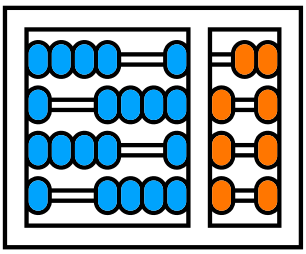
**Universidade Estadual de Campinas**

Instituto de Computação



**Exame de Qualificação Específico**

---

# Variações do Problema de Distância de Rearranjos

---

*Aluno*: Alexsandro Oliveira Alexandrino

*Orientador*: Prof. Dr. Zanoni Dias

*Coorientador*: Prof. Dr. Ulisses Dias

# Sumário

**Resumo**

Na genômica comparativa, uma forma de estimar a distância evolucionária consiste em encontrar uma sequência de custo mínimo de rearranjos de genomas (mutações que afetam um trecho do genoma) que transforma um genoma no outro. O custo dessa sequência é chamado de distância de rearranjos. Dentre os rearranjos mais estudados, podemos citar as reversões, transposições, transposições inversas, revrevs, inserções e deleções. Os primeiros trabalhos da área modelaram os genomas como sequências de genes e trataram a distância de rearranjos como o número mínimo de rearranjos que transforma um genoma em outro. Estudos indicam que alguns rearranjos ocorrem com maior frequência do que outros em algumas espécies, fazendo com que abordagens ponderadas sejam mais realistas. Dentre as funções de custo mais estudadas, temos a que associa diferentes custos dependendo do tipo de rearranjo e a que relaciona o custo de um rearranjo ao tamanho da região afetada por ele. Durante muito tempo, apenas a ordem relativa dos genes de cada genoma foi utilizada na comparação de genomas. Recentemente, estudos mostraram que incorporar o tamanho das regiões intergênicas (material genético entre cada par de genes) pode trazer melhores estimativas para a distância usando genomas reais. Dessa forma, uma das variações do problema de distância de rearranjo a ser estudada considera tanto a ordem relativa dos genes quanto as regiões intergênicas na representação de um genoma. Pretendemos incorporar elementos genômicos distintos em cada variação do problema a fim de torná-los mais relevantes do ponto de vista biológico. As variações consideradas são: (i) estudo da complexidade do problema de distância de rearranjos quando reversões, transposições, transposições inversas e revrevs são os rearranjos permitidos; (ii) estudo do problema de distância de rearranjos com uma restrição no tamanho permitido dos rearranjos e utilizando uma função de custo proporcional ao tamanho do rearranjo; (iii) estudo dos problemas de distância de rearranjos de genomas com reversões, transposições, inserções e deleções de material genético; (iv) estudo dos problemas de distância de rearranjos de genomas com reversões e transposições, além das inserções e deleções de material genético, considerando regiões intergênicas. Para (i) e (ii), consideramos que os genomas a serem comparados possuem o mesmo conjunto de genes e que não existem genes repetidos. Para (iii) e (iv), consideramos genomas sem genes repetidos e genomas com genes repetidos, separadamente.

# 1 Introdução

Um ***rearranjo de genoma*** é uma mutação que altera trechos de um genoma. Na área de genômica comparativa, a distância evolucionária entre espécies é estimada usando a ***distância de rearranjos de genoma***. No cálculo dessa distância, a ocorrência de um rearranjo está associada a um custo que pode simplesmente indicar essa ocorrência (custo unitário) ou pode indicar características do rearranjo. Dados dois genomas $\mathcal{G}_1$ e $\mathcal{G}_2$, a ***distância de rearranjos*** entre $\mathcal{G}_1$ e $\mathcal{G}_2$ é o menor custo possível para uma sequência de rearranjos transformar o genoma $\mathcal{G}_1$ no genoma $\mathcal{G}_2$.

Na genômica comparativa um genoma é usualmente representado como uma sequência de genes e, dependendo da informação genômica disponível, diferentes modelos matemáticos podem ser utilizados. Assumindo que não existem genes repetidos e que os genomas possuem o mesmo conjunto de genes, um genoma pode ser modelado como uma permutação de números inteiros, onde cada elemento representa um gene. Quando a orientação dos genes é conhecida, essa informação é representada pelo sinal (positivo ou negativo) dos elementos da permutação. Quando a orientação dos genes não é conhecida, permutações sem sinais são utilizadas para representação dos genomas. Ao utilizar permutações, o problema de transformar um genoma em outro é equivalente ao problema de ordenação de permutações [25].

Um ***modelo de rearranjo*** define o conjunto de operações (rearranjos de genoma) permitidas para o cálculo da distância. Dois dos rearranjos mais estudados são as ***reversões***, que invertem um segmento do genoma, e as ***transposições***, que trocam as posições relativas de dois segmentos adjacentes do genoma.

O estudo dos problemas de rearranjos de genoma teve início isolando um único tipo de rearranjo, o que gerou os problemas de Ordenação de Permutações por Reversões [21, 33] e Ordenação de Permutações por Transposições [8]. Posteriormente, esses rearranjos foram incorporados em um só modelo, mas sem considerar pesos distintos entre eles [46]. O problema de Ordenação de Permutações com Sinais por Reversões possui algoritmo exato polinomial [28]. Os problemas de Ordenação de Permutações sem Sinais por Reversões ou por Transposições são NP-difíceis [19, 21]. O problema que permite o uso de reversões e

transposições é NP-difícil para permutações com ou sem sinais [41].

Outras operações de rearranjo são as transposições inversas e as revrevs. Dados dois segmentos adjacentes de um genoma, uma ***transposição inversa*** é uma operação que troca as posições relativas dos dois segmentos e inverte um dos segmentos, enquanto uma ***revrev*** inverte cada um dos dois segmentos. Apesar de existirem algoritmos de aproximação para problemas envolvendo essas operações [25], a complexidade desses problemas ainda é desconhecida. O primeiro objetivo desta proposta é apresentar provas de dificuldade para modelos de rearranjo contendo transposições e combinações de reversões, transposições inversas e revrevs.

Os problemas mencionados acima consideram que os rearranjos possuem a mesma probabilidade de acontecer e, portanto, a distância de rearranjos é igual ao número mínimo de rearranjos que ordenam uma permutação. No entanto, estudos indicam que alguns rearranjos são mais raros do que outros em algumas espécies [7, 14] e, a partir dessa motivação, surgiram os primeiros estudos de abordagens ponderadas para o cálculo da distância de rearranjos de genoma [14].

Uma função de custo proporcional ao tamanho da região afetada pelo rearranjo foi motivada pela observação de que, em algumas espécies, os rearranjos que ocorreram no processo evolucionário não afetaram regiões muito grandes [14, 34]. De forma independente da ponderação pelo tamanho, essa observação também motivou os problemas de Ordenação de Permutações por Rearranjos de Tamanho Limitado [31, 38]. O segundo objetivo desta proposta está relacionado ao estudo dos problemas de Ordenação de Permutações por Operações Ponderadas e Limitadas pelo Tamanho, considerando reversões e transposições.

Quando existem genes repetidos nos genomas a serem comparados, ou quando os genomas possuem conjuntos de genes distintos, eles são representados matematicamente como cadeias de caracteres (strings). Assim como na representação por permutações, a orientação dos genes é indicada pelo sinal (positivo ou negativo) de cada elemento da string.

Os rearranjos mencionados anteriormente são operações conservativas, pois não alteram a quantidade de material genético do organismo. Existem também operações não conservativas, como é o caso da inserção, que adiciona um segmento em uma posição do genoma, e

da deleção, que remove um segmento do genoma [22,48]. As inserções e deleções são chamadas de ***indels***. Em modelos que possuem indels, os conjuntos de genes entre cada um dos genomas a serem comparados podem ser distintos e, consequentemente, utilizamos strings para a representação dos genomas. O terceiro objetivo desta proposta está relacionado ao estudo de modelos contendo indels, reversões e transposições, considerando genomas com ou sem repetições de genes.

Além da ordem relativa em que os genes aparecem no genoma, novos estudos incorporaram a informação sobre o tamanho das ***regiões intergênicas*** (sequências de DNA entre cada par de genes) na representação dos genomas, dada a evidência de que as regiões intergênicas ajudam a inferir melhores cenários evolucionários [12,13]. Os modelos intergênicos existentes na literatura contém operações conservativas e indels, no entanto, esses indels podem adicionar ou remover apenas regiões intergênicas, restringindo os genomas comparados a terem o mesmo conjunto de genes. O último objetivo é o estudo da distância de rearranjo incorporando informações intergênicas na representação dos genomas, considerando modelos que possuem operações conservativas e indels de genes e regiões intergênicas.

Para a descrição mais detalhada dos objetivos desta proposta, precisamos da definição dos conceitos e notações das variações dos problemas de distância de rearranjo, além de uma apresentação das variações presentes na literatura e das lacunas que este trabalho pretende suprir. Portanto, o restante desta proposta está dividida da seguinte forma. A Seção 2 introduz conceitos e notações básicas dos problemas de rearranjos de genomas. A Seção 3 apresenta uma revisão da literatura dos problemas relacionados aos objetivos apresentados. As seções 4 e 5 apresentam os objetivos e a metodologia a ser utilizada, respectivamente. A Seção 6 exibe o cronograma de atividades proposto para a realização deste projeto. Por fim, apresentamos os resultados obtidos até o momento na Seção 7.

