Universidade Estadual de Campinas
Instituto de Computação

Klairton de Lima Brito

Modelos com Proporção entre Operações e Regiões Intergênicas Rígidas e Flexíveis

CAMPINAS
1500

Klairton de Lima Brito

Modelos com Proporção entre Operações e Regiões Intergênicas Rígidas e Flexíveis

Tese apresentada ao Instituto de Computação da Universidade Estadual de Campinas como parte dos requisitos para a obtenção do título de Doutor em Ciência da Computação.

Orientador: Prof. Dr. Zanoni Dias
Coorientador: Prof. Dr. Ulisses Martins Dias

Este exemplar corresponde à versão da Tese entregue à banca antes da defesa.

CAMPINAS
1500

Na versão final esta página será substituída pela ficha catalográfica.

De acordo com o padrão da CCPG: “Quando se tratar de Teses e Dissertações financiadas por agências de fomento, os beneficiados deverão fazer referência ao apoio recebido e inserir esta informação na ficha catalográfica, além do nome da agência, o número do processo pelo qual recebeu o auxílio.”
e
“caso a tese de doutorado seja feita em Cotutela, será necessário informar na ficha catalográfica o fato, a Universidade convenente, o país e o nome do orientador.”

Na versão final, esta página será substituída por outra informando a composição da banca e que a ata de defesa está arquivada pela Unicamp.

Agradecimentos

Os agradecimentos devem ocupar uma única página.

Resumo

O resumo deve ter no máximo 500 palavras e deve ocupar uma única página.

Abstract

The abstract must have at most 500 words and must fit in a single page.

Lista de Figuras

Lista de Tabelas

Sumário

Capítulo 1

Introdução

O estudo da evolução dos organismos é uma tarefa de fundamental importância no campo da biologia. Ao decorrer do tempo mudanças podem ocorrer nos organismos, que refletem adaptações desenvolvidas para melhor se adequar e prosperar no ambiente que estão inseridos. Em particular, mudanças genéticas são uma das características utilizadas no campo da *genômica comparativa* para estimar a proximidade de dois organismos com base na similaridade de seus materiais genéticos. O genoma pode sofrer modificações a partir de mutações que podem ser pontuais ou afetar grandes trechos do genoma, que são chamadas de eventos de rearrajos de genomas. Tais eventos podem afetar o genoma modificando, inserindo ou removendo material genético [26]. Uma das formas bem aceita de estimar a proximidade de dois organismos é determinando uma sequência de eventos de rearranjos de genomas com tamanho mínimo e capaz de transformar o genoma de um organismo em outro. O tamanho de tal sequência é chamada de *distância de rearranjo*.

Reversão e transposição são os eventos de rearranjo mais estudados na literatura [7, 6]. Uma reversão atua em um segmento do genoma invertendo a posição e a orientação dos genes contidos no segmento, enquanto uma transposição troca dois segmentos consecutivos do genoma, mas sem afetar a posição e a orientação dos genes nos segmentos. Os eventos de reversão e transposição são chamados de conservativos, pois não alteram a quantidade de material genético do genoma. Existem também eventos não conservativos, como é o caso dos eventos de inserção, deleção e duplicação [37, **?**, **?**, **?**, **?**, **?**], que inserem, removem e duplicam material genético de uma região específica do genoma, respectivamente. Um modelo de rearranjo é caracterizado pelo conjunto de eventos de rearranjo permitidos para transformar um genoma em outro e a representação do genoma utilizada.

Um genoma pode ser representado computacionalmente de diferentes maneiras. Quando o genoma é tratado como uma sequência ordenada de genes, podemos encontrar casos em que determinados genes apresentam múltiplas cópias, sendo comum utilizarmos uma representação na forma de uma cadeia de caracteres, onde cada caractere é associado a um gene. Por outro lado, se existir apenas uma cópia de cada gene, podemos associar um número inteiro para cada gene e a representação é dada na forma de uma permutação. Em ambos os casos, quando a orientação dos genes é conhecida, um sinal de positivo ou negativo é atribuído para cada elemento e a representação é chamada com sinais (string com sinais e permutação com sinais). Caso contrário, o sinal é omitido e a representação é chamada sem sinais (string sem sinais e permutação sem sinais).

Ao utilizar a representação de um genoma como uma permutação, podemos simplificar o problema como sendo um problema de ordenação. Nesse caso, o objetivo consiste em transformar uma permutação π qualquer em uma permutação específica na qual os elementos encontram-se ordenados de maneira crescente e com sinal positivo para o caso com sinais, essa permutação é chamada de identidade.

Quando consideramos um modelo de rearranjo composto apenas pelo evento de reversão e utilizando uma representação do genoma na forma de permutações com sinais, temos o problema de Ordenação de Permutações com Sinais por Reversões. Hannenhalli e Pevzner [27] apresentaram o primeiro algoritmo exato em tempo polinomial para o problema, sendo posteriormente simplificado por Bergeron [7]. Atualmente, temos um algoritmo com complexidade subquadrática para determinar a sequência de reversões capaz de ordenar uma permutação com sinais [35]. Entretanto, se estivermos interessados somente na distância de reversão, existe um algoritmo que executa em tempo linear [2]. Entretanto, quando consideramos uma representação utilizando permutações sem sinais, temos o problema de Ordenação de Permutações sem Sinais por Reversões. Caprara [14] provou que o problema faz parte da classe de problemas NP-Difícil. Um dos primeiros algoritmos para o problema apresentou um fator de aproximação 1.75 [5]. Em seguida, Christie [17] apresentou um algoritmo com fator de aproximação 1.5. Atualmente, o melhor algoritmo para o problema apresenta um fator de aproximação 1.375 [8].

Quando consideramos um modelo de rearranjo composto apenas pelo evento de transposição, a orientação dos genes não é considerada, tendo em vista que o evento de transposição não altera a orientação dos genes. Dessa forma, ao adotar permutações sem sinais, temos o problema de Ordenação de Permutações sem Sinais por Transposições. O problema também pertence à classe de problemas NP-Difícil, sendo a prova apresentada por Bulteau e coautores [12]. O primeiro algoritmo para o problema foi proposto por Bafna e Pevzner [6] com um fator de aproximação 1.5. Posteriormente, Christie [18] apresentou uma implementação mais simples para esse algoritmo. Atualmente, o melhor algoritmo para o problema apresenta um fator de aproximação 1.375 [22] e heurísticas foram apresentadas por Dias e Dias [20] visando a obtenção de resultados práticos melhores.

Ao considerar um modelo de rearranjo composto pelos eventos de reversão e transposição em permutações com e sem sinais, obtemos os problemas de Ordenação de Permutações com Sinais por Reversões e Transposições, e Ordenação de Permutações sem Sinais por Reversões e Transposições, respectivamente. Ambos os problemas pertencem à classe de problemas NP-Difícil [29]. Os melhores algoritmos para os problemas apresentam fatores de aproximação 2 [36] e $2k$ [33] (onde k é o fator de aproximação para a decomposição de ciclos [15]) para os casos com e sem sinais, respectivamente. Diversas heurísticas considerando esses problemas foram apresentadas na literatura [21, 11].

Quando passamos a considerar que o genoma pode apresentar genes repetidos, em 2001, Christie e Irving [19] mostraram que o problema de Distância de Strings sem Sinais por Reversões pertence à classe de problemas NP-Difícil, mesmo se considerarmos um alfabeto binário (os caracteres das strings comparadas pertencem ao conjunto $\{0, 1\}$). Para isso, os autores apresentaram uma redução do problema 3-partition [**?**]. Em 2005, Radcliffe e coautores [32] mostraram que a Distância de Strings com Sinais por Reversões e Distância de Strings sem Sinais por Transposições também pertencem à classe de

problemas NP-Difícil, mesmo se considerarmos um alfabeto binário. Outra contribuição importante do trabalho foi que os autores caracterizaram um conjunto de instâncias em que é possível obter uma solução ótima em tempo polinomial.

Uma relação entre o problema de Distância de Strings por Reversões e o problema de Partição Mínima em Strings foi apresentada por Chen *et al.* [16]. Com essa relação entre os problemas, foi apresentado por Kolman e Waleń [28] um algoritmo de aproximação com fator $\Theta(k)$ para o problema de Distância de Strings com e sem Sinais por Reversões, onde k representa o número máximo de cópias de um caractere nas strings consideradas.

A representação do genoma como uma sequência ordenada de genes é uma abordagem simples e prática, mas acarreta na perda de informação referente às estruturas genéticas que não fazem parte da sequência de genes. Estudos apontaram que considerar informações adicionais contidas no genoma, além da sequência de genes, pode tornar a comparação entre genomas mais realista [9, 10] . Em particular, os pesquisadores abordaram a importância de considerar o tamanho das regiões presentes entre cada par de genes consecutivos e nas extremidades do genoma, chamadas de regiões intergênicas.

Trabalhos que levam em conta a sequência de genes e também consideram os tamanhos das regiões intergênicas começaram a ser apresentados recentemente. Fertin *et al.* [25] apresentaram um modelo que permite o uso do evento de rearranjo Double-Cut and Join (DCJ), mostraram que o problema pertence à classe de problemas NP-Difícil e desenvolveram um algoritmo de aproximação com fator 4/3. O evento de rearranjo DJC [38] atua fragmentando o genoma em dois pontos e, em seguida, as extremidades dos segmentos resultantes são unidas obedecendo certas restrições. Bulteau *et al.* [13] apresentaram um modelo que permite o uso do evento DCJ juntamente com os eventos não conservativos de inserção e deleção restritos a atuarem apenas sobre as regiões intergênicas. Para esse problema, os autores apresentaram um algoritmo exato em tempo polinomial. Oliveira *et al.* [30] apresentaram um modelo que permite o uso apenas de reversões super curtas (esse evento de rearranjo possui uma restrição adicional que faz com que todo evento de reversão afete um segmento com no máximo dois genes). Juntamente com o modelo, os autores desenvolveram algoritmos de aproximação para o problema de forma geral e para instâncias do problema com características específicas.

Trabalhos considerando a ordem dos genes e o tamanho das regiões intergênicas são recentes, sendo uma promissora linha de pesquisa, tendo em vista as melhorias que podem ser obtidas nas estimativas para a distância evolutiva entre os organismos.

Representação Clássica sem Sinais		
Modelo	Complexidade	Aproximação
Reversão		
Transposição		
DCJ		
Reversão e Transposição		

Representação Clássica com Sinais		
Modelo	Complexidade	Aproximação
Reversão		
DCJ		
Reversão e Transposição		

Capítulo 2

Definições

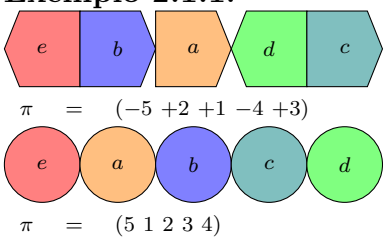
Nesse capítulo, apresentamos as formas como representamos um genoma e como os eventos de rearrajo de genomas podem afetá-los. Além disso, definimos o formato das instâncias que serão utilizadas pelos problemas investigados nos capítulos seguintes e apresentamos definições, conceitos e grafos que serão amplamente utilizados para obtenção de resultados.

2.1 Representação de Genomas

Nessa seção, apresentamos três representações de genomas que diferem nas estruturas genéticas que são incorporadas na representação computacional.

Dado um genoma $\mathcal{G} = (\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_n)$ com n genes não repetidos, utilizamos uma representação através de uma permutação $\pi = (\pi_1\ \pi_2\ \ldots\ \pi_n)$, de forma que cada elemento π_i, com $1 \leq i \leq n$, da permutação π representa o gene $\mathcal{G}_i$ do genoma $\mathcal{G}$. Caso a orientação dos genes no genome $\mathcal{G}$ seja conhecida, associamos um sinal "+" ou "−" em cada elemento π_i de π para representar a orientação de cada um dos genes de $\mathcal{G}$. Caso contrário, o sinal é simplesmente omitido. Quando representamos um genoma utilizando apenas as informações obtidas com base nas características dos genes denominamos de *representação clássica*. Além disso, denotamos por *representação clássica com sinais* quando a orientação dos genes é conhecida e *representação clássica sem sinais* caso contrário. O Exemplo 2.1.1 mostra uma representação clássica com sinais e sem sinais de genomas fictícios. Os elementos coloridos com letras no interior representam os genes, sendo que na parte supeior eles possuem orientação e na parte inferior não.

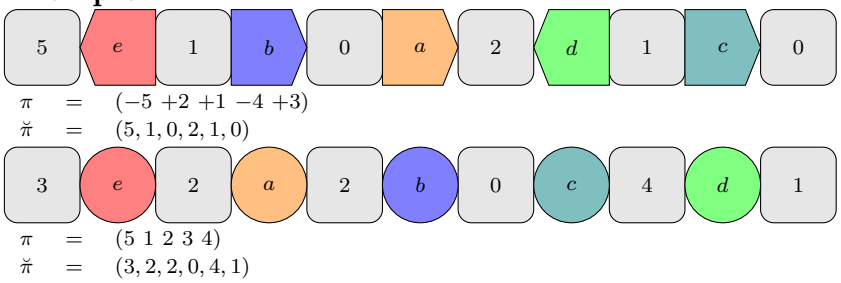
Exemplo 2.1.1.



Dado um genoma $\mathcal{G} = (\mathcal{R}_1, \mathcal{G}_1, \mathcal{R}_2, \mathcal{G}_2 \ldots, \mathcal{R}_n, \mathcal{G}_n, \mathcal{R}_{n+1})$ com n genes não repetidos $\{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_n\}$ e $n+1$ regiões intergênicas $\{\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_{n+1}\}$, utilizamos essas duas características para representar um genoma. As regiões intergênicas estão presentes nas

extremidades do genoma e entre cada par de genes consecutivos. Cada região intergênica possui uma quantidade específica de nucleotídeos, que chamamos de *tamanho*. Dessa forma, denotamos o tamanho de uma região intergênica pela quantidade de nucleotídeos contida nela. Representamos o genoma $\mathcal{G}$ utilizando dois elementos, o primeiro elemento é uma permutação $\pi = (\pi_1\ \pi_2\ \ldots\ \pi_n)$, de forma que cada elemento π_i, com $1 \leq i \leq n$, da permutação π representa o gene $\mathcal{G}_i$ do genoma $\mathcal{G}$. Caso a orientação dos genes no genome $\mathcal{G}$ seja conhecida, associamos um sinal "+" ou "−" em cada elemento π_i de π para representar a orientação de cada um dos genes de $\mathcal{G}$. Caso contrário, o sinal é simplesmente omitido. O segundo elemento é uma lista de inteiros não negativos $\breve{\pi} = (\breve{\pi}_1, \breve{\pi}_2, \ldots, \breve{\pi}_{n+1})$, de forma que cada elemento $\breve{\pi}_i$, com $1 \leq i \leq n+1$, da lista $\breve{\pi}$ representa o tamanho da região intergênica $\mathcal{R}_i$ do genoma $\mathcal{G}$. Quando representamos um genoma utilizando a informação da estrutura genética dos genes e das regiões intergênicas denominamos de *representação intergênica rígida*. Além disso, denotamos por *representação intergênica rígida com sinais* quando a orientação dos genes é conhecida e *representação intergênica rígida sem sinais* caso contrário. O Exemplo 2.1.2 mostra uma representação intergênica rígida com sinais e sem sinais de genomas fictícios. Os elementos coloridos com letras no interior representam os genes, sendo que na parte supeior eles possuem orientação e na parte inferior não. Os retangulos com bordas arredondadas entre cada par de genes e nas extremidades respresentam as regiões intergênicas com o número no interior indicando seu tamanho.

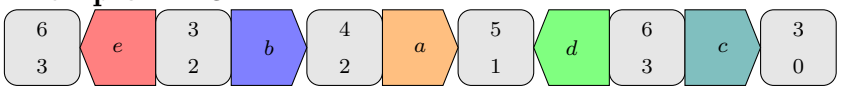
Exemplo 2.1.2.



