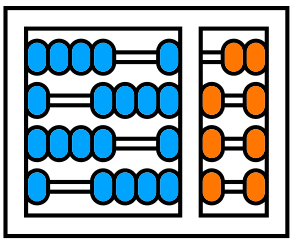
Universidade Estadual de Campinas

Instituto de Computação



Exame de Qualificação Específico

Generalizações de Problemas Envolvendo Partição de Strings e Eventos de Rearranjo

---

Candidato: Gabriel Henriques Siqueira

Orientador: Prof. Dr. Zanoni Dias

Resumo

O problema de Partição Comum Mínima de Strings é um problema de otimização combinatória com grande relevância tanto pelo sua complexidade teórica quanto pela sua aplicação prática em problemas da biologia computacional. Esse problema tem como objetivo encontrar o número mínimo de substrings necessárias para formar duas strings distintas dadas como entrada, mudando apenas a ordem em que as substrings são concatenadas. Algumas relações foram estabelecidas entre suas variações e problemas de rearranjo de genomas, que são importantes para comparação genômica. Em estudos mais recentes, os problemas de rearranjo vêm incorporando informações sobre as regiões intergênicas, o que motiva a incorporação dessas estruturas também nos problemas de partição. Além disso, a maior parte dos resultados conhecidos para os problemas de rearranjo e de partição assumem que as duas strings consideradas no problema são formadas pelos mesmos caracteres, mas existem versões mais gerais dos problemas com strings desbalanceadas, ou seja, strings com composições distintas, tanto em termos do conjunto de caracteres, quanto da quantidade de cada um deles. O objetivo deste trabalho é estudar variações dos problemas de Partição Comum Mínima de Strings e de Rearranjo de Genomas, incluindo generalizações que envolvem a adição de informação sobre as regiões intergênicas e consideram strings desbalanceadas.

1 Introdução

Strings (cadeias de caracteres) são estruturas computacionais largamente utilizadas na modelagem de diversos problemas em áreas como processamento de texto e biologia computacional. Em problemas que podem ser modelados por strings, é comum a necessidade de se obter uma métrica que permita a comparação entre duas strings. A busca por distâncias entre duas strings levou à definição de vários problemas que possuem um aspecto teórico interessante e aplicações em problemas práticos.

Em particular, o problema de Partição Comum Mínima de Strings (PCMS) busca pelo menor número de substrigs em que se deve particionar uma string de origem para que essas substrings possam ser reordenadas de forma a se obter uma string de destino. O problema PCMS foi proposto de forma independente por Chen e coautores [12], em 2005, e Swenson e

coautores [36], em 2008. Em ambos os trabalhos, o problema foi motivado pela sua aplicação na comparação entre genomas. Nessa aplicação os genomas são representados por strings, onde a orientação dos genes é representada por sinais associados a cada caractere. Devido a essa representação, a versão do problema PCMS inicialmente estudada foi o problema de Partição com Sinais Comum Mínima de Strings (PSCMS). Nessa versão do problema as substrings podem ser invertidas para obtermos a string de destino e, se essa inversão for utilizada em uma versão sem sinais do problema, temos uma terceira variação chamada Partição Reversa Comum Mínima de Strings (PRCMS).

Esses problemas de partição pertencem à classe de problemas NP-Difícil [16], ou seja, não existem algoritmos polinomiais exatos para esses problemas, a menos que $P = NP$. Uma forma de se abordar problemas com essa restrição é através de algoritmos que garantam uma fator de aproximação. Dizer que um algoritmo garante um fator de aproximação α significa que a razão entre o valor obtido pelo algoritmo e o valor ótimo é menor do que α. As aproximações conhecidas para os problemas de partição assumem que as strings são compostas pelos mesmos caracteres, sendo que apenas a ordem e orientação dos caracteres podem ser diferentes entre as strings. Com essa restrição, os melhores algoritmos conhecidos [19] possuem fatores de aproximação $4k$, $8k$ e $8k$ para os problemas PCMS, PSCMS e PRCMS, respectivamente, onde k é o maior número de cópias de um caractere nas strings. Existe também um algoritmo com fator de aproximação $O(\log n \log^* n)$ [14] para os problemas, onde n é a quantidade de caracteres presente nas strings.

A principal aplicação dos problemas de partição de strings está em sua relação com os problemas de distância de rearranjo de genomas. Um rearranjo de genoma é uma mutação que afeta uma grande porção do genoma. Dado um conjunto de operações de rearranjo, chamado modelo de rearranjo, a distância de rearranjo é o menor número de operações desse modelo necessárias para transformar um genoma em outro.

Dentre os rearranjos mais estudados na literatura, temos: as reversões, que invertem a ordem e orientação de uma sequência de genes; as transposições, que trocam a posição de duas sequências consecutivas de genes; os indels (inserções ou deleções), que inserem ou removem genes. Encontrar a distância de rearranjo considerando o evento de reversão,

o evento de transposição ou a combinação dos dois eventos são problemas da classe NP-Difícil [13, 23, 28]. Os problemas de partição de strings são a principal forma de se obter aproximações para os problemas de rearranjo [12, 31].

Normalmente, os trabalhos envolvendo problemas de partição de strings ou rearranjo de genomas se limitam a estudar strings compostas pelos mesmos caracteres, mas no caso geral pode haver caracteres presentes em apenas uma das strings. Alguns trabalhos abordaram esse cenário [2, 9, 38]. Também existem problemas de rearranjo de genomas envolvendo regiões intergênicas [7, 24] para os quais ainda não foram propostos problemas correspondentes envolvendo partições de strings. Essas generalizações pertence à classe de problemas NP-Difícil. Para simplificar o problema, vários estudos foram realizados assumindo que não existem dois genes iguais. Nesse caso, dizemos que o genoma não tem repetição de genes. Os problemas de rearranjo são separados em variações com sinais, onde a orientação dos genes é codificada por sinais associados aos caracteres das strings, e variações sem sinais, onde a orientação dos genes é ignorada. As tabelas 1 e 2 apresentam os melhores fatores do aproximação conhecidos para diferentes problemas de rearranjo com e sem sinais, respectivamente. Em negrito estão os resultados obtidos no início do doutorado.

Tabela 1: Aproximações conhecidas para distâncias de rearranjo em genomas sem sinais levando em conta a presença de repetição de genes ou regiões intergênicas (RI). Nas aproximações com repetição de genes, k é o maior número de cópias de um gene no genoma.

Distância de...	**Sem Repetição de Genes**		**Com Repetição de Genes**	
	Sem RI	**Com RI**	**Sem RI**	**Com RI**
Reversão	1.375 [5]	4 [7]	$16k$ [19, 32], **4k**	**8k**
Transposição	1.375 [15]	3.5 [26]	$12k$ [19, 31], **6k**	**6k**[1]
Reversão e Transposição	$2.8334 + \epsilon$ [11, 29]	4 [6]	$24k$ [19, 32], **6k**	**8k**
Reversão e Indels	2 [2]	4 [3]	–	–
Transposição e Indels	3 [2]	4.5 [3]	–	–
Reversão, Transposição e Indels	3 [2]	6 [3]	–	–

[1]aproximação assintótica.

Além do desenvolvimento de algoritmos de aproximação para os problemas de partição, outra abordagem muito utilizada para lidar com esses problemas é o uso de algoritmos parametrizados [10, 18]. Tais algoritmos garantem que a complexidade em termos de tempo de execução de um problema depende exponencialmente apenas de um parâmetro das strings

Tabela 2: Aproximações conhecidas para distâncias de rearranjo em genomas com sinais levando em conta a presença de repetição de genes ou regiões intergênicas (RI). Nas aproximações com repetição de genes, k é o maior número de cópias de um gene no genoma.

Distância de...	**Sem Repetição de Genes**		**Com Repetição de Genes**	
	Sem RI	**Com RI**	**Sem RI**	**Com RI**
Reversão	Exato [17]	2 [24]	$16k$ [12, 19], **4k**	**8k**
Reversão e Transposição	2 [37]	3 [25]	$24k$ [19, 32], **6k**	**9k**
Reversão e Indels	Exato [38]	3 [1]	–	–
Reversão, Transposição e Indels	3 [2]	–	–	–

de entrada, como o número máximo de cópias de algum caractere, o tamanho da partição buscada ou a quantidade de caracteres distintos na string. Dessa forma, caso as instâncias de interesse tenham valores pequenos para o parâmetro utilizado, o problema pode ser resolvido de forma exata ou com um melhor fator de aproximação em um tempo aceitável. Nesse contexto de algoritmos parametrizados, novas variações mais gerais de problemas de partição ou rearranjo foram propostas, como a operação de k-cut [8], que corta um genoma em k partes e reorganiza as partes em qualquer ordem, ou o problema de Partição Comum Mínima de Strings Restrita por uma Permutação [20], que além de buscar uma forma de particionar a string de origem, busca também como ela deve ser reordenada para obtermos a string de destino.

O restante deste documento está dividido da seguinte forma. A Seção 2 apresenta formalmente os problemas estudados e outras definições relacionadas. Em seguida, a Seção 3 apresenta os objetivos deste projeto e a Seção 4 descreve alguns resultados preliminares relacionados a esses objetivos. Por fim, a Seção 5 descreve a metodologia que será utilizada e a Seção 6 apresenta o cronograma proposto.

2 Definições e Problemas

Formalmente, uma *string* S é uma sequência de caracteres pertencentes a um conjunto Σ_S (chamado *alfabeto* de S). Como os elementos de Σ_S podem ter múltiplas cópias em S, utilizamos o termo caractere para falar de uma cópia específica e o termo rótulo para falar dos elementos de Σ_S. O caractere na i-ésima posição da string S é denotado por S_i e o

número de caracteres de S (o tamanho de S) é denotado por $|S|$. A *ocorrência* de um rótulo α em uma string S é a quantidade de caracteres de S com rótulo α, e é denotada por $occ(\alpha, S)$. A *ocorrência máxima* de algum rótulo em S é $occ(S) = \max_{\alpha \in \Sigma_S}(occ(\alpha, S))$.

Exemplo 1 *A string* $S = (A\ D\ B\ B\ A\ C\ B\ D\ E)$ *apresenta os valores* $|S| = 9$, $S_1 = A$, $S_4 = B$, $occ(A, S) = 2$, $occ(B, S) = 3$ *e* $occ(S) = 3$.

Uma *partição direta comum* entre duas strings S e P é um par de sequências de strings $(\mathbb{S}, \mathbb{P})$ tal que:

1. As strings de $\mathbb{S}$ quando concatenadas correspondem a S.
2. As strings de $\mathbb{P}$ quando concatenadas correspondem a P.
3. Existe uma bijeção ϕ entre uma subsequência $\mathbb{S}'$ de strings de $\mathbb{S}$ e uma subsequência $\mathbb{P}'$ de strings de $\mathbb{P}$, tal que: (i) Se $\phi(S_i) = S_j$, onde $\mathbb{S}_i \in \mathbb{S}'$ e $\mathbb{P}_j \in \mathbb{P}'$, então $\mathbb{S}_i = \mathbb{P}_j$; e (ii) Se X é o conjunto de rótulos dos caracteres de $\mathbb{S}$ que não pertencem a $\mathbb{S}'$ e Y é o conjunto de rótulos dos caracteres de $\mathbb{P}$ que não pertencem a $\mathbb{P}'$, então nenhum rótulo pertence simultaneamente a X e a Y.

Dada uma string S, dizemos que a string Q é *reversa* de S, denotada por $Q = rev(S)$, se $Q_i = S_{|S|-i+1}, \forall 1 \leq i \leq |S|$.