# 2 Fundamentação Teórica

Nesta seção, apresentamos formalmente os problemas que serão estudados e as definições relacionadas a esses problemas.

## 2.1 Representação da Ordem Relativa dos Genes

No caso em que um genoma não possui genes repetidos e que os genomas a serem comparados compartilham o mesmo material genético, um genoma $\mathcal{G}$ pode ser modelado como uma permutação de inteiros cujos elementos representam genes. Se a orientação dos genes é conhecida, então $\mathcal{G}$ é representado como uma permutação com sinais. Caso contrário, $\mathcal{G}$ é representado por uma permutação sem sinais.

**Definição 2.1** *Uma* ***permutação com sinais*** *é denotada por* $\pi = (\pi_1\ \pi_2\ \ldots\ \pi_n)$*, com* $\pi_i \in \{-n, -(n-1), \ldots, -1, 1, 2, \ldots, n\}$ *e* $|\pi_i| \neq |\pi_j| \iff i \neq j$*, para todo* $i$ *e* $j$*.*

**Definição 2.2** *Uma* ***permutação sem sinais*** *é denotada por* $\pi = (\pi_1\ \pi_2\ \ldots\ \pi_n)$*, com* $\pi_i \in \{1, 2, \ldots, n\}$ *e* $\pi_i \neq \pi_j \iff i \neq j$*, para todo* $i$ *e* $j$*.*

A ***permutação identidade***, $\iota = (1\ 2\ \ldots\ n)$ ou $\iota = (+1\ +2\ \ldots\ +n)$, é a permutação alvo dos problemas de ordenação. No caso em que os genomas não possuem genes repetidos, o problema de encontrar a distância de rearranjos entre dois genomas $\mathcal{G}_1$ e $\mathcal{G}_2$ é equivalente ao problema de encontrar a distância de ordenação por rearranjos de uma permutação [25], já que podemos considerar que o genoma $\mathcal{G}_2$ é representado como a permutação identidade $\iota = (1\ 2\ \ldots\ n)$ e que os genes do genoma $\mathcal{G}_1$ são mapeados respeitando a representação de $\mathcal{G}_2$.

Quando existem genes repetidos nos genomas a serem comparados ou quando os genomas possuem conjuntos de genes distintos, eles são representados matematicamente como cadeias de caracteres (strings).

**Definição 2.3** *Dado um genoma* $\mathcal{G}$ *com uma sequência de genes* $g_1, g_2, \ldots, g_n$*, representamos* $\mathcal{G}$ *como uma* ***string*** $\sigma = (\sigma_1\ \sigma_2\ \ldots\ \sigma_n)$*, tal que o gene* $g_i$ *é mapeado com o caractere* $\sigma_i$*, para todo* $1 \leq i \leq n$*. Assim como nas permutações, a orientação dos genes é representada pelo sinal positivo ou negativo em cada caractere.*

Denotamos por $\Sigma_\sigma$ o alfabeto da string $\sigma$ e por $\sigma_i$ o caractere da posição $i$ da string $\sigma$. Além disso, definimos o tamanho (quantidade de caracteres) de uma string $\sigma$ como $|\sigma|$.

**Definição 2.4** *A* ***ocorrência*** *de um caractere $\gamma$ na string $\sigma$ indica o número de cópias de $\gamma$ na string $\sigma$ e é denotada por $occ(\gamma, \sigma)$. O número de cópias do caractere com maior ocorrência da string $\sigma$ é denotado por $occ(\sigma) = \max_{\gamma \in \Sigma_\sigma}(occ(\gamma, \sigma))$.*

**Definição 2.5** *Dados um caractere $\gamma$ e duas strings $\sigma'$ e $\sigma''$, definimos $\Delta occ(\gamma, \sigma', \sigma'') = occ(\gamma, \sigma') - occ(\gamma, \sigma'')$.*

**Definição 2.6** *Dizemos que duas strings $\sigma'$ e $\sigma''$ são* ***balanceadas*** *se $\Sigma_{\sigma'} = \Sigma_{\sigma''}$ e $occ(\gamma, \sigma') = occ(\gamma, \sigma'')$, para todo $\gamma \in \Sigma_{\sigma'}$, ou seja, as duas strings possuem o mesmo alfabeto e a ocorrência dos caracteres em ambas strings é a mesma.*

### 2.1.1 Rearranjos de Genomas

Como uma permutação de tamanho $n$ é uma string que possui alfabeto $\Sigma = \{1, 2, \ldots, n\}$ e não possui caracteres repetidos, notamos que todas as definições de rearranjos apresentadas para strings também são válidas para permutações, exceto as definições de rearranjos não conservativos, ou seja, rearranjos que adicionam ou removem material genético do genoma.

Os dois tipos de rearranjo mais comuns são as reversões e transposições. A seguir, definimos formalmente essas operações.

**Definição 2.7** *Dada uma string sem sinais $\sigma$ com $|\sigma| = n$, uma* ***reversão*** *$\rho(i, j)$, com $1 \leq i < j \leq n$, é uma operação que inverte o segmento que inicia na posição $i$ e termina na posição $j$, como mostrado a seguir.*

$$\sigma \cdot \rho(i,j) = (\sigma_1 \ \ldots \ \sigma_{i-1} \ \underline{\sigma_j \ \ldots \ \sigma_i} \ \sigma_{j+1} \ \ldots \ \sigma_n)$$

**Definição 2.8** *Dada uma string com sinais $\sigma$ com $|\sigma| = n$, uma* ***reversão*** *$\bar{\rho}(i, j)$, com $1 \leq i \leq j \leq n$, é uma operação que inverte o segmento que inicia na posição $i$ e termina na posição $j$ e inverte o sinal dos elementos afetados, como mostrado a seguir.*

$$\sigma \cdot \bar{\rho}(i,j) = (\sigma_1 \ \ldots \ \sigma_{i-1} \ \underline{-\sigma_j \ \ldots \ -\sigma_i} \ \sigma_{j+1} \ \ldots \ \sigma_n)$$

**Definição 2.9** *Dada uma string $\sigma$ com $|\sigma| = n$, uma* ***transposição*** *$\tau(i, j, k)$, com $1 \leq i < j < k \leq n + 1$, é uma operação que troca as posições relativas dos segmentos $\sigma_i, \ldots, \sigma_{j-1}$ e $\sigma_j, \ldots, \sigma_{k-1}$, como mostrado a seguir.*

$$\sigma \cdot \tau(i, j, k) = (\sigma_1 \ \ldots \ \sigma_{i-1} \ \underline{\sigma_j \ \ldots \ \sigma_{k-1}} \ \underline{\sigma_i \ \ldots \ \sigma_{j-1}} \ \sigma_k \ \ldots \ \sigma_n)$$

Para dois segmentos adjacentes $A$ e $B$, uma ***transposição inversa*** é uma operação que inverte os elementos de $A$ (Tipo 1) ou $B$ (Tipo 2) e troca as posições relativas dos segmentos $A$ e $B$, além disso, quando aplicada a uma string com sinais, essa operação inverte os sinais dos elementos do segmento invertido.

**Definição 2.10** *Dada uma string sem sinais $\sigma$ com $|\sigma| = n$, uma* ***transposição inversa Tipo 1*** *$\rho\tau_1(i, j, k)$ e uma* ***transposição inversa Tipo 2*** *$\rho\tau_2(i, j, k)$, com $1 \leq i < j < k \leq n + 1$, são operações que transformam $\sigma$ da seguinte forma:*

$$\sigma \cdot \rho\tau_1(i, j, k) = (\sigma_1 \ \ldots \ \sigma_{i-1} \ \underline{\sigma_j \ \ldots \ \sigma_{k-1}} \ \underline{\sigma_{j-1} \ \ldots \ \sigma_i} \ \sigma_k \ \ldots \ \sigma_n)$$
$$\sigma \cdot \rho\tau_2(i, j, k) = (\sigma_1 \ \ldots \ \sigma_{i-1} \ \underline{\sigma_{k-1} \ \ldots \ \sigma_j} \ \underline{\sigma_i \ \ldots \ \sigma_{j-1}} \ \sigma_k \ \ldots \ \sigma_n)$$

**Definição 2.11** *Dada uma string com sinais $\sigma$ com $|\sigma| = n$, uma* ***transposição inversa Tipo 1*** *$\bar{\rho\tau}_1(i, j, k)$ e uma* ***transposição inversa Tipo 2*** *$\bar{\rho\tau}_2(i, j, k)$, com $1 \leq i < j < k \leq n + 1$, são operações que transformam $\sigma$ da seguinte forma:*

$$\sigma \cdot \bar{\rho\tau}_1(i, j, k) = (\sigma_1 \ \ldots \ \sigma_{i-1} \ \underline{\sigma_j \ \ldots \ \sigma_{k-1}} \ \underline{-\sigma_{j-1} \ \ldots \ -\sigma_i} \ \sigma_k \ \ldots \ \sigma_n)$$
$$\sigma \cdot \bar{\rho\tau}_2(i, j, k) = (\sigma_1 \ \ldots \ \sigma_{i-1} \ \underline{-\sigma_{k-1} \ \ldots \ -\sigma_j} \ \underline{\sigma_i \ \ldots \ \sigma_{j-1}} \ \sigma_k \ \ldots \ \sigma_n)$$

Para dois segmentos adjacentes $A$ e $B$, uma ***revrev*** é uma operação que inverte cada um dos dois segmentos adjacentes e, quando aplicada a uma string com sinais, inverte os sinais dos elementos afetados. Ao contrário das transposições (ou transposições inversas), as revrevs não trocam as posições relativas dos segmentos $A$ e $B$.