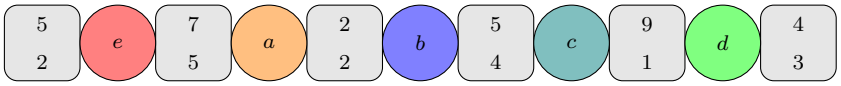
Para tornar a especificação em relação ao tamanho de cada região intergênica menos rígida, criamos uma representação denominamos de *representação intergênica flexível*. Para isso, representamos um genoma $\mathcal{G} = (\mathcal{R}_1, \mathcal{G}_1, \mathcal{R}_2, \mathcal{G}_2 \ldots, \mathcal{R}_n, \mathcal{G}_n, \mathcal{R}_{n+1})$ com n genes não repetidos $\{\mathcal{G}_1,\ \mathcal{G}_2,\ \ldots,\ \mathcal{G}_n\}$ e $n+1$ regiões intergênicas $\{\mathcal{R}_1,\ \mathcal{R}_2,\ \ldots,\ \mathcal{R}_{n+1}\}$ utilizando três elementos. O primeiro elemento é uma permutação $\pi = (\pi_1\ \pi_2\ \ldots\ \pi_n)$, de forma que cada elemento π_i, com $1 \leq i \leq n$, da permutação π representa o gene $\mathcal{G}_i$ do genoma $\mathcal{G}$. Caso a orientação dos genes no genome $\mathcal{G}$ seja conhecida, associamos um sinal "+" ou "−" em cada elemento π_i de π para representar a orientação de cada um dos genes de $\mathcal{G}$. Caso contrário, o sinal é simplesmente omitido. Os demais elementos são duas listas de inteiros não negativos $\breve{\pi}^{\min} = (\breve{\pi}^{\min}_1, \breve{\pi}^{\min}_2, \ldots, \breve{\pi}^{\min}_{n+1})$ e $\breve{\pi}^{\max} = (\breve{\pi}^{\max}_1, \breve{\pi}^{\max}_2, \ldots, \breve{\pi}^{\max}_{n+1})$, de forma que $\breve{\pi}^{\min}_i \leq \mathcal{R}_i \leq \breve{\pi}^{\max}_i$, com $1 \leq i \leq n+1$. Isso faz com que o tamanho de cada região intergênica seja flexível, tornando possível especificar um intervalos de valores aceitáveis para o tamanho de cada uma delas ao invés de apenas um único valor. Por fim, denotamos por *representação intergênica flexível com sinais* quando a orientação dos genes é conhecida e *representação intergênica flexível sem sinais* caso contrário. O Exemplo 2.1.3 mostra uma representação intergênica flexível com sinais e sem sinais de genomas fictícios. Os elemen-

tos coloridos com letras no interior representam os genes, sendo que na parte supeior eles possuem orientação e na parte inferior não. Os retangulos com bordas arredondadas entre cada par de genes e nas extremidades respresentam as regiões intergênicas. O número na parte superior de cada região intergênica indica o tamanho máximo permitido, enquanto o número na parte inferior indica o tamanho mínimo permitido.

Exemplo 2.1.3.



$\pi = (-5\ +2\ +1\ -4\ +3)$
$\breve{\pi}^{\min} = (3, 2, 2, 1, 3, 0)$
$\breve{\pi}^{\max} = (6, 3, 4, 5, 6, 3)$



$\pi = (5\ 1\ 2\ 3\ 4)$
$\breve{\pi}^{\min} = (2, 5, 2, 4, 1, 3)$
$\breve{\pi}^{\max} = (5, 7, 2, 5, 9, 4)$

Dada a representação $\mathcal{R}$ de um genoma $\mathcal{G}$, seja na forma clássica $\mathcal{R} = (\pi)$, intergênica rígida $\mathcal{R} = (\pi, \breve{\pi})$ ou intergênica flexível $\mathcal{R} = (\pi, \breve{\pi}^{\min}, \breve{\pi}^{\max})$, obtemos sua versão estendida adicionando dois novos elementos em π, com $\pi_0 = 0$ e $\pi_{n+1} = n + 1$ inseridos no início e no fim da permutação π, respectivamente. Esses dois novos elementos adicionados em π representam genes fictícios que não serão afetados por nenhum evento de rearranjo de genomas e serão utilizados apenas para tornar algumas definições, que serão apresentadas posteriormente, mais simples. De agora em diante assumimos que qualquer representação de genoma estará na sua forma estendida, a não ser que seja dito expressamente o contrário.

2.2 Eventos de Rearranjo

Nessa seção, apresentamos os eventos de rearrajo considerados nessa tese e como eles podem afetar o genoma dependendo da representação que é utilizada.

Os eventos de rearranjo de genomas são classificados em eventos conservativos ou não conservativos. O eventos de rearranjo conservativos não alteram a quantidade de material genético do genoma, enquanto os eventos de rearranjo não conservativos sim. Dado um evento de rearranjo γ e uma representação $\mathcal{R}$ de um genoma, denotamos por $\mathcal{R} \cdot \gamma$ como sendo o genoma resultante após a aplicação do evento de rearranjo γ em $\mathcal{R}$. De maneira similar, quando temos uma sequência de eventos de rerranjo $S = (\gamma_1, \gamma_2, \ldots, \gamma_k)$ e uma representação $\mathcal{R}$ de um genoma, denotamos por $\mathcal{R} \cdot S = \mathcal{R} \cdot \gamma_1 \cdot \gamma_2 \cdot \ldots \cdot \gamma_k$ como sendo o genoma resultante após a aplicação da sequência S em $\mathcal{R}$ de maneira ordenada. A seguir, mostramos como os eventos de rearranjo conservativos de reversão e transposição afetam a representação clássica de um genoma.

Definição 2.2.1. Dada uma representação clássica $\mathcal{R} = \pi$ de um genoma e sejam i e j números inteiros tal que $1 \leq i \leq j \leq n$. Uma reversão $\rho^{(i,j)}$ inverte o segmento $(\pi_i\ \pi_{i+1}\ \ldots\ \pi_{j-1}\ \pi_j)$ de π. Caso a representação $\mathcal{R}$ do genoma seja clássica com sinais, o sinal de cada elemento no segmento $(\pi_i\ \pi_{i+1}\ \ldots\ \pi_{j-1}\ \pi_j)$ também é invertido.

Os exemplos 2.2.1 e 2.2.2 mostram uma reversão $\rho^{(i,j)}$ sendo aplicada em uma representação clássica $\mathcal{R} = (\pi)$ com e sem sinais de um genoma, respectivamente.

Exemplo 2.2.1.

$$\begin{array}{lcl} \pi & = & (\pi_0\ \pi_1\ \ldots\ \pi_{i-1}\ \underline{\pi_i\ \pi_{i+1}\ \ldots\ \pi_{j-1}\ \pi_j}\ \pi_{j+1}\ \ldots\ \pi_n\ \pi_{n+1}) \\ \pi \cdot \rho^{(i,j)} & = & (\pi_0\ \pi_1\ \ldots\ \pi_{i-1}\ \underline{-\pi_j\ -\pi_{j-1}\ \ldots\ -\pi_{i+1}\ -\pi_i}\ \pi_{j+1}\ \ldots\ \pi_n\ \pi_{n+1}) \end{array}$$

Exemplo 2.2.2.

$$\begin{array}{lcl} \pi & = & (\pi_0\ \pi_1\ \ldots\ \pi_{i-1}\ \underline{\pi_i\ \pi_{i+1}\ \ldots\ \pi_{j-1}\ \pi_j}\ \pi_{j+1}\ \ldots\ \pi_n\ \pi_{n+1}) \\ \pi \cdot \rho^{(i,j)} & = & (\pi_0\ \pi_1\ \ldots\ \pi_{i-1}\ \underline{\pi_j\ \pi_{j-1}\ \ldots\ \pi_{i+1}\ \pi_i}\ \pi_{j+1}\ \ldots\ \pi_n\ \pi_{n+1}) \end{array}$$

O Exemplo 2.2.3 mostra uma uma reversão $\rho^{(2,4)}$ sendo aplicada em uma representação clássica com sinais $\mathcal{R} = \pi = (+0\ -3\ +2\ -4\ +1\ +5\ +6)$ de um genoma, enquanto o Exemplo 2.2.4 mostra uma uma reversão $\rho^{(1,5)}$ sendo aplicada em uma representação clássica sem sinais $\mathcal{R} = \pi = (0\ 4\ 5\ 3\ 2\ 1\ 6)$ de um genoma.

Exemplo 2.2.3.

$$\begin{array}{lcl} \pi & = & (+0\ -3\ \underline{+2\ -4\ +1}\ +5\ +6) \\ \pi \cdot \rho^{(2,4)} & = & (+0\ -3\ \underline{-1\ +4\ -2}\ +5\ +6) \end{array}$$

Exemplo 2.2.4.

$$\begin{array}{lcl} \pi & = & (0\ \underline{4\ 5\ 3\ 2\ 1}\ 6) \\ \pi \cdot \rho^{(1,5)} & = & (0\ \underline{1\ 2\ 3\ 5\ 4}\ 6) \end{array}$$

Definição 2.2.2. Dada uma representação clássica $\mathcal{R} = \pi$ de um genoma e sejam i, j e k números inteiros tal que $1 \leq i < j < k \leq n+1$. Uma transposição $\tau^{(i,j,k)}$ troca a posição dos segmentos consecutivos $(\pi_i\ \pi_{i+1}\ \ldots\ \pi_{j-1})$ e $(\pi_j\ \pi_{j+1}\ \ldots\ \pi_{k-1})$ de π.

O Exemplo 2.2.5 mostra uma uma transposição $\tau^{(i,j,k)}$ sendo aplicada em uma representação clássica $\mathcal{R} = (\pi)$ de um genoma. Note que a transposição pode ser aplicada em ambas as representações clássicas, com e sem sinais.

Exemplo 2.2.5.

$$\begin{array}{lcl} \pi & = & (\pi_0\ \pi_1\ \ldots\ \pi_{i-1}\ \underline{\pi_i\ \pi_{i+1}\ \ldots\ \pi_{j-1}}\ \underline{\pi_j\ \pi_{j+1}\ \ldots\ \pi_{k-1}}\ \pi_k\ \ldots\ \pi_n\ \pi_{n+1}) \\ \pi \cdot \tau^{(i,j,k)} & = & (\pi_0\ \pi_1\ \ldots\ \pi_{i-1}\ \underline{\pi_j\ \pi_{j+1}\ \ldots\ \pi_{k-1}}\ \underline{\pi_i\ \pi_{i+1}\ \ldots\ \pi_{j-1}}\ \pi_k\ \ldots\ \pi_n\ \pi_{n+1}) \end{array}$$

O Exemplo 2.2.6 mostra uma uma transposição $\tau^{(1,3,5)}$ sendo aplicada em uma representação clássica com sinais $\mathcal{R} = \pi = (+0\ -4\ -3\ +1\ +2\ +5\ +6)$ de um genoma, enquanto o Exemplo 2.2.7 mostra uma uma transposição $\tau^{(4,5,6)}$ sendo aplicada em uma representação clássica sem sinais $\mathcal{R} = \pi = (0\ 3\ 2\ 1\ 5\ 4\ 6)$ de um genoma.

Exemplo 2.2.6.

$$\begin{array}{lcl} \pi & = & (+0\ \underline{-4\ -3}\ \underline{+1\ +2}\ +5\ +6) \\ \pi \cdot \tau^{(1,3,5)} & = & (+0\ \underline{+1\ +2}\ \underline{-4\ -3}\ +5\ +6) \end{array}$$

Exemplo 2.2.7.

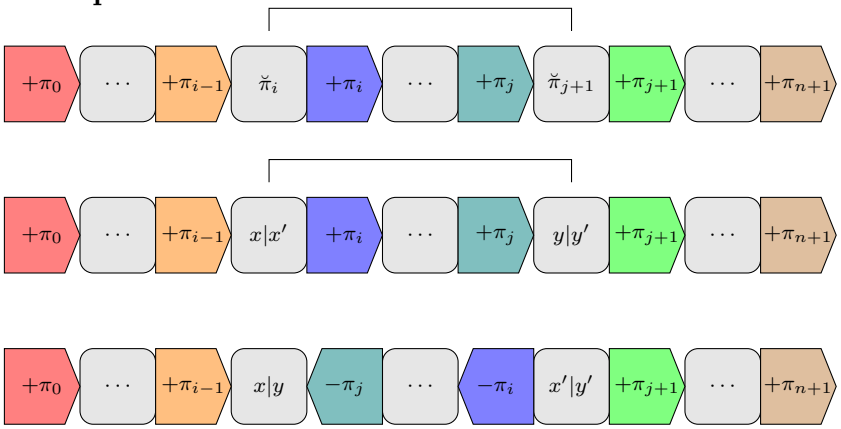
$$\begin{array}{lcl} \pi & = & (0\ 3\ 2\ 1\ \underline{5}\ \underline{4}\ 6) \\ \pi \cdot \tau^{(4,5,6)} & = & (0\ 3\ 2\ 1\ \underline{4}\ \underline{5}\ 6) \end{array}$$

A seguir, mostramos como os eventos de rearranjo conservativos de reversão intergênica, transposição intergênica e move intergênico afetam a representação intergênica de um genoma.

Definição 2.2.3. Dada uma representação intergênica rígida $\mathcal{R} = (\pi, \breve{\pi})$ de um genoma e sejam i, j, x e y números inteiros tal que $1 \leq i \leq j \leq n$, $0 \leq x \leq \breve{\pi}_i$ e $0 \leq y \leq \breve{\pi}_{j+1}$. Uma reversão intergênica $\rho^{(i,j)}_{(x,y)}$ divide as regiões intergênicas $\breve{\pi}_i$ e $\breve{\pi}_{j+1}$ da sequinte forma: $\breve{\pi}_i$ em partes com tamanho x e x', com $x' = \breve{\pi}_i - x$, e $\breve{\pi}_{j+1}$ em partes com tamanho y e y', com $y' = \breve{\pi}_{j+1} - y$. Em seguida, a sequência $(x', \pi_i, \breve{\pi}_{i+1} \dots \breve{\pi}_j, \pi_j, y)$ do genoma é invertida. Caso a representação seja com sinais, o sinal dos elementos de π_i até π_j também é invertida. Por fim os segmentos do genoma são remontados com os pares de partes (x, y) e (x', y') fundindo-se e formando as novas regiões intergênicas $\breve{\pi}_i$ e $\breve{\pi}_{j+1}$ com tamanhos $x + y$ e $x' + y'$, respectivamente.

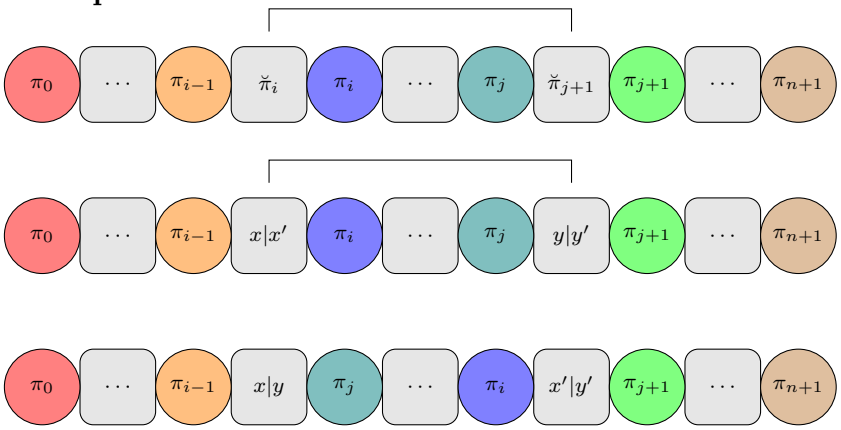
O Exemplo 2.2.8 mostra uma reversão intergênica $\rho^{(i,j)}_{(x,y)}$ genérica sendo aplicada em uma representação intergênica rígida com sinais de um genoma.

Exemplo 2.2.8.



O Exemplo 2.2.9 mostra uma reversão intergênica $\rho^{(i,j)}_{(x,y)}$ genérica sendo aplicada em uma representação intergênica rígida sem sinais de um genoma.

Exemplo 2.2.9.



O Exemplo 2.2.10 mostra uma reversão intergênica $\rho^{(2,4)}_{(2,0)}$ sendo aplicada em uma representação intergênica rígida com sinais $\mathcal{R} = (\pi, \breve{\pi}) = ((+0\ -3\ +2\ -4\ +1\ +5\ +6), (1, 4, 4, 2, 0, 3))$ de um genoma, enquanto o Exemplo 2.2.11 mostra uma reversão intergênica $\rho^{(1,5)}_{(1,2)}$ sendo aplicada em uma representação intergênica rígida sem sinais $\mathcal{R} = (\pi, \breve{\pi}) =$

$((0\ 4\ 5\ 3\ 2\ 1\ 6), (1, 1, 7, 3, 0, 2))$ de um genoma. As regiões intergênicas marcadas com sobrescrito podem ter o tamanho alterado pelo evento, enquanto as regiões intergênicas marcadas com subscrito sofrem apenas uma troca de posição.