Uma *partição reversa comum* entre duas strings S e P é um par de sequências de strings $(\mathbb{S}, \mathbb{P})$ tal que:

1. As strings de $\mathbb{S}$ quando concatenadas correspondem a S.
2. As strings de $\mathbb{P}$ quando concatenadas correspondem a P.
3. Existe uma bijeção ϕ entre uma subsequência $\mathbb{S}'$ de strings de $\mathbb{S}$ e uma subsequência $\mathbb{P}'$ de strings de $\mathbb{P}$, tal que: (i) Se $\phi(\mathbb{S}_i) = \mathbb{P}_j$, onde $\mathbb{S}_i \in \mathbb{S}'$ e $\mathbb{P}_j \in \mathbb{P}'$, então $\mathbb{S}_i = \mathbb{P}_j$ ou $rev(\mathbb{S}_i) = \mathbb{P}_j$; e (ii) Se X é o conjunto de rótulos dos caracteres de $\mathbb{S}$ que não pertencem a $\mathbb{S}'$ e Y é o conjunto de rótulos dos caracteres de $\mathbb{P}$ que não pertencem a $\mathbb{P}'$, então nenhum rótulo pertence simultaneamente a X e a Y.

A Figura 1 mostra uma partição direta comum entre duas strings S e P. A Figura 2 mostra uma partição reversa das mesmas strings.

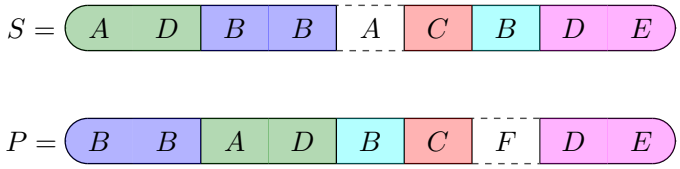


Figura 1: Partição direta comum entre duas strings S e P. Nessa partição temos $\mathbb{S} = [(A\ D), (B\ B), (A), (C), (B), (D\ E)]$ e $\mathbb{P} = [(B\ B), (A\ D), (B), (C), (F), (D\ E)]$. As substrings coloridas estão em $\mathbb{S}'$ e $\mathbb{P}'$ e as cores indicam uma bijeção entre elas. Note que, as substrings em branco (A) e (F), que não estão nessas sequências, não têm rótulos em comum.

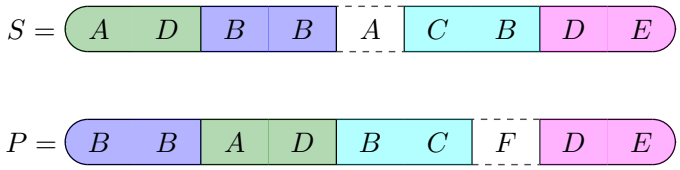


Figura 2: Partição reversa comum entre duas strings S e P. Nessa partição temos $\mathbb{S} = [(A\ D), (B\ B), (A), (C\ B), (D\ E)]$ e $\mathbb{P} = [(B\ B), (A\ D), (B\ C), (F), (D\ E)]$. As substrings coloridas estão em $\mathbb{S}'$ e $\mathbb{P}'$ e as cores indicam uma bijeção entre elas. Note que a substring $(C\ B)$ é a reversa da substring $(B\ C)$.

Em alguns casos, é importante atribuir uma noção de orientação aos caracteres de uma string. Essa noção é dada pela presença de um sinal $+$ ou $-$ associado a cada caractere da string, nesse caso dizemos que temos uma string com sinais. No caso de uma string com sinais S, a string Q é reversa de S se $Q_i = -S_{|S|-i+1}, \forall 1 \leq i \leq |S|$. Com essa definição, a mesma definição de partição reversa comum pode ser aplicada a strings com sinais, nesse caso a partição é chamada *partição com sinais comum*.

Para cada tipo de partição entre duas strings, existe um problema cujo objetivo é minimizar o tamanho da partição, isso é, minimizar o número de substrings nas sequências $\mathbb{S}$ e $\mathbb{P}$. Os três tipos de partição que definimos dão origem aos problemas: Partição Comum Mínima de Strings (PCMS), que considera uma partição direta; Partição Reversa Comum Mínima de Strings (PRCMS), que considera uma partição reversa; e Partição com Sinais Comum Mínima de Strings (PSCMS), que considera uma partição com sinais.

Duas strings S e P são *balanceadas* se $\Sigma_S = \Sigma_P$ e $occ(\alpha, S) = occ(\alpha, P), \forall \alpha \in \Sigma_S$. Caso contrário elas são *desbalanceadas*. A maioria dos resultados para os problemas de partição assumem que as strings S e P são balanceadas [10, 12, 16, 18].

2.1 Rearranjo de Genomas

Algoritmos para os problemas de partição de strings podem ser usados como parte de algoritmos para os problemas de rearranjo de genomas. Um rearranjo é uma mutação que pode afetar grandes trechos do genoma. Vamos limitar esse estudo aos genomas lineares que possuam apenas um cromossomo. Nesse caso, os eventos de rearranjos podem ser conservativos, ou seja, alteram apenas a ordem e orientação do material genético, ou não conservativos, ou seja, podem remover ou inserir nucleotídeos alterando os genes ou as regiões entre genes. Diferentes representações podem ser utilizadas para os genomas, o que afeta as operações utilizadas para representar os eventos de rearranjo.

Na representação que utilizamos, supomos que um genoma $\mathcal{G} = (g_1, \breve{g}_1, g_2, \ldots, \breve{g}_{n-1}, g_n)$ é composto por uma sequência alternada de n genes $(g_1, \ldots, g_n)$ e $n-1$ regiões intergênicas $(\breve{g}_1, \ldots, \breve{g}_{n-1})$. Nós representamos um genoma $\mathcal{G} = (S, \breve{S})$ por:

- Uma string S, onde cada caractere S_i representa o gene g_i.
- Uma lista de inteiros não negativos $\breve{S}$, onde cada inteiro $\breve{S}_i$ representa o tamanho de uma região intergênica $\breve{g}_i$ (o tamanho de uma região intergênicas é a quantidade de nucleotídeos que ela possui).

Quando a orientação dos genes é conhecida, S é uma string com sinais, onde cada sinal de um caractere representa a orientação do gene correspondente. Nesse caso, dizemos que temos um genoma com sinais. Caso a orientação seja desconhecida, S é uma string sem sinais e temos um genoma sem sinais. As figuras 3 e 4 mostram um exemplo de um genoma com sinais e um exemplo de um genoma sem sinais, respectivamente.

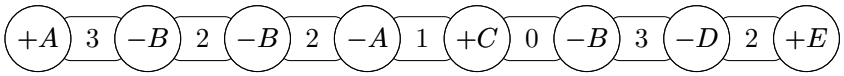


Figura 3: O genoma $\mathcal{G} = (S, \breve{S})$, tal que $S = (+A\ -B\ -B\ -A\ +C\ -B\ -D\ +E)$ e $\breve{S} = [3, 2, 2, 1, 0, 3, 2]$.

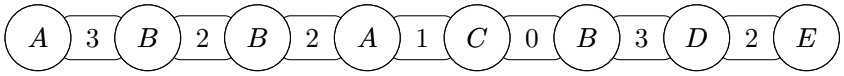


Figura 4: O genoma $\mathcal{G} = (S, \breve{S})$, tal que $S = (A\ B\ B\ A\ C\ B\ D\ E)$ e $\breve{S} = [3, 2, 2, 1, 0, 3, 2]$.

Dado um genoma com sinais $\mathcal{G} = (S, \breve{S})$ de tamanho n e inteiros i, j, x, y, tais que $2 \leq i \leq j \leq n-1$, $0 \leq x \leq \breve{S}_{i-1}$ e $0 \leq y \leq \breve{S}_j$. A *reversão* $\rho^{(i,j)}_{(x,y)}$ é uma operação de rearranjo que transforma $\mathcal{G}$ no genoma $\mathcal{G}.\rho^{(i,j)}_{(x,y)} = (S', \breve{S}')$, tal que:

$$S' = (S_1 \ldots S_{i-1}\ \underline{-S_j \ldots -S_i}\ \ldots S_n)$$
$$\breve{S}' = [\breve{S}_1, \ldots, \breve{S}_{i-2}, \underline{x+y}, \underline{\breve{S}_{j-1}, \ldots, \breve{S}_i}, \underline{x'+y'}, \breve{S}_{j+1}, \ldots, \breve{S}_{n-1}],$$

sendo que $x' = \breve{S}_{i-1} - x$ e $y' = \breve{S}_j - y$. A Figura 5 apresenta um exemplo de reversão em um genoma com sinais.

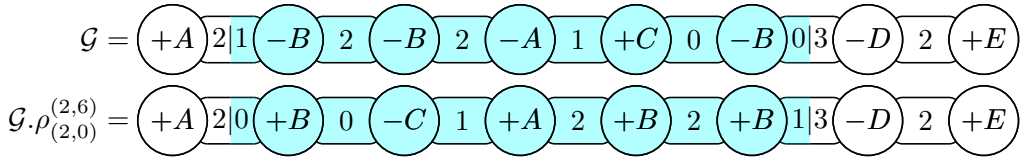


Figura 5: Uma reversão $\rho^{(2,6)}_{(2,0)}$ aplicada no genoma com sinais $\mathcal{G} = ((+A\ -B\ -B\ -A\ +C\ -B\ -D\ +E), [3, 2, 2, 1, 0, 3, 2])$ resultando no genoma $\mathcal{G}.\rho^{(2,6)}_{(2,0)} = ((+A\ +B\ -C\ +A\ +B\ +B\ -D\ +E), [2, 0, 1, 2, 2, 4, 2])$.

Dado um genoma sem sinais $\mathcal{G} = (S, \breve{S})$ de tamanho n e inteiros i, j, x, y, tais que $2 \leq i \leq j \leq n-1$, $0 \leq x \leq \breve{S}_{i-1}$ e $0 \leq y \leq \breve{S}_j$. A *reversão* $\rho^{(i,j)}_{(x,y)}$ é uma operação de rearranjo que transforma $\mathcal{G}$ no genoma $\mathcal{G}.\rho^{(i,j)}_{(x,y)} = (S', \breve{S}')$, tal que:

$$S' = (S_1 \ldots S_{i-1}\ \underline{S_j \ldots S_i}\ \ldots S_n)$$
$$\breve{S}' = [\breve{S}_1, \ldots, \breve{S}_{i-2}, \underline{x+y}, \underline{\breve{S}_{j-1}, \ldots, \breve{S}_i}, \underline{x'+y'}, \breve{S}_{j+1}, \ldots, \breve{S}_{n-1}],$$

sendo que $x' = \breve{S}_{i-1} - x$ e $y' = \breve{S}_j - y$. A Figura 6 apresenta um exemplo de reversão em um genoma sem sinais.

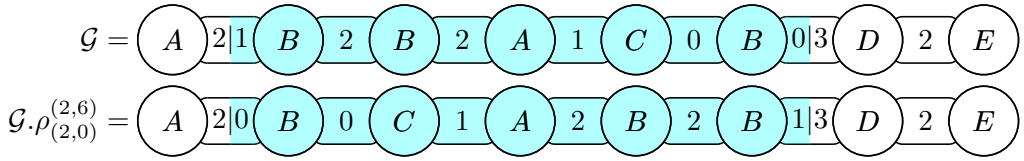


Figura 6: Uma reversão $\rho^{(2,6)}_{(2,0)}$ aplicada no genoma sem sinais $\mathcal{G} = ((A\ B\ B\ A\ C\ B\ D\ E), [3, 2, 2, 1, 0, 3, 2])$ resultando no genoma $\mathcal{G}.\rho^{(2,6)}_{(2,0)} = ((A\ B\ C\ A\ B\ B\ D\ E), [2, 0, 1, 2, 2, 4, 2])$.