**Definição 2.12** *Dada uma string sem sinais $\sigma$ com $|\sigma| = n$, uma* ***revrev*** *$\rho\rho(i, j, k)$, com*

$1 \leq i < j < k \leq n+1$, *é uma operação que transforma* $\sigma$ *da seguinte forma:*

$$\sigma \cdot \rho\rho(i,j,k) = (\sigma_1 \ \ldots \ \sigma_{i-1} \ \underline{\sigma_{j-1} \ \ldots \ \sigma_i} \ \underline{\sigma_{k-1} \ \ldots \ \sigma_j} \ \sigma_k \ \ldots \ \sigma_n)$$

**Definição 2.13** *Dada uma string com sinais* $\sigma$ *com* $|\sigma| = n$*, uma* ***revrev*** $\bar{\rho}\bar{\rho}(i,j,k)$*, com* $1 \leq i < j < k \leq n+1$*, é uma operação que transforma* $\sigma$ *da seguinte forma:*

$$\sigma \cdot \bar{\rho}\bar{\rho}(i,j,k) = (\sigma_1 \ \ldots \ \sigma_{i-1} \ \underline{-\sigma_{j-1} \ \ldots \ -\sigma_i} \ \underline{-\sigma_{k-1} \ \ldots \ -\sigma_j} \ \sigma_k \ \ldots \ \sigma_n)$$

Os rearranjos apresentados são ditos ***conservativos***, pois não alteram a quantidade de material genético do organismo. Os rearranjos não conservativos estudados nesta proposta são as inserções e as deleções. Como esses rearranjos alteram a quantidade de material genético do organismo, ao serem aplicados em uma string, eles alteram o tamanho da string e o número de ocorrências de um ou mais caracteres, além de poderem alterar o alfabeto da string. Nos modelos que possuem apenas rearranjos conservativos, as strings (genomas) a serem comparadas devem ser balanceadas. Já em modelos que possuem rearranjos não conservativos, as strings podem ser não balanceadas. A seguir definimos formalmente essas operações.

**Definição 2.14** *Dada uma string* $\sigma$ *com* $|\sigma| = n$*, uma* ***deleção*** $\psi(i,j)$*, com* $1 \leq i \leq j \leq n$*, é uma operação que remove o segmento* $\sigma_i, \ldots, \sigma_j$*, como mostrado a seguir.*

$$\sigma \cdot \psi(i,j) = (\sigma_1 \ \ldots \ \sigma_{i-1} \ \sigma_{j+1} \ \ldots \ \sigma_n)$$

**Definição 2.15** *Dada uma string* $\sigma$ *com* $|\sigma| = n$*, uma* ***inserção*** $\phi(i,S)$*, com* $0 \leq i \leq n$*, é uma operação que adiciona o segmento* $S = (S_1 \ \ldots \ S_{|S|})$ *após o i-ésimo elemento de* $\sigma$*, como mostrado a seguir.*

$$\sigma \cdot \phi(i,S) = (\sigma_1 \ \ldots \ \sigma_i \ \underline{S_1 \ \ldots \ S_{|S|}} \ \sigma_{i+1} \ \ldots \ \sigma_n)$$

As operações de inserção e deleção são coletivamente chamadas de ***indels***. Seguindo o

modelo apresentado por Willing *et al.* [48], consideramos as seguintes restrições no problema de encontrar uma sequência de rearranjos de custo mínimo que transforma uma string $\sigma'$ em uma string $\sigma''$: (i) um caractere $\gamma$ pode ser removido apenas se $\Delta occ(\gamma, \sigma', \sigma'') > 0$; (ii) um caractere $\gamma$ pode ser inserido apenas se $\Delta occ(\gamma, \sigma', \sigma'') < 0$. Em outras palavras, as operações de indel não devem remover e, posteriormente, inserir (ou vice-versa) o mesmo material genético.

## 2.2 Representação Incluindo Regiões Intergênicas

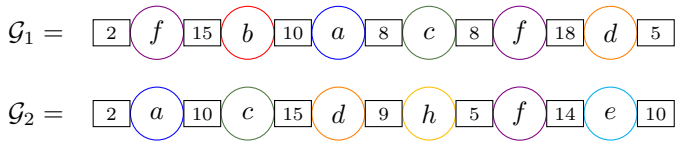
Nesta variação do problema, um genoma $\mathcal{G} = (\breve{g}_1, g_1, \breve{g}_2, g_2, \ldots, \breve{g}_n, g_n, \breve{g}_{n+1})$ é representado como uma sequência de genes $g_1, g_2, \ldots, g_n$ separados por regiões intergênicas $\breve{g}_1, \breve{g}_2 \ldots, \breve{g}_n$, $\breve{g}_{n+1}$, tal que o gene $g_i$ está localizado entre $\breve{g}_i$ e $\breve{g}_{i+1}$. As regiões intergênicas são mais suscetíveis a mudanças do que genes e, além disso, normalmente podemos achar uma associação entre os genes dos genomas sendo comparados, o que geralmente não é possível com regiões intergênicas [12]. Esse fato nos leva a modelar regiões intergênicas usando os seus tamanhos ao invés de usar um marcador, como é feito com os genes.

**Definição 2.16** *Dado um genoma* $\mathcal{G} = (\breve{g}_1, g_1, \breve{g}_2, g_2, \ldots, \breve{g}_n, g_n, \breve{g}_{n+1})$*, representamos* $\mathcal{G}$ *como (i) uma string* $\sigma = (\sigma_1\ \sigma_2\ \ldots\ \sigma_n)$*, tal que cada gene* $g_i$ *é mapeado com o caractere* $\sigma_i$ *para* $1 \leq i \leq n$*, e (ii) uma lista* $\breve{\sigma} = (\breve{\sigma}_1, \breve{\sigma}_2, \ldots, \breve{\sigma}_n, \breve{\sigma}_{n+1})$*, onde o elemento* $\breve{\sigma}_i$ *corresponde ao tamanho da região intergênica* $\breve{g}_i$ *para* $1 \leq i \leq n+1$*.*

Se a orientação dos genes é conhecida, cada elemento $\sigma_i$ possui um sinal positivo ou negativo que representa a orientação do gene $\mathcal{G}_i$. As notações e definições apresentadas para strings na Seção 2.1, como alfabeto e ocorrências, também são utilizadas nos modelos intergênicos.

A seguir apresentamos um exemplo da representação de dois genomas $\mathcal{G}_1$ e $\mathcal{G}_2$, em que genes e regiões intergênicas são representados, respectivamente, por letras em círculos e

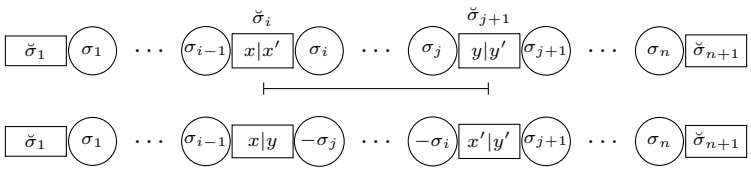
números em retângulos.




Note que o gene $\boldsymbol{b}$ está presente em $\mathcal{G}_1$, mas não em $\mathcal{G}_2$. Já os genes $\boldsymbol{h}$ e $\boldsymbol{e}$ estão presentes apenas em $\mathcal{G}_2$. Além disso, o gene $\boldsymbol{f}$ está repetido em $\mathcal{G}_1$. Dessa forma, a representação desses genomas deve levar em conta que o conjunto de genes dos dois genomas é distinto e que existem genes repetidos. Representamos o genoma $\mathcal{G}_1$ usando a string $\sigma = (5\ 7\ 1\ 2\ 5\ 3)$ e a lista $\breve{\sigma} = (2, 15, 10, 8, 8,\ 18, 5)$, e o genoma $\mathcal{G}_2$ usando a string $\sigma' = (1\ 2\ 3\ 4\ 5\ 6)$ e a lista $\breve{\sigma}' = (2, 10, 15, 9, 5, 14, 10)$.

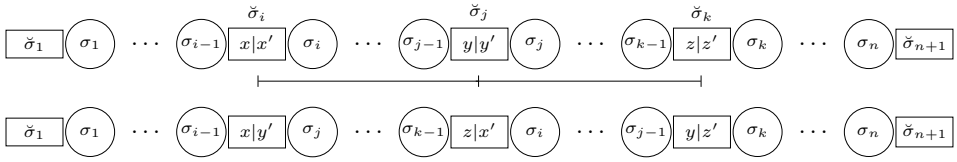
A seguir apresentamos como os rearranjos transformam um genoma $\mathcal{G} = (\sigma, \breve{\sigma})$ com $|\sigma| = n$. Como anteriormente, utilizamos a notação $\mathcal{G} \cdot \beta$ para indicar a aplicação de um rearranjo $\beta$ em $\mathcal{G}$. Também utilizamos $\sigma \cdot \beta$ e $\breve{\sigma} \cdot \beta$ para indicar o efeito de $\beta$ na string e na lista de regiões intergênicas, respectivamente.

