

Análise Algébrica de Problemas de Rearranjo em Genomas: Algoritmos e Complexidade

Este exemplar corresponde à redação final da Tese devidamente corrigida e defendida por Cleber V. G. Mira e aprovada pela Banca Examinadora.

Campinas, 19 de Outubro de 2007.

Prof. Doutor João Meidanis (Orientador)

Tese apresentada ao Instituto de Computação, UNICAMP, como requisito parcial para a obtenção do título de Doutor em Ciência da Computação.

Substitua pela ficha catalográfica

(Esta página deve ser o verso da página anterior mesmo no caso em que não se imprime frente e verso, i.é., até 100 páginas.)

Substitua pela folha com as assinaturas da banca

Análise Algébrica de Problemas de Rearranjo em Genomas: Algoritmos e Complexidade

Cleber V. G. Mira¹

Janeiro de 2008

Banca Examinadora:

- Prof. Doutor João Meidanis (Orientador)
- Profa. Dra. Maria Emília Machado Telles Walter
Departamento de Ciência da Computação –CIC– UnB
- Prof. Dra. Celina M. H. de Figueiredo
Instituto de Matemática – Universidade Federal do Rio de Janeiro
- Prof. Dr. Zaroni Dias
Instituto de Computação –IC– Unicamp
- Prof. Dr. Cid Carvalho de Souza
Instituto de Computação –IC– Unicamp
- Prof. Dr. Flávio Keidi Miyazawa (Suplente)
Instituto de Computação –IC– Unicamp
- Prof. Dr. Arnaldo Mandel (Suplente)
Departamento de Ciência da Computação – USP

¹Suporte financeiro de: Bolsa de Doutorado Direto da FAPESP (processo 03/00731-3) 2003–2006, Bolsa PDEE da CAPES (processo 3041-05-0) 01/2006–08/2006.

Resumo

O sucesso na obtenção de cadeias completas de DNA de alguns organismos tem incentivado a busca de novas técnicas computacionais capazes de analisar esse montante de informação para aplicá-lo na descoberta de novos remédios, aumento da produção de alimentos e investigação do processo de evolução dos seres vivos, entre outras aplicações.

A comparação de seqüências de DNA (ou RNA) de diferentes espécies é uma das técnicas importantes para desvendar novas propriedades biológicas. Uma das maneiras de se comparar dois genomas é analisar como os dois se distinguem com base em certas mutações chamadas eventos de rearranjo em genomas. Nessa técnica de comparação, um genoma é modelado como uma seqüência de regiões que são conservadas em um conjunto de genomas. O problema de rearranjos em genomas consiste genericamente em encontrar, dados dois genomas como entrada e um conjunto de tipos de eventos de rearranjo permitidos, uma seqüência mínima de tais eventos de rearranjo que transforme um genoma em outro.

No formalismo clássico de rearranjos em genomas, um genoma tem sido modelado como um conjunto de seqüências de inteiros. Cada número inteiro representa um gene e o seu sinal representa a orientação do gene no genoma. O problema de rearranjos em genomas nesse modelo é analisado de forma geral por meio de diversos diagramas e grafos que representam certas propriedades do par de genomas na entrada do problema. Neste trabalho, usamos um novo modelo para rearranjos em genomas proposto por Meidanis e Dias [39]: o *formalismo algébrico*. Em vez de se basear na análise de diagramas, o formalismo algébrico usa permutações na modelagem de genomas e, principalmente, utiliza resultados de grupos de permutações para analisar as propriedades de genomas e os efeitos de eventos de rearranjo.

A motivação para o desenvolvimento do formalismo algébrico é a possibilidade de formalização de argumentos sobre rearranjos por meio de métodos algébricos, em vez da utilização de recursos gráficos como é feito no formalismo clássico. Esperamos que, por meio do desenvolvimento de um novo formalismo para o tratamento de problemas de rearranjos em genomas, algoritmos mais eficientes para a resolução desses problemas, ou maneiras mais simples de demonstrar alguns dos resultados clássicos na área sejam encontrados com maior facilidade.

Nesse trabalho, apresentamos duas soluções simples e eficientes derivadas diretamente do formalismo algébrico para dois problemas de rearranjos em genomas (o problema de rearranjos

em genomas por intercâmbio de blocos e reversões com sinais e o problema de rearranjo em genomas por fissões, fusões e reversões com sinais). Também discutimos e propomos um algoritmo polinomial para o problema de rearranjos em genomas por transposições generalizadas. Acreditamos que o sucesso na solução desses problemas possa ser estendido para outros problemas de rearranjos em genomas com a consolidação dos conceitos fundamentais do formalismo algébrico.

Esperamos com essa tese convencer o leitor de que o formalismo algébrico é um modelo representativo e poderoso para tratar genomas compostos por cromossomos circulares e ao lidar com a atribuição de pesos a eventos de rearranjo. Por outro lado, não defendemos que o formalismo clássico seja simplesmente substituído pelo formalismo algébrico. Ambos os formalismos podem ser beneficiados por um processo semelhante, porém em menor escala, ao sucesso do desenvolvimento da Geometria Analítica e da Geometria Tradicional.

Abstract

The success in obtaining complete sequences of DNA of some species has encouraged the search for new computational techniques for the analysis of such huge amount of information. One hopes that the results of this research could be applied for the development of new medicines, increasing food crops productivity, better understanding of the evolutionary process in live beings, among other applications.

One technique for the genome analysis is the comparison of DNA (or RNA) sequences from different species. Such a comparison may reveal the similarities and differences between the genomes, which could be used in phylogeny reconstruction for instance. Two genomes can be compared by the analysis of their differences based on mutational events called genome rearrangements. The genome rearrangement problem (also called a sorting problem) consists of finding a minimum sequence of rearrangement events that transforms one genome into another and the number of rearrangement events in the sequence is called the genomic distance.

In the classical formalism for genome rearrangements, a genome is usually modeled by a set of sequences of integers. Each integer represents a gene and its sign stands for the orientation of the gene in the genome. The genome rearrangement problem in this model is analyzed generally with tools such as diagrams and graphs that convey the properties of the genomes in the problem input. We use instead a new model for genome rearrangements proposed by Meidanis and Dias [39]: the *algebraic formalism*. Instead of being based on the analysis of diagrams, the algebraic formalism uses permutations to model genomes and the results from permutation group theory for the analysis of the properties of genomes and the effects of rearrangement events.

The motivation for the development of the algebraic formalism is the possibility of stating arguments more formally by means of algebraic methods than by using graphical resources as the classical formalism does. We hope that more efficient algorithms for genome rearrangement problems or simpler proofs for classical results in the area will be more easily found due to the development of a new formalism.

We present a simple, efficient solution based on the algebraic formalism for two genome rearrangement problems (the problem of genome rearrangements by block-interchanges and signed reversals and the problem of genome rearrangements by fissions, fusions, and signed

reversals). We also discuss and offer a solution for the problem of genome rearrangements by generalized transpositions. We believe that the success in solving those genome rearrangement problems could be extended to other problems by consolidating the fundamental concepts of the algebraic formalism.

We hope that the reader will be convinced that the algebraic formalism is representative and powerful in dealing with circular chromosomes and modeling the assignment of weights to rearrangement events. On the other hand, we do not argue in favor of a substitution of the classical formalism by the algebraic formalism. Both of these formalisms could profit by a similar, even though on a smaller scale, success of the development of the Analytic Geometry and the Traditional Geometry.

Agradecimentos

Primeiramente, eu gostaria de agradecer aos meus pais (Pedro e Regina), aos meus irmãos (Diógenes e Angélica) pelo carinho e atenção dispensados por todos esses anos. Sua educação, norteamento moral e companheirismo ofereceram a segurança necessária a minha formação pessoal.

Muito obrigado ao meu orientador, João Meidanis, pela atenção e disposição ao me passar sua grande experiência acadêmica. Trabalhar com o professor Meidanis não foi apenas uma jornada de desenvolvimento profissional, mas também uma construção de um profundo respeito e amizade.

Aos membros da banca de defesa do doutorado, Zanoni Dias, Maria Emília Walter, Cid Carvalho e Celina de Figueiredo, pela oportunidade de discussão e pelas sugestões durante o desenvolvimento desse projeto.

Ao professor Pavel Pevzner pela acolhida e pela oportunidade de trabalhar juntamente com um dos seus orientandos (Qian Peng) em um problema relevante da área de Rearranjo em Genomas.

Aos professores do Instituto de Computação que sempre mostraram-se prestativos para discutir sobre os tópicos de suas disciplinas não apenas em sala de aula, mas também sempre que a oportunidade nos era mais favorável. Um agradecimento especial ao professor Tomasz Kowaltowski cuja dedicação durante as aulas serve de modelo do que é ser um bom professor.

Aos funcionários do Instituto de Computação pela prestatividade e paciência para lidar com as questões burocráticas do dia a dia.

Aos meus professores e amigos que cultivei nos cursos de Filosofia e de Japonês da UNICAMP dos quais participei: Sílvia Chibeni, José Carlos Pinto de Oliveira, Marcelo Coniglio, Kanami Hirai. Especialmente gostaria de agradecer aos professores Walter Carnielli e Fumiko Takasu. O professor Walter, meu orientador de iniciação científica, me incentivou a buscar a carreira de cientista por meio de seus exemplos que demonstravam que o valor da pesquisa científica não se restringe apenas ao seu caráter técnico, mas também a possibilidade de desenvolvimento pessoal e compreensão do mundo que ela oferece. À professora Fumiko Takasu agradeço a confiança em meu potencial como aluno. O resultado dessa confiança foi a possibilidade de me tornar um pouco mais cidadão do mundo ao visitar a terra do sol nascente.

Aos meus amigos de graduação na Unicamp: Juliano Pastorelli Dutra, Emilson Margoto, Alessandro Soares dos Santos, Claus Aranha, Fernando Xavier, Júnior Telles e a todos os demais colegas da Turma de Ciência da Computação de 1998. Aos bons amigos que fiz em diversas ocasiões em minha graduação: Gina Trancoso, Rodrigo Tomita, Leonardo Tizzei, Sheila Moraes de Almeida, Diogo Ditzel, Vinícius Fortuna, Marcelo Reis, Ricardo Suyama, Eros, Nilton Volpato, Paulo Petrillo, Juliana Bueno-Soler e a todos os demais que eventualmente não consegui citar.

Aos meus novos amigos que fiz nos últimos quatro anos de pós-graduação: Eduardo “Pará” Candido Xavier, Evandro Bracht, Edson Borin, Juliana Freitag, Luís Meira, Luiz Celso, Evandro Bacarin, Leonardo, Patrick, Amanda, Adilson, Norton, Luciano Digiampietri, Augusto Devigili, Felipe Klein, Neumar, Gordana Manic e Juliano Schimiguel.

Aos meus companheiros de empreitada na compreensão do formalismo algébrico: André Atanásio de Almeida e Hederson Santos. Sou muito grato pelas contribuições de ambos ao meu trabalho.

À Fundação de Amparo à Pesquisa do Estado de São Paulo - FAPESP (processo 03/00731-3) e a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - CAPES (processo 3041-05-0) pelo apoio financeiro ao meu estágio de pesquisa no exterior.