Dado um genoma com ou sem sinais $\mathcal{G} = (S, \breve{S})$ de tamanho n e inteiros i, j, k, x, y, z, tais que $2 \leq i < j < k \leq n$, $0 \leq x \leq \breve{S}_{i-1}$, $0 \leq y \leq \breve{S}_{j-1}$ e $0 \leq z \leq \breve{S}_{k-1}$. A *transposição* $\tau^{(i,j,k)}_{(x,y,z)}$ é uma operação de rearranjo que transforma $\mathcal{G}$ no genoma $\mathcal{G}.\tau^{(i,j,k)}_{(x,y,z)} = (S', \breve{S}')$, tal que:

$$S' = (S_1 \ldots S_{i-1}\ \underline{S_j \ldots S_{k-1}}\ \underline{S_i \ldots S_{j-1}}\ S_k \ldots S_n)$$
$$\breve{S}' = [\breve{S}_1, \ldots, \breve{S}_{i-2}, \underline{x + y'}, \underline{\breve{S}_j, \ldots, \breve{S}_{k-2}}, \underline{z + x'}, \underline{\breve{S}_i, \ldots, S_{j-2}}, \underline{y + z'}, S_k, \ldots, S_{n-1}],$$

sendo que $x' = \breve{S}_{i-1} - x$, $y' = \breve{S}_{j-1} - y$ e $z' = \breve{S}_{k-1} - z$. A Figura 7 apresenta um exemplo de transposição.

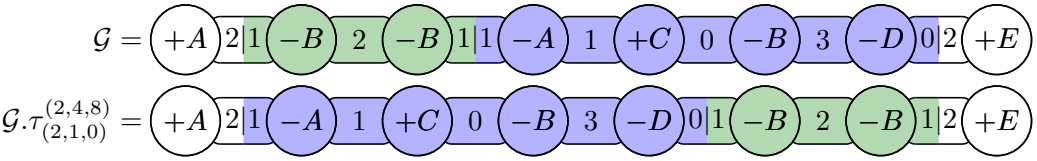


Figura 7: Uma transposição $\tau^{(2,4,8)}_{(2,1,0)}$ aplicada no genoma com sinais $\mathcal{G} = ((+A\ -B\ -B\ -A\ +C\ -B\ -D\ +E), [3, 5, 2, 2, 1, 0, 3, 2])$ resultando no genoma $\mathcal{G}.\tau^{(2,4,8)}_{(2,1,0)} = ((+A\ -A\ +C\ -B\ -D\ -B\ -B\ +E), [3, 1, 0, 3, 1, 2, 3])$.

Dado um genoma com ou sem sinais $\mathcal{G} = (S, \breve{S})$ de tamanho n, um genoma $\mathcal{X} = (A, \breve{A})$ ($\mathcal{X}$ é um genoma com sinais se e somente se $\mathcal{G}$ é um genoma com sinais) e inteiros i, x, y, z, tais que $1 \leq i \leq n - 1$, $0 \leq x \leq \breve{S}_i$, $0 \leq y$ e $0 \leq z$. A *inserção* $\phi^{(i,\mathcal{X})}_{(x,y,z)}$ é uma operação de rearranjo que transforma o genoma $\mathcal{G}$ no genoma $\mathcal{G}.\phi^{(i,\mathcal{X})}_{(x,y,z)} = (S', \breve{S}')$, tal que:

$$S' = (S_1 \ldots S_i\ \underline{A_1 \ldots A_{|A|}}\ S_{i+1} \ldots S_n)$$
$$\breve{S}' = [\breve{S}_1, \ldots, \breve{S}_{i-1}, \underline{x + y, \breve{A}_1, \ldots, \breve{A}_{|\breve{A}|}, z + x'}, \breve{S}_{i+1}, S_{n-1}],$$

sendo que $x' = \breve{S}_i - x$. Note que se adotarmos um genoma vazio para $\mathcal{X}$ a inserção

corresponde a adicionar nucleotídeos na região intergênica $\breve{S}_i$. A Figura 8 apresenta um exemplo de inserção.

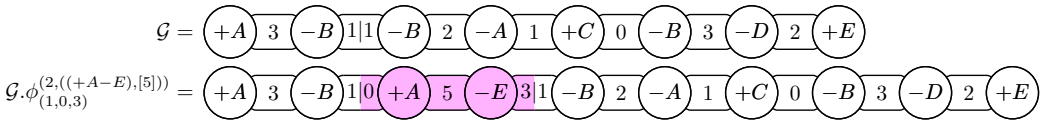


Figura 8: Uma inserção $\phi^{(2,((+A\ -E),[5]))}_{(1,0,3)}$ aplicada no genoma com sinais $\mathcal{G} = ((+A\ -B\ -B\ -A\ +C\ -B\ -D\ +E), [3,2,2,1,0,3,2])$ resultando no genoma $\mathcal{G}.\phi^{(2,((+A\ -E),[5]))}_{(1,0,3)} = ((+A\ -B\ +A\ -E\ -B\ -A\ +C\ -B\ -D\ +E), [3,1,5,4,2,1,0,3,2])$.

Dado um genoma com ou sem sinais $\mathcal{G} = (S, \breve{S})$ de tamanho n e inteiros i, j, x, tais que $2 \leq i \leq j \leq n$ e $0 \leq x \leq \breve{S}_{i-1} + \breve{S}_{j-1}$. A *deleção* $\psi^{(i,j)}_{(x)}$ é uma operação de rearranjo que transforma o genoma $\mathcal{G}$ no genoma $\mathcal{G}.\psi^{(i,j)}_{(x)} = (S', \breve{S}')$, tal que:

$$S' = (S_1 \ \ldots \ S_{i-1}\ S_j \ \ldots \ S_n)$$
$$\breve{S}' = [\breve{S}_1, \ldots, \breve{S}_{i-2}, \underline{x}, \breve{S}_j, S_{n-1}].$$

Note que se $i = j$ a deleção apenas remove nucleotídeos da região intergênica $\breve{S}_i$, nesse caso o valor de x deve satisfazer $0 \leq x < \breve{S}_{i-1}$. A Figura 9 apresenta um exemplo de deleção.

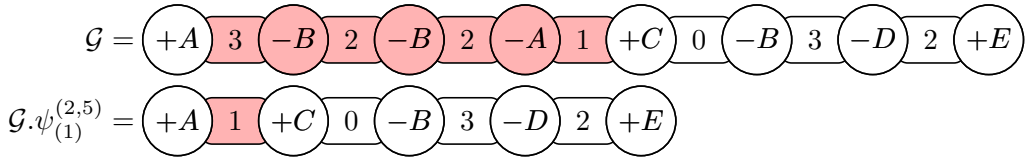


Figura 9: Uma deleção $\psi^{(2,5)}_{(1)}$ aplicada no genoma com sinais $\mathcal{G} = ((+A\ -B\ -B\ -A\ +C\ -B\ -D\ +E), [3,2,2,1,0,3,2])$ resultando no genoma $\mathcal{G}.\psi^{(2,5)}_{(1)} = ((+A\ +C\ -B\ -D\ +E), [1,0,3,2])$.

Dados dois genomas $\mathcal{G}$ e $\mathcal{H}$ e um conjunto de operações de rearranjo, a *distância de rearranjo* é o menor número de operações desse conjunto necessárias para transformar $\mathcal{G}$ em $\mathcal{H}$. Dois genomas $\mathcal{G} = (S, \breve{S})$ e $\mathcal{H} = (P, \breve{P})$ são *balanceados* se as strings S e P são balanceadas e $\sum_{i=1}^{|\breve{S}|} \breve{S}_i = \sum_{i=1}^{|\breve{P}|} \breve{P}_i$. Dizemos que dois genomas $\mathcal{G}$ e $\mathcal{H}$ são *desbalanceados* se eles não são balanceados. Quando o conjunto de operação apresenta apenas eventos conservativos, é necessário que os dois genomas sendo comparados sejam balanceados.

Em trabalhos que não consideram as regiões intergênicas, os genomas são representados apenas como strings. Nesses casos, as definições das operações de rearranjo incluem apenas as transformações aplicadas em S. Algumas relações foram encontradas entre problemas de partição e problemas de rearranjo considerando apenas strings balanceadas:

- Uma aproximação com fator ℓ para o problema PSCMS garante uma aproximação de fator 2ℓ para a distância de rearranjo considerando o evento de reversão em strings com sinais [12] e uma aproximação de fator 3ℓ para a distância de rearranjo considerando os eventos de reversão e transposição em strings com sinais [32].

- Uma aproximação com fator ℓ para o problema PRCMS garante uma aproximação de fator 2ℓ para a distância de rearranjo considerando o evento de reversão em strings sem sinais e uma aproximação de fator 3ℓ para a distância de rearranjo considerando os eventos de reversão e transposição em strings sem sinais [32].

- Uma aproximação com fator ℓ para o problema PCMS garante uma aproximação de fator 3ℓ para a distância de rearranjo considerando o evento de transposição em strings sem sinais [31].

Parte do objetivo deste projeto é estabelecer relações similares entre as seguintes variações dos problemas de partição e problemas de rearranjo de genomas envolvendo regiões intergênicas, além de incluir os eventos de inserção e deleção para generalizar as relações considerando strings não balanceadas.

Para facilitar as definições dos problemas de partição, utilizamos o conceito de quebra. Uma *quebra* de um genoma $\mathcal{G} = (S, \breve{S})$ é uma operação que separa $\mathcal{G}$ em dois genomas $\mathcal{H} = (P, \breve{P})$ e $\mathcal{K} = (Q, \breve{Q})$ com uma região intergênica $\breve{S}_i$, tal que: $|P| = i$; $P_j = S_j, \forall 1 \leq j \leq i$; $|Q| = |S| - i$; $Q_{j-i} = S_j, \forall i < j \leq |S|$; $|\breve{P}| = i - 1$; $\breve{P}_j = \breve{S}_j, \forall 1 \leq j < i$; $|\breve{Q}| = |\breve{S}| - i$; e $\breve{Q}_{j-i} = \breve{S}_j, \forall i < j \leq |\breve{S}|$ (ou seja, $\mathcal{H}$ e $\mathcal{K}$ são compostos pelos genes e regiões intergênicas anteriores a $\breve{S}_i$ e posteriores a $\breve{S}_i$, respectivamente). Chamamos a região intergênica onde a quebra ocorre de *breakpoint*. A Figura 10 mostra um exemplo de quebra. Note que, um genoma pode ser quebrado em mais subgenomas ao realizarmos múltiplas quebras.

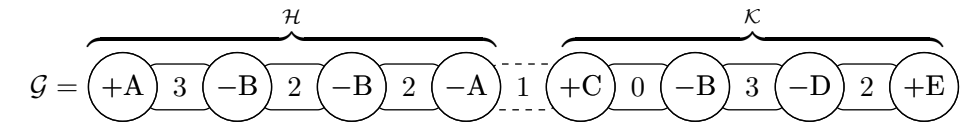


Figura 10: O genoma $\mathcal{G} = (S, \breve{S})$, com $S = (+A\ -B\ -B\ -A\ +C\ -B\ -D\ +E)$ e $\breve{S} = [3, 2, 2, 1, 0, 3, 2])$, quebrado em dois genomas $\mathcal{H} = ((+A\ -B\ -B\ -A), [3, 2, 2])$ e $\mathcal{K} = ((+C\ -B\ -D\ +E), [0, 3, 2])$ pelo *breakpoint* $\breve{S}_4 = 1$.