Considerando que $\sigma$ é uma string sem sinais, uma ***reversão intergênica*** $\rho^{(i,j)}_{(x,y)}$, com $1 \leq i \leq j \leq n$, $0 \leq x \leq \breve{\sigma}_i$ e $0 \leq y \leq \breve{\sigma}_{j+1}$, é uma operação que transforma $\mathcal{G}$ em $\mathcal{G}' = (\sigma', \breve{\sigma}')$, tal que $\sigma' = (\sigma_1\ \ldots\ \sigma_{i-1}\ \underline{\sigma_j\ \ldots\ \sigma_i}\ \sigma_{j+1}\ \ldots\ \sigma_n)$ e $\breve{\sigma}' = (\breve{\sigma}_1, \ldots, \breve{\sigma}_{i-1}, x + y, \underline{\breve{\sigma}_j, \ldots, \breve{\sigma}_{i+1}}$, $x' + y', \breve{\sigma}_{j+2}, \ldots, \breve{\sigma}_{n+1})$, com $x' = \breve{\sigma}_i - x$ e $y' = \breve{\sigma}_{j+1} - y$. Quando $\sigma$ é uma string com sinais o efeito é similar, mas essa operação também inverte os sinais dos elementos do segmento afetado. Apresentamos a seguir o efeito de uma reversão intergênica em um genoma $\mathcal{G}$.

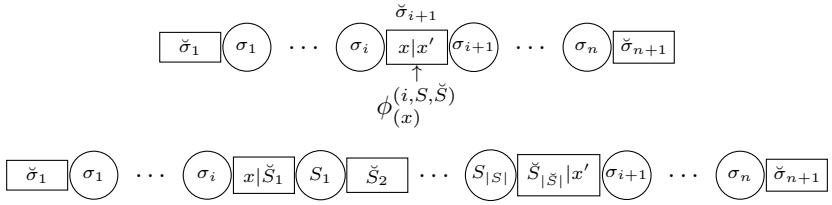



Uma ***transposição intergênica*** $\tau^{(i,j,k)}_{(x,y,z)}$, com $1 \leq i < j < k \leq n + 1$, $0 \leq x \leq \breve{\sigma}_i$, $0 \leq y \leq \breve{\sigma}_j$, e $0 \leq z \leq \breve{\sigma}_k$, é uma operação que transforma $\mathcal{G}$ em $\mathcal{G}' = (\sigma', \breve{\sigma}')$, onde $\sigma' = (\sigma_1\ \ldots\ \sigma_{i-1}\ \underline{\sigma_j\ \ldots\ \sigma_{k-1}}\ \underline{\sigma_i\ \ldots\ \sigma_{j-1}}\ \sigma_k\ \ldots\ \sigma_n)$ e $\breve{\sigma}' = (\breve{\sigma}_1, \ldots, \breve{\sigma}_{i-1}, x +$

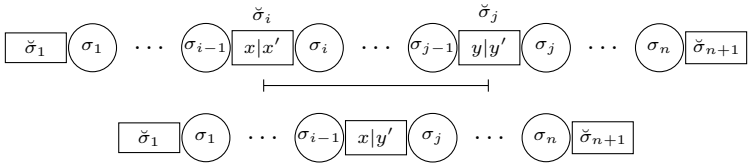
$y', \underline{\breve{\sigma}_{j+1}, \ldots, \breve{\sigma}_{k-1}}, z+x', \underline{\breve{\sigma}_{i+1}, \ldots, \breve{\sigma}_{j-1}}, y+z', \breve{\sigma}_{k+1}, \ldots, \breve{\sigma}_{n+1})$, com $x' = \breve{\sigma}_i - x$, $y' = \breve{\sigma}_j - y$, e $z' = \breve{\sigma}_k - z$. Apresentamos a seguir o efeito de uma transposição intergênica em um genoma $\mathcal{G}$.




Uma ***inserção intergênica*** $\phi_{(x)}^{(i,S,\breve{S})}$, tal que $0 \leq i \leq n$, $0 \leq x \leq \breve{\sigma}_{i+1}$, $S$ é uma sequência de caracteres e $\breve{S}$ é uma lista de inteiros de tamanho $|\breve{S}| = |S| + 1$, é uma operação que transforma $\mathcal{G}$ em $\mathcal{G}' = (\sigma', \breve{\sigma}')$, onde $\sigma' = (\sigma_1 \ \ldots \ \sigma_i \ \underline{S_1 \ \ldots \ S_{|S|}} \ \sigma_{i+1} \ \ldots \ \sigma_n)$ e $\breve{\sigma}' = (\breve{\sigma}_1, \ldots, \breve{\sigma}_i, \underline{x + \breve{S}_1, \breve{S}_2, \ldots, \breve{S}_{|S|}, \breve{S}_{|\breve{S}|} + x'}, \breve{\sigma}_{i+2}, \ldots, \breve{\sigma}_{n+1})$, com $x' = \breve{\sigma}_{i+1} - x$. Quando $S$ é uma sequência vazia, uma inserção intergênica $\phi_{(x)}^{(i,S,\breve{S})}$ altera apenas a região intergênica $\breve{\sigma}_{i+1}$. Apresentamos a seguir o efeito de uma inserção intergênica em um genoma $\mathcal{G}$.




Uma ***deleção intergênica*** $\psi_{(x,y)}^{(i,j)}$, tal que $1 \leq i \leq j \leq n+1$, $0 \leq x \leq \breve{\sigma}_i$, e $0 \leq y \leq \breve{\sigma}_j$, é uma operação que transforma $\mathcal{G}$ em $\mathcal{G}' = (\sigma', \breve{\sigma}')$, onde $\sigma' = (\sigma_1 \ \ldots \ \sigma_{i-1} \ \sigma_j \ \ldots \ \sigma_n)$ e $\breve{\sigma}' = (\breve{\sigma}_1, \ldots, \breve{\sigma}_{i-1}, \underline{x + y'}, \breve{\sigma}_{j+1}, \breve{\sigma}_{j+2}, \ldots, \breve{\sigma}_{n+1})$, com $y' = \breve{\sigma}_j - y$. Quando $i = j$, uma deleção intergênica $\psi_{(x,y)}^{(i,j)}$ deve atender a restrição $0 \leq x \leq y \leq \breve{\sigma}_i$. Nesse caso, $\psi_{(x,y)}^{(i,j)}$ não remove elementos de $\sigma$ e altera apenas $\breve{\sigma}_i$. Apresentamos a seguir o efeito de uma deleção intergênica em um genoma $\mathcal{G}$.



## 2.3 Distância de Ordenação por Rearranjos em Permutações

Um modelo de rearranjo $\mathcal{M}$ define o conjunto de operações permitidas em um problema de rearranjos de genoma. Considerando uma abordagem não ponderada, dados um modelo de rearranjo $\mathcal{M}$ e uma permutação $\pi$, a distância de ordenação $d_{\mathcal{M}}(\pi)$ é igual ao número mínimo de rearranjos pertencentes a $\mathcal{M}$ necessários para ordenar $\pi$.

Considerando uma função de custo $w : \mathcal{M} \to \mathbb{R}$, dados um modelo de rearranjo $\mathcal{M}$ e uma permutação $\pi$, a distância de ordenação $d^w_{\mathcal{M}}(\pi)$ é igual a $\sum_{i=1}^{\ell} w(\beta_i)$ tal que $\beta_i \in \mathcal{M}$, para $1 \leq i \leq \ell$, $\pi \cdot \beta_1 \cdot \ldots \cdot \beta_\ell = \iota$, e $\sum_{i=1}^{\ell} w(\beta_i)$ é mínimo. Para uma sequência de rearranjos $S = \beta_1, \ldots, \beta_\ell$, temos que $w(S) = \sum_{i=1}^{\ell} w(\beta_i)$.

**Definição 2.17** *Considerando um modelo de rearranjo $\mathcal{M}$ e uma função de custo $w$, um problema de ordenação de permutações por rearranjos de genomas recebe como entrada uma permutação $\pi$ e consiste em encontrar o valor de $d^w_{\mathcal{M}}(\pi)$.*

**Definição 2.18** *O **tamanho** de um rearranjo $\beta$ é igual à quantidade de elementos afetados pelo rearranjo e é denotado por $|\beta|$. Por exemplo, uma reversão $\rho(i, j)$ possui tamanho igual a $j - i + 1$ e uma transposição $\tau(i, j, k)$ possui tamanho igual a $k - i$.*

**Definição 2.19** *Dado um valor real $\alpha \geq 0$, na abordagem ponderada pelo tamanho, um rearranjo $\beta$ possui custo $w(\beta) = |\beta|^\alpha$.*

**Definição 2.20** *Dado um inteiro $\lambda$, uma $\lambda$-operação é um rearranjo que possui tamanho menor ou igual a $\lambda$.*

Os problemas de Ordenação de Permutações por $\lambda$-Operações Ponderadas pelo Tamanho consideram um modelo de rearranjo que possui apenas $\lambda$-operações e uma abordagem ponderada pelo tamanho.