Sumário

Resumo	vii
Abstract	ix
Agradecimentos	xi
1 Introdução	1
2 Formalismo Clássico	7
2.1 Reversões com Sinais	10
2.2 Transposições	11
2.3 Diagramas de Ciclos	12
2.4 Grafo de Sobreposição	16
2.5 Obstáculos	19
2.6 Diagramas de ciclos e Transposições	24
2.7 Outros resultados em Rearranjos em Genomas	28
2.8 Discussão	29
3 Formalismo Algébrico	33
3.1 Permutações	33
3.2 Conjugação	38
3.3 Norma e Divisibilidade	40
3.4 Produtos envolvendo 2-ciclos	45
3.5 Produtos envolvendo 3-ciclos	48
3.6 Modelagem de Genomas	59
3.6.1 Genomas Circulares	61
3.6.2 Genomas Lineares	63
3.7 Eventos de Rearranjo em Genomas	63
3.7.1 Reversões com Sinais	64
3.7.2 Transposições	67

3.7.3	Transposições Generalizadas	69
3.7.4	Intercâmbio de Blocos	73
3.7.5	Fusões e Fissões	73
3.8	Rearranjos em Genomas no Formalismo Algébrico	74
3.8.1	Distância Genômica	75
3.8.2	Distância Genômica Ponderada	77
3.9	Considerações Finais	82
4	Nova Estrutura de Dados para Genomas	83
4.1	Genes rotulados como inteiros	83
4.2	Genes rotulados com seus próprios nomes	87
4.3	Considerações Finais	88
5	Algoritmos para Novos Problemas de Rearranjo em Genomas	91
5.1	Distância de Transposição e Transposições Generalizadas	92
5.1.1	Limitantes para a Distância de Transposição	93
5.1.2	Propriedades do Grafo de Transposições Boas	101
5.1.3	Transposições Generalizadas	108
5.2	Fissão, Fusão e Reversão com Sinais	126
5.3	Intercâmbio de Blocos e Reversão com Sinais	134
5.4	Considerações Finais	139
6	Conclusões	141
6.1	Publicações e Resumo de resultados	142
6.2	Considerações Finais e Trabalhos Futuros	144
A	Revisão Bibliográfica	147
A.1	Bafna e Pevzner (1998)	147
A.2	Kaplan, Shamir e Tarjan (2000)	158
A.3	Bergeron (2005)	163
A.4	Hartman e Shamir (2006)	168
A.5	Elias e Hartman (2006)	176
A.5.1	Resultados de Diâmetro de Transposição	179
A.5.2	Algoritmo 1,375-Aproximado	182
A.5.3	Relação entre Configurações e Grafos de Pontos de Quebra	184
A.5.4	Discussão	184
	Bibliografia	185

Lista de Tabelas

5.1	Distância e Tempo de Execução	92
5.2	Proporção de sumidouros no grafo de transposições boas $G(\sigma)$	106

Lista de Figuras

2.1	Exemplo de uma reversão com sinais e uma transposição.	9
2.2	Diagrama de ciclos de uma permutação.	13
2.3	Duas maneiras diferentes de se desenhar o diagrama de ciclos.	14
2.4	Diagrama de ciclos e Grafo de Sobreposição de genoma.	17
2.5	Efeito de uma reversão com sinais em um grafo de sobreposição.	17
2.6	Exemplo de obstáculos de uma permutação.	21
2.7	Classificação de Componentes em Diagrama de Ciclos.	22
2.8	Classificação de componentes de uma permutação.	23
2.9	Fortaleza.	25
2.10	Algoritmo para ordenar uma permutação π	28
3.1	Genoma Mitochondrial Humano.	60
3.2	Resultado da aplicação de uma reversão a um genoma.	65
4.1	Estrutura de Dados para o Problema de Rearranjo em Genomas.	84
4.2	Procedimento para inicializar a Lista de Pares Q	85
4.3	Procedimento usado para construir um par N e incluí-lo em Q	86
4.4	Procedimento usado para inicializar a Tabela de Espalhamento H	87
4.5	Procedimento usado para inicializar os vetores P e S da estrutura de dados. . .	88
5.1	Grafo de Transposições Boas	102
5.2	Lista Q Modificada	115
5.3	Procedimento para construir um par N e incluí-lo na lista Q modificada. . . .	116
5.4	Estrutura de Dados Modificada para Transposições Generalizadas	118
5.5	Procedimento de Atualização de Vetores	120
5.6	Atualização da Lista Q quando há um Par Longo	122
5.7	Atualização da Lista Q quando há apenas Pares Curtos	123
5.8	Atualização da Lista Q	124
5.9	Algoritmo TGSORT	125
5.10	Eventos de rearranjo em genomas vistos como operações de quebra-junção em ciclos.	127

5.11	Atualização de Vetores	130
5.12	Atualização da Lista de Pares Q	130
5.13	Exemplo da ação de fissões, fusões e reversões com sinais sobre a estrutura de dados.	131
5.14	Algoritmo FFSRSort	132
5.15	Algoritmo <i>SRBISort</i>	137
A.1	Diagrama de Ciclos de um genoma	153
A.2	Mudanças de orientação de arestas causadas por shuffling transposition.	153
A.3	Cruzamento e Interferência entre ciclos.	154
A.4	Algoritmo de Ordenação por Transposições.	154
A.5	Um $0 - 2 - 2$ -movimento válido.	157
A.6	Algoritmo 1,5-Aproximado para a Ordenação por Transposições	157
A.7	Ordenação por Reversões de Permutações com sinais.	163
A.8	Configurações de 3-ciclos que podem ser obtidos por meio da aplicação de transposições legais.	170
A.9	Algoritmo de Hartman e Shamir para o problema de Ordenação por Transposições	174
A.10	Algoritmo para a Análise de Casos	181
A.11	Algoritmo de Elias e Hartman para o problema de Ordenação por Transposições	183

Capítulo 1

Introdução

Os avanços obtidos no sequenciamento de macromoléculas (ácidos nucleicos e proteínas) nas últimas décadas nos permitiram encontrar as seqüências de genomas completos de diversos organismos [1, 18, 54]. O sucesso obtido nesse empreendimento científico incentivou a busca por novas técnicas computacionais que fossem capazes de analisar a grande quantidade de dados obtidos dessas seqüências visando a obtenção de informações que sejam relevantes para a pesquisa biológica. Por exemplo, uma compreensão melhor da maneira de se comparar genomas pode trazer à luz novas hipóteses evolutivas, um melhor entendimento do processo de especiação e destacar as distinções e similaridades entre genomas de espécies distintas [51].

Um dos métodos de análise de genomas é o de comparação de cadeias de DNA (ou RNA) de espécies distintas a partir da qual é possível identificar suas diferenças e similaridades, o que pode ser usado na reconstrução de filogenias. A análise das diferenças entre espécies provocada por eventos de rearranjo em genomas é a técnica mais adequada quando ocorrerem nos genomas eventos de mutação de grandes blocos de bases, ao contrário de operações pontuais sobre as bases [48, 50]. O problema de rearranjo em genomas (também chamado de problema da ordenação) consiste em encontrar uma seqüência mínima de eventos de rearranjo que transforme um genoma em outro e o número mínimo de tais eventos de rearranjo na seqüência é chamado de uma distância genômica.

Os eventos de rearranjo em genomas mais estudados na literatura são reversão, transposição, intercâmbio de blocos, fusão, fissão e translocação devido aos interessantes problemas combinatórios definidos a partir dessas operações, bem como ao aparecimento freqüente desses eventos de rearranjo na evolução de alguns genomas. Os eventos de duplicação e deleção possuem grande interesse prático do ponto de vista biológico, mas a investigação desses eventos não tem sido tão acentuada como os demais eventos de rearranjo devido a ausência de um formalismo que permite integrar esses tipos de eventos de rearranjo aos tipos anteriores. A distância de reversão pode ser calculada por meio de um algoritmo de tempo de execução linear de Bader, Moret e Yan [4]. O primeiro algoritmo polinomial (tempo de execução $O(n^4)$, onde n é

o número de genes no genoma) exato para obter uma sequência de reversões que transforma um genoma em outro foi obtida por Hannenhalli e Pevzner [29]. Baseados neste algoritmo, Berman e Hannenhalli [10] apresentaram um algoritmo com complexidade de tempo $O(n^2\alpha(n))$, onde $\alpha(n)$ é a inversa da função de Ackermann. Diversas simplificações na teoria que descreve rearranjos em genomas por reversões resultaram em melhorias no tempo de execução do problema de encontrar a sequência de reversões que transforma um genoma em outro. Kaplan, Shamir e Tarjan [32] propuseram um algoritmo que executa em tempo $O(n^2)$, Bergeron [8] simplificou a teoria envolvendo reversões e ofereceu um algoritmo de tempo de execução $O(n^3)$ e Tannier e Sagot [56] utilizaram uma estrutura de dados projetada por Kaplan e Verbin [33] para projetar um algoritmo cujo tempo de execução é $O(n\sqrt{n \log n})$. Se a orientação dos genes for desconsiderada, isto é, se as reversões não alterarem a orientação dos genes, então o problema de rearranjos em genomas por reversões é *NP*-difícil [15]. Hannenhalli e Pevzner [28] apresentam um algoritmo de tempo polinomial para resolver um problema envolvendo translocação, fusão e fissão em genomas multicromossomais. Hannenhalli e Pevzner [28] pensaram ter encontrado um algoritmo para o problema de rearranjos envolvendo reversões com sinais e translocações por meio da redução desse problema ao problema de rearranjos em genomas por reversões com sinais, porém Ozery-Flato e Shamir [49] demonstraram que aquele algoritmo falha em determinadas circunstâncias e propuseram uma correção para o algoritmo. Dias e Meidanis [20] apresentaram um algoritmo polinomial para o problema de rearranjo em genomas por fusões, fissões e transposições. Hannenhalli [26] apresentou um algoritmo polinomial para encontrar uma sequência de translocações que transforme um genoma em outro, no entanto Bergeron, Mixtacki e Stoye [9] encontraram um erro nesse algoritmo e propuseram um algoritmo de tempo de execução $O(n^3)$ para resolver o problema. Li e colegas [35] apresentaram um algoritmo de tempo de execução linear para determinar a distância de translocações entre dois genomas. Uma sequência de translocações que transforme um genoma em outro pode ser encontrada em tempo de execução quadrático [59], no entanto se restringirmos as operações a translocações que não alterem a orientação dos genes então o problema torna-se *NP*-difícil [62]. Yancopoulos e colegas [61] ofereceram um algoritmo de tempo de execução para o problema de rearranjos em genomas envolvendo translocações, reversões com sinais e intercâmbios de blocos por meio de uma redução desses eventos de rearranjo a operações mais simples (cortes duplos e junções).

A pesquisa no problema de determinar a distância de transposição entre dois genomas resultou em diversos algoritmos aproximativos. Bafna e Pevzner [5, 7] apresentaram um algoritmo aproximativo quadrático para o problema de transposição que possui um fator de aproximação de 1,5 e complexidade quadrática, entretanto esse algoritmo é considerado muito complexo. Christie [17] propõe um outro algoritmo com o mesmo fator de aproximação de 1,5, mas com complexidade $O(n^4)$. Eriksson *et al.* [23] projetaram um algoritmo que ordena qualquer cromossomo com no máximo $\lfloor (2n - 2)/3 \rfloor$ transposições. Hartman e Shamir [30] apresentaram

um algoritmo aproximativo de fator 1,5 e tempo de execução $O(n\sqrt{n \log n})$ pois utiliza a estrutura de dados desenvolvida por Kaplan e Verbin [33]. Elias e Hartman [22] apresentaram um algoritmo de tempo de execução quadrático e fator de aproximação 1,375. Aigner e West [2] e Jerrum [31] apresentaram variantes do problema de rearranjos em genomas por transposições que podem ser resolvidas em tempo de execução polinomial, respectivamente o problema envolvendo transposições que deslocam um bloco que contenha apenas o primeiro gene de um cromossomo linear e o problema envolvendo transposições cujos blocos a serem movidos são dois genes adjacentes apenas. Meidanis, Walter e Dias [41] e, de forma independente, Christie [17] calcularam a distância de transposição entre uma permutação e sua inversa. Dias e Meidanis [21] propuseram um algoritmo para o problema de rearranjo em genomas por transposições de prefixo, ou seja, transposições que movem um bloco contendo o primeiro gene de um cromossomo linear. Fortuna e Meidanis [25] apresentaram um algoritmo para ordenar a permutação reversa por transposições de prefixo e apresentaram um novo limitante superior para esse problema de rearranjo em genomas. Christie [16] também propôs e ofereceu um algoritmo de tempo de execução quadrático para o problema de ordenação por intercâmbio de blocos em genomas — um problema que pode ser visto como uma generalização do problema de transposição. Lin e colegas [37] propuseram um outro algoritmo de tempo de execução quadrático para o problema de rearranjos em genomas por intercâmbio de blocos usando um formalismo algébrico como modelo para genomas. O problema de ordenação por transposições até o momento não pôde ser classificado adequadamente quanto a sua complexidade de tempo de execução. Não sabemos se o problema de ordenação por transposições é *NP*-difícil ou se existe um algoritmo de tempo de execução polinomial que o resolve.