Um genoma com sinais $\mathcal{K} = (Q, \breve{Q})$ é o reverso de um genoma com sinais $\mathcal{G} = (S, \breve{S})$, denotado por $\mathcal{K} = rev(\mathcal{G})$, se $Q_i = -S_{n-i+1}, \forall 1 \leq i \leq |S|$ e $\breve{Q}_i = \breve{S}_{n-i}, \forall 1 \leq i \leq |\breve{S}|$. De forma análoga, um genoma sem sinais $\mathcal{K} = (Q, \breve{Q})$ é o reverso de um genoma sem sinais $\mathcal{G} = (S, \breve{S})$, se $Q_i = S_{n-i+1}, \forall 1 \leq i \leq |S|$ e $\breve{Q}_i = \breve{S}_{n-i}, \forall 1 \leq i \leq |\breve{S}|$.

Uma *partição intergênica com sinais comum* entre dois genomas com sinais $\mathcal{G} = (S, \breve{S})$ e $\mathcal{H} = (P, \breve{P})$ é um par de sequências de genomas com sinais $(\mathbb{S}, \mathbb{P})$ tal que:

1. O genoma $\mathcal{G}$ pode ser quebrado nos genomas de $\mathbb{S}$.

2. O genoma $\mathcal{H}$ pode ser quebrado nos genomas de $\mathbb{P}$.

3. Existe uma bijeção ϕ entre uma subsequência $\mathbb{S}'$ de genomas de $\mathbb{S}$ e uma subsequência $\mathbb{P}'$ de genomas de $\mathbb{P}$, tal que: (i) Se $\phi(\mathbb{S}_i) = \mathbb{P}_j$, onde $\mathbb{S}_i \in \mathbb{S}'$ e $\mathbb{P}_j \in \mathbb{P}'$, então $\mathbb{S}_i = \mathbb{P}_j$ ou $rev(\mathbb{S}_i) = \mathbb{P}_j$; e (ii) Se X é o conjunto de rótulos dos caracteres de $\mathbb{S}$ que não pertencem a $\mathbb{S}'$ e Y é o conjunto de rótulos dos caracteres de $\mathbb{P}$ que não pertencem a $\mathbb{P}'$, então nenhum rótulo pertence simultaneamente a X e a Y.

Note que essa é uma generalização da partição com sinais entre strings S e P, pois além de particionarmos as strings particionamos também as listas $\breve{S}$ e $\breve{P}$. A Figura 11 mostra uma partição com sinais entre dois genomas com sinais $\mathcal{G}$ e $\mathcal{H}$.

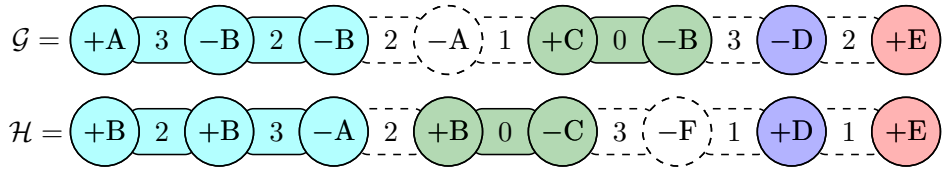


Figura 11: Partição intergênica com sinais comum entre dois genomas $\mathcal{G}$ e $\mathcal{H}$. Nessa partição temos $\mathbb{S} = [((+A\ -B\ -B), [3,2]), ((-A), []), ((+C\ -B), [0]), ((-D), []), ((+E), [])]$ e $\mathbb{P} = [((+B\ +B\ -A), [2,3]), ((+B\ -C), [0]), ((-F), []), ((+D), []), ((+E), [])]$. Os genomas coloridos estão em $\mathbb{S}'$ e $\mathbb{P}'$ e as cores indicam uma bijeção entre eles.

A mesma definição utilizada para partições com sinais pode ser aplicada para genomas sem sinais. Nesse caso, temos uma *partição intergênica reversa comum*, uma generalização da partição reversa comum de strings. De forma similar, podemos generalizar a partição direta comum.

Uma *partição intergênica direta comum* entre dois genomas $\mathcal{G} = (S, \breve{S})$ e $\mathcal{H} = (P, \breve{P})$ é um par de sequências de genomas $(\mathbb{S}, \mathbb{P})$ tal que:

1. O genoma $\mathcal{G}$ pode ser quebrado nos genomas de $\mathbb{S}$.

2. O genoma $\mathcal{H}$ pode ser quebrado nos genomas de $\mathbb{P}$.

3. Existe uma bijeção ϕ entre uma subsequência $\mathbb{S}'$ de genomas de $\mathbb{S}$ e uma subsequência $\mathbb{P}'$ de genomas de $\mathbb{P}$, tal que: (i) Se $\phi(\mathbb{S}_i) = \mathbb{P}_j$, onde $\mathbb{S}_i \in \mathbb{S}'$ e $\mathbb{P}_j \in \mathbb{P}'$, então $\mathbb{S}_i = \mathbb{P}_j$; e (ii) Se X é o conjunto de rótulos dos caracteres de $\mathbb{S}$ que não pertencem a $\mathbb{S}'$ e Y é o conjunto de rótulos dos caracteres de $\mathbb{P}$ que não pertencem a $\mathbb{P}'$, então nenhum rótulo pertence simultaneamente a X e a Y.

A Figura 12 mostra uma partição direta entre dois genomas sem sinais $\mathcal{G}$ e $\mathcal{H}$.

Nas versões do problema de partição apresentadas até aqui, temos uma condição muito restrita ao exigirmos que as regiões intergênicas sejam iguais. Por isso, é interessante considerar uma nova variação onde pequenas diferenças nas regiões intergênicas sejam ignoradas. Nessas variações, usamos a ideia de genomas flexíveis. Um *genoma flexível* $\mathcal{G} = (S, \breve{S})$ é representado por uma string S e uma lista de pares de inteiros $\breve{S}$. Cada par $(\breve{S}_i^i, \breve{S}_i^s)$ de $\breve{S}$ indica que a região intergênica $\breve{g}_i$ correspondente a esse par tem seu tamanho no intervalo

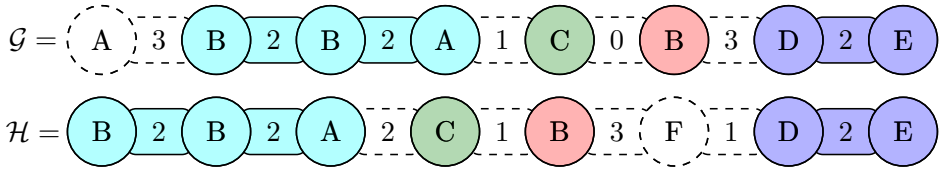


Figura 12: Partição intergênica com sinais comum entre dois genomas $\mathcal{G}$ e $\mathcal{H}$. Nessa partição temos $\mathbb{S} = [((A), []), ((B\ B\ A), [2, 2]), ((C), []), ((B), []), ((D\ E), [2])]$ e $\mathbb{P} = [((B\ B\ A), [2, 2]), ((C), []), ((B), []), ((F), []), ((D\ E), [2])]$. Os genomas coloridos estão em $\mathbb{S}'$ e $\mathbb{P}'$ e as cores indicam uma bijeção entre eles.

$[\breve{S}_i^i, \breve{S}_i^s]$. Um genoma $\mathcal{K} = (Q, \breve{Q})$ é dito *compatível* com um genoma flexível $\mathcal{G} = (S, \breve{S})$, denotado por $\mathcal{K} \approx \mathcal{G}$ se $Q = S$ e $\breve{S}_i^i \leq \breve{Q}_i \leq \breve{S}_i^s, \forall 1 \leq 1 \leq |\breve{S}|$. Se considerarmos que $\mathcal{H}$ e os genomas da sequência $\mathbb{P}$ são flexíveis e substituirmos a igualdade ($=$) por compatibilidade ($\approx$) no terceiro item das definições de partição, temos as definições de partição intergênica flexível com sinais comum, partição intergênica flexível reversa comum e partição intergênica flexível direta comum.

A Tabela 3 mostra as diferentes variações do problema de partição mínima de strings levando em conta o tipo de partição e a representação das regiões intergênicas. Assim como nos problemas em strings, o objetivo desses problemas é minimizar o tamanho das partições. Os problemas sem regiões intergênicas já aviam sido propostos na literatura [9, 12, 19].

Tabela 3: Problemas de partição mínima de strings de acordo com o tipo de partição e com a representação das regiões intergênicas (RI).

	Partição Direta	**Partição Reversa**	**Partição com Sinais**
Sem RI	PCMS	PRCMS	PSCMS
RI Fixas	PICMS	PIRCMS	PISCMS
RI Flexíveis	PFCMS	PFRCMS	PFSCMS

1. PCMS: Partição Comum Mínima de Strings;
2. PRCMS: Partição Reversa Comum Mínima de Strings;
3. PSCMS: Partição com Sinais Comum Mínima de Strings;
4. PICMS: Partição Intergênica Comum Mínima de Strings;
5. PIRCMS: Partição Intergênica Reversa Comum Mínima de Strings;

6. PISCMS: Partição Intergênica com Sinais Comum Mínima de Strings;

7. PFCMS: Partição Intergênica Flexível Comum Mínima de Strings;

8. PFRCMS: Partição Intergênica Flexível Reversa Comum Mínima de Strings;

9. PFSCMS: Partição Intergênica Flexível com Sinais Comum Mínima de Strings.

2.2 Mapeamentos e Empacotamento Máximo de Ciclos

Uma solução para um problema de partição permite simplificar os genomas originais, reduzindo seus tamanhos e o número de genes que apresentam múltiplas cópias (genes multiplicados). Tanto em genomas simplificados como nos genomas originais, uma das principais formas de se obter a distância de rearranjo é a obtenção de uma correspondência um para um entre os genes multiplicados presentes em ambos os genomas, seguida da aplicação de um dos algoritmos conhecidos para a distância de rearranjo quando o genoma não apresenta genes multiplicados [1, 5, 17, 24]. A busca por correspondências ligadas a menores distâncias pode ser feita a partir de alguma heurística que explora o espaço de possíveis correspondências [34].

Alternativamente, essa busca pode ser feita através do problema de *Empacotamento Máximo de Ciclos (EMC)*, utilizando uma adaptação do *grafo de adjacências* das strings [12, 30]. Esse grafo tem como objetivo codificar as relações de adjacência entre genes nos genomas sendo comparados e as possíveis correspondências entre os genes de ambos os genomas.

Dada uma string com sinais S, vamos definir o *grafo de adjacências parcial* $G_S = (V_S, E_S)$ com conjunto de vértices V_S e conjunto de arestas $E_S = E_S^b \cup E_S^r$, que descrevemos a seguir. Inicialmente, considere a adição de dois novos caracteres $+L$ e $+R$ (com rótulos não pertencentes ao alfabeto) nas extremidades de S de forma que $S_0 = +L$ e $S_{|S|+1} = +R$. Para cada i, com $0 \leq i \leq |S|$, existem dois vértices em V_S, representados por $+S_i$ e $-S_{i+1}$, conectados por uma aresta $e_i^b \in E_S^b$. As arestas do conjunto E_S^b são denominadas arestas pretas. Note que, todo S_i de S, exceto $+L$ e $+R$, dá origem a dois vértices $+S_i$ e $-S_i$. Dizemos que esses dois vértices são *gêmeos* e os conectamos por uma aresta $e_i^r \in E_S^r$. As arestas do conjunto E_S^r são denominadas arestas vermelhas.