## 2.4 Distância de Rearranjos em Strings

Dados um modelo de rearranjo $\mathcal{M}$, uma função de custo $w$ e duas strings $\sigma'$ e $\sigma''$, a distância de rearranjos $d^w_M(\sigma', \sigma'')$ é igual a $\sum_{i=1}^{\ell} w(\beta_i)$ tal que $\beta_i \in \mathcal{M}$, para $1 \leq i \leq \ell$, $\sigma' \cdot \beta_1 \cdot \ldots \cdot \beta_\ell = \sigma''$, e $\sum_{i=1}^{\ell} w(\beta_i)$ é mínimo.

Dados dois genomas $\mathcal{G}_1$ e $\mathcal{G}_2$ e considerando que não existem caracteres repetidos, podemos considerar a distância de rearranjos como a distância de ordenação em strings, sendo que podemos representar $\mathcal{G}_2$ como a string $\iota = (1\ 2\ \ldots\ n)$. Nesse caso, a notação da distância de rearranjos $d_M^w(\sigma, \iota)$ é simplificada por $d_M^w(\sigma)$ (distância de ordenação).

**Definição 2.21** *Considerando um modelo de rearranjo $\mathcal{M}$ e uma função de custo $w$, um problema de distância de rearranjos em strings recebe como entrada duas strings $\sigma'$ e $\sigma''$, e consiste em encontrar o valor de $d_{\mathcal{M}}^w(\sigma', \sigma'')$.*

## 2.5 Distância de Rearranjos Considerando Regiões Intergênicas

Dados um modelo de rearranjo $\mathcal{M}$, uma função de custo $w$ e dois genomas $\mathcal{G}_1 = (\sigma, \breve{\sigma})$ e $\mathcal{G}_2 = (\sigma', \breve{\sigma}')$, a distância de rearranjos $d_M^w(\mathcal{G}_1, \mathcal{G}_2)$ é igual a $\sum_{i=1}^{\ell} w(\beta_i)$ tal que $\beta_i \in \mathcal{M}$, para $1 \leq i \leq \ell$, $\mathcal{G}_1 \cdot \beta_1 \cdot \ldots \cdot \beta_\ell = \mathcal{G}_2$, e $\sum_{i=1}^{\ell} w(\beta_i)$ é mínimo.

**Definição 2.22** *Considerando um modelo de rearranjo $\mathcal{M}$ e uma função de custo $w$, um problema de distância de rearranjos em genomas com informação intergênica recebe como entrada dois genomas $\mathcal{G}_1$ e $\mathcal{G}_2$, e consiste em encontrar o valor de $d_{\mathcal{M}}^w(\mathcal{G}_1, \mathcal{G}_2)$.*

# 3 Revisão Bibliográfica

Esta seção apresenta um resumo da bibliografia dos trabalhos existentes, mostrando uma visão geral dos melhores resultados conhecidos para problemas relacionados aos propostos nesse trabalho.

## 3.1 Complexidade dos Problemas de Rearranjos de Genoma

Hannenhalli e Pevzner [28] apresentaram um algoritmo polinomial exato para a ordenação por reversões considerando permutações com sinais. Para permutações sem sinais, Caprara [21] demonstrou que esse problema é NP-difícil. Como transposições não alteram o sinal de elementos, quando consideramos apenas transposições, temos o problema de

Ordenação de Permutações (sem Sinais) por Transposições, que também é um problema NP-difícil [19].

Apesar de existirem algoritmos de aproximação desde 1998 [46], a complexidade dos problemas de Ordenação de Permutações (com ou sem Sinais) por Reversões e Transposições estava em aberto até pouco tempo, quando Oliveira et al. [41] apresentaram uma prova de que esses problemas pertencem à classe NP-difícil. Eles também apresentaram provas de dificuldade para a versão ponderada do problema, onde uma reversão tem custo $w_\rho$, uma transposição tem custo $w_\tau$ e $w_\tau/w_\rho \leq 1.5$.

Outros modelos de rearranjo consideram também as operações de transposições inversas e revrevs. Algoritmos de aproximação e exatos foram propostos para modelos contendo essas operações, apesar da complexidade do problema para esses modelos ser desconhecida [25,30]. Gu *et al.* [27] desenvolveram um algoritmo de 2-aproximação para permutações com sinais considerando reversões, transposições e transposições inversas. Lin e Xue [35] adicionaram a operação de revrev ao modelo e apresentaram uma 1.75-aproximação. Para o modelo contendo transposições, transposições inversas e revrevs, o melhor resultado para permutações com sinais é uma 1.5-aproximação [29]. Lou e Zhu [37] apresentaram uma 2.25-aproximação para o problema de Ordenação de Permutações sem Sinais por Reversões, Transposições e Transposições Inversas.

Na abordagem ponderada por tipo de rearranjo, consideramos que $w_1$ indica o custo de reversões e $w_2$ indica o custo de transposições, transposições inversas e revrevs. Como reversões são operações mais observadas em casos reais [14], nas abordagens ponderadas o valor de $w_1$ tende a ser menor do que o valor de $w_2$. Para valores de $w_1$ e $w_2$ tal que $1 \leq w_2/w_1 \leq 2$, Bader *et al.* [6] apresentaram uma 1.5-aproximação para o modelo contendo reversões, transposições e transposições inversas em permutações com sinais. Para o mesmo problema e considerando que $w_2/w_1 = 2$, Eriksen [23] desenvolveu uma 7/6-aproximação.

## 3.2 Operações Ponderadas pelo Tamanho e Operações Limitadas pelo Tamanho

O problema de Ordenação de Permutações por Operações Ponderadas pelo Tamanho e o problema de Ordenação de Permutações por Operações Limitados pelo Tamanho foram motivados pela observação de que, em algumas espécies, os rearranjos que ocorreram durante o processo evolutivo não agem em regiões grandes [14, 34].

Lembramos que na abordagem ponderada pelo tamanho o custo de um rearranjo $\beta$ é igual a $|\beta|^\alpha$, onde $\alpha \geq 0$ é um parâmetro de valor real. Para $\alpha = 1$, Bender *et al.* [10] desenvolveram uma $O(\log n)$-aproximação para a Ordenação de Permutações (com ou sem Sinais) por Reversões. Considerando transposições e reversões para permutações com ou sem sinais, Lintzmayer *et al.* [36] apresentaram algoritmos de $O(\log^2 n)$-aproximação. Para $\alpha \geq 2$, existem algoritmos com fator de aproximação constante para os modelos com reversões e transposições para permutações com e sem sinais [10, 36]. Além disso, para $\alpha \geq 3$, os problemas envolvendo permutações sem sinais possuem algoritmos polinomiais exatos [10].

Para qualquer valor de $\lambda > 1$ e combinação de rearranjos (reversões e transposições) para permutações com e sem sinais, existem algoritmos de $O(\lambda^2)$-aproximação [39] para o problema de Ordenação de Permutações por $\lambda$-Operações. Para $\lambda = 3$ (operações curtas), existem algoritmos de aproximação com fator constante [26, 45]. Para $\lambda = 2$ (operações super curtas), os problemas possuem algoritmos polinomiais exatos [26, 32].

Nguyen *et al.* [40] combinaram a abordagem ponderada pelo tamanho e a restrição no tamanho das operações permitidas, considerando reversões em permutações com sinais. Os autores apresentaram uma $O(\log n)$-aproximação, para $\lambda = \Omega(n)$ e $\alpha = 1$, e uma $(2\log^2 n + \log n)$-aproximação, para $\lambda = o(n)$ e $\alpha = 1$.

## 3.3 Distância de Rearranjos de Genoma com Inserções e Deleções

Considerando strings com sinais que não possuem caracteres repetidos, El-Mabrouk [22] introduziu o estudo da distância de reversões e indels, apresentando heurísticas baseadas no algoritmo exato para o problema de Ordenação de Permutações com Sinais por Re-

versões [28].

Além dos modelos de rearranjo já mencionados, Yancopoulos *et al.* [49] introduziram um modelo que contém a operação genérica DCJ (*Double-Cut-and-Join*), que simula reversões e outros rearranjos de genomas. Alguns rearranjos de genomas, como as transposições e revrevs, só podem ser simuladas com duas operações de DCJ [25]. Considerando que os genomas não possuem genes repetidos, a distância de DCJs pode ser calculada em tempo linear [11], assim como a distância de DCJs e indels [15].

Os resultados para a distância de DCJs e indels foram utilizados em um algoritmo polinomial exato para a distância de reversões e indels em classes específicas de strings com sinais que não possuem caracteres repetidos [48]. Recentemente, Willing *et al.* [47] estenderam os seus resultados anteriores apresentando um algoritmo polinomial exato para a distância de reversões e indels em strings com sinais que não possuem caracteres repetidos.

## 3.4 Distância de Rearranjos de Genoma Considerando Regiões Intergênicas

Estudos que incorporam regiões intergênicas são relativamente recentes. Esses estudos assumem que não há genes repetidos nos genomas e que inserções e deleções afetam apenas regiões intergênicas. Dessa forma, os genomas possuem o mesmo conjunto de genes e podem ser modelados usando permutações e uma lista de regiões intergênicas. Para permutações com sinais, Fertin *et al.* [24] mostraram que o problema da distância para o modelo que contém apenas DCJs é NP-difícil e apresentaram uma 4/3-aproximação, um esquema de aproximação de tempo polinomial e uma formulação de programação linear inteira. Quando inserções e deleções de regiões intergênicas são incorporadas ao modelo com DCJs, a distância pode ser calculada em tempo polinomial [20].