No formalismo clássico de rearranjos em genomas, um genoma tem sido modelado como um conjunto de seqüências de inteiros. Cada número inteiro representa um gene e o seu sinal representa a orientação do gene no genoma. O problema de rearranjos em genomas nesse modelo é analisado por meio de diversos diagramas e grafos que representam certas propriedades do par de genomas na entrada do problema. Pelo contrário, usamos um novo modelo para rearranjos em genomas proposto por Meidanis e Dias [39]: o *formalismo algébrico*. Ao invés de basear-se na análise de diagramas, o formalismo algébrico usa permutações na modelagem de genomas e, principalmente, utiliza resultados de grupos de permutações para analisar as propriedades de genomas e os efeitos de eventos de rearranjo. A própria teoria de permutações foi estendida com conceitos como norma e divisibilidade, que até onde sabemos eram desconhecidos, para adequar-se ao estudo de genomas. Neste trabalho, definimos um novo conceito, o da 3-norma, útil no estudo de problemas com transposições e afins.

A motivação para o desenvolvimento do formalismo algébrico é a possibilidade de formalizar argumentos sobre rearranjos por meio de métodos algébricos ao invés de utilizar de recursos gráficos como é feito no formalismo clássico. Por exemplo, Bergeron [8], para demonstrar um resultado fundamental do formalismo clássico a respeito do efeito de reversões em um deter-

minado diagrama, descreve informalmente o efeito de uma reversão e baseia-se fortemente em uma figura para ilustrar o seu argumento (a autora menciona que “a picture is worth a thousand words”). Esperamos que, por meio do desenvolvimento de um novo formalismo para o tratamento de problemas de rearranjos em genomas, algoritmos mais eficientes para a resolução desses problemas ou maneiras mais simples de demonstrar alguns dos resultados clássicos na área sejam encontrados com maior facilidade.

Definimos algumas características comuns que um problema de rearranjo em genomas deve possuir e que nos permite aplicar uma estratégia gulosa para solucioná-lo. Apresentamos uma nova estrutura de dados baseada na representação de genomas por permutações. Essa estrutura de dados e a aplicação da estratégia gulosa permite a obtenção de uma solução simples e eficiente para dois problemas de rearranjos em genomas (o problema de rearranjos em genomas ponderado por intercâmbio de blocos e reversões com sinais e o problema de rearranjo em genomas ponderado por fissões, fusões e reversões com sinais). Também analisamos alguns aspectos sobre o limitante inferior do problema de rearranjo em genomas por transposições e apresentamos alguns resultados sobre o grafo formado por um conjunto de genomas (grafo de transposições boas). Além disso, propomos uma solução para o problema de rearranjos em genomas ponderado por transposições generalizadas. Acreditamos que o sucesso na solução desses problemas possa ser estendido para outros problemas de rearranjo em genomas com a consolidação dos conceitos fundamentais do formalismo algébrico. A pesquisa de um novo formalismo para representar genomas e eventos de rearranjo pode levar a identificação de novas propriedades que não estavam explícitas no formalismo clássico e a novos algoritmos mais eficientes. Os formalismos clássico e algébrico provavelmente possuem o mesmo poder de representação de problemas de rearranjo em genomas, embora cada um desses formalismos possa ter vantagens específicas em diferentes circunstâncias, por exemplo ao se modelar cromossomos lineares ou circulares. Pretendemos com essa tese contribuir para demonstrar algumas das vantagens do formalismo algébrico sobre o formalismo clássico em determinados problemas e fundamentar as bases sobre as quais novas pesquisas sobre o relacionamento entre esses formalismos possam ser realizadas.

No Capítulo 2 apresentamos o formalismo clássico e como os problemas de rearranjo envolvendo reversões com sinais e transposições são tratados por esse formalismo. Apresentamos no Capítulo 3 os conceitos algébricos fundamentais que são necessários para modelar genomas e eventos de rearranjo. Apresentamos o conceito de permutações e algumas de suas propriedades fundamentais. Divisibilidade, os efeitos de produtos de ciclos, conjugação e norma são tratados a seguir. Além disso, mostramos como modelar genomas por meio de permutações em sua representação como produto de ciclos. Apresentamos a modelagem de alguns eventos de rearranjo em genomas e a abordagem de problemas de rearranjo em genomas quando representados pelo formalismo algébrico. Por último, discutimos a estratégia gulosa mencionada anteriormente para problemas de rearranjo em genomas. O Capítulo 4 trata de uma nova estrutura de

dados que projetamos para modelar genomas em um problema de rearranjo em genomas. Discutimos as operações básicas permitidas pela nova estrutura de dados e como ela pode ser usada para lidar com o caso no qual os genes são representados com os seus próprios nomes em um genoma. No Capítulo 5 apresentamos soluções para três problemas de rearranjo em genomas por meio do formalismo algébrico. Discutimos primeiramente o limitante inferior do problema de rearranjo por transposições e apresentamos algumas propriedades fundamentais do grafo de transposições boas. Em seguida, apresentamos um algoritmo linear para encontrar uma sequência de transposições generalizadas que transforme um genoma em outro. Apresentamos um algoritmo de tempo de execução quadrático para o problema de rearranjo por intercâmbio de blocos e reversões com sinais e outro algoritmo de tempo de execução linear para o problema envolvendo fusões, fissões e reversões com sinais. Sumarizamos os nossos resultados no Capítulo 6. Segue no apêndice desse trabalho uma revisão bibliográfica de resultados importantes da teoria de rearranjo em genomas.

Adotaremos a convenção de chamar de “proposições” aos resultados cujas demonstrações não são dadas aqui, mas podem ser encontradas nas referências, de “lemas” aos resultados auxiliares demonstrados e de “teoremas” aos resultados demonstrados mais relevantes.

Capítulo 2

Formalismo Clássico

O genoma de uma espécie é o conjunto de macromoléculas que são responsáveis por codificar toda a informação necessária para construir as proteínas e outros produtos usados no metabolismo de um organismo [53, 34]. Essas macromoléculas são ácidos nucleicos. Muitos seres vivos codificam a informação para a construção de proteínas em algumas moléculas de ácido desoxirribonucleico (DNA em inglês) chamados *cromossomos*. O DNA é uma cadeia composta por duas *fitas*. Cada fita é uma sequência de unidades básicas compostas por um açúcar (desoxirribose), um fosfato e uma *base*: adenina (A), guanina (G), citosina (C) e timina (T). A estrutura dessas unidades básicas induz uma *orientação* na cadeia de DNA. As duas fitas do DNA, que obedecem a uma estrutura de dupla hélice, são mantidas juntas pelas conexões (pontes de hidrogênio) entre os pares de bases: A – T e C – G. As bases A e T, bem como C e G, são chamadas *bases complementares* entre si. Um *gene* é um segmento, não necessariamente contínuo, em uma das fitas do DNA que codifica informação para a construção de uma proteína. As fitas de um cromossomo são *complementares reversas*, isto é, os genes que pertencem a mesma fita possuem a mesma orientação, enquanto os genes pertencentes a outra fita possuem orientações opostas à da primeira. Um genoma contendo apenas um cromossomo é chamado *unicromossomal*, caso contrário é chamado *multicromossomal*. Cada cromossomo pode ser visto como uma sequência de genes quando é um *cromossomo linear*, ou seja, com duas extremidades; ou uma lista circular de genes, quando o cromossomo é um *cromossomo circular*, ou seja, sem extremidades, uma cadeia circular.

O genoma pode sofrer alterações causadas por *mutações* ou *recombinações*. Uma mutação é uma modificação em uma pequena região no nível das bases do cromossomo. Por exemplo, mutações podem alterar uma base, inserir uma nova base ou removê-la da cadeia. Uma recombinação envolve a mudança de uma grande região de um cromossomo, geralmente envolvendo diversos genes. Recombinações podem causar, por exemplo, a duplicação, inserção, alteração de orientação, remoção ou movimento em outras posições do cromossomo de segmentos de genes.

Uma das técnicas mais importantes para a reconstrução de filogenias e proposição de novas hipóteses evolutivas é a comparação de genomas. Uma técnica bem estabelecida para a comparação de dois genomas é o *alinhamento de seqüências* que consiste em encontrar a *distância de edição ponderada* entre dois cromossomos [53]. No problema de alinhamento de seqüências, um cromossomo de um genoma é visto como uma cadeia de caracteres sobre o alfabeto de bases $\{A, C, G, T\}$ e supõe-se que há apenas inserções, deleções e substituições de bases como eventos de mutação. Por outro lado, alguns trabalhos em biologia molecular [48, 50] demonstraram que o alinhamento em seqüências não é sempre a técnica mais adequada para o problema de comparação de genomas porque há genomas na natureza que se distinguem mais entre si devido aos eventos de recombinação do que aos eventos de mutação. Um exemplo da relevância das recombinações na diferenciação de genomas foi a descoberta de que os genomas mitocondriais de *Brassica oleracea* (repolho) e *Brassica campestris* (nabo) possuem genes quase idênticos, porém tais genes aparecem em ordens bem distintas em cada genoma [50]. Chamaremos a esses eventos de recombinação de *eventos de rearranjo* em genomas. Eventos de rearranjo em genomas parecem ser raros em algumas espécies [28], logo a análise desse tipo de evento seria mais apropriada para a comparação de genomas de espécies que possuem um ancestral comum distante no tempo de evolução como é o caso dos genomas do homem e do rato por exemplo [48].

A análise de eventos de rearranjo em genomas foca a comparação das posições e orientações dos mesmos genes em diversos genomas distintos. Assumindo-se que a evolução segue um cenário parcimonioso, isto é, os eventos seriam tão raros que dificilmente ocorreriam eventos de rearranjo repetidos ou “retornos” a genomas passados, estamos interessados em, dados dois genomas e os tipos de eventos de rearranjo permitidos, encontrar uma seqüência mínima de tais eventos de rearranjo que transformem um genoma em outro. Esse é o *problema de rearranjo em genomas*. O número mínimo de eventos de rearranjo que transforma um genoma em outro é chamado de *distância genômica*. A distância genômica pode ser usada por exemplo como um parâmetro em reconstrução de filogenias [12].

Alguns exemplos de eventos de rearranjo são as reversões com sinais e transposições. Uma reversão com sinais inverte a ordem e a orientação de uma seqüência contígua de genes, enquanto uma transposição move uma seqüência de genes para outra posição no mesmo cromossomo. Na Figura 2.1, podemos observar o efeito de uma reversão com sinais e de uma transposição sobre um genoma contendo um único cromossomo.

O formalismo clássico introduzido por Hannenhalli e Pevzner [27] modela um genoma enfatizando a visão do genoma como um conjunto de cromossomos e cada cromossomo como uma seqüência de genes. Um *gene* é identificado por um inteiro cujo sinal indica a orientação do gene. Os n genes encontrados em um genoma são representados pelo conjunto de inteiros $SE(n) = \{-n, \dots, -2, -1, +1, +2, \dots, +n\}$. Um *cromossomo* é uma função, chamada *permutação com sinais*, que mapeia uma *posição* do conjunto $\{1, 2, \dots, n\}$ em um

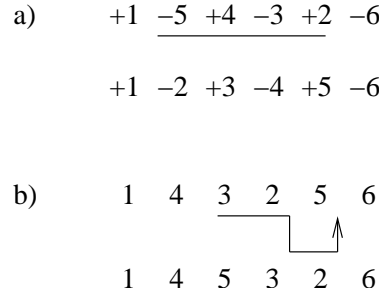


Figura 2.1: Exemplo de uma reversão com sinais e uma transposição, onde cada número inteiro representa um gene (ou um marcador). (a) A reversão com sinais inverte a ordem e modifica a orientação de uma seqüência de genes. Os genes modificados são sublinhados na figura. (b) Uma transposição move uma seqüência de genes para uma posição imediatamente anterior a um certo gene como indicado pela flecha. O mesmo evento pode ser descrito como a troca de lugar de duas seqüências adjacentes de genes.

gene em $SE(n)$ naquela posição no cromossomo. Para um cromossomo π , o gene encontrado na posição i é denotado por π_i . O formalismo clássico assume em sua representação de genomas que um cromossomo é linear. Para que um cromossomo circular possa ser modelado pelo formalismo clássico, ele deve ser aberto entre dois genes adjacentes que são escolhidos para serem as extremidades do cromossomo. Um cromossomo linear possui duas representações equivalentes [28]: $\pi = [\pi_1, \dots, \pi_n]$ e $-\pi = [-\pi_n, \dots, -\pi_1]$. Um *genoma* é um conjunto de cromossomos. Dado o genoma $\Pi = \{\pi(1), \dots, \pi(N)\}$ contendo N cromossomos, existem 2^N maneiras distintas de representar este genoma baseada nas duas representações de cromossomos.