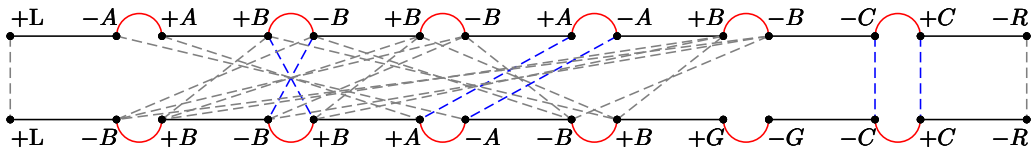


Figura 13: O grafo de adjacências de duas strings $S = (+L\ +A\ -B\ -B\ -A\ -B\ +C\ +R)$ e $P = (+L\ +B\ +B\ -A\ +B\ -G\ +C\ +R)$. Arestas pretas são representadas por linhas contínuas, arestas vermelhas por arcos contínuos e arestas cinzas por linhas pontilhadas. Alguns pares de arestas cinzas gêmeas então indicados em azul.

Dadas duas strings S e P, o *grafo de adjacências* $G(S, P) = (V, E)$ é formado pela união disjunta dos grafos parciais G_S e G_P, e pelo conjunto E_g de arestas cinzas. Para cada par de vértices $u \in V_S$ e $v \in V_P$, existe uma aresta cinza $\{u, v\}$ em E_g se esses vértices têm a mesma representação. Dizemos que duas arestas cinzas $\{u, v\}$ e $\{t, s\}$, com $u, t \in V_S$, são *gêmeas* se tanto u e t quanto v e s são gêmeos. A Figura 13 mostra um exemplo de grafo de adjacências.

Um *ciclo alternado* de um grafo de adjacência $G(S, P)$ é um ciclo formado por uma sequência de arestas $(e_0, e_1, e_2, \ldots, e_t)$, tal que e_i é preta se i for par e e_i é cinza ou vermelha se i for impar. Um *empacotamento de ciclos alternados* do grafo $G(S, P)$ é um conjunto de ciclos alternados disjuntos, tais que:

- Cada vértice pertence a exatamente um ciclo.
- Uma aresta cinza $\{u, v\}$ pertence a um ciclo se e somente se ela não tem uma aresta gêmea ou sua aresta gêmea também pertence a algum ciclo do empacotamento.
- A quantidade de arestas vermelhas de G_S conectando vértices $+\alpha$ e $-\alpha$ em algum ciclo é igual a $max(0, occ(\alpha, S) - occ(\alpha, P))$.
- A quantidade de arestas vermelhas de G_P conectando vértices $+\alpha$ e $-\alpha$ em algum ciclo é igual a $max(0, occ(\alpha, P) - occ(\alpha, S))$.

As duas últimas condições acima garantem que as arestas vermelhas correspondem a genes que devem ser removidos de S ou de P para tornar as duas strings balanceadas. Além disso, as duas primeiras condições garantem que o empacotamento de ciclos estabelece uma

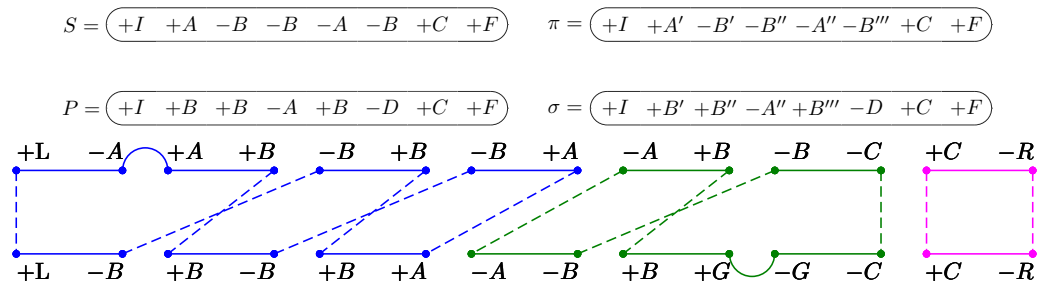


Figura 14: Um empacotamento de ciclos alternados do grafo de adjacências de duas strings S e P e um par de strings π e σ, sem caracteres repetidos, resultantes da correspondência entre os caracteres dada pelo empacotamento. O empacotamento tem três ciclos cada um indicado com uma cor.

correspondência um para um entre os caracteres das strings que não correspondem a arestas vermelhas. A Figura 14 apresenta um empacotamento de ciclos alternados do grafo da Figura 13 e a correspondência entre os caracteres das strings dada pelo empacotamento.

Empacotamentos com mais ciclos estão associados a correspondências que levam a menores distâncias de rearranjo [4]. Isso motiva o problema de *Empacotamento Máximo de Ciclos* que, dado um grafo de adjacências $G(S, P)$, busca por um empacotamento de ciclos com o maior número de ciclos possível.

3 Objetivos

Este projeto tem como propósito encontrar novas solução para as diferentes variações dos problemas de partição, assim como estabelecer novas relações entre esses problemas e os problemas de rearranjo de genomas. Para as versões dos problemas de partição e rearranjo que consideram regiões intergênicas ou genomas desbalanceados, vamos buscar por teoremas que estabeleçam relações similares às existentes entre os problemas de partição e rearranjo em strings balanceadas.

Tanto para os novos problemas de partição quanto para os que já têm relações conhecidas, vamos buscar por novos algoritmos. Quando possível, vamos descrever os algoritmos para as variações mais genéricas dos problemas e detalhar as adaptações necessárias para aplicá-los nas outras variações. Nos problemas de rearranjo, além de utilizar a relação com os

algoritmos de partição, vamos buscar por novas estratégias para obter mapeamentos das strings em permutações, de forma a melhorar a distância de rearranjo obtida.

4 Resultados Preliminares

Obtivemos resultados preliminares para alguns dos objetivos desta proposta. Esta seção apresenta uma breve descrição dos resultados publicados ou aceitos para publicação. No Anexo A, apresentamos testes relativos às soluções sendo exploradas atualmente.

4.1 Heurísticas Considerando Mapeamentos ou Empacotamento Máximo de Ciclos

O objetivo deste projeto relativo a encontrar estratégias para obter mapeamentos das strings em permutação é uma continuidade do trabalho realizado durante o mestrado do aluno Gabriel Henriques Siqueira [32]. No mestrado foram desenvolvidas heurísticas que exploram diretamente o espaço de possíveis mapeamentos de strings em permutações. Nessa busca, os mapeamentos são avaliados por um algoritmo para alguma distância de rearranjo entre permutações. As distâncias estudadas envolviam reversão e transposição. No início do doutorado, esses resultados obtidos no mestrado em conjunto com novos testes envolvendo genomas reais foram apresentados no artigo "*Heuristics for Genome Rearrangement Distance with Replicated Genes*" (Gabriel Siqueira, Klairton Lima Brito, Ulisses Dias e Zanoni Dias) [34], publicado na revista *IEEE/ACM Transactions on Computational Biology and Bioinformatics (2021)*.

Esse trabalho representava os mapeamentos de strings em permutações utilizando um vetor de permutações, cada permutação indica como os caracteres correspondentes a um rótulo devem ser mapeados. Uma forma alternativa de representar esses mapeamentos é utilizar um empacotamento de ciclos alternados do grafo de ciclos conforme apresentado na Seção 2.2. Utilizando essa representação, desenvolvemos heurísticas para estimar a distância de reversão em strings com sinais balanceadas. Esses resultados compõe o artigo "*Heuristics for Cycle Packing of Adjacency Graphs for Genomes with Repeated Genes*" (Gabriel

Siqueira, André Rodrigues Oliveira, Alexsandro Oliveira Alexandrino e Zanoni Dias) [35], apresentado na conferência *Brazilian Symposium on Bioinformatics (BSB'2021)*. Esse trabalho recebeu o prêmio de melhor artigo da conferência.

4.2 Aproximações para Problemas de Partição Envolvendo Regiões Intergênicas

Outros resultados preliminares envolveram os problemas de partição com regiões intergênicas. Considerando apenas pares de genomas balanceados, estabelecemos as seguintes relações entre problemas de rearranjo e os problemas de partição envolvendo regiões intergênicas:

- Uma aproximação com fator ℓ para o problema PISCMS garante uma aproximação de fator 4ℓ para a distância de rearranjo considerando o evento de reversão em genomas com sinais e uma aproximação de fator 4.5ℓ para a distância de rearranjo considerando os eventos de reversão e transposição em genomas com sinais.

- Uma aproximação com fator ℓ para o problema PIRCMS garante uma aproximação de fator 4ℓ para a distância de rearranjo considerando o evento de reversão em genomas sem sinais e uma aproximação de fator 4.5ℓ para a distância de rearranjo considerando os eventos de reversão e transposição em genomas sem sinais.

- Uma aproximação com fator ℓ para o problema PICMS garante uma aproximação assintótica de fator 3ℓ para a distância de rearranjo considerando o evento de transposição em genomas sem sinais.

Além disso, desenvolvemos algoritmos para os problemas PICMS, PIRCMS e PISCMS que garantem um fator de aproximação $2k$, onde k é o maior número de cópias de um caractere nas strings. Esse algoritmo também pode ser aplicado para os problemas de partição sem regiões intergênicas (PCMS, PRCMS e PSCMS) com o mesmo fator de aproximação, assim melhoramos os fatores conhecidos para os problemas ($4k$ para PCMS, $8k$ para PRCMS e $8k$ para PSCMS).

Os resultados para os problemas sem sinais foram apresentados no artigo "*Approximation Algorithm for Rearrangement Distances Considering Repeated Genes and Intergenic Regions*" (Gabriel Siqueira, Alexsandro Oliveira Alexandrino, André Rodrigues Oliveira e Zanoni Dias) [33], publicado na revista *Algorithms for Molecular Biology (2021)*. Os resultados com sinais foram aceitos na conferência *14th International Conference on Bioinformatics and Computational Biology (BICOB'2022)* no artigo "*Signed Rearrangement Distances Considering Repeated Genes and Intergenic Regions*" (Gabriel Siqueira, Alexsandro Oliveira Alexandrino e Zanoni Dias), a conferência será realizada entre os dias 21 e 23 de março de 2022.

5 Metodologia e Análise dos Resultados

Para cada um dos problemas, iniciaremos com um estudo teórico, desenvolvendo heurísticas, algoritmos exatos ou algoritmos que garantam um fator de aproximação. Em seguida, implementaremos os algoritmos descritos e realizaremos experimentos práticos para verificar seu desempenho.

Os testes serão realizados primariamente em bases de dados construídas artificialmente para avaliar diferentes características das soluções propostas. Quando possível utilizaremos bases já presentes na literatura, mas como estamos estudando problemas que ainda não foram explorados na literatura vamos precisar construir bases novas. Para os problemas com aplicação na comparação de genomas, pretendemos realizar alguns testes com genomas reais para validar o uso dos algoritmos desenvolvidos nesse cenário. Durante os testes experimentais, realizaremos, quando possível, uma comparação entre os algoritmos desenvolvidos e outros algoritmos existentes na literatura.

Apresentaremos os resultados obtidos durante este projeto na forma de artigos para publicação em anais de congressos e revistas das áreas de biologia computacional e teoria da computação.

6 Plano de Trabalho

A seguir apresentamos um cronograma com as atividades realizadas e previstas para o programa de doutorado do aluno Gabriel Henriques Siqueira iniciado em março de 2021. A Tabela 4 mostra o cronograma com uma breve descrição das atividades a serem realizadas.

Tabela 4: Cronograma das atividades.

