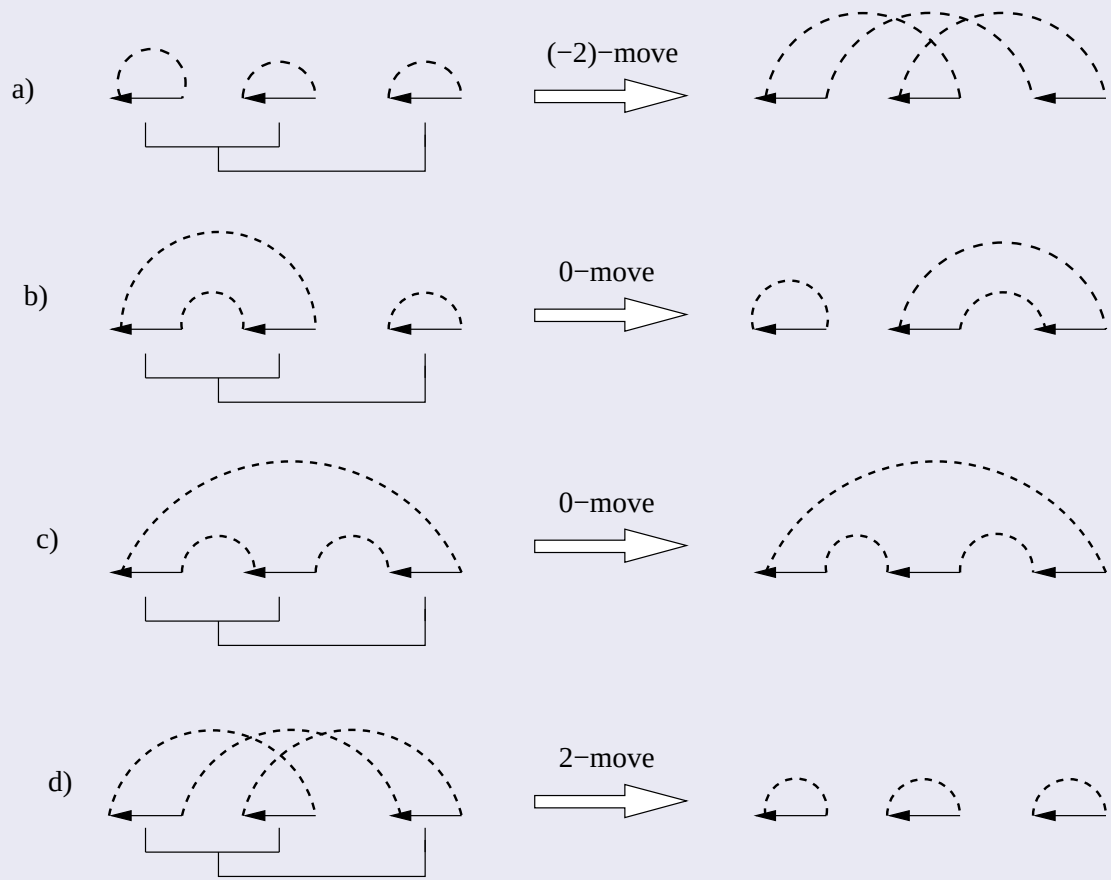
$\iota = (1\ 2\ 3\ 4\ 5)$



- Sejam $\Delta c(G(\pi), \rho)$ e $\Delta c_{impar}(G(\pi), \rho)$ as variações no número total de ciclos e no número de ciclos ímpares em $G(\pi)$ após ser aplicada a transposição ρ.
  - $\Delta c(G(\pi), \rho), \Delta c_{impar}(G(\pi), \rho) \in \{2, 0, -2\}$

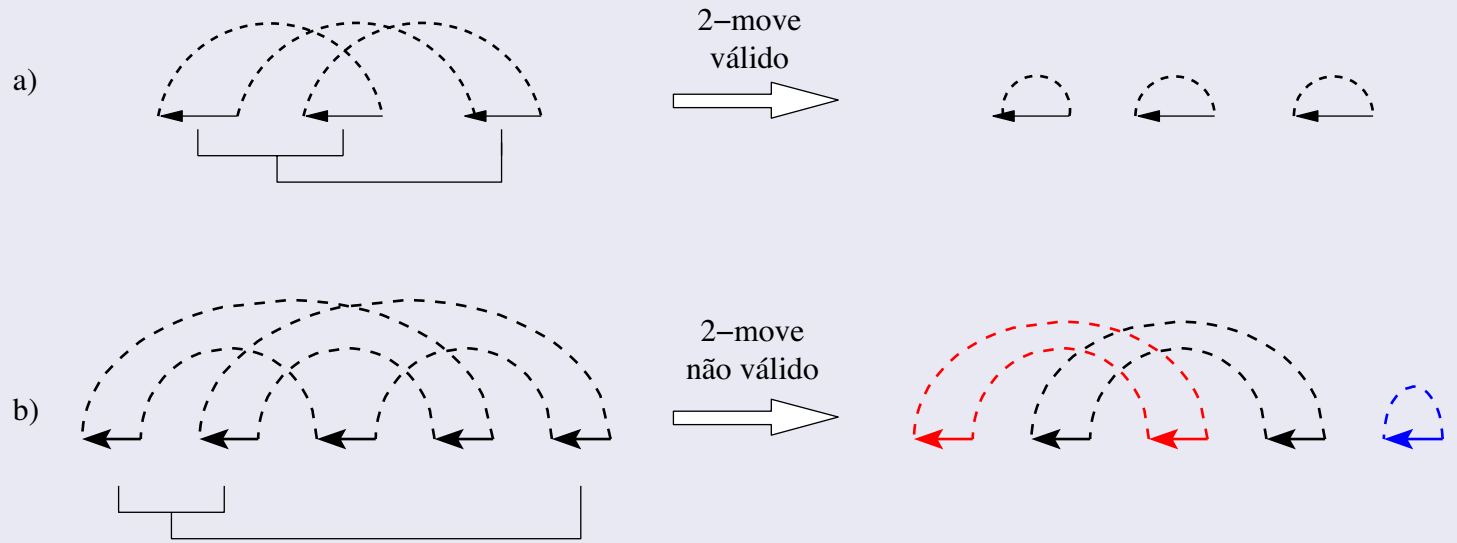
Grafo de Ciclos

- Define-se um x-move como uma transposição ρ tal que $\Delta c(G(\pi), \rho) = x$, $x \in \{2,0,-2\}$.