Exemplo 1. Dado o conjunto de genes $SE(10) = \{x \in \mathbb{Z} \mid 1 \leq |x| \leq 10\}$, um exemplo de genoma com esses genes é $\Pi = \{\pi, \theta\}$ onde

$$\pi = [+1, -6, +4, -3, +2]$$

e

$$\theta = [+7, -5, +10, -9, +8]$$

são os cromossomos de Π .

Em problemas de rearranjo em genomas cujos genomas são unicromossomais e os eventos de rearranjo envolvidos não criam nem removem cromossomos, costuma-se simplificar a notação do genoma usando-se apenas uma das representações equivalentes do único cromossomo como modelo para o genoma. Nessa apresentação do formalismo clássico, nos concentraremos na exposição dos problemas de rearranjo cujos genomas na entrada são unicromossomais.

2.1 Reversões com Sinais

Uma *reversão com sinais* $\rho(i, j)$, para $1 \leq i \leq j \leq n$, é o evento de rearranjo que transforma o genoma

$$\pi = [\pi_1, \pi_2, \dots, \pi_{i-1}, \pi_i, \dots, \pi_{j-1}, \pi_j, \pi_{j+1}, \dots, \pi_n]$$

no genoma

$$\pi\rho(i, j) = [\pi_1, \pi_2, \dots, \pi_{i-1}, -\pi_j, \dots, -\pi_i, \pi_{j+1}, \dots, \pi_n].$$

Exemplo 2. Seja $\pi = [+1, -5, +4, -3, +2]$ um genoma. A reversão com sinais $\rho(3, 4)$ inverte a ordem e modifica a orientação dos elementos π_3 e π_4 :

$$[+1, -5, +4, -3, +2]\rho(3, 4) = [+1, -5, +3, -4, +2]$$

Dados dois genomas π e σ , o *problema de rearranjo em genomas por reversões com sinais* consiste em encontrar uma seqüência de reversões com sinais $\rho_1, \rho_2, \dots, \rho_k$ tal que

$$\pi\rho_1 \dots \rho_{k-1}\rho_k = \sigma$$

e k é mínimo. Chamamos o número mínimo k de reversões na seqüência de *distância de reversão com sinais* de π a σ , denotada por $d_r(\pi, \sigma)$.

Dado o genoma π , o *problema de ordenação por reversões com sinais* consiste em encontrar uma seqüência mínima de reversões com sinais $\rho_1, \rho_2, \dots, \rho_k$ tal que

$$\pi\rho_1 \dots \rho_{k-1}\rho_k = [1, 2, \dots, n].$$

Chamamos o número mínimo k de reversões na seqüência de *distância de reversão com sinais* de π , denotada por $d_r(\pi)$.

Exemplo 3. Dados os genomas π e σ abaixo para um conjunto de genes $SE(5)$:

$$\pi = [+1, -5, +4, -3, +2] \quad \text{e} \quad \sigma = [+1, +2, +3, +4, +5].$$

Mostramos uma seqüência de reversões com sinais que transformam o genoma π no genoma σ .

Os passos da seqüência de reversões com sinais são mostrados a seguir:

$$\pi\rho(5, 5) = [+1, -5, +4, -3, -2]$$

$$\pi\rho(5, 5)\rho(3, 3) = [+1, -5, -4, -3, -2]$$

$$\pi\rho(5, 5)\rho(3, 3)\rho(2, 5) = [+1, +2, +3, +4, +5]$$

Para um dado genoma π , uma *reversão com sinais ordenante*, é uma reversão com sinais ρ tal que $d_r(\pi\rho) = d_r(\pi) - 1$.

Seja $Gen(SE(n))$ o conjunto de genomas definidos para $SE(n)$. O valor máximo de $d_r(\pi)$ para qualquer $\pi \in Gen(SE(n))$ é chamado de *diâmetro de reversão com sinais* e o denotamos por $d_r(n)$.

2.2 Transposições

Uma *transposição* $\tau(i, j, k)$, para $1 \leq i < j < k \leq n+1$, é o evento de rearranjo que transforma o genoma

$$\pi = [\pi_1, \dots, \pi_n],$$

no genoma

$$\pi\tau(i, j, k) = [\pi_1, \dots, \pi_{i-1}, \pi_j, \dots, \pi_{k-1}, \pi_i, \dots, \pi_{j-1}, \pi_k, \dots, \pi_n].$$

Observe que transposições podem ser vistas como permutações como segue-se.

$$\tau(i, j, k) = [1, \dots, i-1, j, \dots, k-1, i, \dots, j-1, k, \dots, n].$$

para $1 \leq i < j < k \leq n+1$.

Dados dois genomas e um conjunto de genes sem sinais (i.e. com a mesma orientação) π e σ , o *problema de rearranjo em genomas por transposições* consiste em encontrar uma seqüência de transposições $\tau_1, \tau_2, \dots, \tau_k$ tal que $\pi\tau_1 \dots \tau_{k-1}\tau_k = \sigma$ e k é mínimo. Chamamos ao valor mínimo k de *distância de transposição* de π para σ , denotada por $d_t(\pi, \sigma)$.

Dado um genoma no conjunto de genes sem sinais $E(n) = \{1, \dots, n\}$, o *problema de ordenação por transposições* consiste em encontrar uma seqüência τ_1, \dots, τ_k de transposições que transformem um genoma π em $[+1, \dots, +n]$ e k é mínimo. O número mínimo de transposições k é chamado de *distância de transposição* de π e é denotado por $d_\tau(\pi)$.

Para um dado genoma π , uma *transposição ordenante* é uma transposição τ tal que $d_t(\pi\tau) = d_t(\pi) - 1$.

Denotamos por $d_t(n)$ o *diâmetro de transposição* de $Gen(E(n))$, que é a máxima distância de transposição $d_\tau(\pi)$ para todo $\pi \in Gen(E(n))$.

Em um cenário evolutivo, diferentes tipos de rearranjo em eventos podem ocorrer em uma seqüência de eventos de rearranjo que transformam um genoma em outro. Logo, é natural procurar por seqüências de eventos de rearranjo que transformem um genoma em outro usando um número mínimo de eventos de rearranjo de tipos distintos. Por exemplo, poderíamos considerar reversões com sinais e transposições como eventos de rearranjos possíveis. Dados os genomas π e σ e um conjunto de genes, o *problema de rearranjo em genomas por reversões com sinais e transposições* consiste em encontrar uma seqüência de eventos $\rho_1, \rho_2, \dots, \rho_k$, cada um sendo uma reversão com sinal ou uma transposição, tal que $\pi\rho_1 \dots \rho_{k-1}\rho_k = \sigma$ e k é mínimo. Chamamos ao número mínimo de eventos de rearranjo k de *distância de reversão com sinais e transposição* entre π e σ , denotado como $d_{rt}(\pi, \sigma)$. O problema de ordenação (apenas um genoma como entrada) e o diâmetro podem ser definidos de maneira semelhante a dos problemas anteriores.

2.3 Diagramas de Ciclos

Uma vez que problemas de rearranjo em genomas (envolvendo dois genomas) são sempre equivalentes aos problemas de ordenação (envolvendo um genoma), o formalismo clássico foi desenvolvido principalmente para problemas de ordenação. Os conceitos e resultados apresentados a seguir podem ser encontrados nos trabalhos de diversos autores [28, 29, 32, 8, 53]. Esses conceitos são fundamentais para a compreensão de como o formalismo clássico lida com os problemas de ordenação por reversões com sinais e o problema de ordenação por transposições.

Dada uma permutação (sem sinais) $\pi = [\pi_1, \pi_2, \dots, \pi_n]$ sobre um conjunto $\{x \in \mathbb{Z} \mid 1 \leq x \leq n\}$, a *versão estendida* de π adiciona $\pi_0 = 0$ e $\pi_{n+1} = n + 1$ à permutação π . Um par de elementos (π_i, π_{i+1}) , para $0 \leq i \leq n$, de π é uma *adjacência* quando $|\pi_{i+1} - \pi_i| = 1$, caso contrário o par é um *ponto de quebra* [28]. O número de pontos de quebra é denotado por $b(\pi)$.

Uma permutação com sinais π pode ser também representada por uma permutação sobre o conjunto $\{1, \dots, 2n\}$, chamada de *imagem* de π , que consiste na sequência obtida de π ao substituir-se π_i por $2|\pi_i| - 1$ e $2|\pi_i|$ quando π_i é um inteiro positivo, e por $2|\pi_i|$ e $2|\pi_i| - 1$, caso contrário. Por exemplo, a imagem de $[-3, 2, -1, -4, -5]$ é a permutação $[6, 5, 3, 4, 2, 1, 8, 7, 10, 9]$. A imagem de um cromossomo π é denotada por π' . Alguns autores definem a imagem de uma permutação usando $-\pi_i$ e $+\pi_i$ ao invés de $2|\pi_i| - 1$ de $2|\pi_i|$ [53]. Ambas as representações são equivalentes. O número de pontos de quebra $b(\pi)$ de uma permutação com sinais π é o número de pontos de quebra da permutação (sem sinais) π' imagem de π .

Dada uma reversão com sinais $\rho(i, j)$ para uma permutação com sinais π , a reversão com sinais com efeito equivalente para a imagem de π é $\rho^* = \rho(2i - 1, 2j)$ [6]. Nesse caso temos $(\pi\rho)' = \pi'\rho^*$. Logo, qualquer sequência de reversões com sinais que ordene π pode ser vista como uma sequência de reversões com sinais que ordena π' .

Exemplo 4. Para $\pi = [-1, 4, 2, -5, -3]$ a reversão com sinais $\rho(2, 4)$ tem efeito equivalente a $\rho(3, 8)$, que inverte os elementos nas posições de 3 a 8, incluindo ambos 3 e 8:

$$\pi'\rho^* = \pi'\rho(3, 8) = [0, 2, 1, 9, 10, 4, 3, 8, 7, 6, 5, 11]$$

Dada uma permutação com sinais π , denotamos por $\Delta b(\rho, \pi)$ a diferença no número de pontos de quebra $b(\pi\rho) - b(\pi)$. Uma vez que uma reversão inverte um segmento contíguo de uma permutação com sinais, a reversão pode criar ou eliminar no máximo dois pontos de quebra em π . Logo, temos $-2 \leq \Delta b(\rho, \pi) \leq 2$. A desigualdade anterior sugere um limitante inferior para a distância de reversão com sinais:

$$\frac{b(\pi)}{2} \leq d_r(\pi)$$

O limitante inferior baseado em pontos de quebra não é muito justo [28]. Bafna e Pevzner [6] identificaram um parâmetro mais adequado ao observar a estrutura de um diagrama

definido a partir de uma permutação.

O *diagrama de ciclos* de uma permutação π em sua versão estendida (também chamado de *diagrama de pontos de quebra*), denotado por $B(\pi)$, é construído tomando-se os $n+2$ elementos de π como vértices e dispondo esses vértices na ordem em que aparecem na permutação. Os vértices π_i e π_j são ligados por uma *aresta preta* quando (π_i, π_j) é um ponto de quebra em π (i.e. $|\pi_i - \pi_j| \neq 1$ e $|i - j| = 1$) e por uma *aresta cinza* quando $|i - j| \neq 1$ e $|\pi_i - \pi_j| = 1$. A Figura 2.2 mostra o diagrama de ciclos para uma permutação. O diagrama de ciclos de uma permutação com sinais π é o diagrama de ciclos obtido a partir da imagem de π , ou seja, o diagrama $B(\pi')$. A Figura 2.3 ilustra um diagrama de ciclos para uma permutação com sinais.