Exemplo 2.2.10.

$$\begin{aligned}(\pi, \breve{\pi}) &= ((+0\ -3\ \underline{+2\ -4\ +1}\ +5\ +6), (1, \overline{4}, \underline{4, 2}, \overline{0}, 3)) \\ (\pi, \breve{\pi}) \cdot \rho_{(2,0)}^{(2,4)} &= ((+0\ -3\ \underline{-1\ +4\ -2}\ +5\ +6), (1, \overline{2}, \underline{2, 4}, \overline{2}, 3))\end{aligned}$$

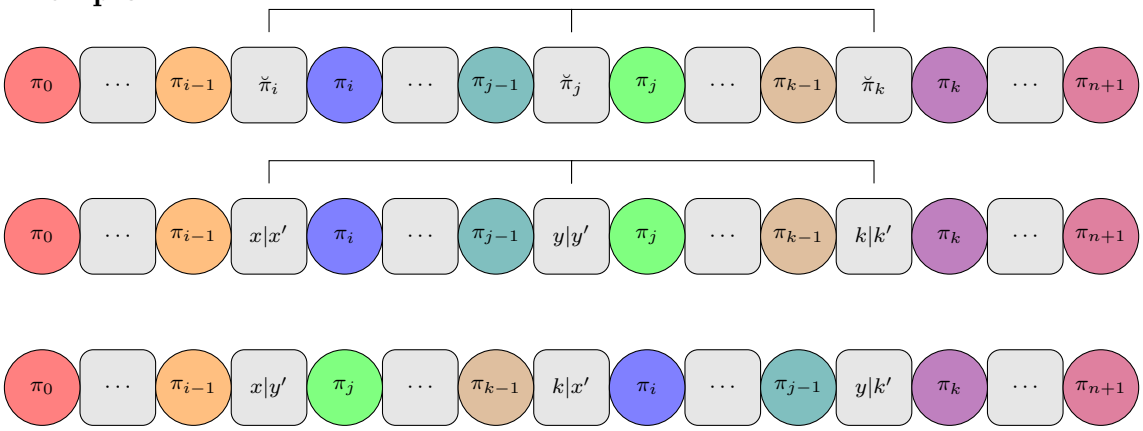
Exemplo 2.2.11.

$$\begin{aligned}(\pi, \breve{\pi}) &= ((0\ \underline{4\ 5\ 3\ 2\ 1}\ 6), (\overline{1}, \underline{1, 7, 3, 0}, \overline{2})) \\ (\pi, \breve{\pi}) \cdot \rho_{(1,2)}^{(1,5)} &= ((0\ \underline{1\ 2\ 3\ 5\ 4}\ 6), (\overline{3}, \underline{0, 3, 7, 1}, \overline{0}))\end{aligned}$$

Definição 2.2.4. Dada uma representação intergênica rígida $\mathcal{R} = (\pi, \breve{\pi})$ de um genoma e sejam i, j, k, x, y e z números inteiros tal que $1 \leq i < j < k \leq n+1$, $0 \leq x \leq \breve{\pi}_i$, $0 \leq y \leq \breve{\pi}_j$ e $0 \leq z \leq \breve{\pi}_k$. Uma transposição intergênica $\tau_{(x,y,z)}^{(i,j,k)}$ divide as regiões intergênicas $\breve{\pi}_i$, $\breve{\pi}_j$ e $\breve{\pi}_k$ da sequinte forma: $\breve{\pi}_i$ em partes com tamanho x e x', com $x' = \breve{\pi}_i - x$, $\breve{\pi}_j$ em partes com tamanho y e y', com $y' = \breve{\pi}_j - y$, e $\breve{\pi}_k$ em partes com tamanho z e z', com $z' = \breve{\pi}_k - z$. Em seguida, as sequências consecutivas $(x', \pi_i, \breve{\pi}_{i+1}, \ldots \breve{\pi}_{j-1}, \pi_{j-1}, y)$ e $(y', \pi_j, \breve{\pi}_{j+1} \ldots \breve{\pi}_{k-1}, \pi_{k-1}, z)$ trocam de posição sem alterar a orientação dos genes contidos nos segmentos. Por fim os segmentos do genoma são remontados com os pares de partes (x, y'), (z, x') e (y, z') fundindo-se e formando as novas regiões intergênicas $\breve{\pi}_i$, $\breve{\pi}_{k+i-j}$, e $\breve{\pi}_k$ com tamanhos $x + y'$, $z + x'$ e $y + z'$, respectivamente.

O Exemplo 2.2.12 mostra uma transposição intergênica $\tau_{(x,y,z)}^{(i,j,k)}$ genérica sendo aplicada em uma representação intergênica rígida de um genoma. Note que caso a representação utilizada seja com sinais o evento não altera a orientação dos genes nos segmentos afetados.

Exemplo 2.2.12.



O Exemplo 2.2.13 mostra uma transposição intergênica $\tau_{(1,1,3)}^{(1,3,6)}$ sendo aplicada em uma representação intergênica rígida com sinais $\mathcal{R} = (\pi, \breve{\pi}) = ((+0\ -4\ -3\ +1\ +2\ +5\ +6), (3, 0, 2, 2, 4, 7))$ de um genoma, enquanto o Exemplo 2.2.14 mostra uma transposição intergênica $\tau_{(0,0,1)}^{(4,5,6)}$ sendo aplicada em uma representação intergênica rígida sem sinais $\mathcal{R} = (\pi, \breve{\pi}) = ((0\ 3\ 2\ 1\ 5\ 4\ 6), (3, 2, 4, 1, 0, 2))$ de um genoma. As regiões intergênicas marcadas com sobrescrito podem ter o tamanho alterado pelo evento, enquanto as regiões intergênicas marcadas com subscrito sofrem apenas uma troca de posição.

Exemplo 2.2.13.

$$\begin{array}{lcl}(\pi,\breve{\pi}) & = & ((+0\ \underline{-4\ -3}\ \underline{+1\ +2\ +5}\ +6),(\overline{3},\underline{0},\overline{2},\underline{2,4},\overline{7})) \\ (\pi,\breve{\pi})\cdot\tau_{(1,1,3)}^{(1,3,6)} & = & ((+0\ \underline{+1\ +2\ +5}\ \underline{-4\ -3}\ +6),(\overline{2},\underline{2,4},\overline{5},\underline{0},\overline{5}))\end{array}$$

Exemplo 2.2.14.

$$\begin{array}{lcl}(\pi,\breve{\pi}) & = & ((0\ 3\ 2\ 1\ \underline{5}\ \underline{4}\ 6),(3,2,4,\overline{1},\overline{0},\overline{2})) \\ (\pi,\breve{\pi})\cdot\tau_{(0,0,1)}^{(4,5,6)} & = & ((0\ 3\ 2\ 1\ \underline{4}\ \underline{5}\ 6),(3,2,4,\overline{0},\overline{2},\overline{1}))\end{array}$$

Definição 2.2.5. Dada uma representação intergênica rígida $\mathcal{R}=(\pi,\breve{\pi})$ de um genoma e sejam i, j e x números inteiros tal que $1\leq i,j\leq n$ e $0\leq x\leq\breve{\pi}_i$. Um move intergênico $\mu_{(x)}^{(i,j)}$ transfere x nucleotídeos da região intergênica $\breve{\pi}_i$ para a região intergênica $\breve{\pi}_j$.

O Exemplo 2.2.15 mostra um move intergênico $\mu_{(3)}^{(2,5)}$ sendo aplicado em uma representação intergênica rígida com sinais $\mathcal{R}=(\pi,\breve{\pi})=((+0\ -3\ +2\ -4\ +1\ +5\ +6),(1,4,4,2,0,3))$ de um genoma, enquanto o Exemplo 2.2.16 mostra um indel intergênico $\mu_{(5)}^{(3,5)}$ sendo aplicado em uma representação intergênica rígida sem sinais $\mathcal{R}=(\pi,\breve{\pi})=((0\ 4\ 5\ 3\ 2\ 1\ 6),(1,1,7,3,0,2))$ de um genoma. As regiões intergênicas marcadas com sobrescrito sofrem alteração no tamanho causado pelo evento.

Exemplo 2.2.15.

$$\begin{array}{lcl}(\pi,\breve{\pi}) & = & ((+0\ -3\ +2\ -4\ +1\ +5\ +6),(1,\overline{4},4,2,\overline{0},3)) \\ (\pi,\breve{\pi})\cdot\mu_{(3)}^{(2,5)} & = & ((+0\ -3\ -1\ +4\ -2\ +5\ +6),(1,\overline{1},4,2,\overline{3},3))\end{array}$$

Exemplo 2.2.16.

$$\begin{array}{lcl}(\pi,\breve{\pi}) & = & ((0\ 4\ 5\ 3\ 2\ 1\ 6),(1,1,\overline{7},3,\overline{0},2)) \\ (\pi,\breve{\pi})\cdot\mu_{(5)}^{(3,5)} & = & ((0\ 1\ 2\ 3\ 5\ 4\ 6),(1,1,\overline{2},3,\overline{5},2))\end{array}$$

A seguir, mostramos como o evento de rearranjo não conservativo de indel intergênico afeta a representação intergênica de um genoma.

Definição 2.2.6. Dada uma representação intergênica rígida $\mathcal{R}=(\pi,\breve{\pi})$ de um genoma e sejam i e x números inteiros tal que $1\leq i\leq n$ e $x\geq-\breve{\pi}_i$. Um indel intergênico $\delta_{(x)}^{(i)}$ remove x nucleotídeos da região intergênica $\breve{\pi}_i$ caso x seja negativo. Caso contrário, um indel intergênico $\delta_{(x)}^{(i)}$ insere x nucleotídeos na região intergênica $\breve{\pi}_i$.

Note que o evento de rearranjo indel intergênico é uma forma compacta para definir os eventos de inserção e deleção utilizando a mesma notação. O Exemplo 2.2.17 mostra um indel intergênico $\delta_{(9)}^{(5)}$ sendo aplicado em uma representação intergênica rígida com sinais $\mathcal{R}=(\pi,\breve{\pi})=((+0\ -3\ +2\ -4\ +1\ +5\ +6),(3,5,1,0,2,1))$ de um genoma, enquanto o Exemplo 2.2.18 mostra um indel intergênico $\delta_{(-6)}^{(6)}$ sendo aplicado em uma representação intergênica rígida sem sinais $\mathcal{R}=(\pi,\breve{\pi})=((0\ 4\ 5\ 3\ 2\ 1\ 6),(3,3,2,1,0,7))$ de um genoma. As regiões intergênicas marcadas com sobrescrito sofrem alteração no tamanho causado pelo evento.

Exemplo 2.2.17.

$$\begin{array}{lcl}(\pi,\breve{\pi}) & = & ((+0\ -3\ +2\ -4\ +1\ +5\ +6),(3,5,1,0,\overline{2},1)) \\ (\pi,\breve{\pi})\cdot\delta_{(9)}^{(5)} & = & ((+0\ -3\ -1\ +4\ -2\ +5\ +6),(3,5,1,0,\overline{11},1))\end{array}$$

Exemplo 2.2.18.

$$\begin{array}{lcl}(\pi,\breve{\pi}) & = & ((0\ 4\ 5\ 3\ 2\ 1\ 6),(3,3,2,1,0,\overline{7})) \\ (\pi,\breve{\pi})\cdot\delta_{(-6)}^{(6)} & = & ((0\ 1\ 2\ 3\ 5\ 4\ 6),(3,3,2,1,0,\overline{1}))\end{array}$$

2.3 Caracterização das Instâncias

Os problemas investigados nessa tese tem como principal objetivo transformar uma representação de um genoma de origem $\mathcal{R}_o$ em uma representação de um genoma alvo $\mathcal{R}_a$ utilizando eventos de rearranjo de genoma para realizar essa tarefa. Um *modelo de rearranjo* $\mathcal{M}$ é um conjunto de eventos de rearranjo que podem ser utilizados para transformar um genoma em outro. Os problemas de distância entre genomas buscam por a menor sequência de eventos de rearranjo $S = (\gamma_1, \gamma_2, \ldots, \gamma_k)$ pertencentes a um modelo $\mathcal{M}$ de forma que $\mathcal{R}_o \cdot S = \mathcal{R}_a$. A *distância* entre $\mathcal{R}_o$ e $\mathcal{R}_a$ no modelo $\mathcal{M}$ é o tamanho da menor sequência de eventos de rearranjo capaz de transformar $\mathcal{R}_o$ em $\mathcal{R}_a$, e é denotada por $d_{\mathcal{M}}(\mathcal{R}_o, \mathcal{R}_a)$. Os problemas de distância entre genomas assumem que cada evento de rearranjo em um modelo possui a mesma probabilidade de ocorrer em um cenário evolutivo. Entretanto, outra abordagem utiliza é associar um peso para cada tipo de evento de rearranjo pertencente a um modelo de rearranjo. Com isso, temos os problemas de distância ponderada entre genomas, que buscam por uma sequência de eventos de rearranjo $S = (\gamma_1, \gamma_2, \ldots, \gamma_k)$ pertencentes a um modelo $\mathcal{M}$ de forma que $\mathcal{R}_o \cdot S = \mathcal{R}_a$ e que o valor de $\sum_{\gamma_i \in S} p(\gamma_i)$ seja mínimo, onde $p(\gamma_i)$ representa o peso associado ao tipo do evento γ_i no modelo $\mathcal{M}$. A *distância ponderada* entre $\mathcal{R}_o$ e $\mathcal{R}_a$ no modelo $\mathcal{M}$ é o menor valor de $\sum_{\gamma_i \in S} p(\gamma_i)$ para uma sequência de eventos de rearrajo S e que $\mathcal{R}_o \cdot S = \mathcal{R}_a$, e é denotada por $dp_{\mathcal{M}}(\mathcal{R}_o, \mathcal{R}_a)$. A seguir descrevemos os tipos de instâncias que os problemas investigados posteriormente podem receber como entrada.

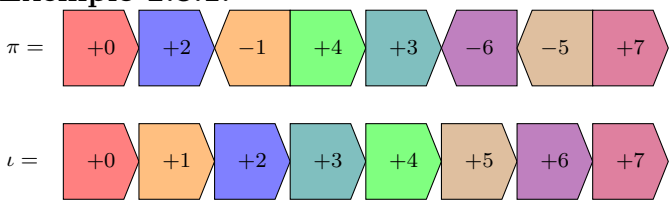
- Uma *instância clássica* é caracterizada por um par de representações clássicas de genomas (π, ι) que compartilham o mesmo conjunto de genes, sendo que ambas as representações podem ser com ou sem sinais. Por padrão, em uma instância clássica utilizaremos π e ι como sendo a representação do genoma de origem e alvo, respectivamente. O objetivo principal dos problemas que utilizam esse tipo de instância consiste em transformar π em ι.

- Uma *instância intergênica rígida* é caracterizada por um par de representações intergênicas rígidas de genomas $((\pi, \breve{\pi}), (\iota, \breve{\iota}))$ que compartilham o mesmo conjunto de genes, sendo que ambas as representações podem ser com ou sem sinais. Por padrão, em uma instância intergênica rígida utilizaremos $(\pi, \breve{\pi})$ e $(\iota, \breve{\iota})$ como sendo a representação do genoma de origem e alvo, respectivamente. O objetivo principal dos problemas que utilizam esse tipo de instância consiste em transformar $(\pi, \breve{\pi})$ em $(\iota, \breve{\iota})$.

- Uma *instância intergênica flexível* é caracterizada por um par de representações de genomas $((\pi, \breve{\pi}), (\iota, \breve{\iota}^{\min}, \breve{\iota}^{\max}))$ que compartilham o mesmo conjunto de genes, sendo a primeira representação intergênica rígida e a segunda intergênica flexível. Ambas as representações podem ser com ou sem sinais. Por padrão, em uma instância intergênica flexível utilizaremos $(\pi, \breve{\pi})$ e $(\iota, \breve{\iota}^{\min}, \breve{\iota}^{\max})$ como sendo a representação do genoma de origem e alvo, respectivamente. O objetivo principal dos problemas que utilizam esse tipo de instância consiste em transformar $(\pi, \breve{\pi})$ em $(\iota, \breve{\pi}')$, tal que $\breve{\iota}_i^{\min} \leq \breve{\pi}'_i \leq \breve{\iota}_i^{\max}$, com $1 \leq i \leq n+1$.

Pelo fato de utilizarmos a representação dos genes de um genoma através de uma permutação e os genomas origem e alvo compartilharem o mesmo conjunto de genes, podemos determinar uma permutação padrão ι para os genes do genoma alvo e mapear a permutação do genoma de origem π de acordo com os valores utilizados em ι. A permutação padrão para os genes do genoma alvo é $\iota = (+1\ +2\ \ldots\ +n)$ para uma representação com sinais e $\iota = (1\ 2\ \ldots\ n)$ para uma representação sem sinais. O exemplo 2.3.1 mostram uma instância clássica com sinais.