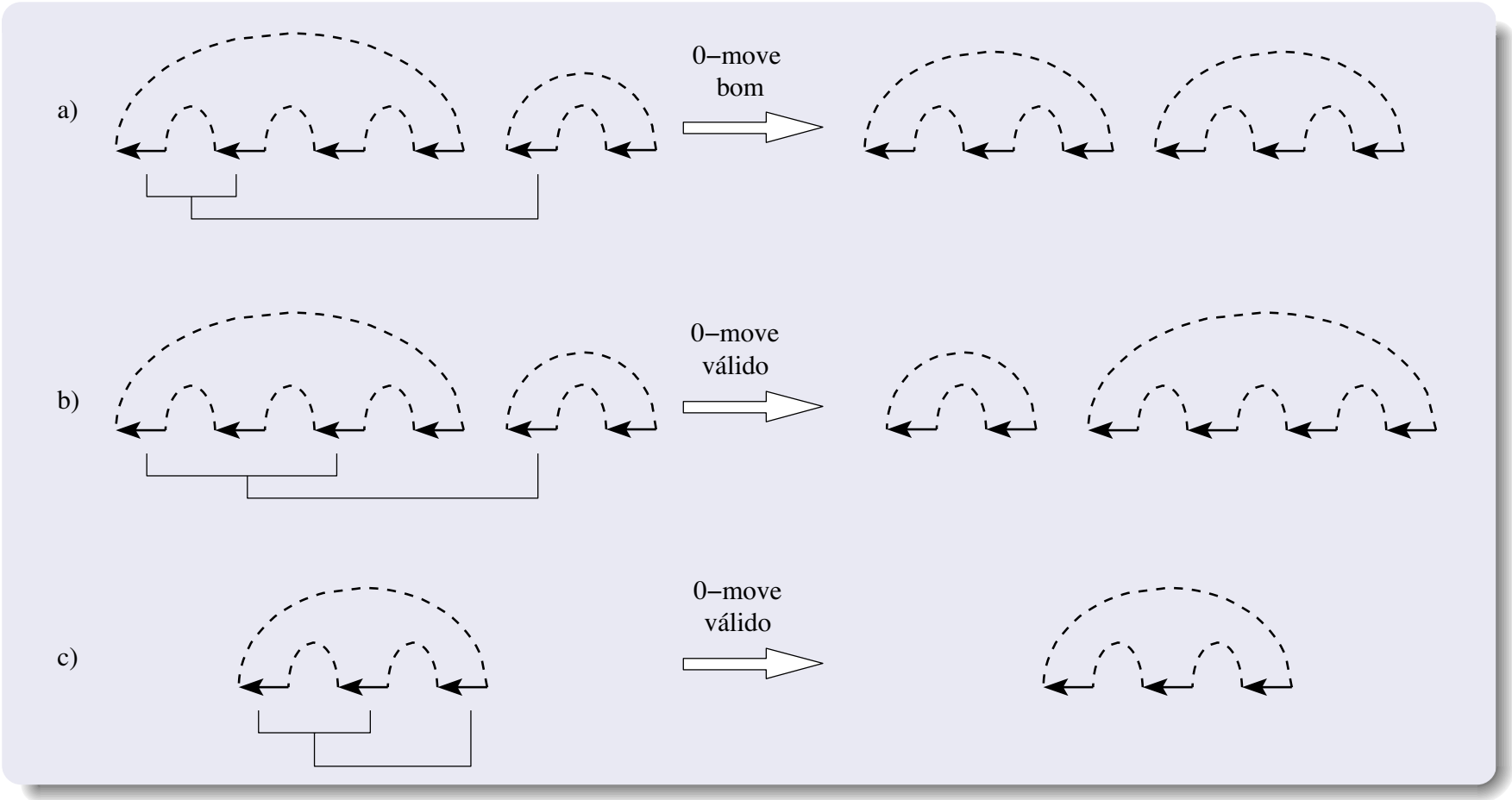
Grafo de Ciclos

- Um x-move é válido se $\Delta c(G(\pi), \rho) = \Delta c_{impar}(G(\pi), \rho)$.



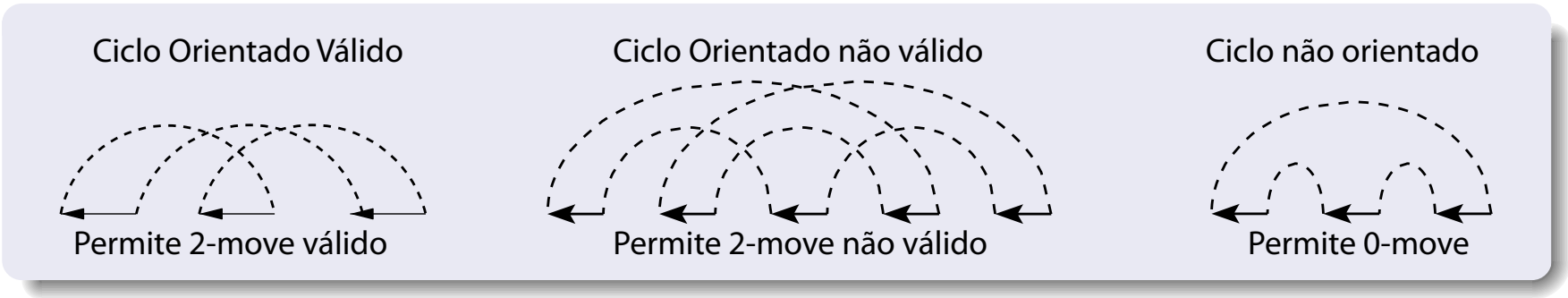
Grafo de Ciclos

- Um 0-move é dito bom se $\Delta c_{impar}(G(\pi), \rho) = 2$.