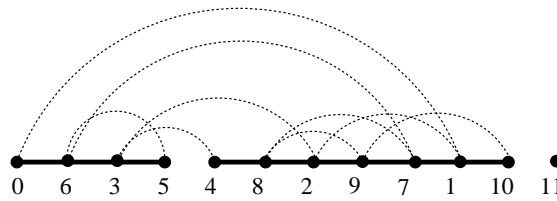


Figura 2.2: Diagrama de ciclos de $[0, 6, 3, 5, 4, 8, 2, 9, 7, 1, 10, 11]$ já em sua versão estendida.

Bergeron [8] define o diagrama de ciclos de maneira um pouco diferente da definição original de Hannenhalli e Pevzner [28]. Na definição de Bergeron é permitido que existam múltiplas arestas entre dois vértices. Nesse caso o diagrama de ciclos é construído ao se dispor os elementos da imagem de uma permutação com sinais π como vértices, incluindo 0 e $2n + 1$, na ordem em que aparecem na permutação (ver a parte superior da Figura 2.3). Há uma aresta cinza (x, y) quando $x = 2i$, $y = 2i + 1$ para $0 \leq i \leq n$. Há uma aresta preta (x, y) quando $x = \pi'_{2i}$ e $y = \pi'_{2i+1}$ para $0 \leq i \leq n$. A Figura 2.3 mostra um exemplo de um diagrama de ciclos para a permutação com sinais $\pi = [-3, 2, -5, -4, 1]$. Devido a sua simplicidade, iremos adotar a representação do diagrama de ciclos usando a definição de Bergeron. Além da variação de definição do diagrama de ciclos, alguns autores consideram uma maneira conveniente de representar graficamente o diagrama. O diagrama é tradicionalmente representado com os vértices dispostos em uma linha, como mostrado no topo da Figura 2.3. No entanto, pode ser mais conveniente apresentar o diagrama de ciclos de um modo “circular”, como ilustrado na parte inferior da Figura 2.3. Como Hannenhalli e Pevzner [28], e Kaplan *et al.* [32] já destacaram em seus trabalhos, circularizar a imagem de uma permutação permite maior uniformidade para lidar com obstáculos, que serão definidos na Seção 2.5.

Dado um diagrama de ciclos $B(\pi)$, um ciclo de $B(\pi)$ é chamado *alternante* quando quaisquer duas arestas incidentes em um mesmo vértice possuem cores distintas. De agora em diante, quando nos referirmos a ciclos do diagrama de ciclos estaremos falando de ciclos alternantes e também poderemos nos referir aos “ciclos da permutação” quando estamos de fato falando

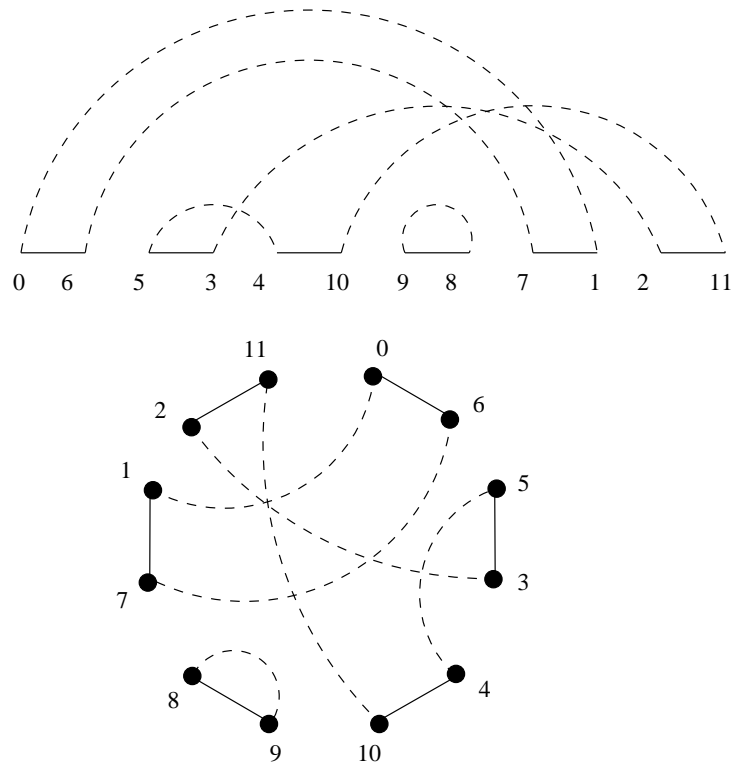


Figura 2.3: Duas maneiras diferentes de se desenhar o diagrama de ciclos da permutação com sinais $\pi = [-3, 2, -5, -4, 1]$. Na parte superior da figura, o diagrama é desenhado com as aresta pretas alinhadas ao longo de uma reta horizontal (as arestas cinzas são representadas por linhas tracejadas). Na parte inferior da figura, O mesmo diagrama é desenhado com as arestas pretas dispostas em torno de um círculo. Em ambos os casos a ordem dos vértices é dada por π' .

dos ciclos no diagrama de ciclos definido pela permutação. O diagrama de ciclos possui uma propriedade fundamental: como cada vértice tem grau exatamente 2, há uma única decomposição em ciclos no diagrama. O *comprimento* de um ciclo é o número de arestas pretas no mesmo. Seja $c(\pi)$ o número de ciclos em $B(\pi)$ (esse número é bem definido devido a única decomposição em ciclos do diagrama). Como uma consequência, por permitir ciclos de tamanho unitário na representação do diagrama de ciclos de Bergeron, o valor $c(\pi)$ definido por Bergeron é maior que o número de ciclos definidos por Hannenhalli e Pevzner pelo valor de $n + 1 - b(\pi)$. Denotando por $c_{\geq 2}(\pi)$ o número de ciclos com comprimento de pelo menos 2, temos

$$c(\pi) = n + 1 - b(\pi) + c_{\geq 2}(\pi).$$

Adotamos nessa apresentação do formalismo clássico o número de ciclos $c(\pi)$ que inclui os ciclos unitários.

Para analisar a estrutura complexa do diagrama de ciclos e os efeitos de uma reversão com sinais em uma permutação e no diagrama, Hannenhalli e Pevzner constroem uma permutação equivalente por meio de *operações de padding* [28]. O diagrama de ciclos da permutação obtida por meio das operações de padding é composto somente por ciclos de tamanho dois. Essa permutação equivalente contém a mesma informação da permutação original e simplifica a análise da influência de reversões com sinais no número de ciclos. Mais tarde, Kaplan *et al.* [32] descobriram que ao se focar nos *arcos*, i.e. arestas cinzas, ao invés de ciclos, é possível abandonar as operações de padding e obter a informação desejada sobre os efeitos de uma reversão com sinais diretamente da permutação original. Devido a esse desenvolvimento teórico, não apresentamos os efeitos de operações de padding em permutações.

Dado um diagrama de ciclos $B(\pi)$, seja $g = (\pi'_i, \pi'_j)$ com π'_i par, uma aresta cinza no diagrama de ciclos $B(\pi)$. Definimos o *intervalo induzido por g* como o conjunto de inteiros não negativos $[i, j] = \{k \mid i \leq k \leq j\}$. Arestas cinzas (π'_i, π'_j) e (π'_a, π'_b) são *entrelaçadas* quando os intervalos $[i, j]$ e $[a, b]$ são tais que $[i, j] \cap [a, b] \neq \emptyset$, $[i, j] \not\subseteq [a, b]$ e $[a, b] \not\subseteq [i, j]$. Dois ciclos C_1 e C_2 são *entrelaçados* quando há arestas cinzas entrelaçadas $g_1 \in C_1$ e $g_2 \in C_2$ no diagrama de ciclos.

Denotamos por $\Delta c(\rho, \pi)$ a diferença no número de ciclos $c(\pi\rho) - c(\pi)$. Uma reversão com sinais é *própria* quando $\Delta c(\rho, \pi) = 1$. Cada par de arestas pretas (π'_{2i}, π'_{2i+1}) e (π'_{2j}, π'_{2j+1}) , para $0 \leq i < j \leq n$, no diagrama de ciclos define uma reversão com sinais $\rho(i, j)$. Uma reversão com sinais ρ *atua em* uma aresta cinza g quando ela é definida pelas arestas pretas incidentes à aresta cinza g . Uma aresta cinza é *orientada* quando a reversão com sinais atuando nela é própria, caso contrário ela é *não orientada*. Um ciclo é *orientado* quando possuir uma aresta cinza orientada entre as suas arestas, caso contrário é *não orientado*. Se existe uma sequência de reversões com sinais próprias que ordena π , então $d_r(\pi) = n + 1 - c(\pi)$. Em qualquer caso, temos $\Delta c(\rho, \pi) \leq 1$ [6]. Logo, um limitante inferior mais justo para a distância de reversão

com sinais é:

$$n + 1 - c(\pi) \leq d_\rho(\pi),$$

uma vez que o diagrama de ciclos da permutação identidade possui $n + 1$ ciclos. A permutação identidade $[0, 1, 2, \dots, 2n + 1]$ é a única permutação cujo diagrama de ciclos possui $n + 1$ ciclos e por esta razão poderíamos acreditar que para encontrar uma seqüência mínima de reversões com sinais bastaria encontrar reversões que aumentem o número de ciclos do diagrama de ciclos. Infelizmente, reversões com sinais que aumentam o número de ciclos (reversões com sinais próprias) nem sempre pertencem a uma seqüência mínima que ordena uma permutação.

Exemplo 5. Considere a permutação com sinais $\pi = [4, 3, 1, -5, -2]$, que possui $c(\pi) = 3$. Existem duas reversões com sinais próprias que podemos aplicar em π : $\rho_1 = \rho(4, 5)$ e $\rho_2 = \rho(2, 4)$. No primeiro caso, a permutação resultante $\pi\rho_1$ não possui nenhuma reversão com sinais próprias e portanto sua distância é pelo menos 3. Por outro lado, a permutação $\pi\rho_2$ não apenas tem uma reversão com sinais própria mas também admite uma seqüência de reversões com sinais que a ordena, o que atesta que $d_r(\pi) = 3$. Podemos concluir que ρ_2 é uma reversão com sinais ordenante enquanto ρ_1 não é.

2.4 Grafo de Sobreposição

Kaplan, Shamir e Tarjan [32] definem um grafo baseado na relação de entrelaçamento entre arestas cinzas (arcos) do diagrama de ciclos. Dada uma permutação com sinais π , o *grafo de sobreposição* de π , denotado por $OV(\pi)$, é o grafo cujo conjunto de vértices é o conjunto de arcos de $B(\pi)$ e dois vértices são ligados por uma aresta em $OV(\pi)$ se as arestas cinzas correspondentes em $B(\pi)$ são entrelaçadas (adotamos uma nomenclatura distinta da de Kaplan, Shamir e Tarjan [32], pois o que chamamos de grafo de sobreposição, aqueles autores chamam de grafo de entrelaçamento). Um vértice em $OV(\pi)$ é *orientado* se a aresta cinza correspondente em $B(\pi)$ é orientada, caso contrário o vértice é *não orientado*. Um componente em $OV(\pi)$ é *orientado* quando ele contém um vértice orientado, caso contrário ele é *não orientado*. A Figura 2.4 ilustra um grafo de sobreposição.

O grafo $OV(\pi\rho)$ obtido por meio da aplicação de uma reversão com sinais ρ definida por um vértice orientado v no grafo de sobreposição $OV(\pi)$ consiste no grafo com o mesmo conjunto de vértices, mas com o subgrafo contendo v e seus vértices vizinhos complementado e a orientação dos vértices nesse subgrafo trocadas [32]. A Figura 2.5 ilustra o efeito de uma reversão com sinais em um grafo de sobreposição.

Dado o grafo $OV(\pi)$ definido para uma permutação com sinais π , uma reversão com sinais que não aumenta o número de componentes não orientados em $OV(\pi\rho)$ em relação a $OV(\pi)$, exceto por vértices isolados, é chamada de reversão *segura*.