Atividades	Semestres							
	2021/1	2021/2	2022/1	2022/2	2023/1	2023/2	2024/1	2024/2
1	X		X		X		X	
2		X	X					
3		X						
4			X					
5					X	X		
6	X	X						
7		X	X	X				
8			X	X	X			
9					X	X	X	
10							X	X
11								X

1. Revisão da literatura;
2. Participação no Programa de Estágio Docente (PED);
3. Escrita da proposta de doutorado;
4. Exame de Qualificação Específico (EQE);
5. Estágio de Pesquisa no Exterior;
6. Investigação dos problemas de partição e distância de rearranjo com e sem regiões intergênicas em genomas balanceados;
7. Investigação dos problemas de partição e distância de rearranjo com e sem regiões intergênicas em genomas não balanceados;
8. Investigação dos problemas de partição e distância de rearranjo com regiões intergênicas flexíveis em genomas balanceados;
9. Investigação dos problemas de partição e distância de rearranjo com regiões intergênicas flexíveis em genomas não balanceados;

10. Escrita e revisão da tese;

11. Defesa da tese.

A revisão bibliográfica será uma atividade constante que se intensificará em alguns momentos. Os créditos obrigatórios em disciplinas foram obtidos durante o mestrado e a graduação. Vale ressaltar que os tempos alocados em algumas atividades podem sofrer alterações no decorrer da pesquisa, uma vez que alguns resultados obtidos podem ser mais promissores que outros, fazendo com que mais tempo seja despendido em uma atividade em detrimento de outra.

Conforme previsto no cronograma, pretendemos realizar um estágio de pesquisa no exterior. Planejamos que o estágio seja realizado no *Laboratoire Des Sciences Du Numérique de Nantes* (LS2N) da Universidade de Nantes com o grupo *Combinatoire et Bioinformatique* (COMBI). Vários orientandos do prof. Dr. Zanoni Dias realizaram estágio com esse grupo: Carla Negri Lintzmayer (BEPE), Andre Rodrigues Oliveira (CAPES-COFECUB), Klairton Lima Brito (CAPES-COFECUB) e Alexsandro Oliveira Alexandrino (SWE-CNPq). Esses estágios geraram diversas publicações em conferências e revistas internacionais [7, 21, 22, 24, 25, 27].

Referências

[1] Alexsandro Oliveira Alexandrino, Klairton Lima Brito, Andre Rodrigues Oliveira, Ulisses Dias e Zanoni Dias. "Reversal Distance on Genomes with Different Gene Content and Intergenic Regions Information". Em: *Proceedings of the 8th International Conference on Algorithms for Computational Biology (AlCoB'2021)*. Cham: Springer International Publishing, 2021, pp. 121–133.

[2] Alexsandro Oliveira Alexandrino, Andre Rodrigues Oliveira, Ulisses Dias e Zanoni Dias. "Genome Rearrangement Distance with Reversals, Transpositions, and Indels". Em: *Journal of Computational Biology* 28.3 (2021), pp. 235–247.

[3] Alexsandro Oliveira Alexandrino, Andre Rodrigues Oliveira, Ulisses Dias e Zanoni Dias. “Incorporating intergenic regions into reversal and transposition distances with indels”. Em: *Journal of Bioinformatics and Computational Biology* 19.06 (2021), p. 2140011.

[4] Vineet Bafna e Pavel A. Pevzner. “Genome Rearrangements and Sorting by Reversals”. Em: *SIAM Journal on Computing* 25.2 (1996), pp. 272–289.

[5] Piotr Berman, Sridhar Hannenhalli e Marek Karpinski. “1.375-Approximation Algorithm for Sorting by Reversals”. Em: *Proceedings of the 10th Annual European Symposium on Algorithms (ESA'2002)*. London, UK: Springer-Verlag, 2002, pp. 200–210.

[6] Klairton L. Brito, Andre R. Oliveira, Alexsandro O. Alexandrino, Ulisses Dias e Zanoni Dias. “An improved approximation algorithm for the reversal and transposition distance considering gene order and intergenic sizes”. Em: *Algorithms for Molecular Biology* 16.1 (2021), pp. 1–21.

[7] Klairton Lima Brito, Géraldine Jean, Guillaume Fertin, Andre Rodrigues Oliveira, Ulisses Dias e Zanoni Dias. “Sorting by Genome Rearrangements on Both Gene Order and Intergenic Sizes”. Em: *Journal of Computational Biology* 27.2 (2020), pp. 156–174.

[8] Laurent Bulteau, Guillaume Fertin, Géraldine Jean e Christian Komusiewicz. “Sorting by Multi-Cut Rearrangements”. Em: *Algorithms* 14.6 (2021), p. 169.

[9] Laurent Bulteau, Guillaume Fertin, Christian Komusiewicz e Irena Rusu. “A Fixed-Parameter Algorithm for Minimum Common String Partition with Few Duplications”. Em: *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2013, pp. 244–258.

[10] Laurent Bulteau e Christian Komusiewicz. “Minimum Common String Partition Parameterized by Partition Size Is Fixed-Parameter Tractable”. Em: *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'2013)*. Berlin, Heidelberg: Society for Industrial e Applied Mathematics, 2013, pp. 244–258.

[11] Xin Chen. “On sorting unsigned permutations by double-cut-and-joins”. Em: *Journal of Combinatorial Optimization* 25.3 (2010), pp. 339–351.

[12] Xin Chen, Jie Zheng, Zheng Fu, Peng Nan, Yang Zhong, Stefano Lonardi e Tao Jiang. “Assignment of Orthologous Genes via Genome Rearrangement”. Em: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 2.4 (2005), pp. 302–315.

[13] David A. Christie e Robert W. Irving. “Sorting Strings by Reversals and by Transpositions”. Em: *SIAM Journal on Discrete Mathematics* 14.2 (2001), pp. 193–206.

[14] Graham Cormode e S. Muthukrishnan. “The string edit distance matching problem with moves”. Em: *ACM Transactions on Algorithms* 3.1 (2007), pp. 1–19.

[15] Isaac Elias e Tzvika Hartman. “A 1.375-Approximation Algorithm for Sorting by Transpositions”. Em: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 3.4 (2006), pp. 369–379.

[16] Avraham Goldstein, Petr Kolman e Jie Zheng. “Minimum Common String Partition Problem: Hardness and Approximations”. Em: *Proceedings of the 15th International Symposium on Algorithms and Computation (ISAAC’2004)*. Berlin, Heidelberg, 2005, pp. 484–495.

[17] Sridhar Hannenhalli e Pavel A. Pevzner. “Transforming cabbage into turnip. polynomial algorithm for sorting signed permutations by reversals”. Em: *Journal of ACM* 46.1 (1999), pp. 1–27.

[18] Haitao Jiang, Binhai Zhu, Daming Zhu e Hong Zhu. “Minimum common string partition revisited”. Em: *Journal of Combinatorial Optimization* 23.4 (2010), pp. 519–527.

[19] Petr Kolman e Tomasz Waleń. “Reversal Distance for Strings with Duplicates: Linear Time Approximation Using Hitting Set”. Em: *Proceedings of the 4th International Workshop on Approximation and Online Algorithms (WAOA’2006)*. Vol. 4368. Lecture Notes in Computer Science. Berlin, Heidelberg, 2007, pp. 279–289.

[20] Manuel Lafond e Binhai Zhu. "Permutation-Constrained Common String Partitions with Applications". Em: *String Processing and Information Retrieval (SPIRE'2021)*. Cham: Springer International Publishing, 2021, pp. 47–60.

[21] Carla Negri Lintzmayer, Guillaume Fertin e Zanoni Dias. "Sorting permutations and binary strings by length-weighted rearrangements". Em: *Theoretical Computer Science* 715 (2018), pp. 35–59.

[22] Andre R. Oliveira, Géraldine Jean, Guillaume Fertin, Ulisses Dias e Zanoni Dias. "Super short operations on both gene order and intergenic sizes". Em: *Algorithms for Molecular Biology* 14.1 (2019), pp. 1–17.

[23] Andre Rodrigues Oliveira, Klairton Lima Brito, Ulisses Dias e Zanoni Dias. "On the Complexity of Sorting by Reversals and Transpositions Problems". Em: *Journal of Computational Biology* 26.11 (2019), pp. 1223–1229.

[24] Andre Rodrigues Oliveira, Geraldine Jean, Guillaume Fertin, Klairton Lima Brito, Laurent Bulteau, Ulisses Dias e Zanoni Dias. "Sorting Signed Permutations by Intergenic Reversals". Em: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 18.6 (2021), pp. 2870–2876.

[25] Andre Rodrigues Oliveira, Geraldine Jean, Guillaume Fertin, Klairton Lima Brito, Ulisses Dias e Zanoni Dias. "Sorting Permutations by Intergenic Operations". Em: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 18.6 (2021), pp. 2080–2093.

[26] Andre Rodrigues Oliveira, Géraldine Jean, Guillaume Fertin, Klairton Lima Brito, Ulisses Dias e Zanoni Dias. "A 3.5-Approximation Algorithm for Sorting by Intergenic Transpositions". Em: *Proceedings of the 7th International Conference on Algorithms for Computational Biology (AlCoB'2020)*. Springer International Publishing, 2020, pp. 16–28.

[27] Andre Rodrigues Oliveira, Géraldine Jean, Guillaume Fertin, Ulisses Dias e Zanoni Dias. "Super Short Reversals on Both Gene Order and Intergenic Sizes". Em: *Proce-*

edings of the 11th Brazilian Symposium on Bioinformatics (BSB'2018). Heidelberg, Germany: Springer International Publishing, 2018, pp. 14–25.

[28] Andrew J. Radcliffe, Alex D. Scott e Elizabeth Wilmer. "Reversals and Transpositions Over Finite Alphabets". Em: *SIAM Journal on Discrete Mathematics* 19.1 (2005), pp. 224–244.

[29] Atif Rahman, Swakkhar Shatabda e Masud Hasan. "An approximation algorithm for sorting by reversals and transpositions". Em: *Journal of Discrete Algorithms* 6.3 (2008), pp. 449–457.

[30] Diego P. Rubert, Pedro Feijão, Marília Dias Vieira Braga, Jens Stoye e Fábio Henrique Viduani Martinez. "Approximating the DCJ distance of balanced genomes in linear time". Em: *Algorithms for Molecular Biology* 12.1 (2017), p. 3.

[31] Dana Shapira e James A. Storer. "Edit distance with move operations". Em: *Journal of Discrete Algorithms* 5.2 (2007), pp. 380–392.

[32] Gabriel Siqueira. "Heurísticas para Problemas de Rearranjo de Genomas com Genes Multiplicados". Dissertação de mestrado. Instituto de Computação, Universidade Estadual de Campinas, 2021.

[33] Gabriel Siqueira, Alexsandro Oliveira Alexandrino, Andre Rodrigues Oliveira e Zanoni Dias. "Approximation algorithm for rearrangement distances considering repeated genes and intergenic regions". Em: *Algorithms for Molecular Biology* 16.1 (2021), p. 21.

[34] Gabriel Siqueira, Klairton Lima Brito, Ulisses Dias e Zanoni Dias. "Heuristics for Genome Rearrangement Distance with Replicated Genes". Em: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 18.6 (2021), pp. 2094–2108.

[35] Gabriel Siqueira, Andre Rodrigues Oliveira, Alexsandro Oliveira Alexandrino e Zanoni Dias. "Heuristics for Cycle Packing of Adjacency Graphs for Genomes with Repeated Genes". Em: *Proceedings of the 14th Brazilian Symposium on Bioinformatics (BSB'2021)*. Cham: Springer International Publishing, 2021, pp. 93–105.

[36] Krister M. Swenson, Mark Marron, Joel V. Earnest-Deyoung e Bernard M. E. Moret. "Approximating the true evolutionary distance between two genomes". Em: *ACM Journal of Experimental Algorithmics* 12 (2008), pp. 1–17.

[37] Maria Emília M.T. Walter, Zanoni Dias e João Meidanis. "Reversal and transposition distance of linear chromosomes". Em: *Proceedings of the String Processing and Information Retrieval: A South American Symposium (SPIRE'1998)*. Los Alamitos, CA, USA, 1998, pp. 96–102.