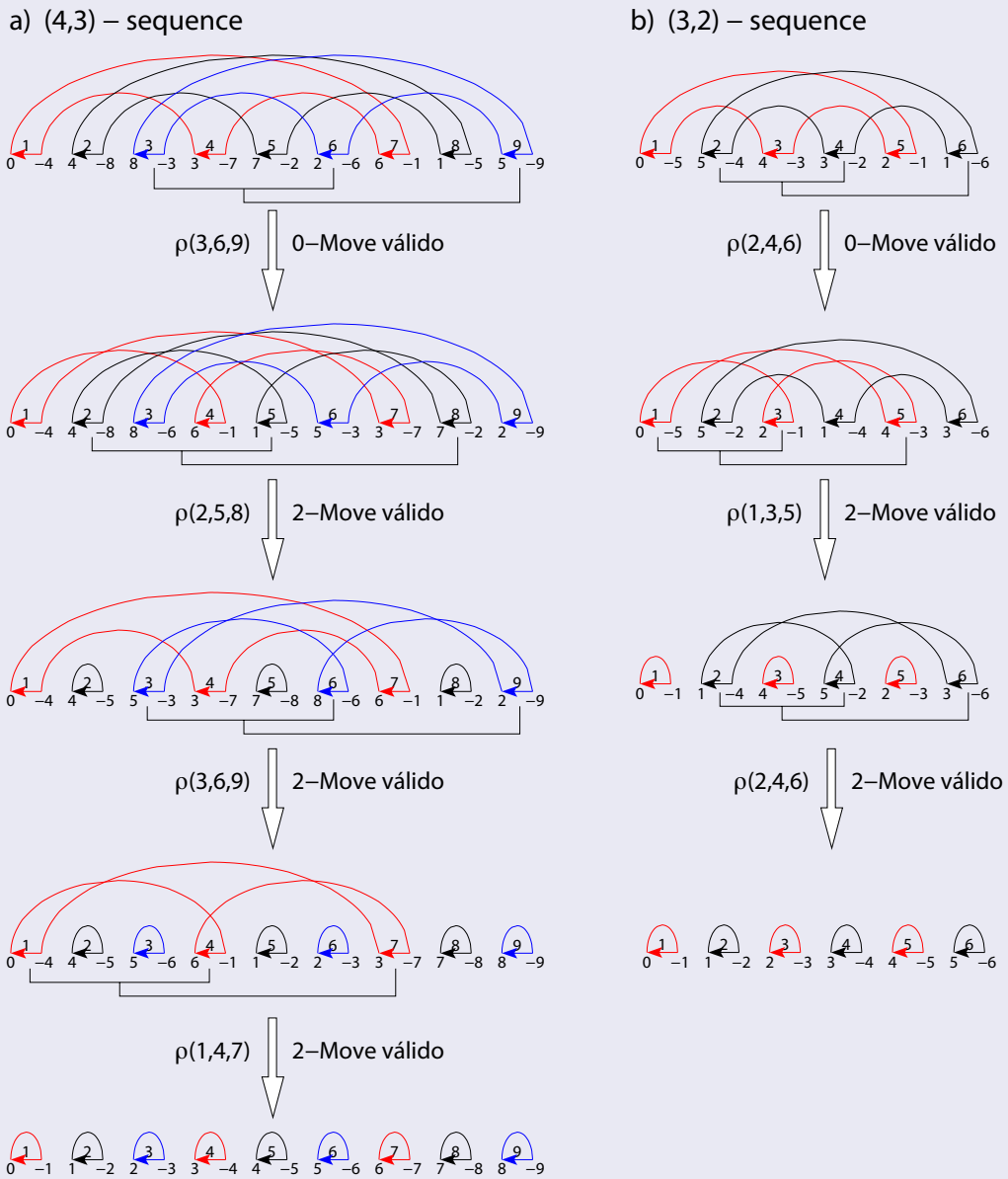
Grafo de Ciclos

- Os ciclos são classificados de acordo com as transposições que podem ser aplicadas a eles.