Oliveira *et al.* [42] apresentaram uma 2-aproximação para reversões intergênicas em permutações com sinais, além da prova de NP-dificuldade para esse problema. Considerando permutações sem sinais, Brito *et al.* [17] apresentaram uma 4-aproximação e uma 4.5-aproximação para reversões intergênicas e reversões e transposições intergênicas, respectivamente. Para transposições intergênicas, Oliveira *et al.* [43] desenvolveram uma 3.5-

aproximação. O problema de distância de rearranjos para os últimos três modelos mencionados também é NP-difícil [17, 43].

# 4 Objetivos

Nesta seção, apresentamos os objetivos desta proposta. Os objetivos são divididos em quatro etapas, sendo que cada etapa consiste na investigação de uma variação do problema de distância de rearranjos de genoma.

## 4.1 Complexidade dos Problemas de Ordenação de Permutações por Transposições e Outros Rearranjos

Pretendemos investigar a complexidade dos problemas de Ordenação de Permutações por Rearranjos considerando os seguintes modelos:

- $\mathcal{M}_1 = \{\tau, \rho\tau\}$ ou $\bar{\mathcal{M}}_1 = \{\tau, \bar{\rho\tau}\}$: Transposições e Transposições Inversas;
- $\mathcal{M}_2 = \{\rho, \tau, \rho\tau\}$ ou $\bar{\mathcal{M}}_2 = \{\bar{\rho}, \tau, \bar{\rho\tau}\}$: Reversões, Transposições, e Transposições Inversas;
- $\mathcal{M}_3 = \{\tau, \rho\rho\}$ ou $\bar{\mathcal{M}}_3 = \{\tau, \bar{\rho\rho}\}$: Transposições e Revrevs;
- $\mathcal{M}_4 = \{\rho, \tau, \rho\rho\}$ ou $\bar{\mathcal{M}}_4 = \{\bar{\rho}, \tau, \bar{\rho\rho}\}$: Reversões, Transposições, e Revrevs;
- $\mathcal{M}_5 = \{\tau, \rho\tau, \rho\rho\}$ ou $\bar{\mathcal{M}}_5 = \{\tau, \bar{\rho\tau}, \bar{\rho\rho}\}$: Transposições, Transposições Inversas, e Revrevs.
- $\mathcal{M}_6 = \{\rho, \tau, \rho\tau, \rho\rho\}$ ou $\bar{\mathcal{M}}_6 = \{\bar{\rho}, \tau, \bar{\rho\tau}, \bar{\rho\rho}\}$: Reversões, Transposições, Transposições Inversas, e Revrevs.

Iniciaremos o estudo considerando a versão não ponderada desses problemas. Posteriormente, investigaremos versões ponderadas desses problemas, como a ponderação por tipo [14, 29] e pelo número de fragmentações (quebras de elementos adjacentes) [1, 2].

## 4.2 Ordenação de Permutações por Operações Ponderadas e Limitadas pelo Tamanho

Pretendemos estudar a variação do problema em que as operações são ao mesmo tempo ponderadas e limitadas pelo tamanho. Para essa variação do problema, estudaremos os seguintes modelos de rearranjo (versões com e sem sinais):

- $\mathcal{M}_1^\lambda = \{\rho\}$ ou $\bar{\mathcal{M}}_1^\lambda = \{\bar{\rho}\}$: $\lambda$-Reversões.
- $\mathcal{M}_2^\lambda = \{\tau\}$: $\lambda$-Transposições.
- $\mathcal{M}_3^\lambda = \{\rho, \tau\}$ ou $\bar{\mathcal{M}}_3^\lambda = \{\bar{\rho}, \tau\}$: $\lambda$-Reversões e $\lambda$-Transposições.

Essa variação reflete genomas de organismos em que operações que afetam grandes regiões do genoma possuem baixa probabilidade de ocorrer. O foco dessa investigação será a proposta de limitantes inferiores e algoritmos de aproximação para cada um dos modelos de rearranjo. Também pretendemos analisar os problemas para valores de $\lambda$ específicos e quando o custo de uma operação é igual a $\ell^\alpha$, onde $\ell$ é o tamanho da operação e $\alpha \geq 0$ é um parâmetro de valor real.

## 4.3 Distância de Rearranjos de Genoma com Inserções e Deleções

Como nos dois objetivos anteriores, para essa variação do problema os genomas são representados apenas pela ordem relativa dos seus genes. Pretendemos estudar as versões com os seguintes modelos de rearranjo (versões com e sem sinais):

- $\mathcal{M}_1^{indel} = \{\rho, \phi, \psi\}$ e $\bar{\mathcal{M}}_1^{indel} = \{\bar{\rho}, \phi, \psi\}$: Reversões, Inserções e Deleções.
- $\mathcal{M}_2^{indel} = \{\tau, \phi, \psi\}$: Transposições, Inserções e Deleções.
- $\mathcal{M}_3^{indel} = \{\rho, \tau, \phi, \psi\}$ ou $\bar{\mathcal{M}}_3^{indel} = \{\bar{\rho}, \tau, \phi, \psi\}$: Reversões, Transposições, Inserções e Deleções.

Essa variação reflete genomas de organismos que possuem conjuntos distintos de genes. O foco dessa investigação será a proposta de limitantes inferiores e algoritmos de aproximação para cada um dos modelos de rearranjo.

## 4.4 Distância de Rearranjos de Genoma Considerando Regiões Intergênicas

Estudaremos os mesmos modelos da Seção 4.3, mas os genomas são representados utilizando tanto a ordem relativa dos genes quanto o tamanho das regiões intergênicas. O foco dessa investigação será a proposta de limitantes inferiores e algoritmos de aproximação para cada um dos modelos de rearranjo.

Iniciaremos com o estudo da distância entre genomas sem genes repetidos e, posteriormente, a distância entre genomas com genes repetidos, o que também é válido para o objetivo da Seção 4.3.

Para os dois últimos objetivos (seções 4.3 e 4.4), além da versão não ponderada, também investigaremos a ponderação que relaciona o custo de uma operação ao seu tipo. Esses problemas refletem genomas de organismos que possuem conjuntos distintos de genes. O foco dessa investigação será a proposta de limitantes inferiores e algoritmos de aproximação para cada um dos modelos de rearranjo.

# 5 Metodologia e Análise dos Resultados

O primeiro objetivo desta proposta, apresentada na Seção 4.1, tem um foco teórico. Já os demais objetivos possuem uma parte teórica seguida por uma parte prática. Para esses objetivos, temos várias etapas que se repetem e podem ser agrupadas da seguinte forma. Observe que no momento desta qualificação, algumas dessas etapas já foram concluídas.

- Definição formal das variações da distância de rearranjos: os problemas foram obtidos a partir de características biológicas relevantes. Em geral, essas características foram ignoradas previamente na literatura para facilitar o estudo do problema.

- Definição de estrutura de dados: formalizado um problema, a estrutura de dados que trabalharemos precisa conter as informações importantes e ter propriedades factíveis de serem exploradas. Tentamos sempre criar extensões de estruturas já conhecidas, como *breakpoints* [9] e grafos de ciclos [8, 28], adicionando as novas propriedades e

estudando o resultado das operações de rearranjo.

- Definição de limitantes para a distância: o estudo do impacto das operações na estrutura de dados definida é o primeiro passo para gerar limitantes para o problema. Esses limitantes serão úteis para informar a qualidade das soluções retornadas pelos nossos algoritmos em um contexto prático e teórico.

- Geração de algoritmos: utilizando características das estruturas de dados definidas, pretendemos criar algoritmos de aproximação para cada variação do problema. No entanto, para algumas variações (como as que envolvem genes repetidos), criaremos algoritmos heurísticos sem necessariamente provar um fator de aproximação para esses algoritmos. Nesses casos, utilizaremos testes práticos para avaliar a qualidade das soluções retornadas.

- Criação de uma base simulada de testes: os algoritmos desenvolvidos serão implementados e experimentos computacionais serão realizadas para medir a qualidade das soluções. Para a realização dos experimentos, criaremos bases de dados com as representações dos genomas, considerando cada um dos modelos, utilizando métodos propostos na literatura [18, 39]. As bases de dados serão criadas de forma a obter características esperadas para cada conjunto de operações permitidas pelo modelo.

- Análise dos resultados: em todas as etapas, os resultados práticos serão comparados com resultados já conhecidos na literatura, caso existam, ou com limitantes teóricos apresentados em cada etapa. Utilizaremos os limitantes teóricos definidos na etapa anterior para a avaliação da qualidade das soluções.

Todos os resultados teóricos e práticos serão apresentados em formato de artigos para publicação em anais de congressos e revistas da área de biologia computacional e teoria da computação. Além disso, os códigos produzidos e as bases de dados utilizadas em experimentos serão disponibilizados em um repositório público[1].

[1] https://github.com/compbiogroup

# 6 Plano de Trabalho

Nesta seção, apresentamos um cronograma das atividades realizadas e previstas para o programa de doutorado, iniciado em março de 2019. A Tabela 1 detalha o cronograma das atividades, seguida por uma breve descrição de cada atividade.