[38] Eyla Willing, Jens Stoye e Marilia Braga. "Computing the Inversion-Indel Distance". Em: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 18.06 (2021), pp. 2314–2326.

A Resultados com Genomas Desbalanceados

Os resultados já publicados apresentados na Seção 4 consideram apenas genomas balanceados. Recentemente, utilizando as definições apresentadas nesta proposta, adaptados nossos algoritmos para lidar com genomas quaisquer. Inicialmente testamos as adaptações em bases de dados geradas artificialmente com genomas com sinais. Esta seção apresenta os resultados preliminares obtidos. Os experimentos foram executados em uma maquina equipada com um processador Intel® Xeon® CPU E5-2470 v2 de 2.3GHz, com 40 cores e com 32 GB de memória RAM. O sistema operacional usado foi Ubuntu 18.04.2.

A.1 Base de Dados

Construímos bases de dados contendo strings (bases sem regiões intergênicas) ou string e listas de inteiros (bases com regiões intergênicas) para testar os algoritmos desenvolvidos. Nessas bases, as strings têm tamanho 100 (nos algoritmos são adicionados dois caracteres nas extremidades o que gera strings de tamanho 102) e as listas de inteiros têm tamanho 101. As bases foram geradas a partir de um parâmetro $O \in \{25, 50, 75, 100\}$, que controla o número de operações aplicadas, e de um parâmetro $L \in \{10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$ que controla o número de caracteres distintos usados para gerar as strings. As seguintes bases de dados foram construídas para testar os algoritmos desenvolvidos:

- R_O^L: Para a construção dessas bases geramos uma string S escolhendo 100 caracteres de um alfabeto de tamanho L. Em seguida transformamos S em uma nova string P aplicando O operações de reversão, seguidas de $\lfloor \frac{O}{2} \rfloor$ operações de deleção de um único caractere, seguidas de $\lfloor \frac{O}{2} \rfloor$ operações de inserção de um único caractere. Os índices das operações são escolhidos aleatoriamente, com uma distribuição uniforme, e cada caractere inserido tem um rótulo que ainda não esteja presente no alfabeto da string.

- $\breve{R}_O^L$: Essas bases são construídas por um processo similar ao usado nas bases R_O^L, mas além da string S foi gerada também um lista de inteiros $\breve{S}$. Para a lista $\breve{S}$ foram escolhidos aleatoriamente, com uma distribuição uniforme, 101 inteiros no intervalo

$[0, 100]$. Além disso as operações aplicadas em P envolvem também a lista $\breve{S}$, o que produz uma lista $\breve{P}$.

A.2 Algoritmos

Para obter os resultados apresentados nessa seção, construímos partições com sinais para cada par de strings. Para isso, utilizamos uma generalização de um algoritmo para o problema de partição entre genomas balanceados [33], que utiliza a definição de partição para genomas quaisquer apresentada na Seção 2.1. Em seguida, geramos mapeamentos aleatórios entre os genes multiplicados dos genomas origem e destino. Geramos 100 mapeamentos considerando as partições geradas e 100 mapeamentos sem considerar as partições. Com os mapeamentos gerados, utilizamos um algoritmo que aproxima a distância de reversão no caso onde os genomas não têm genes multiplicados [2]. A distância reportada no final é a menor distância encontrada entre todos os mapeamentos. Chamamos essa estratégia de Mapeamentos Aleatórios.

Comparamos as distâncias obtidas pelas correspondências com e sem a aplicação do algoritmo de partição. Para as bases de dados sem regiões intergênicas aplicamos os mesmos algoritmos considerando que as regiões intergênicas são vazias (tamanho 0). Além disso, nesse caso, utilizamos também algoritmos baseados em Empacotamento Máximo de Ciclos como uma forma alternativa de obter as correspondências entre os genes. Os algoritmos usados foram adaptações das heurísticas de Empacotamentos Aleatórios e Algoritmo Genético previamente apresentados [35]. Essas adaptações consideram a definição de grafos de ciclos para strings quaisquer apresentada na Seção 2.2.

O algoritmo para o problema de partição foi implementado em Haskell, o algoritmo para calcular a distância de reversão foi implementado em Python e as heurísticas baseadas em Empacotamento Máximo de Ciclos foram implementadas em C++.

A.3 Resultados sem Regiões Intergênicas

As tabelas 5 até 12 apresentam as distâncias e tempos de execução médios para as bases de dados que não consideram regiões intergênicas. Podemos ver que, na maior parte dos

conjuntos de strings, ambas as heurísticas baseadas em Empacotamento Máximo de Ciclos encontraram em média distâncias menores com um tempo de execução também menor se comparadas com a estratégia de Mapeamentos Aleatórios. A exceção são os conjuntos criados com $O = 25$. Como podemos observar na Tabela 5, para esses conjuntos a heurística baseada em mapeamentos obtém distâncias menores no caso onde utilizamos o algoritmo de partição.

Olhando para o efeito do algoritmo de partição, vemos que, em todos os casos, aplicar o algoritmo de partição antes de calcular as distâncias causa uma melhora significativa. Além disso, em geral, os tempos de execução também melhoram na maior parte dos casos. Nos casos onde o tempo de execução com o algoritmo de partição é maior, a diferença não passa de 2 segundos.

É interessante notar que, com o aumento de O, a diferença entre os resultados com e sem o uso do algoritmo de partição diminui. O mesmo ocorre com o aumento de L. Já a diferença entre as distâncias obtidas pelas heurísticas baseadas em Empacotamento Máximo de Ciclos e a heurística Mapeamentos Aleatórios aumenta com a aumento de O.

Comparando a heurística de Algoritmo Genético (AG) com a heurística de Empacotamentos Aleatórios (EA), vemos que a heurística AG obtém em média distâncias menores com um tempo de execução menor. Essa diferença entre as distâncias é maior nos casos onde o algoritmo de partição não foi utilizado.

Tabela 5: Média das distâncias, para os conjuntos sem regiões intergênicas gerados com $O = 25$, calculadas usando as heurísticas Algoritmo Genético (AG), Empacotamentos Aleatórios (EA) e Mapeamentos Aleatórios (MAP). As colunas SP e SMCISP indicam os resultados sem e com o uso do algoritmo de partição, respectivamente. A coluna DIF mostra a diferença entre os valores das colunas SP e SMCISP.

L	AG			EA			MAP		
	SP	SMCISP	DIF	SP	SMCISP	DIF	SP	SMCISP	DIF
10	87.05	68.19	18.86	88.98	68.73	20.25	95.69	68.59	27.10
20	84.22	68.29	15.93	88.92	68.74	20.18	94.11	67.86	26.25
30	81.98	67.33	14.65	86.86	67.78	19.08	92.07	66.62	25.45
40	81.10	66.75	14.35	85.54	67.23	18.31	90.16	65.85	24.31
50	79.95	66.07	13.88	83.85	66.27	17.58	88.30	64.89	23.41
60	78.94	65.18	13.76	81.95	65.55	16.40	86.59	64.14	22.45
70	78.24	64.88	13.36	80.59	65.14	15.45	84.95	63.59	21.36
80	78.30	64.62	13.68	80.23	64.80	15.43	84.16	63.34	20.82
90	78.33	64.70	13.63	79.69	64.90	14.79	83.37	63.38	19.99
100	78.04	64.17	13.87	79.28	64.28	15.00	82.48	62.90	19.58

Tabela 6: Média dos tempos de execução (em segundos), para os conjuntos sem regiões intergênicas gerados com $O = 25$, das heurísticas Algoritmo Genético (AG), Empacotamentos Aleatórios (EA) e Mapeamentos Aleatórios (MAP). As colunas SP e SMCISP indicam os resultados sem e com o uso do algoritmo de partição, respectivamente. A coluna DIF mostra a diferença entre os valores das colunas SP e SMCISP.

L	AG			EA			MAP		
	SP	SMCISP	DIF	SP	SMCISP	DIF	SP	SMCISP	DIF
10	4.15	2.69	1.46	13.57	7.44	6.13	19.83	12.89	6.94
20	3.37	2.46	0.91	12.36	6.71	5.65	19.57	13.25	6.32
30	2.97	2.31	0.66	11.84	6.09	5.75	19.18	13.20	5.98
40	2.75	2.21	0.54	11.60	5.69	5.91	18.68	13.14	5.54
50	2.57	2.13	0.44	11.24	5.27	5.97	19.09	12.76	6.33
60	2.44	2.09	0.35	10.91	4.96	5.95	18.77	12.76	6.01
70	2.33	2.02	0.31	10.59	4.66	5.93	19.03	12.60	6.43
80	2.21	2.08	0.13	10.19	4.49	5.70	18.76	12.67	6.09
90	2.12	2.04	0.08	9.73	4.30	5.43	18.86	12.49	6.37
100	2.06	1.97	0.09	9.39	4.01	5.38	18.67	12.43	6.24

Tabela 7: Média das distâncias, para os conjuntos sem regiões intergênicas gerados com $O = 50$, calculadas usando as heurísticas Algoritmo Genético (AG), Empacotamentos Aleatórios (EA) e Mapeamentos Aleatórios (MAP). As colunas SP e SMCISP indicam os resultados sem e com o uso do algoritmo de partição, respectivamente. A coluna DIF mostra a diferença entre os valores das colunas SP e SMCISP.

L	AG			EA			MAP		
	SP	SMCISP	DIF	SP	SMCISP	DIF	SP	SMCISP	DIF
10	84.80	70.85	13.95	85.16	71.07	14.09	92.36	77.05	15.31
20	84.45	75.86	8.59	85.92	76.69	9.23	91.56	81.32	10.24
30	84.06	77.36	6.70	85.81	78.21	7.60	90.75	82.19	8.56
40	83.96	77.94	6.02	85.31	78.63	6.68	89.79	82.38	7.41
50	83.36	77.91	5.45	84.62	78.58	6.04	89.03	81.99	7.04
60	82.90	77.65	5.25	84.07	78.20	5.87	88.21	81.80	6.41
70	82.63	77.54	5.09	83.84	78.24	5.60	87.67	81.49	6.18
80	82.88	77.83	5.05	83.55	78.38	5.17	87.48	81.56	5.92
90	82.19	77.24	4.95	83.08	77.92	5.16	86.65	81.01	5.64
100	82.10	77.37	4.73	82.84	77.91	4.93	86.52	81.02	5.50

Tabela 8: Média dos tempos de execução (em segundos), para os conjuntos sem regiões intergênicas gerados com $O = 50$, das heurísticas Algoritmo Genético (AG), Empacotamentos Aleatórios (EA) e Mapeamentos Aleatórios (MAP). As colunas SP e SMCISP indicam os resultados sem e com o uso do algoritmo de partição, respectivamente. A coluna DIF mostra a diferença entre os valores das colunas SP e SMCISP.

L	AG			EA			MAP		
	SP	SMCISP	DIF	SP	SMCISP	DIF	SP	SMCISP	DIF
10	4.74	3.80	0.94	17.25	11.93	5.32	18.72	15.24	3.48
20	3.89	3.69	0.20	15.22	11.93	3.29	18.65	17.05	1.60
30	3.52	3.52	0.00	14.20	11.51	2.69	18.01	17.34	0.67
40	3.20	3.34	-0.14	13.57	11.07	2.50	18.70	17.72	0.98
50	3.00	3.21	-0.21	12.92	10.66	2.26	18.00	17.76	0.24
60	2.80	3.05	-0.25	12.50	10.14	2.36	17.82	17.64	0.18
70	2.65	2.94	-0.29	11.93	9.65	2.28	18.44	17.72	0.72
80	2.55	2.89	-0.34	11.65	9.39	2.26	17.88	17.75	0.13
90	2.47	2.81	-0.34	11.17	9.01	2.16	18.15	17.73	0.42
100	2.40	2.74	-0.34	10.83	8.83	2.00	18.09	17.65	0.44

Tabela 9: Média das distâncias, para os conjuntos sem regiões intergênicas gerados com $O = 75$, calculadas usando as heurísticas Algoritmo Genético (AG), Empacotamentos Aleatórios (EA) e Mapeamentos Aleatórios (MAP). As colunas SP e SMCISP indicam os resultados sem e com o uso do algoritmo de partição, respectivamente. A coluna DIF mostra a diferença entre os valores das colunas SP e SMCISP.