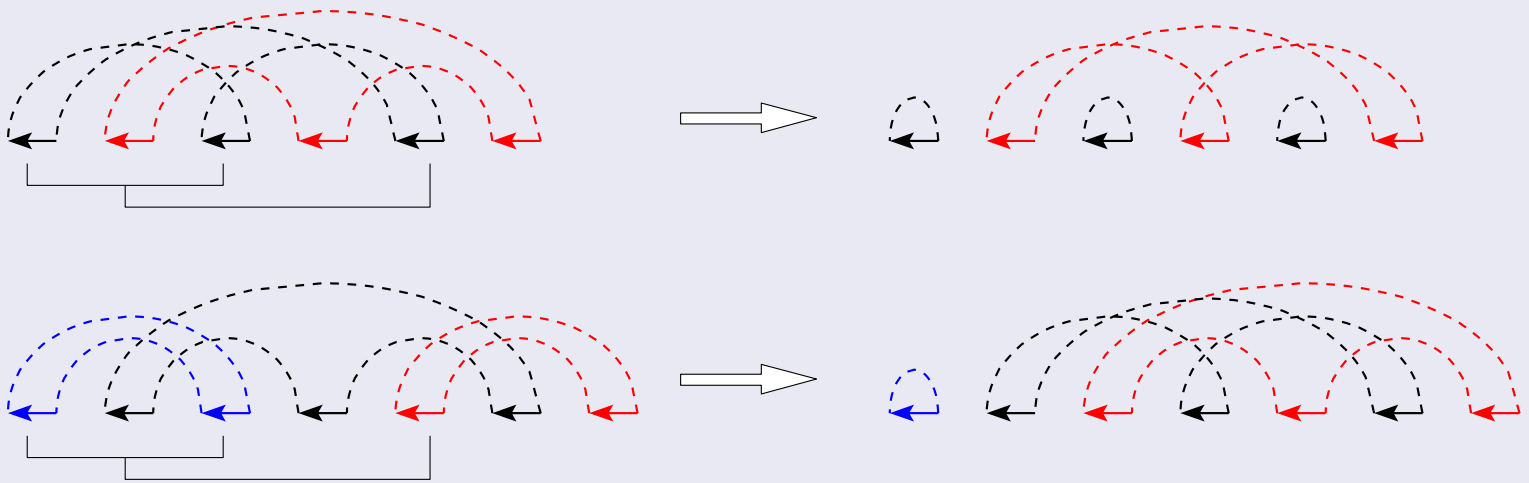
Grafo de Ciclos

Ordenação por Transposições
Aumentar o número de ciclos



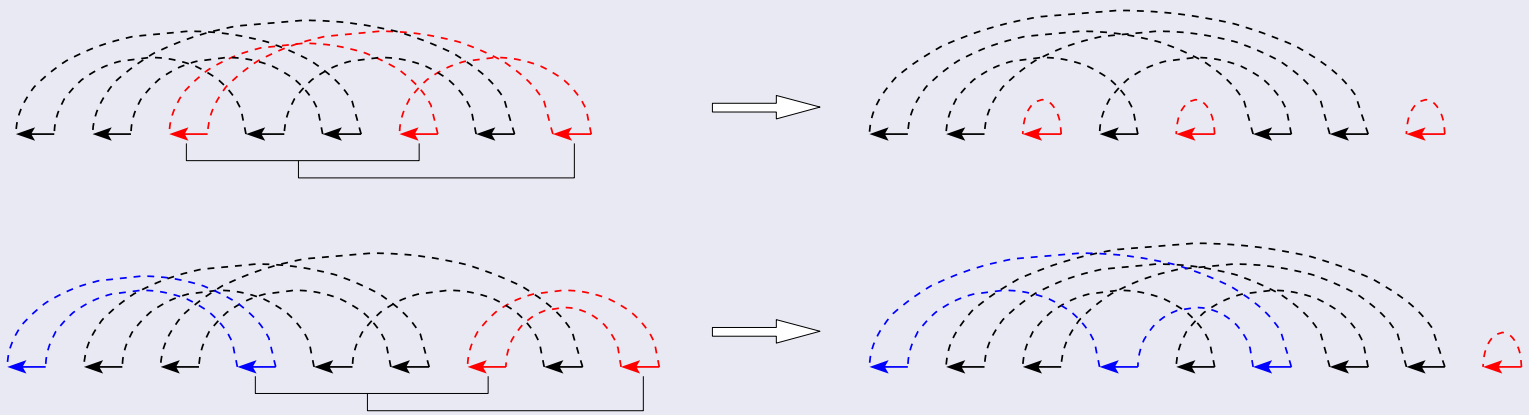
Heurística 1

- Seja $\rho(i,j,k)$ uma transposição que aumenta o número de ciclos ímpares em $G(\pi)$, então existe 2-move válido em $\pi\rho(i,j,k)$ se (i,j,k) está entrelaçado com três arestas pretas (a,b,c) em um ciclo não orientado.



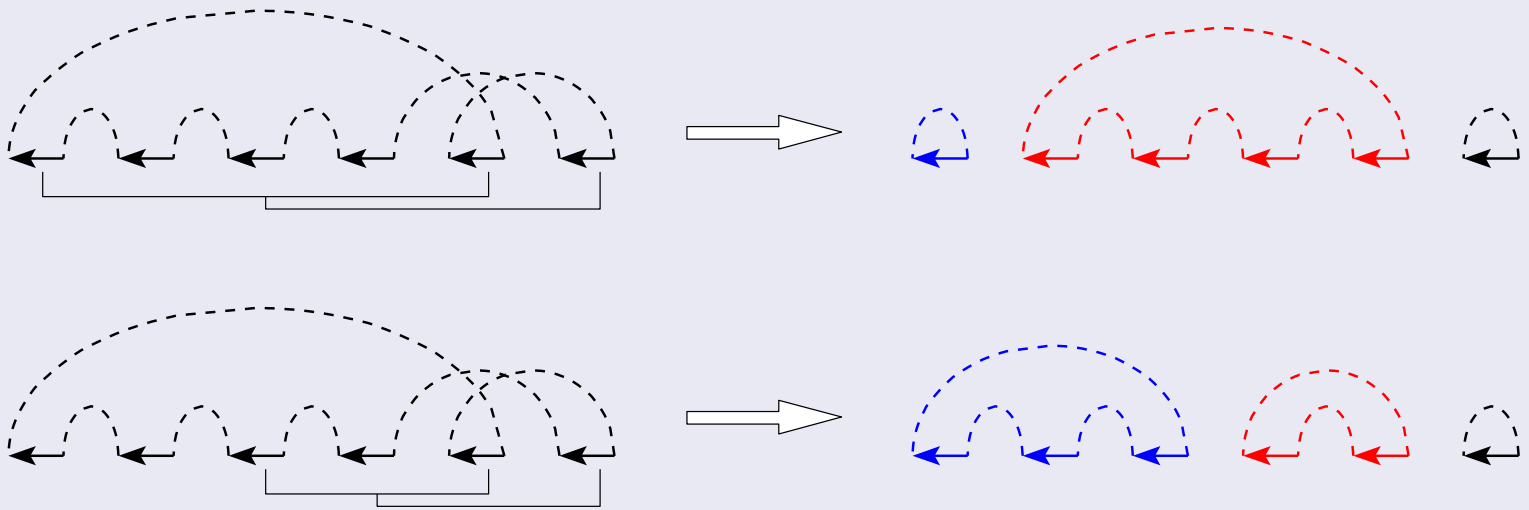
Heurística 2

- Seja $\rho(i,j,k)$ uma transposição que aumenta o número de ciclos ímpares em $G(\pi)$, então existe 2-move válido em $\pi\rho(i,j,k)$ se (i,j,k) transforma um ciclo orientado não válido em um ciclo orientado válido.