Exemplo 2.3.1.



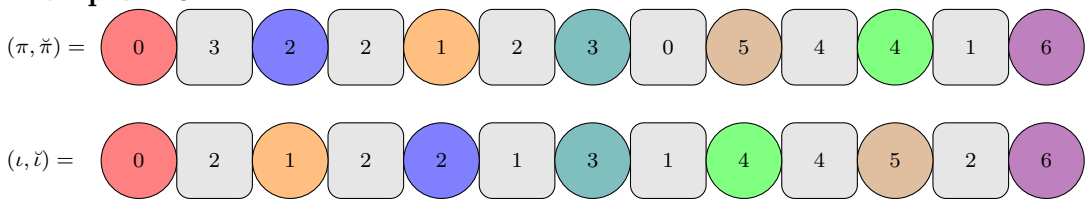
Definição 2.3.1. Dada uma instância intergênica rígida $\mathcal{I} = ((\pi, \breve{\pi}), (\iota, \breve{\iota}))$, $\mathcal{I}$ é chamada de *balanceada* se a seguinte igualdade é satisfeita:

$$\sum_{\breve{\pi}_i \ \in \ \breve{\pi}} \breve{\pi}_i = \sum_{\breve{\iota}_i \ \in \ \breve{\iota}} \breve{\iota}_i.$$

Caso contrário, $\mathcal{I}$ é chamada de *desbalanceada*.

O exemplo 2.3.2 mostra uma instância intergênica rígida balanceada sem sinais.

Exemplo 2.3.2.



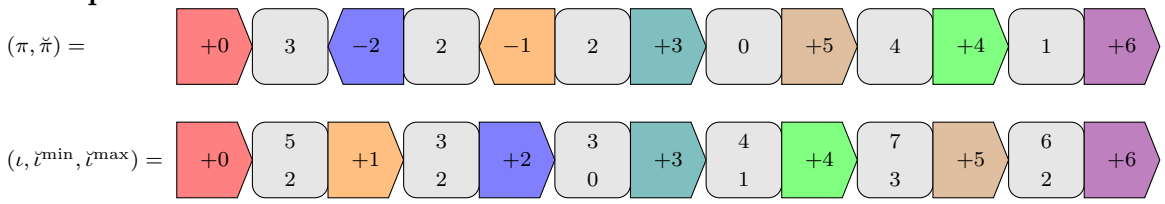
Definição 2.3.2. Dada uma instância intergênica flexível $\mathcal{I} = ((\pi, \breve{\pi}), (\iota, \breve{\iota}^{\min}, \breve{\iota}^{\max}))$, $\mathcal{I}$ é chamada de *balanceada* se a seguinte desigualdade é satisfeita:

$$\sum_{\breve{\iota}_i^{\min} \ \in \ \breve{\iota}^{\min}} \breve{\iota}_i^{\min} \leq \sum_{\breve{\pi}_i \ \in \ \breve{\pi}} \breve{\pi}_i \leq \sum_{\breve{\iota}_i^{\max} \ \in \ \breve{\iota}^{\max}} \breve{\iota}_i^{\max}.$$

Caso contrário, $\mathcal{I}$ é chamada de *desbalanceada*.

O exemplo 2.3.3 mostram uma instância intergênica flexível balanceada com sinais.

Exemplo 2.3.3.



Note que instâncias intergênicas rígidas e flexíveis balanceadas possuem, no genoma de origem, um total de nucleotídeos em que é possível atender todas as retrições referentes

aos tamanhos permitidos para cada região intergênica no genoma alvo. Por outro lado, em instâncias intergênicas rígidas e flexíveis desbalanceadas é necessário inserir ou remover nucleotídeos do genoma de origem para tornar possível transformá-lo no genoma alvo.

2.4 Breakpoints

Nessa seção, apresentamos os conceitos de breakpoints em instâncias clássicas e intergênicas rígidas. Esses conceitos são importantes para obtenção de limitantes inferiores e desenvolvimento de algoritmos para problemas que serão investigados nos capítulos seguintes.

2.4.1 Breakpoint Clássico

Nessa seção, apresentamos o conceito de breakpoint clássico.

Definição 2.4.1. Dada uma instância clássica $\mathcal{I} = (\pi, \iota)$, um par de elementos (π_i, π_{i+1}), de forma que $0 \leq i \leq n$, é um *breakpoint clássico tipo um* se $|\pi_{i+1} - \pi_i| \neq 1$.

Definição 2.4.2. Dada uma instância clássica $\mathcal{I} = (\pi, \iota)$, um par de elementos (π_i, π_{i+1}), de forma que $0 \leq i \leq n$, é um *breakpoint clássico tipo dois* se $\pi_{i+1} - \pi_i \neq 1$.

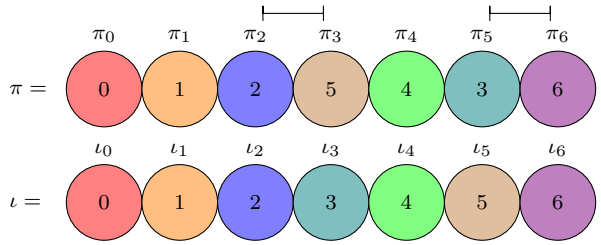
Dada uma instância clássica $\mathcal{I} = (\pi, \iota)$, o número total de breakpoints clássicos tipo um é denotado por $b_1(\mathcal{I})$. A variação no número de breakpoints clássicos tipo um após aplicar uma sequência de eventos de rearranjo S em π é denotada por $\Delta b_1(\mathcal{I}, S) = b_1(\mathcal{I}') - b_1(\mathcal{I})$, onde $\mathcal{I}' = (\pi', \iota)$ com $\pi' = \pi \cdot S$. O número total de breakpoints clássicos tipo dois é denotado por $b_2(\mathcal{I})$. A variação no número de breakpoints clássicos tipo dois após aplicar uma sequência de eventos de rearranjo S em π é denotada por $\Delta b_2(\mathcal{I}, S) = b_2(\mathcal{I}') - b_2(\mathcal{I})$, onde $\mathcal{I}' = (\pi', \iota)$ com $\pi' = \pi \cdot S$.

Definição 2.4.3. Dada uma instância clássica $\mathcal{I} = (\pi, \iota)$, *strips* são sequências maximais de elementos de π sem breakpoints clássicos.

Uma strip obtida de uma instância clássica sem sinais $\mathcal{I} = (\pi, \iota)$ com apenas um elemento π_i é chamada de *singleton* e é definida como crescente caso $i \in \{0, n\}$. Caso contrário, é definida como decrescente. Strips com mais de um elemento são chamadas de crescentes caso os elementos formem uma sequência crescente. Caso contrário, são chamadas de decrescentes. Uma strip obtida de uma instância clássica com sinais $\mathcal{I} = (\pi, \iota)$ é definida como positiva caso todos os elementos da stips tenham sinal positivo. Caso contrário, a strip é definida como negativa.

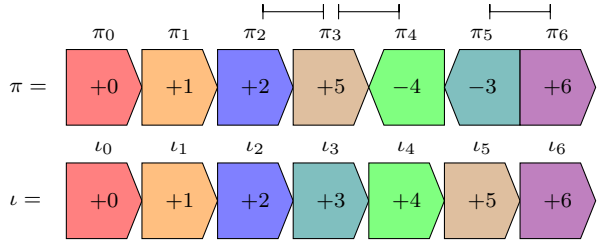
O Exemplo 2.4.1 mostra uma instância clássica sem sinais $\mathcal{I}$ = ((0 1 2 5 4 3 6), (0 1 2 3 4 5 6)). Note que a instância possui dois breakpoints clássicos tipo um ($b_1(\mathcal{I}) = 2$), sendo eles (π_2, π_3) e (π_5, π_6). Além disso, obtemos as seguintes strips da instância $\mathcal{I}$: (0 1 2), (5 4 3) e (6), sendo que (0 1 2) e (6) são strips crescentes enquanto (5 4 3) é uma strip decrescente.

Exemplo 2.4.1.



O Exemplo 2.4.2 mostra uma instância clássica com sinais $\mathcal{I} = ((+0\ +1\ +2\ +5\ -4\ -3\ +6), (+0\ +1\ +2\ +3\ +4\ +5\ +6))$. Note que a instância possui três breakpoints clássicos tipo dois ($b_2(\mathcal{I}) = 3$), sendo eles (π_2, π_3), (π_3, π_4) e (π_5, π_6). As strips obtidas dessa instância com esses breakpoints clássicos tipo dois são: $(+0\ +1\ +2)$, $(+5)$, $(-4\ -3)$ e $(+6)$. Sendo que $(+0\ +1\ +2)$, $(+5)$ e $(+6)$ são strips positivas enquanto $(-4\ -3)$ é uma strip negativa.

Exemplo 2.4.2.



2.4.2 Breakpoint Intergênico

Nessa seção, apresentamos o conceito de breakpoint intergênico.

Definição 2.4.4. Dada uma instância intergênica rígida $\mathcal{I} = ((\pi, \breve{\pi}), (\iota, \breve{\iota}))$, um par de elementos (π_i, π_{i+1}), de forma que $0 \leq i \leq n$, é um *breakpoint intergênico tipo um* se um dos seguintes casos ocorrer:

- $|\pi_{i+1} - \pi_i| \neq 1$
- $|\pi_{i+1} - \pi_i| = 1$ e $\breve{\pi}_{i+1} \neq \breve{\iota}_x$, tal que $x = \max(\pi_i, \pi_{i+1})$.

Definição 2.4.5. Dada uma instância intergênica rígida $\mathcal{I} = ((\pi, \breve{\pi}), (\iota, \breve{\iota}))$, um par de elementos (π_a, π_b) é uma *adjacência intergênica* se $|a - b| = 1$ e o par $(\pi_{\min(a,b)}, \pi_{\max(a,b)})$ não é um breakpoint intergênico tipo um.

Note que um breakpoint intergênico tipo um indica um ponto no genoma de origem que deve ser afeto por algum rearranjo de genoma com o objetivo de transformá-lo no genoma alvo. Por outro lado, uma adjacência intergênica mostra um ponto no genoma de origem em que o par de genes considerados também são consecutivos no genoma alvo. Além disso, a região integênica entre os genes tem o mesmo tamanho no genoma origem e alvo.

Definição 2.4.6. Dada uma instância intergênica rígida $\mathcal{I} = ((\pi, \breve{\pi}), (\iota, \breve{\iota}))$ e breakpoint intergênico tipo um (π_i, π_{i+1}), tal que $|\pi_{i+1} - \pi_i| = 1$, é chamado de *sobrecarregado* se $\breve{\pi}_{i+1} > \breve{\iota}_x$, com $x = \max(\pi_i, \pi_{i+1})$. Caso contrário, o breakpoint intergênico tipo um (π_i, π_{i+1}) é chamado de *subcarregado*.

Observe que um breakpoint intergênico sobrecarregado é formado por um par de genes que são consecutivos no genoma de origem e alvo. Contudo, o tamanho da região intergênica entre o par de genes do genoma origem é maior do que entre o mesmo par de genes no genoma alvo. Já um breakpoint intergênico subcarregado é justamente o cenário oposto, o par de genes são consecutivos no genoma origem e alvo, mas a região intergênica entre o par de genes do genoma origem é menor do que entre o mesmo par de genes no genoma alvo.

Definição 2.4.7. Um breakpoint intergênico tipo um (π_i, π_{i+1}) é chamado de *forte* se (π_i, π_{i+1}) é um breakpoint intergênico sobrecarregado ou subcarregado. Caso contrário, o breakpoint intergênico tipo um (π_i, π_{i+1}) é chamado de *suave*.

Definição 2.4.8. Um breakpoint intergênico forte (π_i, π_{i+1}) é chamado de *super forte* se um dos seguintes casos ocorrer:

- $i \in \{0, n\}$
- (π_{i-1}, π_i) ou (π_{i+1}, π_{i+2}) são um breakpoint intergênico forte ou uma adjacência intergênica.

Note que um breakpoint intergênico super forte está em uma das extremidades do genoma de origem ou imediatamente antes ou depois existe um breakpoint intergênico forte ou uma adjacência intergênica.

Definição 2.4.9. Dada uma instância intergênica rígida $\mathcal{I} = ((\pi, \breve{\pi}), (\iota, \breve{\iota}))$, um par de breakpoints intergênicos tipo um (π_i, π_{i+1}) e (π_j, π_{j+1}) é chamado de *conectado* se ambas as condições a seguir são satisfeitas:

1. O par de elementos (π_i, π_{i+1}), (π_j, π_{j+1}), (π_i, π_j), (π_i, π_{j+1}), (π_{i+1}, π_j) ou (π_{i+1}, π_{j+1}) são consecutivos em ι e não forma uma adjacência intergênica.
2. $\breve{\pi}_{i+1} + \breve{\pi}_{j+1} \geq \breve{\iota}_k$, tal que $\breve{\iota}_k$ é o tamanho da região intergênica entre o par de elementos consecutivos (que satisfaz a condição 1) em ι.

Um par de breakpoints intergênicos conectados indica a possibilidade de criar uma adjacência intergênica utilizando apenas o material de ambos os breakpoints intergênicos tipo um (genes e nucleotídeos das regiões intergênicas).

Definição 2.4.10. Dada uma instância intergênica rígida $\mathcal{I} = ((\pi, \breve{\pi}), (\iota, \breve{\iota}))$, um par de breakpoints intergênicos conectados (π_i, π_{i+1}) e (π_j, π_{j+1}) é chamado de *suavemente conectado* se ambos os breakpoints intergênicos são suaves.

Definição 2.4.11. Dada uma instância intergênica rígida $\mathcal{I} = ((\pi, \breve{\pi}), (\iota, \breve{\iota}))$, *strips suaves* são sequências maximais de elementos de π sem breakpoints intergênicos suaves.

Uma strip suave com apenas um elemento π_i é chamada de *singleton* e é definida como crescente caso $i \in \{0, n\}$. Caso contrário, é definida como decrescente. Strips suaves com mais de um elemento são chamadas de crescentes caso os elementos formem uma sequência crescente. Caso contrário, são chamadas de decrescentes.

Definição 2.4.12. Dada uma instância intergênica rígida $\mathcal{I} = ((\pi, \breve{\pi}), (\iota, \breve{\iota}))$, um par de elementos (π_i, π_{i+1}), de forma que $0 \leq i \leq n$, é um *breakpoint intergênico tipo dois* se um dos seguintes casos ocorrer:

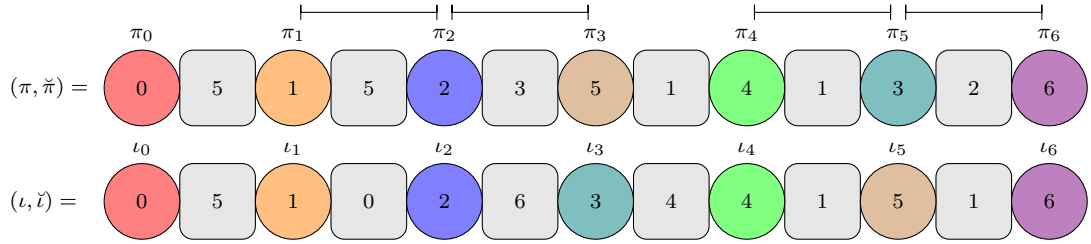
- $\pi_{i+1} - \pi_i \neq 1$
- $\pi_{i+1} - \pi_i = 1$ e $\breve{\pi}_{i+1} \neq \breve{\iota}_x$, tal que $x = \max(|\pi_i|, |\pi_{i+1}|)$.

Os breakpoints intergênicos tipo um e dois são utilizados dependendo do tipo da instância intergênica rígida (com o sem sinais) e do modelo de rearranjo que é considerado, mas ambos os conceitos indicam a mesma informação: os pontos que devem ser afetados no genoma de origem para transformá-lo no genoma alvo.

Dada uma instância intergênica rígida $\mathcal{I} = ((\pi, \breve{\pi}), (\iota, \breve{\iota}))$, o número total de breakpoints fortes e suaves são denotados por $ib_f(\mathcal{I})$ e $ib_s(\mathcal{I})$, respectivamente. O número total de breakpoints intergênicos tipo um é denotado por $ib_1(\mathcal{I}) = ib_f(\mathcal{I}) + ib_s(\mathcal{I})$. A variação no número de breakpoints intergênicos tipo um após aplicar uma sequência de eventos de rearranjo S em $(\pi, \breve{\pi})$ é denotada por $\Delta ib_1(\mathcal{I}, S) = ib_1(\mathcal{I}') - ib_1(\mathcal{I})$, onde $\mathcal{I}' = ((\pi', \breve{\pi}'), (\iota, \breve{\iota}))$ com $(\pi', \breve{\pi}') = (\pi, \breve{\pi}) \cdot S$. O número total de breakpoints intergênicos tipo dois é denotado por $ib_2(\mathcal{I})$. A variação no número de breakpoints intergênicos tipo dois após aplicar uma sequência de eventos de rearranjo S em $(\pi, \breve{\pi})$ é denotada por $\Delta ib_2(\mathcal{I}, S) = ib_2(\mathcal{I}') - ib_2(\mathcal{I})$, onde $\mathcal{I}' = ((\pi', \breve{\pi}'), (\iota, \breve{\iota}))$ com $(\pi', \breve{\pi}') = (\pi, \breve{\pi}) \cdot S$.