L	AG			EA			MAP		
	SP	SMCISP	DIF	SP	SMCISP	DIF	SP	SMCISP	DIF
10	72.06	59.88	12.18	72.01	60.18	11.83	85.81	75.24	10.57
20	72.14	66.38	5.76	72.77	66.98	5.79	85.54	80.22	5.32
30	72.40	68.52	3.88	72.90	69.03	3.87	85.05	81.38	3.67
40	72.03	68.94	3.09	72.89	69.90	2.99	84.73	81.82	2.91
50	71.84	69.33	2.51	72.73	70.09	2.64	84.35	81.80	2.55
60	71.75	69.40	2.35	72.51	69.94	2.57	84.11	81.80	2.31
70	71.76	69.58	2.18	72.80	70.15	2.65	83.98	81.90	2.08
80	71.51	69.58	1.93	72.29	70.32	1.97	83.75	81.84	1.91
90	71.72	69.88	1.84	72.55	70.23	2.32	83.52	81.72	1.80
100	71.66	69.59	2.07	72.29	70.21	2.08	83.23	81.52	1.71

Tabela 10: Média dos tempos de execução (em segundos), para os conjuntos sem regiões intergênicas gerados com $O = 75$, das heurísticas Algoritmo Genético (AG), Empacotamentos Aleatórios (EA) e Mapeamentos Aleatórios (MAP). As colunas SP e SMCISP indicam os resultados sem e com o uso do algoritmo de partição, respectivamente. A coluna DIF mostra a diferença entre os valores das colunas SP e SMCISP.

L	AG			EA			MAP		
	SP	SMCISP	DIF	SP	SMCISP	DIF	SP	SMCISP	DIF
10	5.03	4.49	0.54	19.78	14.44	5.34	16.36	14.72	1.64
20	4.09	4.38	-0.29	17.03	14.76	2.27	17.24	16.52	0.72
30	3.57	4.11	-0.54	15.37	14.18	1.19	16.94	17.08	-0.14
40	3.29	3.90	-0.61	14.26	14.01	0.25	16.99	17.33	-0.34
50	3.00	3.67	-0.67	13.42	13.07	0.35	16.86	17.47	-0.61
60	2.84	3.51	-0.67	12.84	12.32	0.52	17.25	17.42	-0.17
70	2.67	3.43	-0.76	12.27	11.95	0.32	17.11	17.76	-0.65
80	2.50	3.29	-0.79	11.61	11.51	0.10	16.42	17.43	-1.01
90	2.42	3.25	-0.83	11.17	10.88	0.29	16.90	17.63	-0.73
100	2.30	3.11	-0.81	10.80	10.53	0.27	17.26	17.54	-0.28

Tabela 11: Média das distâncias, para os conjuntos sem regiões intergênicas gerados com $O = 100$, calculadas usando as heurísticas Algoritmo Genético (AG), Empacotamentos Aleatórios (EA) e Mapeamentos Aleatórios (MAP). As colunas SP e SMCISP indicam os resultados sem e com o uso do algoritmo de partição, respectivamente. A coluna DIF mostra a diferença entre os valores das colunas SP e SMCISP.

L	AG			EA			MAP		
	SP	SMCISP	DIF	SP	SMCISP	DIF	SP	SMCISP	DIF
10	52.64	43.54	9.10	52.52	43.33	9.19	75.32	68.25	7.07
20	52.35	48.89	3.46	53.24	49.35	3.89	74.99	72.04	2.95
30	52.78	50.64	2.14	53.46	51.36	2.10	74.96	73.20	1.76
40	52.39	51.01	1.38	53.35	51.50	1.85	74.90	73.64	1.26
50	52.52	51.66	0.86	53.26	52.29	0.97	74.83	73.86	0.97
60	52.71	51.74	0.97	53.60	52.37	1.23	74.64	73.72	0.92
70	52.48	51.53	0.95	53.31	52.54	0.77	74.59	73.82	0.77
80	52.85	51.84	1.01	53.57	52.56	1.01	74.63	73.92	0.71
90	53.08	52.27	0.81	53.78	53.05	0.73	74.77	74.18	0.59
100	52.95	51.95	1.00	53.31	52.74	0.57	74.47	73.89	0.58

Tabela 12: Média dos tempos de execução (em segundos), para os conjuntos sem regiões intergênicas gerados com $O = 100$, das heurísticas Algoritmo Genético (AG), Empacotamentos Aleatórios (EA) e Mapeamentos Aleatórios (MAP). As colunas SP e SMCISP indicam os resultados sem e com o uso do algoritmo de partição, respectivamente. A coluna DIF mostra a diferença entre os valores das colunas SP e SMCISP.

L	AG			EA			MAP		
	SP	SMCISP	DIF	SP	SMCISP	DIF	SP	SMCISP	DIF
10	5.07	4.83	0.24	20.55	16.43	4.12	14.38	13.38	1.00
20	4.07	4.70	-0.63	17.40	16.36	1.04	14.47	14.50	-0.03
30	3.52	4.40	-0.88	15.73	15.64	0.09	14.60	14.89	-0.29
40	3.13	4.06	-0.93	14.35	14.54	-0.19	14.62	15.21	-0.59
50	2.86	3.83	-0.97	13.39	13.79	-0.40	14.41	15.29	-0.88
60	2.69	3.64	-0.95	12.73	12.98	-0.25	14.46	15.81	-1.35
70	2.51	3.51	-1.00	12.07	12.52	-0.45	14.51	15.34	-0.83
80	2.36	3.39	-1.03	11.45	11.97	-0.52	14.41	15.31	-0.90
90	2.30	3.30	-1.00	10.99	11.48	-0.49	14.58	15.30	-0.72
100	2.25	3.17	-0.92	10.49	11.05	-0.56	14.53	15.42	-0.89

A.4 Resultados com Regiões Intergênicas

As tabelas 13 até 16 apresentam as distâncias e tempos de execução médios para as bases de dados que consideram regiões intergênicas. Similar ao caso sem regiões intergênicas, o uso do algoritmo de partição gerou distâncias em média menores. Na maior parte dos casos, o tempo de execução também foi menor com o uso do algoritmo de partição. A exceção são os conjuntos criados com $O = 100$, e com $O = 75$ e $L = 50$. Nesses casos, o tempo de execução foi em média menos de um segundo menor. Podemos notar que a diferença das distâncias obtidas com e sem o algoritmo de partição diminui com o aumento do valor de O, assim como com o aumento do valor de L.

Tabela 13: Média das distâncias e dos tempos de execução (em segundos), para os conjuntos com regiões intergênicas gerados com $O = 25$, utilizado a heurística Mapeamentos Aleatórios (MAP). As colunas SP e SMCISP indicam os resultados sem e com o uso do algoritmo de partição, respectivamente. A coluna DIF mostra a diferença entre os valores das colunas SP e SMCISP.

L	MAP Distance			MAP Time		
	SP	SMCISP	DIF	SP	SMCISP	DIF
10	111.50	80.28	31.22	22.24	15.16	7.08
20	109.56	79.00	30.56	22.04	15.34	6.7
30	107.26	77.66	29.6	22.33	15.47	6.86
40	104.98	76.17	28.81	22.27	15.08	7.19
50	102.84	75.10	27.74	21.83	14.93	6.9
60	100.85	74.47	26.38	21.65	15.02	6.63
70	99.17	73.59	25.58	21.80	14.60	7.2
80	98.11	73.22	24.89	21.92	14.70	7.22
90	96.85	73.01	23.84	21.43	14.47	6.96
100	95.40	72.19	23.21	21.40	14.25	7.15

Tabela 14: Média das distâncias e dos tempos de execução (em segundos), para os conjuntos com regiões intergênicas gerados com $O = 50$, utilizado a heurística Mapeamentos Aleatórios (MAP). As colunas SP e SMCISP indicam os resultados sem e com o uso do algoritmo de partição, respectivamente. A coluna DIF mostra a diferença entre os valores das colunas SP e SMCISP.

L	MAP Distance			MAP Time		
	SP	SMCISP	DIF	SP	SMCISP	DIF
10	105.11	95.77	9.34	20.70	18.36	2.34
20	104.27	95.86	8.41	21.37	18.48	2.89
30	103.28	95.38	7.90	20.46	18.70	1.76
40	102.34	94.85	7.49	20.35	18.42	1.93
50	101.42	93.94	7.48	21.10	18.60	2.50
60	100.53	93.47	7.06	20.65	18.24	2.41
70	99.83	92.66	7.17	20.57	18.39	2.18
80	99.27	92.46	6.81	20.22	18.22	2.00
90	98.72	91.96	6.76	20.18	18.25	1.93
100	98.44	92.09	6.35	20.57	18.17	2.40

Tabela 15: Média das distâncias e dos tempos de execução (em segundos), para os conjuntos com regiões intergênicas gerados com $O = 75$, utilizado a heurística Mapeamentos Aleatórios (MAP). As colunas SP e SMCISP indicam os resultados sem e com o uso do algoritmo de partição, respectivamente. A coluna DIF mostra a diferença entre os valores das colunas SP e SMCISP.

L	MAP Distance			MAP Time		
	SP	SMCISP	DIF	SP	SMCISP	DIF
10	95.11	91.81	3.30	18.08	17.72	0.36
20	94.66	92.06	2.60	18.83	17.72	1.11
30	94.30	91.80	2.50	18.83	18.04	0.79
40	94.04	91.80	2.24	19.22	18.08	1.14
50	93.42	91.29	2.13	18.19	18.29	-0.10
60	93.09	91.04	2.05	18.41	18.08	0.33
70	92.99	91.09	1.90	19.12	18.18	0.94
80	92.68	90.86	1.82	18.90	18.16	0.74
90	92.41	90.61	1.80	18.50	17.70	0.80
100	92.25	90.40	1.85	19.03	17.75	1.28

Tabela 16: Média das distâncias e dos tempos de execução (em segundos), para os conjuntos com regiões intergênicas gerados com $O = 100$, utilizado a heurística Mapeamentos Aleatórios (MAP). As colunas SP e SMCISP indicam os resultados sem e com o uso do algoritmo de partição, respectivamente. A coluna DIF mostra a diferença entre os valores das colunas SP e SMCISP.

L	MAP Distance			MAP Time		
	SP	SMCISP	DIF	SP	SMCISP	DIF
10	81.17	80.11	1.06	15.98	16.39	-0.41
20	81.02	80.20	0.82	16.04	16.34	-0.30
30	80.85	80.13	0.72	15.93	16.27	-0.34
40	80.69	80.06	0.63	15.76	16.41	-0.65
50	80.72	80.17	0.55	16.19	16.23	-0.04
60	80.78	80.25	0.53	15.81	16.31	-0.50
70	80.56	80.15	0.41	16.01	16.09	-0.08
80	80.45	79.93	0.52	15.84	16.40	-0.56
90	80.25	79.86	0.39	15.79	16.45	-0.66
100	80.29	79.73	0.56	15.94	16.64	-0.70