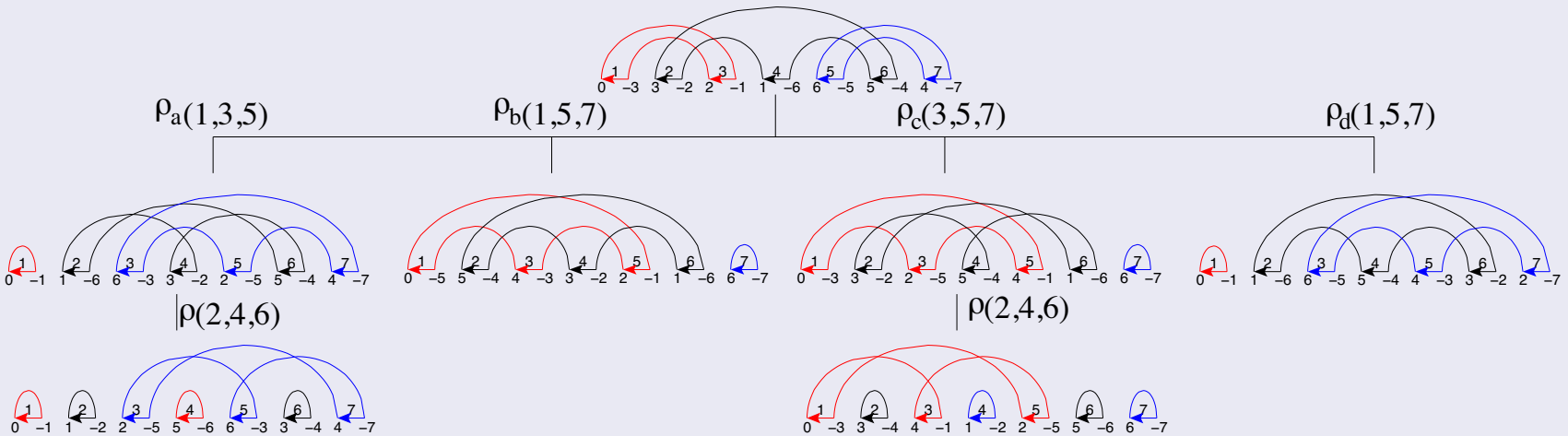
Heurística 3

- Seja $\rho(i,j,k)$ um 2-move válido, os tamanhos dos novos ciclos gerados por ρ estão no conjunto $\{dist(i,j), dist(j,k), dist(k,i)\}$, sendo que, por definição, no máximo uma das distâncias do conjunto pode ser par. Dessa forma, para garantir a criação do maior ciclo par possível, deve-se verificar qual transposição possui a maior distância par.



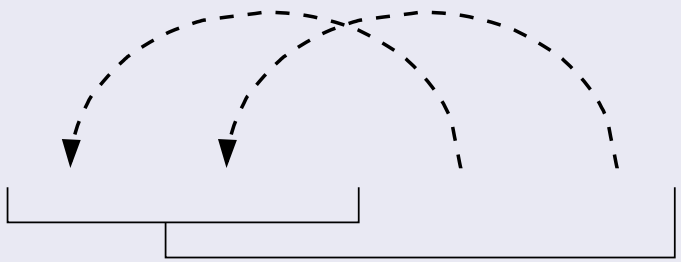
Heurística 4

- Dados a permutação de entrada π e um parâmetro t pré-definido, obtém-se todas as possíveis permutações após t transposições, que podem ser ou 2-move válidos ou 0-moves bons.



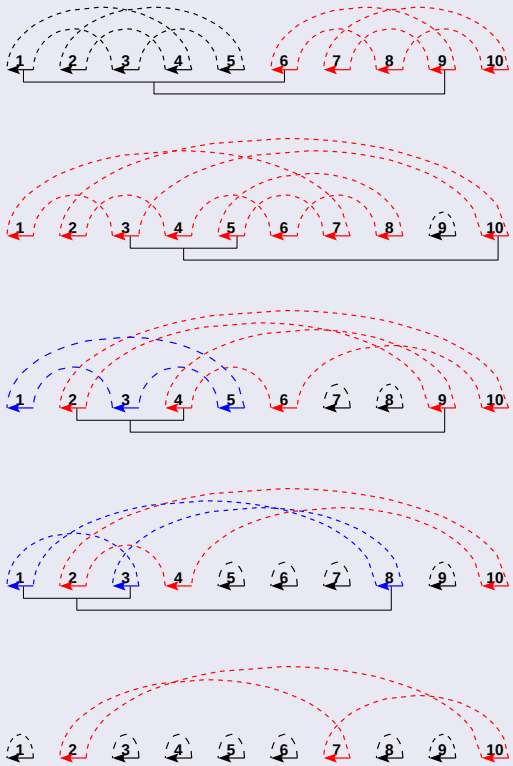
Heurística 5

- Seja $E_{cd}(\pi)$ o conjunto de arestas cinzas $(+(i-1), -i)$ direcionadas para a direita em $G(\pi)$, tais que $\pi^{-1}(i-1) < \pi^{-1}(i)$, a permutação identidade é a única em que $E_{cd}(\pi) = E_c(\pi)$, onde $E_c(\pi)$ é o conjunto de todas as arestas cinzas em $G(\pi)$. Assim, escolhe-se a transposição que mais aumenta o número de arestas em E_{cd} de forma a se aproximar da identidade.