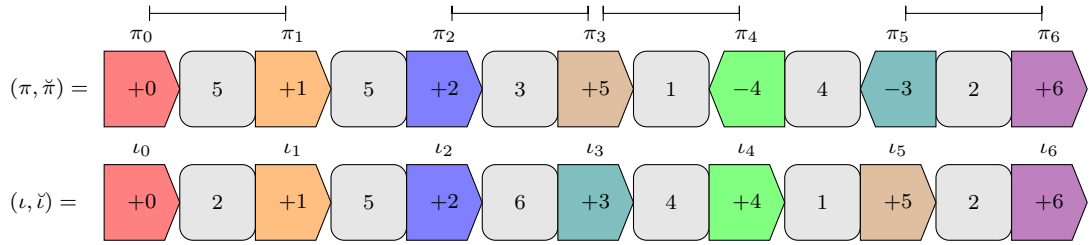
O Exemplo 2.4.3 mostra uma instância intergênica rígida sem sinais $\mathcal{I} = (((0\ 1\ 2\ 5\ 4\ 3\ 6), (5, 5, 3, 1, 1, 2)), ((0\ 1\ 2\ 3\ 4\ 5\ 6), (5, 0, 6, 4, 1, 1)))$. Note que a instância possui quatro breakpoints intergênicos tipo um ($ib_1(\mathcal{I}) = 4$), sendo que $ib_f(\mathcal{I}) = 2$ e $ib_s(\mathcal{I}) = 2$. Os breakpoints intergênicos tipo um (π_1, π_2) e (π_4, π_5) são fortes, sendo que (π_1, π_2) é super forte e sobrecarregado enquanto (π_4, π_5) é subcarregado. Os breakpoints intergênicos tipo um (π_2, π_3) e (π_5, π_6) são suaves. Entre os pares de breakpoints intergênicos que estão conectados na instância, podemos citar o par de breakpoints intergênicos tipo um $((\pi_1, \pi_2), (\pi_2, \pi_3))$, que está conectado, e o par de breakpoints intergênicos tipo um $((\pi_1, \pi_2), (\pi_4, \pi_5))$, que está suavemente conectado. Além disso, obtemos as seguintes strips suaves da instância $\mathcal{I}$: (0 1 2), (5 4 3) e (6), sendo que (0 1 2) e (6) são strips suaves crescentes enquanto (5 4 3) é uma strip suave decrescente.

Exemplo 2.4.3.



O Exemplo 2.4.4 mostra uma instância intergênica rígida com sinais $\mathcal{I} = (((+0\ +1\ +2\ +5\ -4\ -3\ +6), (5, 5, 3, 1, 4, 2)), ((+0\ +1\ +2\ +3\ +4\ +5\ +6), (2, 5, 6, 4, 1, 2)))$. Note que a instância possui quatro breakpoints intergênicos tipo dois ($ib_2(\mathcal{I}) = 4$), sendo eles (π_0, π_1), (π_2, π_3), (π_3, π_4) e (π_5, π_6).

Exemplo 2.4.4.



2.5 Regiões Intergênicas

Nessa seção apresentamos alguns conceitos relacionados a regiões intergênicas em instâncias intergênicas flexíveis. Esses conceitos são importantes para o desenvolvimento de algoritmos e limitantes inferiores para os problemas investigados nos capítulos seguintes.

Definição 2.5.1. Dada uma instância intergênica flexível $\mathcal{I} = ((\pi, \breve{\pi}), (\iota, \breve{\iota}^{\min}, \breve{\iota}^{\max}))$, uma região intergênica $\breve{\pi}_i$ é chamada de *durável* se $\breve{\iota}_k^{\min} \leq \breve{\pi}_i \leq \breve{\iota}^{\max}$, tal que $k = \max(\pi_{i-1}, \pi_i)$. Caso contrário, a região intergênica $\breve{\pi}_i$ é chamada de *temporária*.

Uma região intergênica temporária deve necessariamente ser afetada por um evento de rearranjo, seja para unir genes que são consecutivos no genoma alvo ou para alterar a quantidade de nucleotídeos na região intergênica. Os conjuntos de regiões intergênicas duráveis e temporárias são definidos como $\mathcal{D}$ e $\mathcal{T}$, respectivamente. Dada uma instância intergênica flexível $\mathcal{I} = ((\pi, \breve{\pi}), (\iota, \breve{\iota}^{\min}, \breve{\iota}^{\max}))$, o número total de regiões intergênicas temporárias é denotado por $t(\mathcal{I})$. A variação no número de regiões intergênicas temporárias após aplicar uma sequência de eventos de rearranjo S em $(\pi, \breve{\pi})$ é denotada por $\Delta t(\mathcal{I}, S) = t(\mathcal{I}') - t(\mathcal{I})$, onde $\mathcal{I}' = ((\pi', \breve{\pi}'), (\iota, \breve{\iota}^{\min}, \breve{\iota}^{\max}))$ com $(\pi', \breve{\pi}') = (\pi, \breve{\pi}) \cdot S$.

Dada uma instância intergênica flexível $\mathcal{I} = ((\pi, \breve{\pi}), (\iota, \breve{\iota}^{\min}, \breve{\iota}^{\max}))$ e seja $\breve{\pi}_i$ uma região intergênica durável, denotamos por $gap_{\min}(\breve{\pi}_i) = \breve{\pi}_i - \breve{\iota}_k^{\min}$ e $gap_{\max}(\breve{\pi}_i) = \breve{\iota}_k^{\max} - \breve{\pi}_i$, tal que $k = \max(\pi_{i-1}, \pi_i)$. Os valores de $gap_{\min}$ e $gap_{\max}$ indicam, para cada região intergênica durável, a quantidade de nucleotídeos que podem ser removidos ou adicionados, respectivamente, ainda mantendo-a durável.

De agora em diante, as definições e conceitos que serão apresentados referem-se à instâncias intergênicas flexíveis balanceadas. Note que dada uma instância intergênica flexível $\mathcal{I} = ((\pi, \breve{\pi}), (\iota, \breve{\iota}^{\min}, \breve{\iota}^{\max}))$ e utilizando um modelo apenas com eventos de rearranjo conservativos, todos as regiões intergênicas temporárias precisam ser removidas para transformar $(\pi, \breve{\pi})$ em $(\iota, \breve{\pi}')$, tal que $\forall \breve{\pi}_i' \in \breve{\pi}', \breve{\iota}_i^{\min} \leq \breve{\pi}_i' \leq \breve{\iota}_i^{\max}$. Além disso, algumas regiões intergênicas duráveis podem ser afetadas com esse objetivo dependendo do total de nucleotídeos nas regiões intergênicas temporárias. Regiões intergênicas duráveis devem obrigatoriamente ser afetadas por algum evento de rearranjo se algum dos seguintes cenários ocorrer:

$$\texttt{(i)} \sum_{\breve{\pi}_i \in \mathcal{T}} \breve{\pi}_i < \sum_{\breve{\iota}_i^{\min} \in \breve{\iota}^{\min}} \breve{\iota}_i^{\min} - \sum_{\breve{\pi}_i \in \mathcal{D}} (\breve{\pi}_i - gap_{\min}(\breve{\pi}_i))$$

$$\texttt{(ii)} \sum_{\breve{\pi}_i \in \mathcal{T}} \breve{\pi}_i > \sum_{\breve{\iota}_i^{\max} \in \breve{\iota}^{\max}} \breve{\iota}_i^{\max} - \sum_{\breve{\pi}_i \in \mathcal{D}} (\breve{\pi}_i + gap_{\max}(\breve{\pi}_i))$$

No cenário (i), chamado de *fonte*, a quantidade de nucleotídeos nas regiões intergênicas temporárias não é suficiente para torná-las duráveis. Dessa forma, nucleotídeos das regiões intergênicas duráveis devem ser transferidos para as regiões intergênicas temporárias. No cenário (ii), chamado de *sorvedouro*, a quantidade de nucleotídeos nas regiões intergênicas temporárias excede o limite total permitido para essas regiões intergênicas. Dessa forma, nucleotídeos nucleotídeos das regiões intergênicas temporárias devem ser transferidos para as regiões intergênicas duráveis. Perceba que uma instância intergênica flexível pode não pertencer a nenhum desses cenários. Entretanto, não existe uma instância intergênica flexível que pertence aos dois cenários. Dada uma instância intergênica flexível que pertence ao cenário fonte ou sorvedouro, temos a seguinte definição.

Definição 2.5.2. Dada uma instância intergênica flexível $\mathcal{I} = ((\pi, \breve{\pi}), (\iota, \breve{\iota}^{\min}, \breve{\iota}^{\max}))$ que pertence ao cenário fonte ou sorvedouro, uma região intergênica durável $\breve{\pi}_i$ é chamada de *auxiliar* se deve receber nucleotídeos de regiões intergênicas temporárias ou transferir nucleotídeos para regiões intergênicas temporárias.

O número total de regiões intergênicas auxiliares depende do cenário da instância. No caso do cenário fonte, o conjunto de regiões intergênicas auxiliares $\mathcal{A}$ é tal que seu tamanho é mínimo e a seguinte restrição é satisfeita:

$$\sum_{\breve{\pi}_i \in \mathcal{A}} gap_{\min}(\breve{\pi}_i) \geq \sum_{\breve{\iota}_i^{\min} \in \breve{\iota}^{\min}} \breve{\iota}_i^{\min} - \sum_{\breve{\pi}_i \in \mathcal{D}} (\breve{\pi}_i - gap_{\min}(\breve{\pi}_i)) - \sum_{\breve{\pi}_i \in \mathcal{T}} \breve{\pi}_i$$

Note que o conjunto $\mathcal{A}$ com tamanho mínimo pode ser facilmente obtido ordenando as regiões intergênicas duráveis em ordem decrescente por $gap_{\min}(\breve{\pi}_i)$ e rotulando-os como auxiliares até que a restrição anterior seja satisfeita. No caso do cenário sorvedouro, o conjunto de regiões intergênicas auxiliares $\mathcal{A}$ é tal que seu tamanho é mínimo e a seguinte restrição é satisfeita:

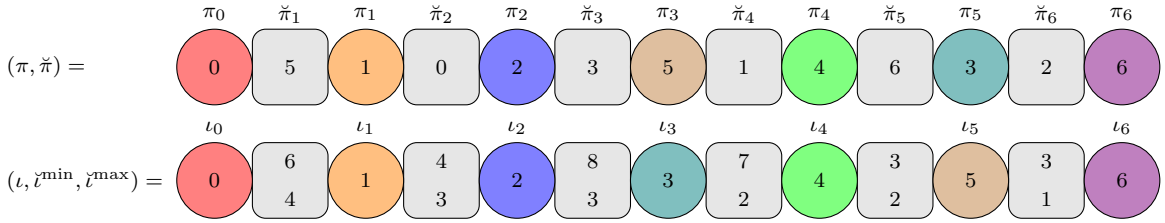
$$\sum_{\breve{\pi}_i \in \mathcal{A}} gap_{\max}(\breve{\pi}_i) \geq \sum_{\breve{\pi}_i \in \mathcal{T}} \breve{\pi}_i - \sum_{\breve{\iota}_i^{\max} \in \breve{\iota}^{\max}} \breve{\iota}_i^{\max} - \sum_{\breve{\pi}_i \in \mathcal{D}} (\breve{\pi}_i + gap_{\max}(\breve{\pi}_i))$$

Semelhante ao caso anterior, o conjunto $\mathcal{A}$ com tamanho mínimo pode ser facilmente obtido classificando as regiões intergênicas estáveis em ordem decrescente de $gap_{\max}(\breve{\pi}_i)$ e rotulando-as como auxiliares até que a restrição seja satisfeita.

Dada uma instância intergênica flexível $\mathcal{I} = ((\pi, \breve{\pi}), (\iota, \breve{\iota}^{\min}, \breve{\iota}^{\max}))$, o número total de regiões intergênicas auxiliares é denotado por $a(\mathcal{I})$. A variação no número de regiões intergênicas auxiliares após aplicar uma sequência de eventos de rearranjo S em $(\pi, \breve{\pi})$ é denotada por $\Delta a(\mathcal{I}, S) = a(\mathcal{I}') - a(\mathcal{I})$, onde $\mathcal{I}' = ((\pi', \breve{\pi}'), (\iota, \breve{\iota}^{\min}, \breve{\iota}^{\max}))$ com $(\pi', \breve{\pi}') = (\pi, \breve{\pi}) \cdot S$.

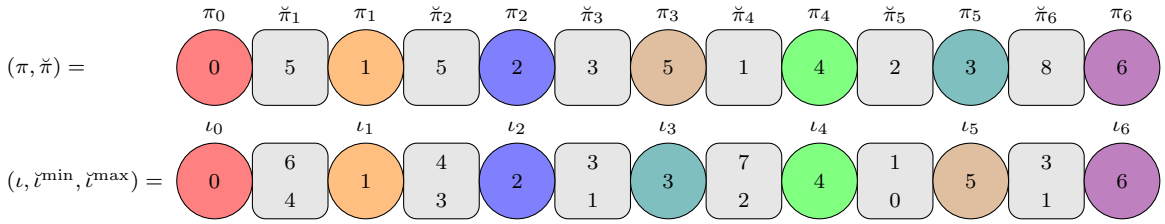
O Exemplo 2.5.1 mostra uma instância intergênica flexível sem sinais $\mathcal{I} = (((0\ 1\ 2\ 5\ 4\ 3\ 6), (5, 0, 3, 1, 6, 2)), ((0\ 1\ 2\ 3\ 4\ 5\ 6), (4, 3, 3, 2, 2, 1), (6, 4, 8, 7, 3, 3)))$ que pertence ao cenário fonte. Note que a instância $\mathcal{I}$ possui quatro regiões intergênicas temporárias ($t(\mathcal{I}) = 4$, com $\mathcal{T} = \{\breve{\pi}_2, \breve{\pi}_3, \breve{\pi}_4, \breve{\pi}_6\}$) e duas regiões intergênicas duráveis ($\mathcal{D} = \{\breve{\pi}_1, \breve{\pi}_5\}$). No exemplo, temos apenas uma região intergênica auxiliar ($a(\mathcal{I}) = 1$, com $\mathcal{A} = \{\breve{\pi}_5\}$). Note que $gap_{\min}(\breve{\pi}_1) = 1$ e $gap_{\min}(\breve{\pi}_5) = 4$.

Exemplo 2.5.1.



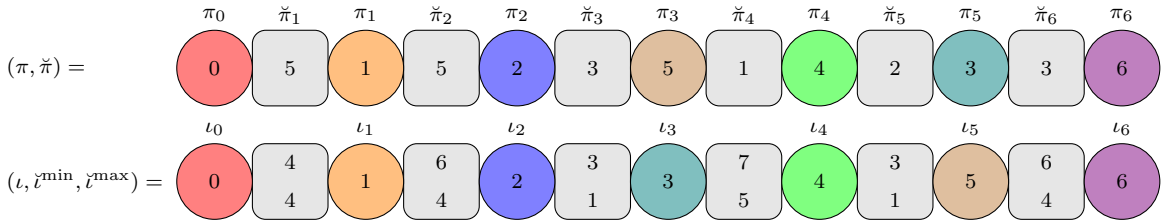
O Exemplo 2.5.2 mostra uma instância intergênica flexível sem sinais $\mathcal{I} = (((0\ 1\ 2\ 5\ 4\ 3\ 6), (5, 5, 3, 1, 2, 8)), ((0\ 1\ 2\ 3\ 4\ 5\ 6), (4, 3, 1, 2, 0, 1), (6, 4, 3, 7, 1, 3)))$ que pertence ao cenário sorvedouro. Note que a instância $\mathcal{I}$ possui quatro regiões intergênicas temporárias ($t(\mathcal{I}) = 4$, com $\mathcal{T} = \{\breve{\pi}_2, \breve{\pi}_3, \breve{\pi}_4, \breve{\pi}_6\}$) e duas regiões intergênicas duráveis ($\mathcal{D} = \{\breve{\pi}_1, \breve{\pi}_5\}$). No exemplo, temos duas região intergênica auxiliar ($a(\mathcal{I}) = 2$, com $\mathcal{A} = \{\breve{\pi}_1, \breve{\pi}_5\}$). Note que $gap_{\max}(\breve{\pi}_1) = 1$ e $gap_{\max}(\breve{\pi}_5) = 5$.