Tabela 1: Cronograma de atividades desta proposta de doutorado.

| Atividades | Semestres | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | x | x | | | | | | |
| 2 | | x | | x | | | | |
| 3 | x | x | x | x | | | | |
| 4 | | | | | x | | | |
| 5 | x | x | x | | | | | |
| 6 | | x | x | x | x | | | |
| 7 | | | | x | x | x | | |
| 8 | | | | x | x | x | x | |
| 9 | | | | | | | x | x |
| 10 | | | | | | | | x |
| 11 | | | | | | | | x |

1. Obtenção dos créditos obrigatórios em disciplinas do programa de doutorado;
2. Participação no Programa de Estágio Docente (PED);
3. Revisão da literatura;
4. Exame de Qualificação Específico (EQE);
5. Investigação da complexidade dos problemas de Ordenação de Permutações por Transposições e Outros Rearranjos;
6. Investigação dos problemas de Ordenação de Permutações por Operações Ponderadas e Limitadas pelo Tamanho;
7. Investigação dos problemas de Distância de Rearranjos de Genoma com modelos incluindo inserções e deleções;
8. Investigação dos problemas de Distância de Rearranjos de Genoma com modelos incluindo inserções e deleções considerando regiões intergênicas;

9. Escrita da tese;
10. Revisão da tese;
11. Defesa da tese.

Apesar de ser uma atividade constante durante a pesquisa, a revisão da literatura é intensificada em alguns semestres específicos. O tempo e a ordem alocados para as atividades podem mudar no decorrer do desenvolvimento da pesquisa, uma vez que alguns resultados podem ser mais promissores do que outros, fazendo com que algumas atividades sejam realocadas ou novas atividades sejam adicionadas ao cronograma.

# 7 Resultados Preliminares

Nesta seção, apresentamos os resultados obtidos até o momento para cada objetivo descrito na Seção 4.

## 7.1 Complexidade dos Problemas de Ordenação de Permutações por Transposições e Outros Rearranjos

Para esse objetivo, provamos que o problema de distância de rearranjos é NP-difícil para todos os modelos descritos na Seção 4.1, que contêm transposições e a combinação de reversões, transposições inversas e revrevs, considerando $w_2/w_1 \leq 1.5$, onde $w_1$ é o custo de uma reversão e $w_2$ é o custo de uma transposição, transposição inversão ou revrev. Além disso, provamos que o problema de distância de rearranjos considerando a ponderação pelo número de fragmentações é NP-difícil para transposições e a combinação de reversões e transposições. Esses resultados estão presentes no artigo "*On the Complexity of Some Variations of Sorting by Transpositions*" (Alexsandro O. Alexandrino, Andre R. Oliveira, Ulisses Dias, Zanoni Dias) [4] que foi publicado no *Journal of Universal Computer Science* (J.UCS).

Tabela 2: Sumário dos resultados obtidos para o problema de ordenação de permutações por operações ponderadas e limitadas pelo tamanho.

| **Parâmetros** | **Fator de Aproximação Obtido** | | | | |
|---|---|---|---|---|---|
| | $\mathcal{M}_1^\lambda$ | $\mathcal{M}_2^\lambda$ | $\mathcal{M}_3^\lambda$ | $\bar{\mathcal{M}}_1^\lambda$ | $\bar{\mathcal{M}}_3^\lambda$ |
| $\lambda > 1$ e $\alpha = 1$ | $\lambda - 1$ | $\lambda/2$ | $\lambda - 1$ | $\lambda$ | $\lambda$ |
| $\lambda = \Theta(n)$ e $\alpha = 1$ | $O(\lg n)$ | $O(\lg^2 n)$ | $O(\lg^2 n)$ | $O(\lg n)$ | $O(\lg^2 n)$ |
| $\lambda = 3$ e $\alpha = 1$ | 2 | 4/3 | 2 | 3 | 7/3 |
| $\lambda > 1$ e $\alpha \geq 2$ | 2 | 2 | 2 | 3 | 3 |
| $\lambda > 1$ e $\alpha \geq 3$ | Exato | Exato | Exato | Exato | Exato |

## 7.2 Ordenação de Permutações por Operações Ponderadas e Limitadas pelo Tamanho

Para essa variação do problema, apresentamos algoritmos de aproximação para cada um dos modelos descritos na Seção 4.2, considerando algumas configurações de valores para os parâmetros $\lambda$ e $\alpha$. Os fatores de aproximação para cada modelo e configuração de parâmetros são detalhados na Tabela 2. Esses resultados são apresentados no artigo "*Length-Weighted $\lambda$-Rearrangement Distance*" (Alexsandro O. Alexandrino, Guilherme H. S. Miranda, Carla N. Lintzmayer, Zanoni Dias) [3] que foi publicado no *Journal of Combinatorial Optimization*.

## 7.3 Distância de Rearranjos de Genoma com Inserções e Deleções

Inicialmente, adaptamos a definição de *breakpoints* para genomas sem genes repetidos e que possuem conjuntos distintos de genes. Com isso, conseguimos desenvolver algoritmos de 2-aproximação para reversões e algoritmos de 3-aproximação para transposições e a combinação de reversões e transposições. Esses resultados estão presentes no artigo "*Genome Rearrangement Distance with Reversals, Transpositions, and Indels*" (Alexsandro O. Alexandrino, Andre R. Oliveira, Ulisses Dias, Zanoni Dias) [5] que foi publicado no *Journal of Computational Biology* (JCB).

Além disso, apresentamos uma adaptação da estrutura grafo de ciclos chamada de grafo de ciclos rotulado. Com essa estrutura, conseguimos algoritmos de 2-aproximação para

*block-interchanges* (operação que generaliza transposições, sendo que essa operação troca a posição relativa de quaisquer dois segmentos de um genoma) e para transposições. Esses resultados estão presentes no artigo "*Transposition and Block Interchange Distances with Indels*" (Alexsandro O. Alexandrino, Andre R. Oliveira, Ulisses Dias, Zanoni Dias) que foi submetido para uma revista internacional em maio de 2020.

## 7.4 Distância de Rearranjos de Genoma Considerando Regiões Intergênicas

Para essa variação do problema, considerando strings com sinais, desenvolvemos uma 3-aproximação para reversões usando uma adaptação da estrutura grafo de ciclos chamada de grafo de ciclos ponderado e rotulado. Esses resultados estão presentes no artigo "*Reversal Distance on Genomes with Different Gene Content and Intergenic Regions Information*" (Alexsandro Oliveira Alexandrino, Klairton Lima Brito, Andre Rodrigues Oliveira, Ulisses Dias, Zanoni Dias) que foi aceito na *International Conference on Algorithms for Computational Biology* (AlCoB'2021).

Considerando strings sem sinais, desenvolvemos uma 4-aproximação para reversões e uma 4.5-aproximação para transposições e a combinação de reversões e transposições. Esses resultados estão presentes no artigo "*Incorporating Intergenic Regions into Reversal and Transposition Distances with Indels*" (Alexsandro O. Alexandrino, Andre R. Oliveira, Ulisses Dias, Zanoni Dias) que foi submetido para um congresso internacional.

## 7.5 Outros Resultados

Além das atividades descritas no cronograma, o aluno pretende continuar colaborando em outras pesquisas relacionadas à área de Teoria da Computação, não relacionadas diretamente aos objetivos desta proposta.

Até o momento, o aluno foi coautor dos seguintes artigos:

- *"Heuristics for Breakpoint Graph Decomposition with Applications in Genome Rearrangement Problems"* (Pedro Olímpio Pinheiro, Alexsandro Oliveira Alexandrino,

Andre Rodrigues Oliveira, Cid Carvalho de Souza, Zanoni Dias) [44], apresentado no *Brazilian Symposium on Bioinformatics* (BSB'2020);

- *"Sorting by Reversals and Transpositions with Proportion Restriction"* (Klairton Lima Brito, Alexsandro Oliveira Alexandrino, Andre Rodrigues Oliveira, Ulisses Dias, Zanoni Dias) [16], apresentado no *Brazilian Symposium on Bioinformatics* (BSB'2020);

- *"Reversal and Transposition Distance of Genomes Considering Flexible Intergenic Regions"* (Klairton Lima Brito, Andre Rodrigues Oliveira, Alexsandro Oliveira Alexandrino, Ulisses Dias, Zanoni Dias), aceito no *Latin and American Algorithms, Graphs and Optimization Symposium* (LAGOS'2021);

- *"Reversals Distance Considering Flexible Intergenic Regions Sizes"* (Klairton Lima Brito, Alexsandro Oliveira Alexandrino, Andre Rodrigues Oliveira, Ulisses Dias, Zanoni Dias), aceito na *International Conference on Algorithms for Computational Biology* (AlCoB'2021).