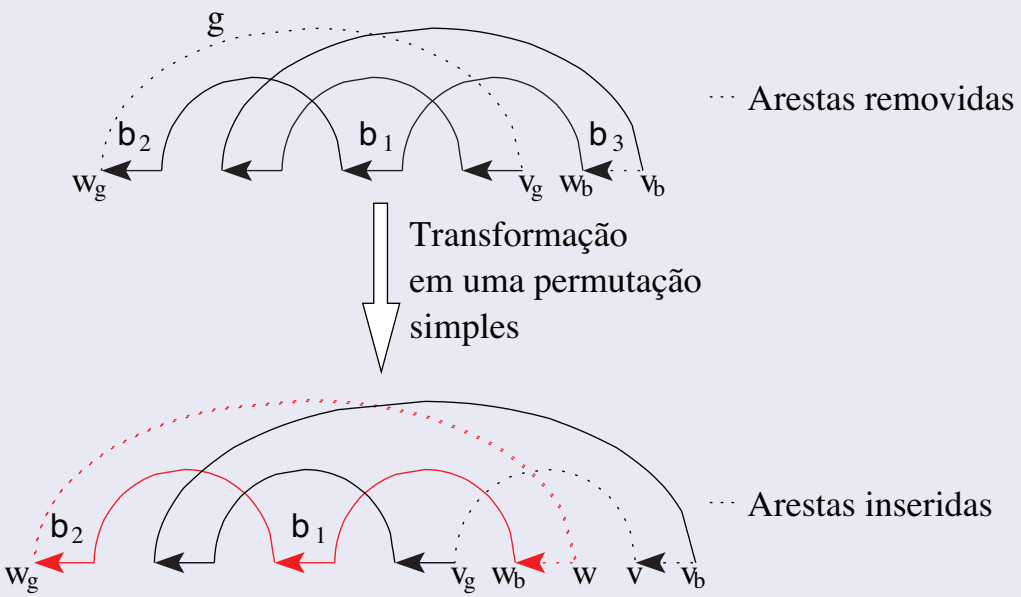
Heurística 6

- Quando não há transposições capazes de aumentar o número de ciclos ímpares, é possível garantir uma (4,3)-sequence se no mínimo dois ciclos orientados existirem.



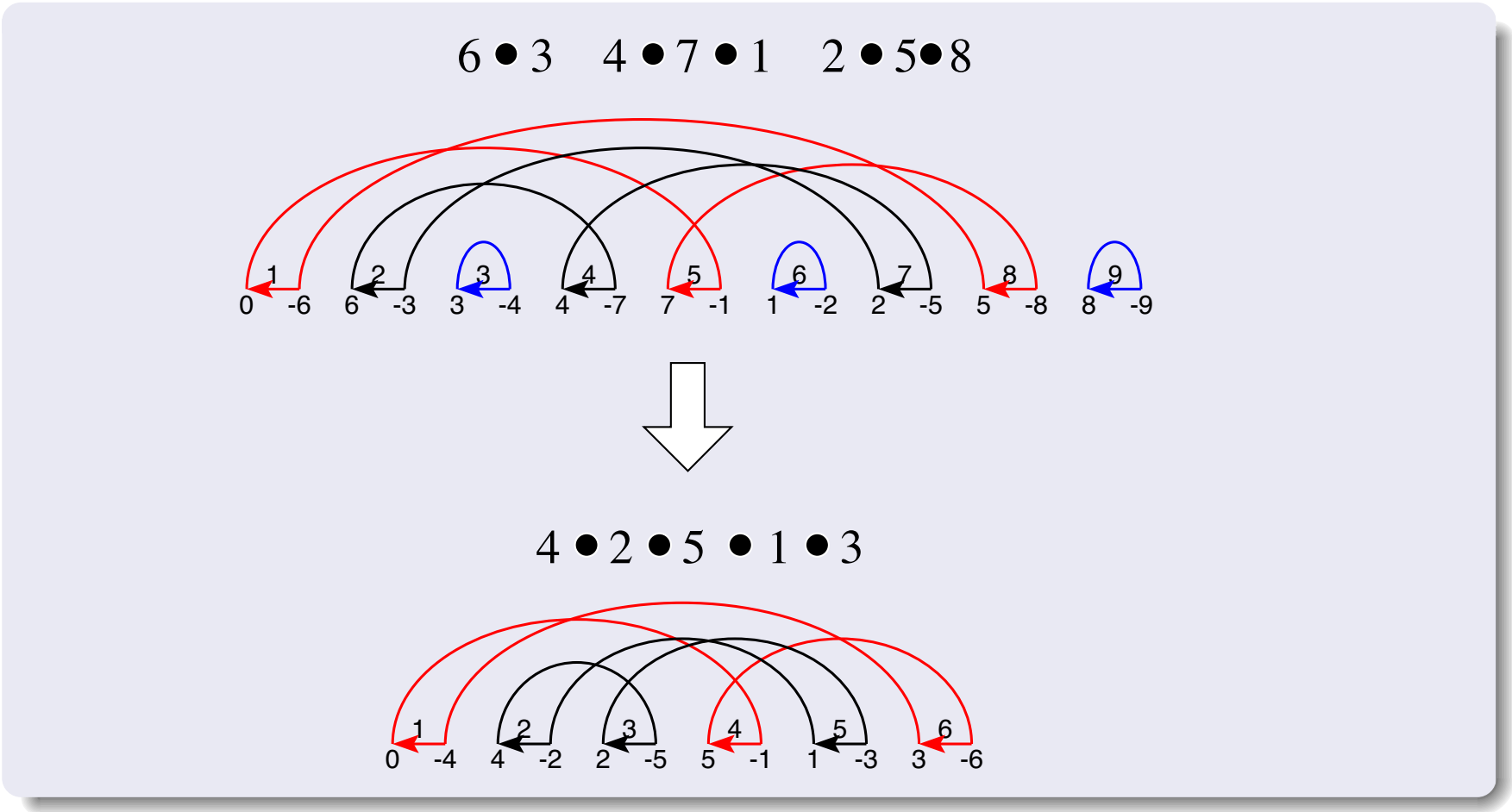
Heurística 7

- Se $G(\pi)$ possui no máximo 1 ciclo orientado não válido e um conjunto de ciclos orientados ímpares, então deve-se transformar a permutação de entrada em uma permutação simples e usar o algoritmo de Elias e Hartman, que garante uma razão de aproximação 1.375.