Exemplo 2.5.2.



O Exemplo 2.5.3 mostra uma instância intergênica flexível sem sinais $\mathcal{I} = (((0\ 1\ 2\ 5\ 4\ 3\ 6), (5, 5, 3, 1, 2, 8)), ((0\ 1\ 2\ 3\ 4\ 5\ 6), (4, 4, 1, 5, 1, 4), (4, 6, 3, 7, 3, 6)))$ que não pertence ao cenário fonte ou sorvedouro. Note que por esse motivo a instância não possui regiões intergênicas auxiliares, ou seja, $a(\mathcal{I}) = 0$ e $\mathcal{A} = \varnothing$. A instância $\mathcal{I}$ possui cinco regiões intergênicas temporárias ($t(\mathcal{I}) = 5$, com $\mathcal{T} = \{\breve{\pi}_1, \breve{\pi}_3, \breve{\pi}_4, \breve{\pi}_5, \breve{\pi}_6\}$) e uma região intergênica durável ($\mathcal{D} = \{\breve{\pi}_2\}$).

Exemplo 2.5.3.



2.6 Grafo de Ciclos

Grafos são estruturas amplamentes utilizadas em problemas de rearranjo de genomas para obtenção de limitantes inferiores e algoritmos. Nessa seção, apresentamos os grafos de ciclos clássico, ponderado e poderado flexível.

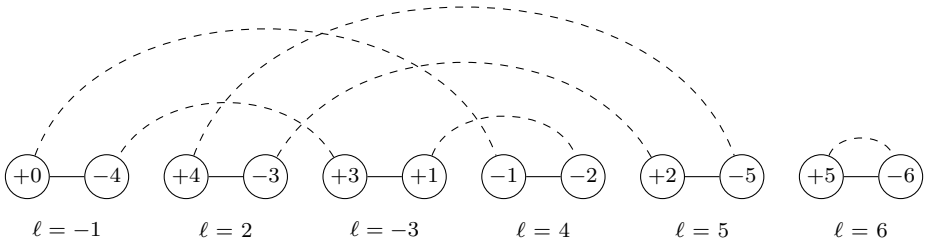
2.6.1 Grafo de Ciclos Clássico

O grafo de ciclos clássico, também chamado de grafo de breakpoints, tem seu uso bastante difundido em problemas de rearranjo de genomas que utilizam instâncias clássicas. Esse grafo evidência em uma mesma estrutura as adjacências presentes no genoma de origem e as adjacências desejadas no genoma alvo. A seguir definimos formalmente o grafo de ciclos clássico.

Dada uma instância clássica $\mathcal{I} = (\pi, \iota)$, definimos o gráfico de ciclos clássico por $G(\mathcal{I}) = (V, E, \ell)$, tal V, E e ℓ representam o conjunto de vértices, o conjunto de arestas e uma função de rotulação de arestas, respectivamente. O conjunto de vértices V é dado por $\{+\pi_0, -\pi_1, +\pi_1, -\pi_2, +\pi_2, \ldots, -\pi_n, +\pi_n, -\pi_{n+1}\}$. Note que para cada elemento π_i, com $0 < i < n + 1$, adicionamos em V os vértices $-\pi_i$ e $+\pi_i$. Por fim, adicionamos em V os vértices $+\pi_0$ e $-\pi_{n+1}$. O conjunto de arestas $E = E_p \cup E_c$ é divido nos conjuntos de arestas pretas (E_p) e arestas cinzas (E_c), onde $E_p = \{(-\pi_i, +\pi_{i-1}) \,|\, 1 \leq i \leq n + 1\}$ e $E_c = \{(+(i-1), -i) \,|\, 1 \leq i \leq n + 1\}$. Perceba que as arestas pretas representam os elementos que são adjacentes na permutação π, enquanto as arestas cinzas representam os elementos que são adjacentes em ι.

Existem diferentes formas de desenhar o grafo de ciclos clássico. Entretanto, utilizaremos a forma que chamamos de *padrão*. Para essa forma de desenhar o grafo os vértices são posicionados horizontalmente da esquerda para direita e seguindo a ordem $+\pi_0, -\pi_1, +\pi_1, -\pi_2, +\pi_2, \ldots, -\pi_n, +\pi_n, -\pi_{n+1}$. As arestas pretas são desenhadas formando uma linha horizontal contínua, enquanto as arestas cinzas formam arcos com linhas tracejadas sobre os vértices. O Exemplo 2.6.1 mostra o grafo de ciclos clássico construído a partir da instância clássica $\mathcal{I} = ((+0\ +4\ +3\ -1\ +2\ +5\ +6), (+0\ +1\ +2\ +3\ +4\ +5\ +6))$.

Exemplo 2.6.1.



Pelo Exemplo 2.6.1, podemos perceber que o grafo de ciclos clássico possui $2n + 2$ vértices e $2n + 2$ arestas ($n + 1$ pretas e $n + 1$ cinzas), sendo que em cada vértice duas arestas são incidentes, uma preta e uma cinza. Por esse motivo, há uma decomposição única de $G(\mathcal{I})$ em ciclos com arestas de cores alternadas.

A função de rotulação $\ell : E_p \to \{-(n+1), -n, \ldots, -2, -1, 1, 2, \ldots, n, (n+1)\}$ atribui um rótulo para cada aresta preta no grafo em função da direção em que a aresta é percorrida. Dada uma aresta preta $e_p = (-\pi_i, +\pi_{i-1}) \in E_p$, a função ℓ atribui o rótulo i em e_p caso ela seja percorrida de $-\pi_i$ até $+\pi_{i-1}$. Caso contrário, e_p é rotulada com $-i$. Por padrão, cada ciclo de $G(\mathcal{I})$ é representado pela sequência de rótulos de suas arestas pretas na ordem em que elas são percorridas, sendo que a primeira aresta preta de um ciclo é aquela que encontra-se mais a direita no grafo e é percorrida da direita para esquerda, ou seja, de $-\pi_i$ até $+\pi_{i-1}$. Essa representação utilizada para os ciclos faz

com que eles sejam representados unicamente. No Exemplo 2.6.1, $G(\mathcal{I})$ possui três ciclos: $C_1 = (4, -1, -3)$, $C_2 = (5, 2)$ e $C_3 = (6)$.

O tamanho de um ciclo $C \in G(\mathcal{I})$ é dado pela quantidade de arestas pretas do ciclo. Um ciclo de tamanho um é chamado de *trivial*. Um Ciclo com tamanho menor que três é chamado de *curto*. Caso contrário, é chamado de *longo*.

Definição 2.6.1. Duas arestas pretas de um ciclo $C \in G(\mathcal{I})$ são chamadas de *divergentes* se elas são percorridas em direções opostas. Caso contrário, são chamadas de *convergentes*.

Definição 2.6.2. Um ciclo $C \in G(\mathcal{I})$ é chamado de *divergente* se pelo menos uma par de arestas pretas de C são divergentes. Caso contrário, C é chamado de *convergente*.

Podemos ainda classificar ciclos convergentes como *orientados* ou *não orientados*.

Definição 2.6.3. Um ciclo convergente $C = (c_1, c_2, \ldots, c_k) \in G(\mathcal{I})$ é classificado como *não orientado* se $c_i > c_{i+1}$, para todo i com $1 \leq i < k$. Caso contrário, C é classificado como *orientado*.

Dois ciclos $C = (c_1, c_2, \ldots, c_k)$ e $D = (d_1, d_2, \ldots, d_k)$, ambos pertencentes ao grafo $G(\mathcal{I})$, são entrelaçados se $|c_1| > |d_1| > |c_2| > |d_2| > \ldots > |c_k| > |d_k|$ ou $|d_1| > |c_1| > |d_2| > |c_2| > \ldots > |d_k| > |c_k|$. Seja g_1 uma aresta cinza adjacente às arestas pretas com rótulos x_1 e y_1, tal que $|x_1| < |y_1|$ e que g_2 seja uma aresta cinza adjacente às arestas pretas com rótulos x_2 e y_2, tal que $|x_2| < |y_2|$. Dizemos que duas arestas cinzas g_1 e g_2 cruzam-se caso $|x_1| < |x_2| \leq |y_1| < |y_2|$. Dois ciclos C e D cruzam-se caso uma aresta cinza de C cruza-se com uma aresta cinza de D. Um *open gate* é uma aresta cinza de um ciclo não trivial $C \in G(\mathcal{I})$ que não se cruza com nenhuma outra aresta cinza de C. Um open gate g_1 de C é fechado se outra aresta cinza (que não seja de C) cruza com g_1.

Observação 2.6.1. Todos os open gates de ciclos não tiviais em $G(\mathcal{I})$ são fechados [31].

No Exemplo 2.6.1, os ciclos $C_1 = (4, -1, -3)$, $C_2 = (5, 2)$ e $C_3 = (6)$ são, respectivamente, longo divergente, curto convergente orientado e trivial. Note que o ciclo C_1 possui o open gate $(+3, -4)$, enquanto o ciclo C_2 possui os seguintes open gates: $(+2, -3)$ e $(+4, -5)$.

Dada uma instância clássica $\mathcal{I} = (\pi, \iota)$, denotamos por $c(G(\mathcal{I}))$ o número de ciclos em $G(\mathcal{I})$. Dada uma sequência de eventos de rearranjo S, denotamos por $\Delta c(G(\mathcal{I}), S) = c(G(\mathcal{I}')) - c(G(\mathcal{I}))$, tal que $\mathcal{I}' = (\pi \cdot S, \iota)$, a variação no número de ciclos após aplicar a sequência S no genoma de origem π de $\mathcal{I}$.

Observação 2.6.2. A única instância clássica $\mathcal{I}$ com $c(G(\mathcal{I})) = n + 1$ é $\mathcal{I} = (\iota, \iota)$.

2.6.2 Grafo de Ciclos Ponderado Rígido

O grafo de ciclos ponderado rígido é uma extensão do grafo de ciclos clássico. O grafo de ciclos ponderado rígido incorpora na sua estrutura, através de pesos nas arestas, informações referentes ao tamanho das regiões intergênicas do genoma de origem e alvo. A seguir definimos formalmente o grafo de ciclos poderado.

Dada uma instância intergênica rígida $\mathcal{I} = ((\pi, \breve{\pi}), (\iota, \breve{\iota}))$, definimos o gráfico de ciclos ponderado rígido por $G(\mathcal{I}) = (V, E = E_p \cup E_c, \ell, w_p, w_c)$, tal que V, E e ℓ representam,

respectivamente, o conjunto de vértices, o conjunto de arestas e uma função de rotulação de arestas, w_p e w_c são funções de peso. Pelo fato do grafo de ciclos ponderado rígido tratar-se de uma extensão do grafo de ciclos clássico, V, E e ℓ comportam-se exatamente como descrito no grafo de ciclos clássico. Além disso, todos os conceitos, definições e representações que foram apresentados no contexto de grafo de ciclos clássico também são válidas e utilizadas no grafo de ciclos ponderado rígido.

A função de peso $w_p : E_p \to \mathbb{N}_0$ associa os tamanhos das regiões intergênicas no genoma de origem com pesos nas arestas pretas do grafo. A função de peso $w_c : E_c \to \mathbb{N}_0$ funciona de uma maneira similar, mas associando os tamanhos das regiões intergênicas no genoma alvo com pesos nas arestas cinzas do grafo. Para cada aresta preta $e_i = (-\pi_i, +\pi_{i-1}) \in E_p$, temos que $w_p(e_i) = \breve{\pi}_i$. Para cada aresta cinza $e'_i = (+(i-1), -i) \in E_c$, temos que $w_c(e'_i) = \breve{\iota}_i$. Dado um ciclo $C \in G(\mathcal{I})$, denotamos por $E_p(C)$ e $E_c(C)$, respectivamente, os conjuntos de arestas pretas e cinzas que pertencem ao ciclo C.

Definição 2.6.4. Um ciclo $C \in G(\mathcal{I})$ é chamado de *balanceado* caso $\sum_{e'_i \in E_c(C)}[w_c(e'_i)] - \sum_{e_i \in E_p(C)}[w_p(e_i)] = 0$. Caso contrário, o ciclo C é chamado de *desbalanceado*.

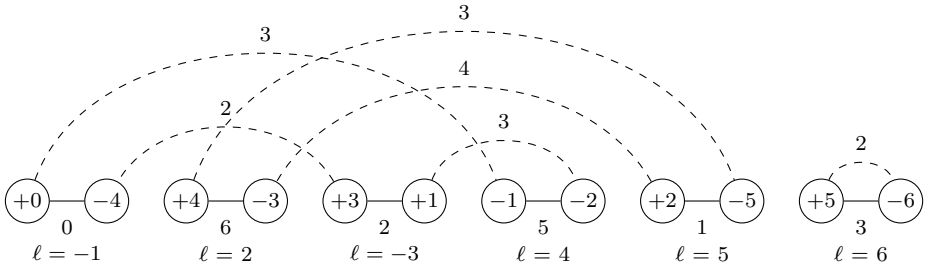
Em outras palavras, um ciclo balanceado indica que a soma dos pesos em suas aretas pretas é a mesma que a soma dos pesos em suas arestas cinzas.

Definição 2.6.5. Um ciclo desbalanceado $C \in G(\mathcal{I})$ é chamado de *negativo* quando $\sum_{e'_i \in E_c(C)}[w_c(e'_i)] - \sum_{e_i \in E_p(C)}[w_p(e_i)] < 0$. Caso contrário, o ciclo C é chamado de *positivo*.

Note que um ciclo negativo possui a soma dos pesos em suas arestas pretas maior que a soma dos pesos em suas arestas cinzas, já é um ciclo positivo acontece justamento o oposto. Dada uma instância intergênica rígida $\mathcal{I} = ((\pi, \breve{\pi}), (\iota, \breve{\iota}))$, denotamos por $c(G(\mathcal{I}))$ e $c_b(G(\mathcal{I}))$ o número de ciclos e ciclos balanceados em $G(\mathcal{I})$, respectivamente. Dada uma sequência de eventos de rearranjo S, denotamos por $\Delta c(G(\mathcal{I}), S) = c(G(\mathcal{I}')) - c(G(\mathcal{I}))$ e $\Delta c_b(G(\mathcal{I}), S) = c_b(G(\mathcal{I}')) - c_b(G(\mathcal{I}))$, tal que $\mathcal{I}' = ((\pi, \breve{\pi}) \cdot S, (\iota, \breve{\iota}))$, a variação no número de ciclos e ciclos balanceados, respectivamente, após aplicar a sequência S no genoma de origem $(\pi, \breve{\pi})$ de $\mathcal{I}$.

O Exemplo 2.6.2 mostra o grafo de ciclos ponderado rígido construído a partir da instância intergênica rígida $\mathcal{I} = (((+0\ +4\ +3\ -1\ +2\ +5\ +6), (0, 6, 2, 5, 1, 3)), ((+0\ +1\ +2\ +3\ +4\ +5\ +6), (3, 3, 4, 2, 3, 2)))$.

Exemplo 2.6.2.



No Exemplo 2.6.2, os ciclos $C_1 = (4, -1, -3)$, $C_2 = (5, 2)$ e $C_3 = (6)$ são, respectivamente, longo positivo, curto balanceado e trivial negativo.

Observação 2.6.3. A única instância intergênica rígida $\mathcal{I}$ com $c(G(\mathcal{I})) = n+1$ e $c_b(G(\mathcal{I})) = n+1$ é $\mathcal{I} = ((\iota, \breve{\iota}), (\iota, \breve{\iota}))$.

2.6.3 Grafo de Ciclos Ponderado Flexível

O grafo de ciclos ponderado flexível é uma extensão do grafo de ciclos clássico. O grafo de ciclos ponderado rígido incorpora na sua estrutura, através de pesos nas arestas, informações referentes ao tamanho das regiões intergênicas do genoma de origem e os tamanhos mínimos e máximos permitidos para cada região intergênica no genoma alvo. A seguir definimos formalmente o grafo de ciclos poderado flexível.