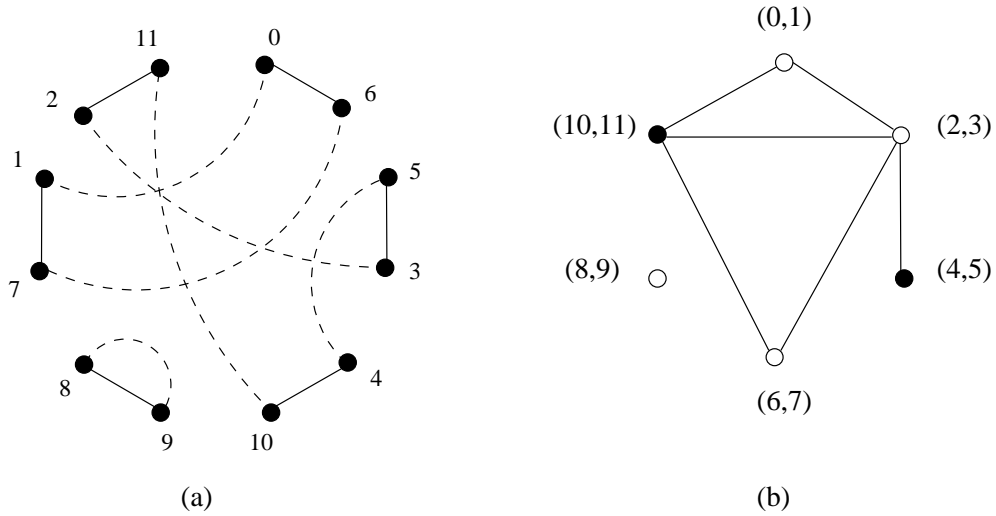


Figura 2.4: a) O diagrama de ciclos do genoma $\pi = [-3, 2, -5, -4, 1]$. b) O grafo de sobreposição correspondente $OV(\pi)$ de π . Para cada vértice no grafo de sobreposição há uma aresta cinza correspondente no diagrama de ciclos. Vértices orientados são pretos, enquanto vértices não orientados são brancos.

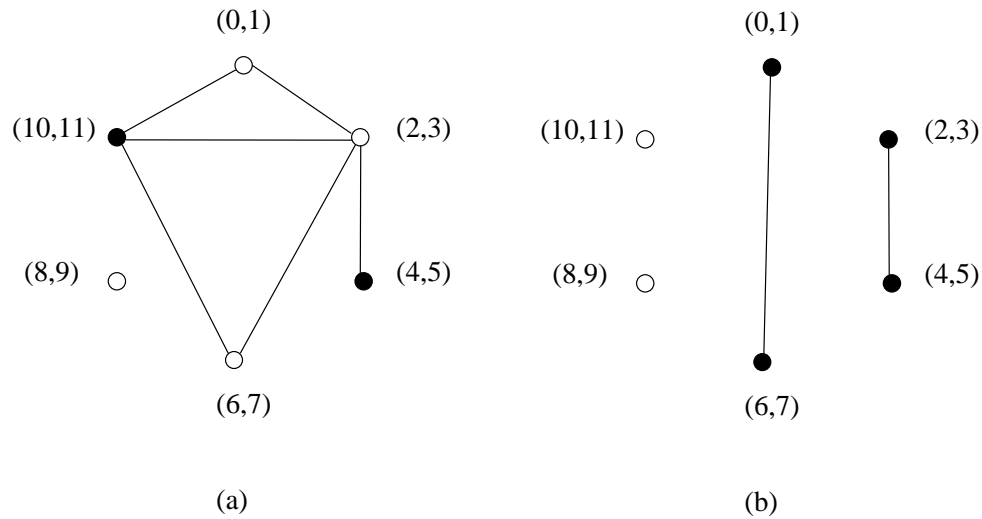


Figura 2.5: Efeito de uma reversão com sinais em um grafo de sobreposição. a) Grafo de sobreposição do genoma $\pi = [-3, 2, -5, -4, 1]$. b) O grafo de sobreposição $OV(\pi\rho)$ onde $\pi\rho = [-3, 2, -1, 4, 5]$ e ρ é definido pelo vértice $v = (10, 11)$. O grafo $OV(\pi\rho)$ é obtido por meio da seguinte transformação de $OV(\pi)$: o grafo $OV(\pi\rho)$ possui o mesmo conjunto de vértices de $OV(\pi)$, mas o subgrafo contendo v e seus vértices vizinhos em $OV(\pi)$ aparece complementado e a orientação dos vértices nesse subgrafo são trocadas em $OV(\pi\rho)$.

Anne Bergeron [8] propôs uma simplificação da teoria utilizada no problema de ordenação por reversões com sinais. Em seu trabalho, a permutação com sinais é colocada em foco ao invés das estruturas derivadas como o diagrama de ciclos. Bergeron identificou alguns dos conceitos e parâmetros apresentados anteriormente na própria permutação, como por exemplo as reversões próprias e obstáculos (veja Seção 2.5).

Faremos uma breve descrição dos principais pontos dessa nova visão sobre a teoria. Seja π uma permutação com sinais em sua versão estendida. Um *par orientado* (π_i, π_j) de π é um par de inteiros com sinais opostos que seriam consecutivos se tomados em valor absoluto, i.e., temos $||\pi(i)| - |\pi(j)|| = 1$, caso contrário o par de inteiros é chamado *par não orientado*. Por exemplo, na permutação com sinais

$$[0, -2, 3, -5, 4, 1, 6],$$

os pares orientados são $(-2, 1)$, $(-2, 3)$, $(-5, 4)$ e $(-5, 6)$. Para cada par orientado de uma permutação há uma única aresta cinza correspondente em $B(\pi)$. Logo, um par orientado induz uma reversão com sinais própria que é a reversão com sinais que atua na aresta cinza correspondente. A reversão com sinais induzida pelo par orientado (π_i, π_j) , que Bergeron chama de *reversão orientada*, é

$$\rho(i, j - 1), \text{ se } \pi_i + \pi_j = +1$$

ou

$$\rho(i + 1, j), \text{ se } \pi_i + \pi_j = -1.$$

Observando-se o efeito de uma reversão com sinais induzida ρ quando aplicada a π , pode-se verificar que ao menos uma adjacência é criada em $\pi\rho$; em outros termos, o número de ciclos em relação a $B(\pi)$ aumenta de ao menos um, uma vez que um ciclo novo de comprimento um aparece em $B(\pi\rho)$. Por exemplo, considere a permutação seguinte

$$\pi = [0, -2, 3, -5, 4, 1, 6].$$

A reversão com sinais induzida pelo par orientado $(-2, 1)$ é $\rho(2, 5)$. Temos:

$$\pi\rho(2, 5) = [0, -2, -1, -4, 5, -3, 6].$$

A permutação π acima possui 6 pontos de quebra e 4 pares orientados, enquanto $\pi\rho(2, 5)$ possui 5 pontos de quebra e 2 pares orientados. Toda reversão com sinais orientada é própria, embora há reversões com sinais próprias que não são orientadas no sentido oferecido por Bergeron [8]. Comentamos anteriormente que seria tentador utilizarmos reversões com sinais próprias, muito embora elas possam não ser reversões ordenantes. A questão crucial que se coloca é: quais reversões com sinais orientadas também são reversões ordenantes? Bergeron

propõe um novo parâmetro associado a reversão com sinais próprias que parcialmente responde a essa questão. Dada a permutação π , a *pontuação* de uma reversão com sinais orientada ρ , denotada por $p(\rho)$, é o número de pares orientados em $\pi\rho$. A pontuação de uma reversão com sinais orientada ρ pode ser obtida por meio do grafo de sobreposição da seguinte forma. Seja v o vértice orientado no qual a reversão com sinais orientada ρ atua. O valor da pontuação de ρ é $p(\rho) = T + U - O - 1$, onde T é o número de vértices orientados no grafo de sobreposição de π , o valor U é o número de vértices não orientados adjacentes a v e o valor O é o número de vértices orientados adjacentes a v . Uma reversão com sinais orientada e com pontuação máxima é segura [8].

Bergeron afirma que escolher uma reversão com sinais orientada com a pontuação máxima em cada passo do processo de ordenação é uma estratégia ótima. Esse fato é afirmado no seguinte resultado:

Proposição 6 (Bergeron [8]). Se a aplicação sucessiva de uma sequência de k reversões com sinais orientadas com máxima pontuação à permutação π leva à permutação θ com nenhum elemento negativo, então $d(\pi) = d(\theta) + k$.

A Proposição 6 afirma que reversões orientadas com máxima pontuação são reversões ordenantes. Aplicando-se reversões desse tipo na permutação π enquanto for possível irá faticamente levar a uma permutação que não contém nenhum par orientado. Se a permutação resultante não é a permutação identidade, precisamos encontrar outra maneira para escolher reversões ordenantes. Na Seção 2.5, discutiremos o conceito de obstáculos e o tipo de reversão com sinais que pode eliminá-los.

2.5 Obstáculos

O conceito de obstáculos foi apresentado primeiramente por Hannenhalli e Pevzner [29], no entanto, iremos apresentar uma definição equivalente e mais simples de obstáculos apresentada por Kaplan, Shamir e Tarjan [32].

Dada uma permutação π e seu grafo de sobreposições $OV(\pi)$, seja x_1, \dots, x_k uma subsequência dos elementos $\pi_0, \pi_1, \dots, \pi_{n+1}$ que incidem em arcos de componentes não orientados com mais de um vértice de $OV(\pi)$. Seja CR uma circunferência na qual os elementos da subsequência são dispostos tal que x_{i+1} segue a x_i para $1 \leq i \leq k-1$ e x_1 segue a x_k . Para um dado componente não orientado M de $OV(\pi)$ com mais de um nó, seja $V(M) \subseteq \{x_1, \dots, x_k\}$ o conjunto dos vértices em $B(\pi)$ que sejam incidentes aos arcos de M . Um componente não orientado M com mais de um nó é um *obstáculo* quando os elementos de $V(M)$ aparecem consecutivamente na circunferência CR . Ao se utilizar a representação circular do diagrama de ciclos, é possível identificar quais componentes em $B(\pi)$ correspondem a obstáculos em $OV(\pi)$, pois o diagrama de ciclos $B(\pi)$, descartados os ciclos de comprimento unitário e os

componentes bons, dispõe a subsequência de elementos x_1, \dots, x_k como em CR . Componentes não orientados de $OV(\pi)$ que não são obstáculos são chamados *não obstáculos*. Denotamos por $h(\pi)$ o número de obstáculos de π . De maneira semelhante a pontos de quebra e ciclos, a variação no número de obstáculos resultante da aplicação de uma reversão com sinais é denotada por $\Delta h(\rho, \pi) = h(\pi\rho) - h(\pi)$.

Exemplo 7. Considere a permutação com sinais $\pi = [11, 10, 1, 8, 7, 2, 5, 4, 3, 6, 9]$ definida no conjunto $\{x \in \mathbb{Z} \mid 1 \leq |x| \leq 11\}$.

A Figura 2.6 mostra o diagrama de ciclos $B(\pi)$ e o grafo de sobreposição $OV(\pi)$ de π . Os vértices que incidem nos nós dos componentes A' e C' aparecem consecutivamente ao longo da representação circular de $B(\pi)$ e portanto A' e C' são obstáculos, ao contrário do componente B' que não é um obstáculo pois os vértices incidentes nos arcos de B' não aparecem consecutivamente ao longo da representação circular de $B(\pi)$.