Permutações Reduzidas

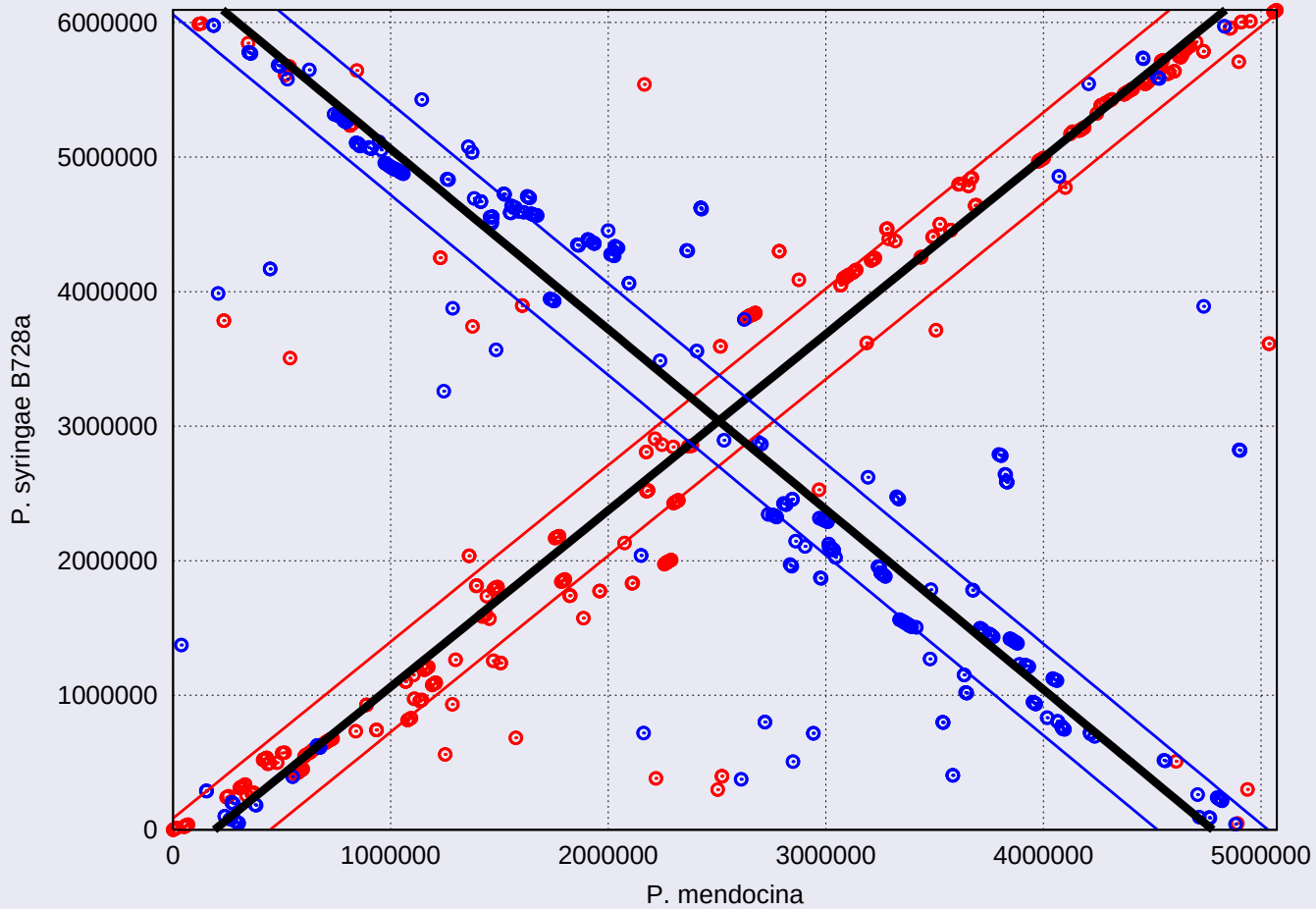
- Permutações reduzidas têm no máximo o mesmo tamanho da permutação original, o que tende a diminuir o espaço de busca



Recuperando informações dos *dotplots*

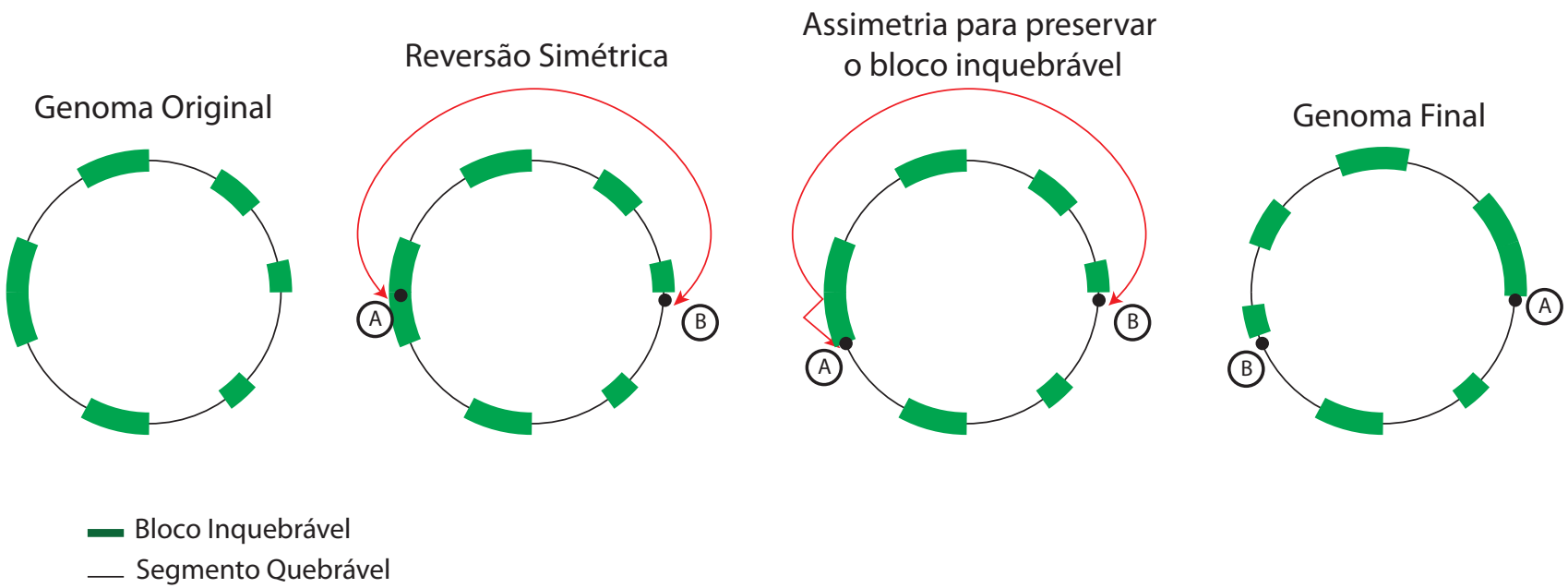
- Os blocos conservados identificados pelo MUMmer são "quebrados" em segmentos de 1kbp (tamanho médio de um gene).
- Cada segmento corresponde a um ponto no *dotplot*.