Dada uma instância intergênica flexível $\mathcal{I} = ((\pi, \breve{\pi}), (\iota, \breve{\iota}^{\min}, \breve{\iota}^{\max}))$, definimos o gráfico de ciclos ponderado flexível por $G(\mathcal{I}) = (V, E = E_p \cup E_c, \ell, w_p, w_c^{\min}, w_c^{\max})$, tal que V, E e ℓ representam, respectivamente, o conjunto de vértices, o conjunto de arestas e uma função de rotulação de arestas, w_p, $w_c^{\min}$ e $w_c^{\max}$ são funções de peso. Pelo fato do grafo de ciclos ponderado flexível também tratar-se de uma extensão do grafo de ciclos clássico, V, E e ℓ comportam-se exatamente como descrito no grafo de ciclos clássico. Além disso, todos os conceitos, definições e representações que foram apresentados no contexto de grafo de ciclos clássico também são válidas e utilizadas no grafo de ciclos ponderado flexível.

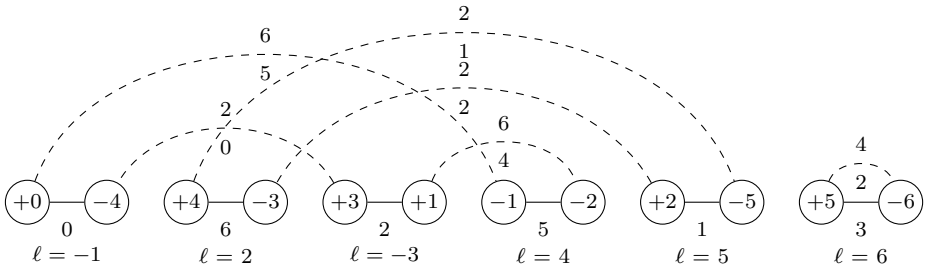
A função de peso $w_p : E_p \to \mathbb{N}_0$ associa os tamanhos das regiões intergênicas no genoma de origem com pesos nas arestas pretas do grafo. As funções de peso $w_c^{\min} : E_c \to \mathbb{N}_0$ e $w_c^{\max} : E_c \to \mathbb{N}_0$ associam, respectivamente, os tamanhos mínimos e máximos permitidos para as regiões intergênicas no genoma alvo com pesos nas arestas cinzas do grafo. Para cada aresta preta $e_i = (-\pi_i, +\pi_{i-1}) \in E_p$, temos que $w_p(e_i) = \breve{\pi}_i$. Para cada aresta cinza $e'_i = (+(i-1), -i) \in E_c$, temos que $w_c^{\min}(e'_i) = \breve{\iota}_i^{\min}$ e $w_c^{\max}(e'_i) = \breve{\iota}_i^{\max}$. Dado um ciclo $C \in G(\mathcal{I})$, denotamos por $E_p(C)$ e $E_c(C)$, respectivamente, os conjuntos de arestas pretas e cinzas que pertencem ao ciclo C. Dado um ciclo $C \in G(\mathcal{I})$, denotamos por $W_p(C) = \sum_{e_i \in E_p(C)} w_p(e_i)$, $W_c^{\min}(C) = \sum_{e'_i \in E_c(C)} w_c^{\min}(e'_i)$ e $W_c^{\max}(C) = \sum_{e'_i \in E_c(C)} w_c^{\max}(e'_i)$ o *peso total*, *peso mínimo total* e *peso máximo total* de C, respectivamente. Note que o peso total de um ciclo é a soma dos pesos em suas arestas pretas, já os pesos mínimo total e máximo total são a soma dos pesos mínimos e máximos em suas arestas cinzas, respectivamente.

Definição 2.6.6. Um ciclo $C \in G(\mathcal{I})$ é chamado de *verdadeiro* caso $W_g^{\min}(C) \leq W_b(C) \leq W_g^{\max}(C)$. Caso contrário, o ciclo C é chamado de *falso*.

Em outras palavras, um ciclo verdadeiro indica que o peso total é suficiente para satisfazer as restrições relativas aos pesos mínimos e máximos em cada uma de suas arestas cinzas. Definimos os conjuntos de ciclos verdadeiros e falsos em $G(\mathcal{I})$ como $\mathcal{V}$ e $\mathcal{F}$, respectivamente. Dado um ciclo $C \in G(\mathcal{I})$, denotamos por $gap_{\min}(C) = W_b(C) - W_g^{\min}(C)$ e $gap_{\max}(C) = W_g^{\max}(C) - W_b(C)$ como valores que se subtraídos e adicionados do peso total de C resultam, respectivamente, nos pesos mínimo total e máximo total de C.

O Exemplo 2.6.3 mostra o grafo de ciclos ponderado flexível construído a partir da instância intergênica flexível $\mathcal{I} = (((+0\ +4\ +3\ -1\ +2\ +5\ +6), (0, 6, 2, 5, 1, 3)), ((+0\ +1\ +2\ +3\ +4\ +5\ +6), (5, 4, 2, 0, 1, 2), (6, 6, 2, 2, 2, 4)))$.

Exemplo 2.6.3.



No Exemplo 2.6.3, os ciclos $C_1 = (4, -1, -3)$, $C_2 = (5, 2)$ e $C_3 = (6)$ são, respectivamente, longo falso, curto falso e trivial verdadeiro.

De agora em diante, as definições e conceitos que serão apresentados referem-se à instâncias intergênicas flexíveis balanceadas. Note que dada uma instância intergênica flexível $\mathcal{I} = ((\pi, \breve{\pi}), (\iota, \breve{\iota}^{\min}, \breve{\iota}^{\max}))$ e utilizando um modelo apenas com eventos de rearranjo conservativos, todos os ciclos falsos devem ser removidos e $G(\mathcal{I})$ deve possuir $n+1$ ciclos verdadeiros para transformar $(\pi, \breve{\pi})$ em $(\iota, \breve{\pi}')$, tal que $\forall \breve{\pi}'_i \in \breve{\pi}', \breve{\iota}_i^{\min} \leq \breve{\pi}'_i \leq \breve{\iota}_i^{\max}$. Dependendo da distribuição dos nucleotídeos e das restrições de tamanho mínimo e máximo nas arestas cinzas, alguns dos ciclos verdadeiros também devem ser afetados para realizar essa tarefa. Na verdade, existem dois casos em que isso ocorre:

$$\texttt{(i)} \sum_{C \in \mathcal{F}} W_p(C) < \sum_{C \in \mathcal{F}} W_c^{\min}(C)$$

$$\texttt{(ii)} \sum_{C \in \mathcal{F}} W_p(C) > \sum_{C \in \mathcal{F}} W_c^{\max}(C)$$

No caso `(i)`, a soma do peso total de todos os ciclos falsos é menor que a soma do peso mínimo total dos mesmos ciclos, ou seja, não é possível atender todas as retrições de peso mínimo e máximo nas arestas cinzas dos ciclos falsos sem que alguns ciclos verdadeiros transfiram uma determinada quantidade de peso de suas arestas pretas para os ciclos falsos. No caso `(ii)`, a soma do peso total de todos os ciclos falsos é maior que a soma do peso máximo total dos mesmos ciclos. Nesse caso, alguns ciclos falsos precisam transferir uma determinada quantidade de peso de suas arestas pretas para alguns os ciclos verdadeiros.

Definição 2.6.7. Um ciclo verdadeiro $C \in G(\mathcal{I})$ é chamado de *ruim* se ele deve receber ou transferir peso de suas arestas pretas para outro ciclo, e é chamado de *bom* caso contrário.

Definimos os conjuntos de ciclos ruins e bons em $G(\mathcal{I})$ como $\mathcal{R}$ e $\mathcal{B}$, respectivamente. Observe que os casos `(i)` e `(ii)` podem não acorrer se $\sum_{C \in \mathcal{F}} W_c^{\min}(C) \leq \sum_{C \in \mathcal{F}} W_p(C) \leq \sum_{C \in \mathcal{F}} W_c^{\max}(C)$, em ambos os casos temos que $\mathcal{R} = \varnothing$ e $\mathcal{B} = \mathcal{V}$ (Exemplo 2.6.3). Note que se o caso `(i)` ou `(ii)` ocorrer é necessário determinar os conjuntos $\mathcal{R}$ e $\mathcal{B}$, e isso depende do caso em que a instância se encaixa.

Se for o caso `(i)`, então um conjunto $\mathcal{R}$ de tamanho mínimo pode ser composto do menor número de ciclos em que a seguinte restrição seja cumprida:

$$\sum_{C \in \mathcal{R}} gap_{\min}(C) + \sum_{C \in \mathcal{F}} gap_{\min}(C) \geq 0$$

Se for o caso (ii), então um conjunto $\mathcal{R}$ de tamanho mínimo pode ser composto do menor número de ciclos em que a seguinte restrição seja cumprida:

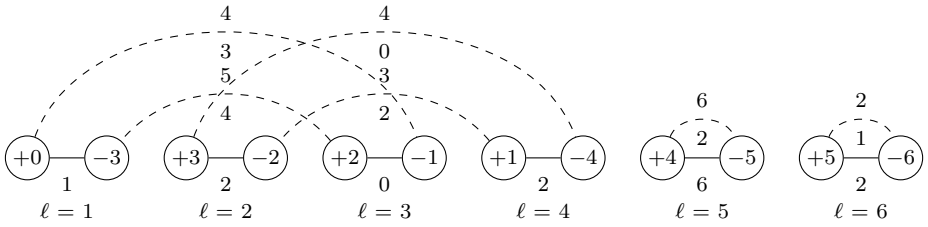
$$\sum_{C\in\mathcal{R}} gap_{\max}(C) + \sum_{C\in\mathcal{F}} gap_{\max}(C) \geq 0$$

Observe que em ambos os casos, o conjunto $\mathcal{R}$ pode ser facilmente obtido após a ordenação, de forma decrescente, dos ciclos verdadeiros pelos valores $gap_{\min}$ e $gap_{\max}$ considerando os casos (i) e (ii), respectivamente. Então, seguindo a ordem decrescente, os ciclos são rotulados como ruins até satisfazerem a restrição. O conjunto de ciclos bons $\mathcal{B}$ é obtido por $\mathcal{V} - R$.

Dada uma instância intergênica flexível $\mathcal{I} = ((\pi, \breve{\pi}), (\iota, \breve{\iota}^{\min}, \breve{\iota}^{\max}))$, denotamos por $c_v(G(\mathcal{I}))$ e $c_b(G(\mathcal{I}))$ o número de ciclos verdadeiros e bons em $G(\mathcal{I})$, respectivamente. Dada uma sequência de eventos de rearranjo S, denotamos por $\Delta c_v(G(\mathcal{I}), S) = c_v(G(\mathcal{I}'))$ $- c_v(G(\mathcal{I}))$ e $\Delta c_b(G(\mathcal{I}), S) = c_b(G(\mathcal{I}')) - c_b(G(\mathcal{I}))$, tal que $\mathcal{I}' = ((\pi, \breve{\pi})\cdot S, (\iota, \breve{\iota}))$, a variação no número de ciclos verdadeiros e bons, respectivamente, após aplicar a sequência S no genoma de origem $(\pi, \breve{\pi})$ de $\mathcal{I}$.

O Exemplo 2.6.4 mostra o grafo de ciclos ponderado flexível construído a partir da instância intergênica flexível $\mathcal{I} = (((+0\ +3\ +2\ +1\ +4\ +5\ +6), (1, 2, 0, 2, 6, 2)), ((+0\ +1\ +2$ $+3\ +4\ +5\ +6), (3, 2, 4, 0, 2, 1), (4, 3, 5, 4, 6, 2)))$.

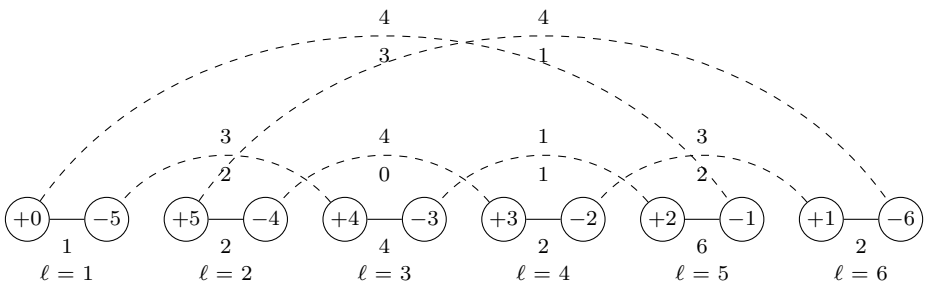
Exemplo 2.6.4.



No Exemplo 2.6.4, $G(\mathcal{I})$ possui quatro ciclos, sendo eles: $C_1 = (3, 1)$, $C_2 = (4, 2)$, $C_3 = (5)$ e $C_4 = (6)$. Além disso, temos os conjuntos $\mathcal{F} = \{C_1\}$ e $\mathcal{V} = \{C_2, C_3, C_4\}$. Observe que a instância intergênica flexível $\mathcal{I}$ do Exemplo 2.6.4 pertence ao caso (i): $1 = W_b(C_1) < W_g^{\min}(C_1) = 7$, onde apenas o ciclo falso C_1 precisa aumentar o seu peso total para ser transformado em um ciclo verdadeiro. Note que $gap_{\min}(C_2) = 2$, $gap_{\min}(C_3) = 4$ e $gap_{\min}(C_4) = 1$. Portanto, temos que $\mathcal{R} = \{C_2, C_3\}$ e $\mathcal{B} = \{C_4\}$.

O Exemplo 2.6.5 mostra o grafo de ciclos ponderado flexível construído a partir da instância intergênica flexível $\mathcal{I} = (((+0\ +5\ +4\ +3\ +2\ +1\ +6), (1, 2, 4, 2, 6, 2)), ((+0\ +1\ +2$ $+3\ +4\ +5\ +6), (3, 2, 1, 0, 2, 1), (4, 3, 1, 4, 3, 4)))$.

Exemplo 2.6.5.



No Exemplo 2.6.5, $G(\mathcal{I})$ possui dois ciclos, sendo eles: $C_1 = (5, 3, 1)$ e $C_2 = (6, 4, 2)$. Além disso, temos os conjuntos $\mathcal{F} = \{C_1\}$ e $\mathcal{V} = \{C_2\}$. Observe que a instância intergênica flexível $\mathcal{I}$ do Exemplo 2.6.5 pertence ao caso `(ii)`: $11 = W_b(C_1) > W_g^{\max}(C_1) = 8$, onde apenas o ciclo falso C_1 precisa reduzir o seu peso total para ser transformado em um ciclo verdadeiro. Note que $gap_{\max}(C_2) = 5$. Portanto, temos que $\mathcal{R} = \{C_2\}$ e $\mathcal{B} = \varnothing$.

Observação 2.6.4. Uma instância intergênica flexível $\mathcal{I} = ((\pi, \breve{\pi}), (\iota, \breve{\iota}^{\min}, \breve{\iota}^{\max}))$ tal que $c_v(G(\mathcal{I}) = c_b(G(\mathcal{I}) = n + 1$ implica que $\pi = \iota$ e $\breve{\iota}_i^{\min} \leq \breve{\pi}_i \leq \breve{\iota}_i^{\max}$ para todo $\breve{\pi}_i \in \breve{\pi}$.

Capítulo 3

Modelos com Porporção entre Operações

Os problemas de distância entre genomas podem utilizar uma abordagem *não ponderada*, ou seja, cada evento de rearranjo utilizado para transformar o genoma de origem no genoma alvo contribui em uma unidade para a distância. Essa abordagem tem como característica que cada tipo de evento de rearranjo, pertencente ao modelo de rearranjo adotado, possui a mesma probabilidade de ocorrer em um cenário evolutivo. Outra abordagem que surgiu para possibilitar uma contribuição diferente para cada evento de rearranjo é chamada de *ponderada*. Nesse abordagem, cada tipo de evento de rearranjo possui um peso associado que é contabilizado na distância evolutiva entre os genomas. A abordagem ponderada geralmente é utilizada para mapear um cenário em que queremos que determinado eventos de rearrajo tenham uma possibilidade maior de ocorrer do que outros. Para isso, basta atribuir um peso menor nos eventos de rearranjo que esperados que ocorram mais. Esses pesos podem ser atribuídos com base em observações empíricas de determinados organismos ou através de análises realizadas especificamente para esse objetivo [3, 23].

Os eventos de rearranjo de reversão e transposição são dois dos eventos mais estudados na literatura [8, 22, 34]. Considerando uma representação clássica e uma abordagem não ponderada, temos o problema de Ordenção de Permutações por Reversões e Transposições (**SbRT**), sendo que o problema possui a variação com e sem sinais. Ambas as variações pertencem à classe NP-difícil de problemas [29], para a variação com sinais do problema existe um algoritmo de aproximação com fator 2 [36]. Para a variação sem sinais, existe um algoritmo de aproximação com fator $2k$ [33], onde k [15] é o fator de aproximação do algoritmo utilizado para a decomposição de ciclos do Grafo de Ciclos [14].