Hannenhalli e Pevzner [29] observaram que uma reversão com sinais ρ atuando em um arco de um obstáculo K de uma permutação π , quando aplicada a π , transforma o obstáculo K em um componente orientado com o mesmo número de ciclos em $B(\pi)$ e cada ciclo com o mesmo comprimento anterior. O obstáculo K foi eliminado de $OV(\pi\rho)$, isto é, o componente torna-se orientado, no entanto outros obstáculos podem surgir em $OV(\pi\rho)$. Tal operação para eliminar um obstáculo é chamada de *corte de obstáculo*, e dizemos que a reversão com sinais ρ *corta* o obstáculo. Essas definições nos ajudam a distinguir entre dois tipos de obstáculos. Um obstáculo para o qual existe uma reversão com sinais atuando em um dos seus arcos tal que $\Delta h(\rho, \pi) = -1$ é chamado de *obstáculo simples*, caso contrário ele é um *super obstáculo*. A Figura 2.7 ilustra os diversos tipos de componentes na permutação com sinais

$$\pi = [1, 3, 5, 7, 6, 8, 4, 9, 11, 10, 12, 2, 13, 15, 14, 16].$$

Dada uma permutação com sinais π , de acordo com Hannenhalli e Pevzner [29], uma reversão com sinais ρ é uma *reversão segura* se $\Delta b(\rho, \pi) - \Delta c_{\geq 2}(\rho, \pi) + \Delta h(\rho, \pi) = -1$, o que corresponde a $\Delta h(\rho, \pi) - \Delta c(\rho, \pi) = -1$ em nossa notação. Kaplan *et al.* [32] definem uma reversão segura de maneira mais restritiva, uma vez que esses autores requerem que a reversão com sinais deve ser própria para ser uma reversão segura. Relembrando que uma reversão própria satisfaz $\Delta c(\rho, \pi) = 1$. No caso em que uma reversão segura atua em componentes orientados, as duas definições de reversões seguras são equivalentes. Uma reversão com sinais atuando em um arco de um obstáculo simples é uma reversão segura [29].

Seja π uma permutação com pelo menos dois obstáculos K e L no grafo $OV(\pi)$. O *conjunto de pontos finais* de um obstáculo K , denotado por $EP(K)$, é formado pelos vértices mais extremos do arco em CR que contém todos os vértices incidentes a arcos de K em $OV(\pi)$. Chamamos os obstáculos K e L de *consecutivos* quando ambos os conjuntos de pontos finais $EP(K)$ e $EP(L)$ são consecutivos em CR , isto é, não há um obstáculo M tal que $EP(M)$

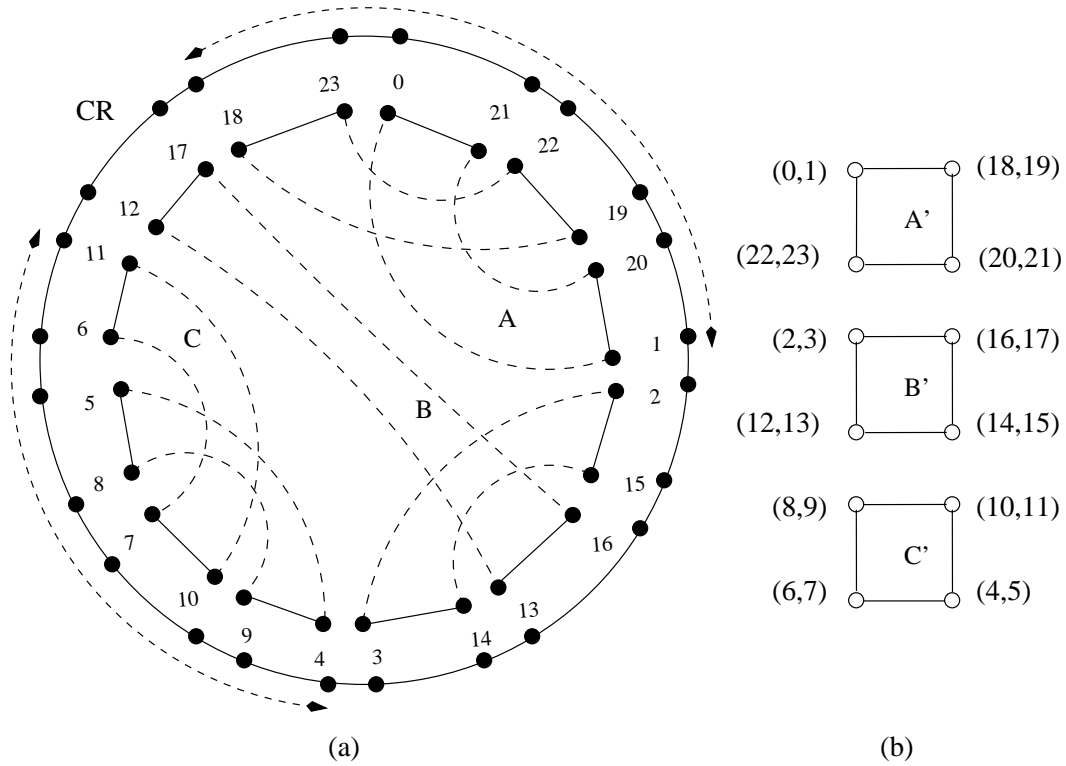


Figura 2.6: Exemplo de obstáculos de uma permutação. (a) Diagrama de ciclos da permutação $\pi = [11, 10, 1, 8, 7, 2, 5, 4, 3, 6, 9]$ e correspondente circunferência CR com os vértices incidentes aos arcos dos componentes não orientados. O diagrama de ciclos é composto por três componentes não orientados A , B , e C . Os vértices desses componentes são representados por pontos de mesmo nome na circunferência CR . Os pontos em CR que pertencem ao mesmo componente de $B(\pi)$ e aparecem consecutivamente em CR são indicados pelos arcos tracejados. (b) Grafo de sobreposição $OV(\pi)$ obtido do diagrama de ciclos anterior. Os componentes não orientados A' , B' e C' de $OV(\pi)$ correspondem aos componentes A , B e C de $B(\pi)$. Os componentes A' e C' são obstáculos de π , pois os pontos correspondentes a vértices dos componentes A e C aparecem consecutivamente em CR .

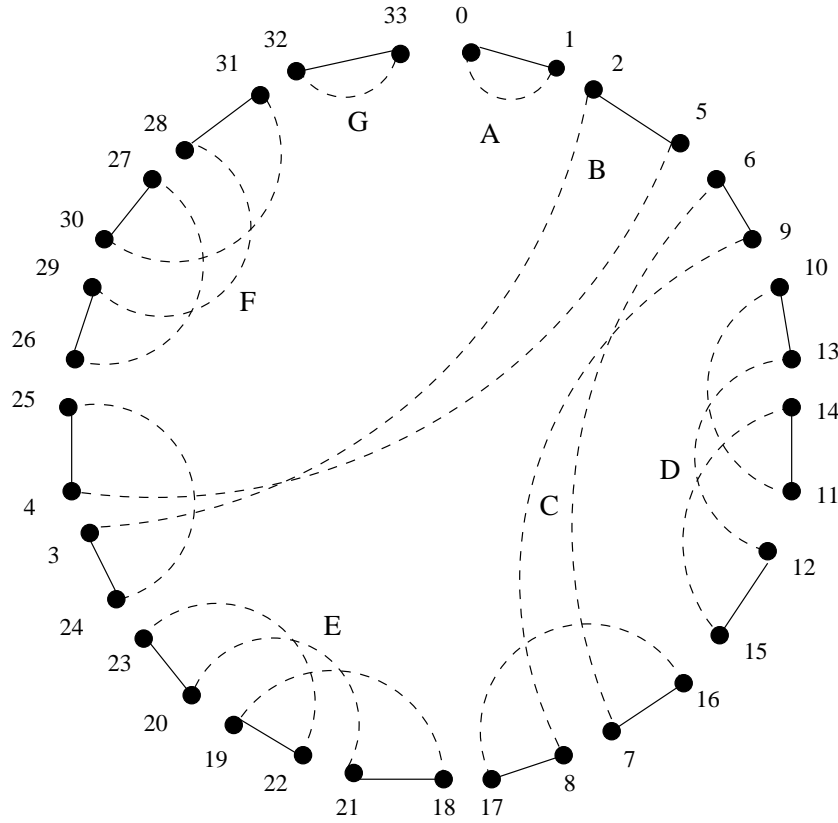


Figura 2.7: Diagrama de ciclos da permutação com sinais π . Os componentes em $B(\pi)$ que correspondem a obstáculos e não obstáculos, bem como obstáculos simples e super obstáculos são mostrados no diagrama. O diagrama de ciclos $B(\pi)$ não contém nenhum componente orientado. Os vértices do componente não orientado B não aparecem consecutivamente no diagrama logo o componente correspondente em $OV(\pi)$ é um não obstáculo. Pelo mesmo motivo, o componente em $OV(\pi)$ correspondente a C também é um não obstáculo. Por outro lado, os componentes em $OV(\pi)$ correspondentes a D , E e F são obstáculos, uma vez que os vértices de cada um deles aparecem consecutivamente no diagrama. Se o obstáculo correspondente a D é cortado, o componente correspondente a C em $OV(\pi)$ torna-se um obstáculo, logo D é um super obstáculo. O mesmo ocorre com o componente correspondente a F que também é um super obstáculo. Por outro lado, o componente correspondente a E pode ser cortado causando uma redução no número de obstáculos em $OV(\pi)$, então aquele componente correspondente a E é um obstáculo simples. Os componentes correspondentes a A e G são não obstáculos porque possuem apenas um arco cada (são componentes com um único vértice em $OV(\pi)$).

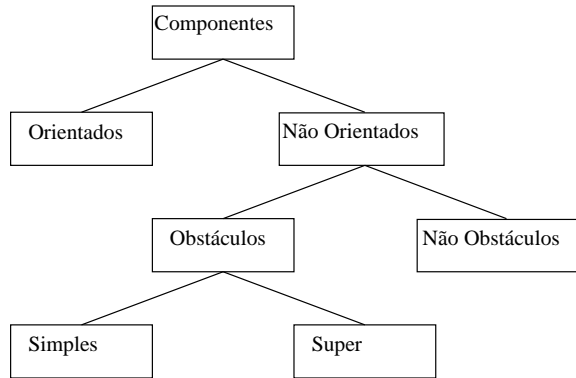


Figura 2.8: Classificação de componentes de uma permutação.

separe $EP(K)$ de $EP(L)$; caso contrário, os obstáculos K e L são *não consecutivos*. Por exemplo, os obstáculos A' e C' mostrados na Figura 2.6 são consecutivos, pois o componente não orientado B em $B(\pi)$ não corresponde a um obstáculo em $OV(\pi)$. Dados os obstáculos K e L em $OV(\pi)$, uma reversão com sinais ρ definida por uma aresta preta incidente a um arco de K e uma aresta preta incidente a um arco de L em $B(\pi)$ é chamada de *junção de obstáculos*. A junção de obstáculos ρ elimina os obstáculos não consecutivos K e L em $\pi\rho$ [29, 32]. Hannenhalli e Pevzner [29] demonstram, usando outros termos, a Proposição 8 e Proposição 9 a seguir:

Proposição 8. Seja π uma permutação com um número par $2k$ de obstáculos. Qualquer sequência de $k - 1$ junções de obstáculos que elimine dois obstáculos não consecutivos seguida por uma junção dos dois obstáculos restantes transforma π em uma permutação θ tal que $d_r(\theta) = d_r(\pi) - k$ e θ possui apenas componentes orientados.

Uma permutação π é chamada de uma *fortaleza* quando π possui um número ímpar de obstáculos e todos eles são super obstáculos (ver Figura 2.9). Seja $f(\pi)$ igual a 1 quando π é uma fortaleza e 0, caso contrário.

Proposição 9. Seja π uma permutação com um número ímpar $2k + 1$ de obstáculos.

- se houver ao menos um obstáculo simples H em π então uma reversão ρ atuante em um arco de H corta o obstáculo H , enquanto que $\pi\rho$ tem $2k$ obstáculos e $d_r(\pi\rho) = d_r(\pi) - 1$.
- se π é uma fortaleza então uma sequência de $k - 1$ junções de obstáculos que elimine dois obstáculos não consecutivos seguida por duas junções de dois obstáculos consecutivos (uma junção elimina dois obstáculos originais enquanto a seguinte elimina dois obstáculos criados pelo primeiro e último obstáculos originais) transforma a permutação π em θ tal que $d_r(\theta) = d_r(\pi) - (k + 1)$ e θ possui apenas componentes orientados.