Recuperando informações dos *dotplots*

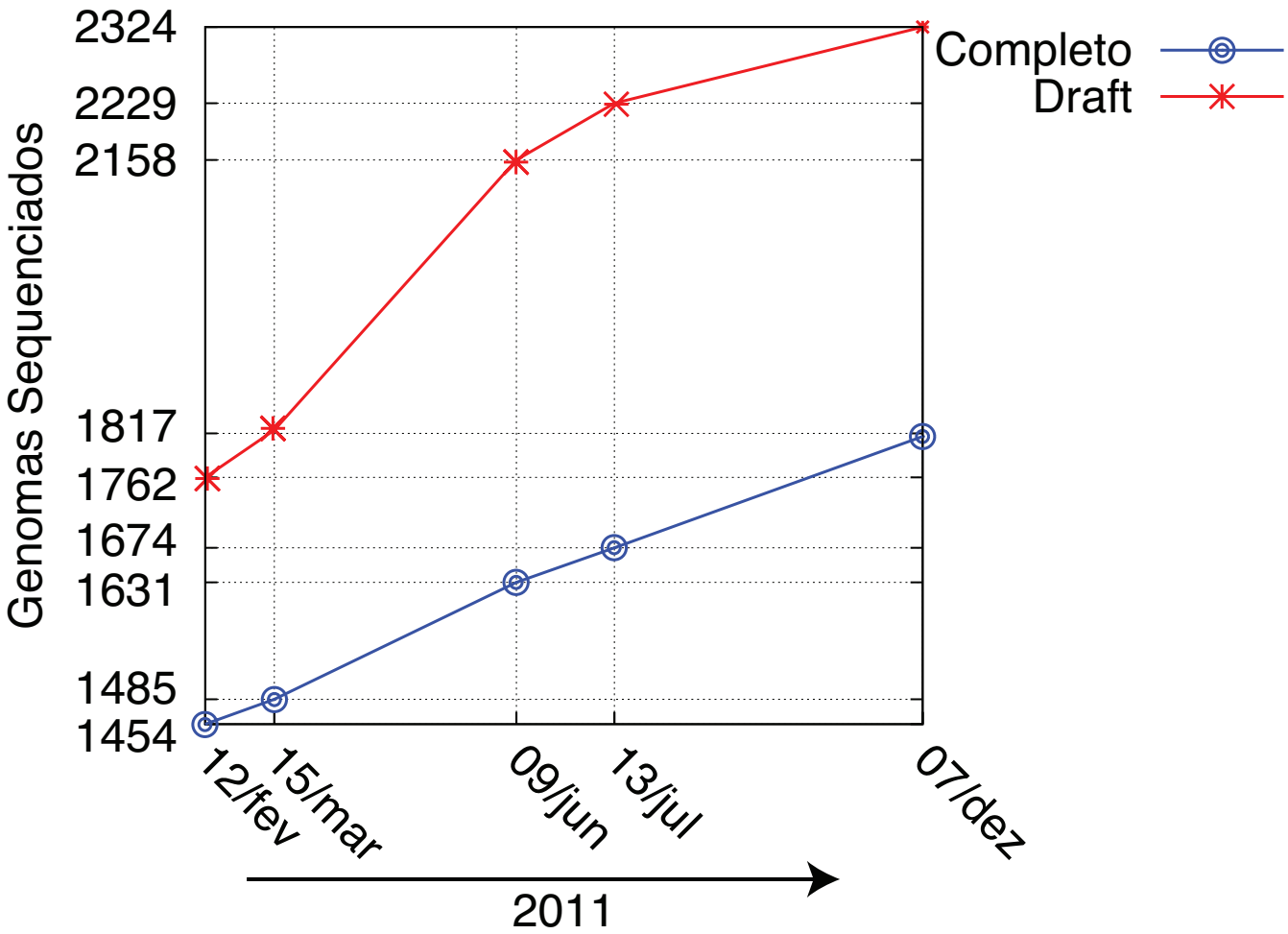


Parâmetros do SIB

- Largura da reversão: distância entre os pontos *breakpoints* e a origem de replicação.
- Blocos inquebráveis: Quando os pontos i ou j são localizados em um bloco inquebrável, uma assimetria é permitida de modo a incluir o bloco.



Construção de *scaffolds* usando genoma de referência



Exemplo

Os IS pairs são:

$$
\begin{array}{lccc}
\text{IS Pair 1:} & [+03,-23] & \times & [-04,+24] \\
\text{IS Pair 2:} & [-22,+19] & \times & [+21,-18] \\
\text{IS Pair 3:} & [-16,+09] & \times & [+15,-08] \\
\text{IS Pair 4:} & [+11,-13] & \times & [-12,+14] \\
\text{IS Pair 5:} & [+25,-26] & \times & [-26,+27] \\
\text{IS Pair 6:} & [+28,-35] & \times & [-29,+36] \\
\text{IS Pair 7:} & [-34,+31] & \times & [+33,-30]
\end{array}
$$

Exemplo

Apenas os pares seguintes são observáveis nos contigs.

IS Pair 1:		$\times$	$[-04, +24]$
IS Pair 2:	$[-22, +19]$	$\times$	$[+21, -18]$
IS Pair 3:	$[-16, +09]$	$\times$	
IS Pair 4:	$[+11, -13]$	$\times$	$[-12, +14]$
IS Pair 5:	$[+25, -26]$	$\times$	$[-26, +27]$
IS Pair 6:		$\times$	
IS Pair 7:		$\times$	$[+33, -30]$

O IS pair 6 está ausente.