Considerando um abordagem poderada, temos o problema de Ordenção de Permutações por Reversões e Transposições Ponderadas ($\mathbf{Sb_WRT}$) na variação com e sem sinais. In 2002, Eriksen [24] apresentou um algoritmo com factor de aproximação 7/6 para a variação com sinais do problema utilizando os pesos 1 e 2 para os eventos de reversão e transposição, respectivamente. Oliveira *et al.*[31] desenvolveram um algoritmo de aproximação com fator 1.5 para a variação com sinais do problema $\mathbf{Sb_WRT}$ utilizando os pesos 2 e 3 para os eventos de reversão e transposição, respectivamente. Além disso, os autores mostraram que as variações com e sem sinais do problema $\mathbf{Sb_WRT}$ pertencem à classe

NP-difícil quando a razão entre os pesos dos eventos de transposição e reversão é maior ou igual a 1.5.

Em 2007, Bader e Ohlebusch [4] apresentaram o problema de Ordenção de Permutações por Reversões, Transposições e Transposições Inversa Ponderadas ($\mathbf{Sb_WRTIT}$). A transposição inversa é um evento similar ao evento de transposição, mas com um dos segmentos adjacentes afetados sendo invertido. Para a variação com sinais do problema os autores apresentaram um algoritmo de aproximação com fator 1.5 utilizando o peso 1 para o evento de reversão e o mesmo peso, no intervalo [1..2], para os eventos de transposição e transposição inversa. Em 2020, Alexandrino *et al.*[1] mostraram que as variações com e sem sinais do problema $\mathbf{Sb_WRTIT}$ pertencem à classe NP-difícil quando os eventos de transposição e transposição inversa possuem o mesmo peso e a razão entre os pesos dos eventos de transposição e reversão é maior ou igual a 1.5.

A abordagem ponderada possui vantagens em comparação com a abordagem não ponderada quando queremos mapear um cenário evolutivo dando mais prioridade para determinados tipos de eventos de rearranjo. Entretanto, ela não garantem que os rearranjos de menor custo, que são supostamente os mais frequentes em um cenário evolutivo, serão os mais utilizados pelos algoritmos. Para contornar esse ponto, propomos e investigamos o problema de Ordenção de Permutações por Reversões e Transposições com Restrição de Proporção ($\mathbf{Sb_PRT}$) em instâncias clássicas com e sem sinais. Neste cenário, buscamos uma sequência de reversões e transposições S capaz de transformar o genoma de origem no genoma alvo com uma restrição adicional na qual a relação entre o número de reversões e o tamanho da sequência S deve ser maior ou igual a um determinado parâmetro $k \in [0..1]$.

Observe que tanto as abordagens ponderada e proporcional tentam incorporar no modelo a frequência na qual os eventos de rearranjo afetam o genoma de um determinado organismo. É importante notar que, do ponto de vista biológico, a frequência e o conjunto de eventos de rearranjo podem variar dependendo do organismo considerado. De um ponto de vista teórico, as abordagens possuem objetivos diferentes, apesar de compartilharem características comuns. Uma característica que difere da abordagem de proporção é que uma vez conhecida a frequência na qual os eventos afetam o genoma, a proporção pode ser facilmente derivada dessa informação, enquanto que na abordagem ponderada o peso associado a cada tipo de evento precisa ser ajustado e validado através de testes experimentais.

O Exemplo 3.0.1 mostra uma solução ótimo S para a instância clássica com sinais $((+0\ -1\ +4\ -8\ +3\ +5\ +2\ -7\ -6\ +9), (+0\ +1\ +2\ +3\ +4\ +5\ +6\ +7\ +8\ +9))$ considerando os problemas $\mathbf{SbRT}$ e $\mathbf{Sb_WRT}$ (utilizando os pesos 2 e 3 para os eventos de reversão e transposição, respectivamente). Note que metade dos eventos de rearranjo de S são reversões e a outra metade transposições, mesmo utilizando um custo maior para o evento de transposição.

Exemplo 3.0.1.

$$
\begin{array}{lcll}
\pi & = & & (+0\ -1\ +4\ -8\ +3\ +5\ +2\ -7\ -6\ +9) \\
\pi^1 & = & \pi \cdot \rho^{(1,5)} & = (+0\ \underline{-5\ -3\ +8\ -4\ +1}\ +2\ -7\ -6\ +9) \\
\pi^2 & = & \pi^1 \cdot \tau^{(2,4,9)} & = (+0\ -5\ \underline{-4\ +1\ +2\ -7\ -6}\ \underline{-3\ +8}\ +9) \\
\pi^3 & = & \pi^2 \cdot \tau^{(1,3,7)} & = (+0\ \underline{+1\ +2\ -7\ -6}\ \underline{-5\ -4}\ -3\ +8\ +9) \\
\pi^4 & = & \pi^3 \cdot \rho^{(3,7)} & = (+0\ +1\ +2\ \underline{+3\ +4\ +5\ +6\ +7}\ +8\ +9) \\
S & = & (\rho^{(1,5)}, \tau^{(2,4,9)}, \tau^{(1,3,7)}, \rho^{(3,7)}) &
\end{array}
$$

O Exemplo 3.0.2 mostra uma solução ótima S' para a mesma instância clássica com sinais apresentada no Exemplo 3.0.1 considerando o problem $\mathbf{Sb_PRT}$ e adotando um valor de $k = 0.6$, ou seja, pelo menos 60% dos eventos de rearranjo em S' devem ser reversões. Quando comparamos com o Exemplo 3.0.1, podemos perceber que S' possui apenas um evento a mais que S, mas a proporção mínima de reversões em relação ao tamanho da sequência S' é garantida.

Exemplo 3.0.2.

$$
\begin{array}{lcll}
\pi & = & & (+0\ -1\ +4\ -8\ +3\ +5\ +2\ -7\ -6\ +9) \\
\pi^1 & = & \pi \cdot \rho^{(2,8)} & = (+0\ -1\ \underline{+6\ +7\ -2\ -5\ -3\ +8\ -4}\ +9) \\
\pi^2 & = & \pi^1 \cdot \rho^{(2,4)} & = (+0\ -1\ \underline{+2\ -7\ -6}\ -5\ -3\ +8\ -4\ +9) \\
\pi^3 & = & \pi^2 \cdot \tau^{(6,8,9)} & = (+0\ -1\ +2\ -7\ -6\ -5\ \underline{-4}\ \underline{-3\ +8}\ +9) \\
\pi^4 & = & \pi^3 \cdot \rho^{(1,1)} & = (+0\ \underline{+1}\ +2\ -7\ -6\ -5\ -4\ -3\ +8\ +9) \\
\pi^5 & = & \pi^4 \cdot \rho^{(3,7)} & = (+0\ +1\ +2\ \underline{+3\ +4\ +5\ +6\ +7}\ +8\ +9) \\
S' & = & (\rho^{(2,8)}, \rho^{(2,4)}, \tau^{(6,8,9)}, \rho^{(1,1)}, \rho^{(3,7)}) &
\end{array}
$$

Dada uma sequência de eventos de rearranjo S, denotamos por $|S|$ o tamanho da sequência S, ou seja, a quantidade de eventos em S. Além disso, denotamos por $|S_\rho|$ pela quantidade de eventos de reversão em S. A seguir, descrevemos formalmente o problema de Ordenção de Permutações por Reversões e Transposições com Restrição de Proporção.

Ordenção de Permutações por Reversões e Transposições com Restrição de Proporção ($\mathbf{Sb_PRT}$)
Entrada: Uma instância clássica com ou sem sinais $\mathcal{I} = (\pi, \iota)$ e um número racional $k \in [0..1]$.
Objetivo: Com base no modelo de rearranjo $\mathcal{M} = \{\rho, \tau\}$, determinar uma sequência de eventos de rearranjo S de tamanho mínimo capaz de transformar π em ι e $\frac{|S_\rho|}{|S|} \geq k$.

Nesse capítulo, provamos que o problema $\mathbf{Sb_PRT}$ pertence à classe NP-difícil em instâncias clássicas sem sinais para qualquer valor de k. Em instâncias clássicas com sinais mostramos que existe um algoritmo exato polinomial para o problema quando $k = 1$ e provamos que o problema pertence à classe NP-difícil quando $k < 1$. Para as variações com e sem sinais do problema $\mathbf{Sb_PRT}$ apresentamos algoritmos de aproximação com fatores $3 - \frac{3k}{2}$ e $3 - k$, respectivamente. Além disso, apresentamos um algoritmo de aproximação assintótico com um fator teórico melhor para instâncias clássicas com sinais. Por fim, realizamos experimentos comparando o desempenho práticos dos algoritmo propostos.

3.1 Análise de Complexidade

Nessa seção, apresentamos uma análise sobre a complexidade do problema **Sb$_\mathbf{P}$RT** em instâncias com e sem sinais para todos os possível valores de k.

Sb$_\mathbf{P}$RT(Versão de Decisão)
Entrada: Uma instância clássica com ou sem sinais $\mathcal{I} = (\pi, \iota)$, um número racional $k \in [0..1]$ e um número natural t.
Pergunta: Existe uma sequência de eventos de rearranjo S, com base no modelo de rearranjo $\mathcal{M} = \{\rho, \tau\}$, capaz de transformar π em ι, com $\frac{|S_\rho|}{|S|} \geq k$ e $|S| \leq t$.

Capítulo 4

Modelos Intergênicos Rígidos

Capítulo 5

Modelos Intergênicos Flexíveis

Capítulo 6

Outras Contribuições

Capítulo 7

Conclusões

Referências Bibliográficas

[1] Alexsandro Oliveira Alexandrino, Guilherme Henrique Santos Miranda, Carla Negri Lintzmayer, and Zanoni Dias. Length-weighted λ-rearrangement Distance. *Journal of Combinatorial Optimization*, pages 1–24, 2020.

[2] 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.

[3] Martin Bader, Mohamed I. Abouelhoda, and Enno Ohlebusch. A Fast Algorithm for the Multiple Genome Rearrangement Problem with Weighted Reversals and Transpositions. *BMC Bioinformatics*, 9(1):1–13, 2008.

[4] Martin Bader and Enno Ohlebusch. Sorting by Weighted Reversals, Transpositions, and Inverted Transpositions. *Journal of Computational Biology*, 14(5):615–636, 2007.

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

[6] Vineet Bafna and Pavel A. Pevzner. Sorting by Transpositions. *SIAM Journal on Discrete Mathematics*, 11(2):224–240, 1998.

[7] Anne Bergeron. A Very Elementary Presentation of the Hannenhalli-Pevzner Theory. *Discrete Applied Mathematics*, 146(2):134–145, 2005.

[8] Piotr Berman, Sridhar Hannenhalli, and Marek Karpinski. 1.375-Approximation Algorithm for Sorting by Reversals. In R. Möhring and R. Raman, editors, *Proceedings of the 10th Annual European Symposium on Algorithms (ESA'2002)*, volume 2461 of *Lecture Notes in Computer Science*, pages 200–210. Springer-Verlag Berlin Heidelberg New York, Berlin/Heidelberg, Germany, 2002.

[9] Priscila Biller, Laurent Guéguen, Carole Knibbe, and Eric Tannier. Breaking Good: Accounting for Fragility of Genomic Regions in Rearrangement Distance Estimation. *Genome Biology and Evolution*, 8(5):1427–1439, 2016.

[10] Priscila Biller, Carole Knibbe, Guillaume Beslon, and Eric Tannier. Comparative Genomics on Artificial Life. In *Pursuit of the Universal*, pages 35–44. Springer International Publishing, 2016.

[11] Klairton Lima Brito, Andre Rodrigues Oliveira, Ulisses Dias, and Zanoni Dias. Heuristics for the Sorting Signed Permutations by Reversals and Transpositions Problem. In *Proceedings of the 5th International Conference on Algorithms for Computational Biology (AlCoB'2018)*, volume 10849, pages 65–75. Springer International Publishing, Heidelberg, Germany, 2018.

[12] Laurent Bulteau, Guillaume Fertin, and Irena Rusu. Sorting by Transpositions is Difficult. *SIAM Journal on Discrete Mathematics*, 26(3):1148–1180, 2012.

[13] Laurent Bulteau, Guillaume Fertin, and Eric Tannier. Genome Rearrangements with Indels in Intergenes Restrict the Scenario Space. *BMC Bioinformatics*, 17(14):426, 2016.

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

[15] Xin Chen. On Sorting Unsigned Permutations by Double-Cut-and-Joins. *Journal of Combinatorial Optimization*, 25(3):339–351, 2013.

[16] Xin Chen, Jie Zheng, Zheng Fu, Peng Nan, Yang Zhong, Stefano Lonardi, and Tao Jiang. Assignment of Orthologous Genes via Genome Rearrangement. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2(4):302–315, 2005.

[17] David A. Christie. A 3/2-Approximation Algorithm for Sorting by Reversals. In H. Karloff, editor, *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'1998)*, pages 244–252, Philadelphia, PA, USA, 1998. Society for Industrial and Applied Mathematics.

[18] David A. Christie. *Genome Rearrangement Problems*. PhD thesis, Department of Computing Science, University of Glasgow, 1998.

[19] David A. Christie and Robert W. Irving. Sorting Strings by Reversals and by Transpositions. *SIAM Journal on Discrete Mathematics*, 14(2):193–206, 2001.

[20] Ulisses Dias and Zanoni Dias. Extending Bafna-Pevzner Algorithm. In *Proceedings of the 1st International Symposium on Biocomputing (ISB'2010)*, pages 1–8, New York, NY, USA, 2010. ACM.

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

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

[23] Niklas Eriksen. *Combinatorics of Genome Rearrangements and Phylogeny*. Teknologie licentiat thesis, Kungliga Tekniska Högskolan, Stockholm, 2001.

[24] Niklas Eriksen. (1+ϵ)-Approximation of Sorting by Reversals and Transpositions. *Theoretical Computer Science*, 289(1):517–529, 2002.

[25] Guillaume Fertin, Géraldine Jean, and Eric Tannier. Algorithms for Computing the Double Cut and Join Distance on both Gene Order and Intergenic Sizes. *Algorithms for Molecular Biology*, 12(1):16, 2017.

[26] Guillaume Fertin, Anthony Labarre, Irena Rusu, Éric Tannier, and Stéphane Vialette. *Combinatorics of Genome Rearrangements*. Computational Molecular Biology. The MIT Press, London, England, 2009.

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

[28] Petr Kolman and Tomasz Waleń. Reversal Distance for Strings with Duplicates: Linear Time Approximation Using Hitting Set. In *International Workshop on Approximation and Online Algorithms*, pages 279–289, 2006.

[29] Andre Rodrigues Oliveira, Klairton Lima Brito, Ulisses Dias, and Zanoni Dias. On the Complexity of Sorting by Reversals and Transpositions Problems. *Journal of Computational Biology*, 26:1223–1229, 2019.

[30] Andre Rodrigues Oliveira, Klairton Lima Brito, Zanoni Dias, and Ulisses Dias. Sorting by Weighted Reversals and Transpositions. In *Proceedings of the 11th Brazilian Symposium on Bioinformatics (BSB'2018)*, pages 38–49. Springer International Publishing, Heidelberg, Germany, 2018.

[31] Andre Rodrigues Oliveira, Klairton Lima Brito, Zanoni Dias, and Ulisses Dias. Sorting by Weighted Reversals and Transpositions. *Journal of Computational Biology*, 26:420–431, 2019.

[32] Andrew J. Radcliffe, Alex D. Scott, and Elizabeth L. Wilmer. Reversals and Transpositions Over Finite Alphabets. *SIAM Journal on Discrete Mathematics*, 19(1):224–244, 2005.

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

[34] Luiz Augusto G. Silva, Luis Antonio B. Kowada, Noraí Romeu Rocco, and Maria Emília M. T. Walter. A new 1.375-approximation algorithm for sorting by transpositions. *Algorithms for Molecular Biology*, 17(1):1–17, 2022.

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

[36] Maria E. M. T. Walter, Zanoni Dias, and João Meidanis. Reversal and Transposition Distance of Linear Chromosomes. In *Proceedings of the 5th International Symposium on String Processing and Information Retrieval (SPIRE'1998)*, pages 96–102, Los Alamitos, CA, USA, 1998. IEEE Computer Society.

[37] Eyla Willing, Simone Zaccaria, Marília DV Braga, and Jens Stoye. On the Inversion-Indel Distance. *BMC Bioinformatics*, 14:S3, 2013.

[38] Sophia Yancopoulos, Oliver Attie, and Richard Friedberg. Efficient Sorting of Genomic Permutations by Translocation, Inversion and Block Interchange. *Bioinformatics*, 21(16):3340–3346, 2005.