- *"Reversals and Transpositions Distance with Proportion Restriction"* (Klairton Lima Brito, Alexsandro Oliveira Alexandrino, Andre Rodrigues Oliveira, Ulisses Dias, Zanoni Dias), aceito para publicação no *Journal of Bioinformatics and Computational Biology.*

# Referências


[1] Alexandrino, A.O., Lintzmayer, C.N., Dias, Z.: Approximation Algorithms for Sorting Permutations by Fragmentation-Weighted Operations. In: Algorithms for Computational Biology, vol. 10849, pp. 53–64. Springer International Publishing, Heidelberg, Germany (2018)

[2] Alexandrino, A.O., Lintzmayer, C.N., Dias, Z.: Sorting Permutations by Fragmentation-Weighted Operations. Journal of Bioinformatics and Computational Biology **18**(2), 2050006.1–2050006.31 (2020)

[3] Alexandrino, A.O., Miranda, G.H.S., Lintzmayer, C.N., Dias, Z.: Length-Weighted $\lambda$-Rearrangement Distance. Journal of Combinatorial Optimization **41**(3), 579–602 (2021)

[4] Alexandrino, A.O., Oliveira, A.R., Dias, U., Dias, Z.: On the Complexity of Some Variations of Sorting by Transpositions. Journal of Universal Computer Science **26**(9), 1076–1094 (2020)

[5] Alexandrino, A.O., Oliveira, A.R., Dias, U., Dias, Z.: Genome Rearrangement Distance with Reversals, Transpositions, and Indels. Journal of Computational Biology **28**(3), 235–247 (2021)

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

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

[8] Bafna, V., Pevzner, P.A.: Sorting Permutations by Transpositions. In: Proceedings of the 6th ACM-SIAM Symposium on Discrete Algorithms (SODA'1995). pp. 614–623. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (1995)

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

[10] Bender, M.A., Ge, D., He, S., Hu, H., Pinter, R.Y., Skiena, S.S., Swidan, F.: Improved Bounds on Sorting by Length-Weighted Reversals. Journal of Computer and System Sciences **74**(5), 744–774 (2008)

[11] Bergeron, A., Mixtacki, J., Stoye, J.: A unifying view of genome rearrangements. In: Proceedings of the 6th Workshop on Algorithms in Bioinformatics (WABI'2006). pp. 163–173. Springer International Publishing, Heidelberg, Germany (2006)

[12] Biller, P., Guéguen, L., Knibbe, C., Tannier, E.: Breaking Good: Accounting for Fragility of Genomic Regions in Rearrangement Distance Estimation. Genome Biology and Evolution **8**(5), 1427–1439 (2016)

[13] Biller, P., Knibbe, C., Beslon, G., Tannier, E.: Comparative Genomics on Artificial Life. In: Pursuit of the Universal. pp. 35–44. Springer International Publishing (2016)

[14] Blanchette, M., Kunisawa, T., Sankoff, D.: Parametric Genome Rearrangement. Gene **172**(1), GC11–GC17 (1996)

[15] Braga, M.D., Willing, E., Stoye, J.: Double cut and join with insertions and deletions. Journal of Computational Biology **18**(9), 1167–1184 (2011)

[16] Brito, K.L., Alexandrino, A.O., Oliveira, A.R., Dias, U., Dias, Z.: Sorting by Reversals and Transpositions with Proportion Restriction. In: Advances in Bioinformatics and Computational Biology. vol. 12558, pp. 117–128. Springer (2020)

[17] Brito, K.L., Jean, G., Fertin, G., Oliveira, A.R., Dias, U., Dias, Z.: Sorting by Genome Rearrangements on both Gene Order and Intergenic Sizes. Journal of Computational Biology **27**(2), 156–174 (2020)

[18] Brito, K.L., Oliveira, A.R., Dias, U., Dias, Z.: Heuristics for the Sorting Signed Permutations by Reversals and Transpositions Problem. In: Algorithms for Computational Biology, vol. 10849, pp. 65–75. Springer International Publishing, Heidelberg, Germany (2018)

[19] Bulteau, L., Fertin, G., Rusu, I.: Sorting by Transpositions is Difficult. SIAM Journal on Discrete Mathematics **26**(3), 1148–1180 (2012)

[20] Bulteau, L., Fertin, G., Tannier, E.: Genome Rearrangements with Indels in Intergenes Restrict the Scenario Space. BMC Bioinformatics **17**(14), 426 (2016)

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

[22] El-Mabrouk, N.: Genome rearrangement by reversals and insertions/deletions of contiguous segments. In: Proceedings of the 11th Symposium on Combinatorial Pattern Matching (CPM'2000). pp. 222–234. Springer (2000)

[23] Eriksen, N.: $(1+\epsilon)$-Approximation of Sorting by Reversals and Transpositions. Theoretical Computer Science **289**(1), 517–529 (2002)

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

[25] Fertin, G., Labarre, A., Rusu, I., Tannier, É., Vialette, S.: Combinatorics of Genome Rearrangements. Computational Molecular Biology, The MIT Press, London, England (2009)

[26] Galvão, G.R., Lee, O., Dias, Z.: Sorting Signed Permutations by Short Operations. Algorithms for Molecular Biology **10**(1), 1–17 (2015)

[27] Gu, Q.P., Peng, S., Sudborough, I.H.: A 2-Approximation Algorithm for Genome Rearrangements by Reversals and Transpositions. Theoretical Computer Science **210**(2), 327–339 (1999)

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

[29] Hartman, T., Sharan, R.: A 1.5-Approximation Algorithm for Sorting by Transpositions and Transreversals. Journal of Computer and System Sciences **70**(3), 300–320 (2005)

[30] Hartmann, T., Wieseke, N., Sharan, R., Middendorf, M., Bernt, M.: Genome rearrangement with ILP. IEEE/ACM Transactions on Computational Biology and Bioinformatics **15**(5), 1585–1593 (2017)

[31] Heath, L.S., Vergara, J.P.C.: Sorting by Short Swaps. Journal of Computational Biology **10**(5), 775–789 (2003)

[32] Jerrum, M.R.: The Complexity of Finding Minimum-length Generator Sequences. Theoretical Computer Science **36**(2-3), 265–289 (1985)

[33] Kececioglu, J.D., Sankoff, D.: Exact and Approximation Algorithms for Sorting by Reversals, with Application to Genome Rearrangement. Algorithmica **13**, 180–210 (1995)

[34] Lefebvre, J.F., El-Mabrouk, N., Tillier, E.R.M., Sankoff, D.: Detection and validation of single gene inversions. Bioinformatics **19**(1), i190–i196 (2003)

[35] Lin, G.H., Xue, G.: Signed Genome Rearrangement by Reversals and Transpositions: Models and Approximations. Theoretical Computer Science **259**(1-2), 513–531 (2001)

[36] Lintzmayer, C.N., Fertin, G., Dias, Z.: Approximation Algorithms for Sorting by Length-Weighted Prefix and Suffix Operations. Theoretical Computer Science **593**, 26–41 (2015)

[37] Lou, X., Zhu, D.: A 2.25-Approximation Algorithm for Cut-and-Paste Sorting of Unsigned Circular Permutations. In: Computing and Combinatorics. vol. 5092, pp. 331–341. Springer International Publishing, Heidelberg, Germany (2008)

[38] Miranda, G.H.S., Alexandrino, A.O., Lintzmayer, C.N., Dias, Z.: Sorting $\lambda$-Permutations by $\lambda$-Operations. In: Proceedings of the 11th Brazilian Symposium on Bioinformatics (BSB'2018), pp. 1–13. Springer International Publishing, Heidelberg, Germany (2018)

[39] Miranda, G.H.S., Lintzmayer, C.N., Dias, Z.: Sorting Permutations by Limited-Size Operations. In: Algorithms for Computational Biology, vol. 10849, pp. 76–87. Springer International Publishing, Heidelberg, Germany (2018)

[40] Nguyen, T.C., Ngo, H.T., Nguyen, N.B.: Sorting by Restricted-Length-Weighted Reversals. Genomics Proteomics & Bioinformatics **3**(2), 120–127 (2005)

[41] Oliveira, A.R., Brito, K.L., Dias, U., Dias, Z.: On the Complexity of Sorting by Reversals and Transpositions Problems. Journal of Computational Biology **26**, 1223–1229 (2019)

[42] Oliveira, A.R., Jean, G., Fertin, G., Brito, K.L., Bulteau, L., Dias, U., Dias, Z.: Sorting Signed Permutations by Intergenic Reversals. IEEE/ACM Transactions on Computational Biology and Bioinformatics (2020)

[43] Oliveira, A.R., Jean, G., Fertin, G., Brito, K.L., Dias, U., Dias, Z.: A 3.5-Approximation Algorithm for Sorting by Intergenic Transpositions. In: Algorithms for Computational Biology. vol. 12099, pp. 16–28. Springer International Publishing, Heidelberg, Germany (2020)

[44] Pinheiro, P.O., Alexandrino, A.O., Oliveira, A.R., de Souza, C.C., Dias, Z.: Heuristics for Breakpoint Graph Decomposition with Applications in Genome Rearrangement Problems. In: Advances in Bioinformatics and Computational Biology. vol. 12558, pp. 129–140. Springer (2020)

[45] Vergara, J.P.C.: Sorting by Bounded Permutations. Ph.D. thesis, Virginia Polytechnic Institute and State University (1998)

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

[47] Willing, E., Stoye, J., Braga, M.: Computing the inversion-indel distance. IEEE/ACM Transactions on Computational Biology and Bioinformatics (2020)

[48] Willing, E., Zaccaria, S., Braga, M.D., Stoye, J.: On the inversion-indel distance. In: BMC Bioinformatics. vol. 14, p. S3. BioMed Central (2013)

[49] Yancopoulos, S., Attie, O., Friedberg, R.: Efficient sorting of genomic permutations by translocation, inversion and block interchange. Bioinformatics **21**(16), 3340–3346 (2005)