Temos agora todos os dados necessários para resolver o problema de ordenação por reversões com sinais. Como exemplo, temos o algoritmo proposto por Kaplan, Shamir e Tarjan [32] que consiste em primeiramente aplicar reversões que eliminem os obstáculos para transformar a permutação de entrada π em uma permutação θ que contenha apenas componentes orientados em $OV(\theta)$, seguido da aplicação de uma seqüência de reversões orientadas com máxima pontuação (reversões seguras). A próxima proposição contém a fórmula para encontrar a distância de reversão com sinais envolvendo os parâmetros discutidos anteriormente: o número de ciclos de $B(\pi)$, o número de obstáculos em $OV(\pi)$ e o fato de π ser ou não uma fortaleza.

Proposição 10 (Hannenhalli e Pevzner[28]). Dada a permutação π , então:

$$d_r(\pi) = n + 1 - c(\pi) + h(\pi) + f(\pi)$$

A fórmula para a distância de reversão com sinais nos permite calcular o diâmetro de reversão com sinais.

Proposição 11. O valor do diâmetro $d_r(n)$ para todos os genomas em $Gen(E)$ satisfaz:

$$\begin{aligned} d_r(1) &= 1, \\ d_r(2) &= 2, \\ d_r(n) &= n + 1, \text{ para } n \geq 3. \end{aligned}$$

2.6 Diagramas de ciclos e Transposições

O problema de ordenação por transposições demonstrou ser até o momento um problema mais difícil do que o problema de ordenação por reversões com sinais. Não sabemos se existe um algoritmo polinomial que resolva o problema de ordenação por transposições ou se o problema é *NP*-difícil. Bafna e Pevzner [7] apresentaram o primeiro algoritmo aproximado de tempo de execução polinomial para o problema de ordenação por transposições. Nessa seção, apresentamos a abordagem de Bafna e Pevzner para o problema e mostramos um algoritmo com fator de aproximação de 1,5.

Bafna e Pevzner [7] e Hartman e Shamir [30] apresentam versões diferentes do diagrama de ciclos de uma permutação π . Bafna e Pevzner definem um vértice para cada elemento em π e o resultado final é um diagrama de ciclos de uma permutação, tal que os vértices podem ter até grau quatro. Hartman e Shamir definem o seu diagrama de ciclos supondo que o genoma unicromossomal é uma permutação com sinais e precisa ser convertido para a sua imagem para obtermos o diagrama de ciclos, como na Seção 2.3. Uma diferença relevante é que Hartman e Shamir consideram cromossomos circulares, observando que há uma correspondência direta entre as versões linear e circular do problema. Como vimos na seção anterior ao lidar com os

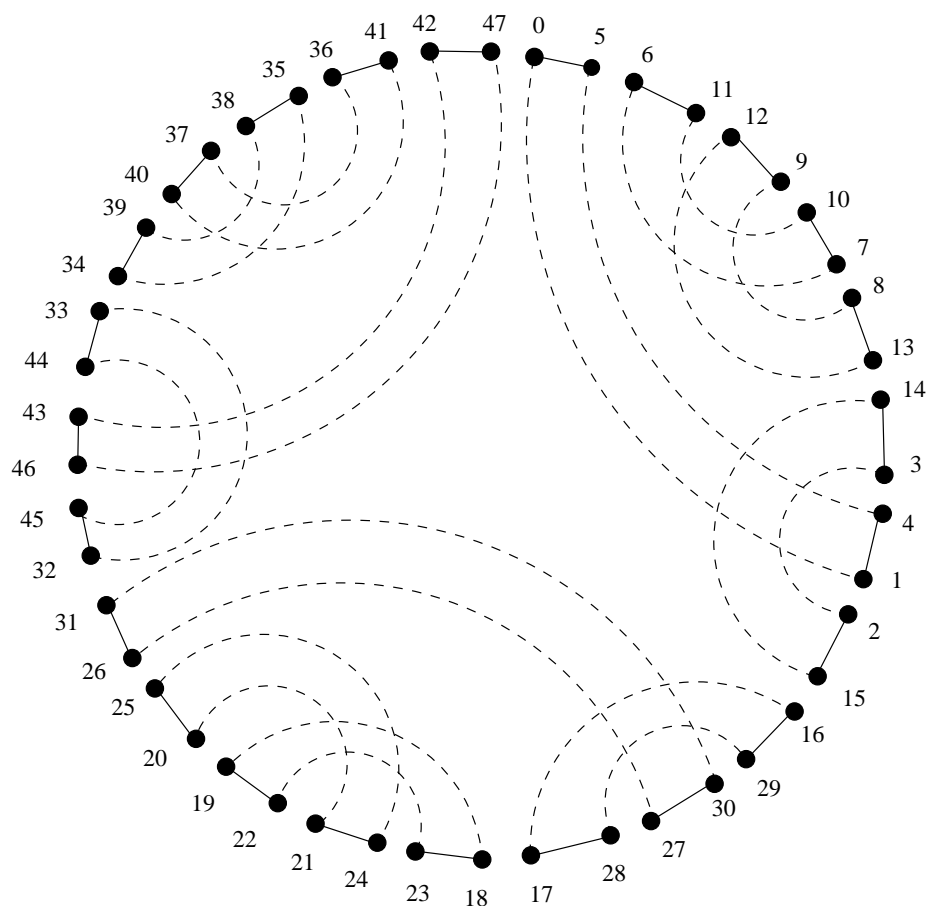


Figura 2.9: O genoma π é uma fortaleza: todos os três obstáculos no diagrama de ciclos são super obstáculos.

obstáculos de uma permutação, certas vezes é mais conveniente lidar com uma lista circular de elementos do que uma sequência (permutação) de elementos.

Seguiremos a mesma linha de apresentação desenvolvida para o caso de reversões com sinais. Como antes, um genoma unicromossomal é uma permutação com sinais. Como transposições nunca trocam sinais, iremos considerar que todos os elementos são não negativos no problema de ordenação por transposições. Logo, para um genoma π a sua imagem π' é obtida ao se substituir cada elemento $\pi(i)$, para $1 \leq i \leq n$ por $2\pi(i) - 1$ e $2\pi(i)$ nessa ordem. Além disso, a versão estendida da imagem de π possui os elementos $\pi'(0) = 0$ e $\pi'(2n+1) = 2n+1$. Uma transposição $\tau(k, l, m)$ em π' é chamada *legal* quando k, l, m são ímpares não negativos. Observe que esse conceito de transposição legal é análogo a uma noção implícita no trabalho de Hannehalli e Pevzner [29], que proíbe cortes em π' em posições que não existem em π . Para qualquer transposição τ em uma permutação com sinais π , existe uma única transposição legal correspondente τ' tal que $(\pi\tau)' = \pi'\tau'$. Uma vez que o diagrama de ciclos é definido do mesmo modo que foi definido para o caso de reversões com sinais, mantemos a mesma notação para o número de ciclos $c(\pi)$ (incluindo ciclos de comprimento unitário). Um ciclo é um *k-ciclo* quando o seu comprimento é k . Um *k-ciclo* é um *ciclo longo* quando $k > 2$, caso contrário ele é um *ciclo curto*.

Dada a permutação com sinais π e uma transposição τ , seja $\Delta c(\tau, \pi) = c(\pi\tau) - c(\pi)$ a diferença no número de ciclos quando τ é aplicada a π . Bafna e Pevzner [7] provam que $\Delta c(\tau, \pi) \in \{-2, 0, 2\}$. A partir dessa observação, eles deduziram o seguinte limitante inferior para a distância de transposição:

Proposição 12. Para qualquer permutação com sinais π definida num conjunto de n genes, temos

$$d_t(\pi) \geq \frac{n+1-c(\pi)}{2}.$$

Bafna e Pevzner [7] melhoraram ainda mais o limitante inferior dado pela Proposição 12 ao levarem em conta a paridade dos ciclos de $B(\pi)$. Um ciclo é *ímpar* se o seu comprimento é ímpar, caso contrário o ciclo é *par*. Dada uma permutação π , sejam $c_{\text{odd}}(\pi)$ e $c_{\text{even}}(\pi)$ respectivamente o número de ciclos ímpares e o número de ciclos pares em $B(\pi)$. Um limitante inferior mais justo para a distância de transposição segue-se:

Proposição 13. Para qualquer permutação com sinais π definida no conjunto de n genes, temos

$$d_t(\pi) \geq \frac{n+1-c_{\text{odd}}(\pi)}{2}.$$

Dada uma permutação com sinais π , uma transposição τ é chamada de um *x-movimento* quando $\Delta c(\pi\tau, \pi) = x$ onde $x \in \{-2, 0, 2\}$.

Proposição 14. Se π é uma permutação com sinais distinta da identidade, então existe um 2-movimento ou um 0-movimento seguido por um 2-movimento em π .

Dada a permutação π , no pior caso, o número de ciclos em $B(\pi\tau)$ seria aumentado de um ciclo apenas comparado a $B(\pi)$. Logo, um limitante superior para a distância de transposição segue diretamente da Proposição 14:

Proposição 15. Qualquer permutação com sinais π definida no conjunto de n genes pode ser ordenada com $n + 1 - c(\pi)$ transposições.

Proposição 15 permite projetar um algoritmo aproximativo para o problema de ordenação por transposições cuja solução não é maior que o dobro da solução ótima. O algoritmo de ordenação é baseado em uma estratégia gulosa de encontrar 2-movimentos ou seqüências de 0 – 2-movimentos. Observe que um 2-movimento não é necessariamente uma transposição ordenante. Bafna e Pevzner [7] mostram que uma melhoria significativa no algoritmo aproximativo pode ser obtida se tentarmos encontrar transposições que aumentem o número de ciclos ímpares no diagrama de ciclos.

Dada uma permutação com sinais π e uma transposição τ , dizemos que τ é uma *transposição válida* quando $\Delta c(\tau, \pi) = \Delta c_{odd}(\tau, \pi)$. Investigando o papel de algumas estruturas auto-entrelaçadas no diagrama de ciclos, Bafna e Pevzner [7] determinaram as condições suficientes para se encontrar movimentos válidos em uma permutação.

Proposição 16. Se existe um ciclo longo em $B(\pi)$, onde π é uma permutação com sinais, então existe um 2-movimento válido em π ou um 0-movimento válido seguido por dois 2-movimentos válidos consecutivos.

Observe que a Proposição 16 vale apenas para permutações que possuam ciclos longos. Uma análise diversa é necessária para permutações que são compostas somente por ciclos curtos. Bafna e Pevzner [7] encontraram transposições atuando em dois ciclos curtos que aumentam o número de ciclos ímpares, embora o número de ciclos na permutação mantenha-se o mesmo. Dada uma permutação π , um 0-movimento τ é chamado *bom* se ele cria dois novos ciclos ímpares em $B(\pi\tau)$.

Proposição 17. Se $B(\pi)$ tem apenas ciclos curtos, então existe um 0-movimento bom seguido de um 2-movimento válido em π .

Combinando a Proposição 16 e a Proposição 17, Bafna e Pevzner [7] propuseram o seguinte algoritmo aproximativo de fator de aproximação 1,5:

Proposição 18. O algoritmo *TransSort* ordena π em tempo de execução $O(n^2)$ ao usar não mais que $\frac{3}{4}(n + 1 - c_{odd}(\pi))$ transposições, sendo um algoritmo de fator de aproximação 1,5.

O algoritmo *TransSort* ordena qualquer permutação com no máximo $\frac{3}{4}(n + 1 - c_{odd}(\pi))$ transposições. Logo, um limitante superior mais justo para o diâmetro de transposição pode ser obtido.

Algoritmo TransSort

```

1: while  $\pi$  não está ordenado do
2:   if  $B(\pi)$  tem um ciclo longo then
3:     Aplique um 2-movimento válido ou um 0–2–2-movimento válido a  $\pi$ .
4:   else  $\{B(\pi)$  tem apenas ciclos curtos $\}$ 
5:     Aplique um 0-movimento bom seguido por um 2-movimento válido a  $\pi$ .
6:   end if
7: end while

```

Figura 2.10: Algoritmo para ordenar uma permutação π .

Corolário 19 (Diâmetro de Transposição). O diâmetro de transposição de $Gen(E)$ é no máximo $\frac{3n}{4}$.

Bafna e Pevzner [7] verificaram que $d_\tau(n) = \lfloor n/2 \rfloor + 1$, para $3 \leq n \leq 10$ e que a *permutação reversa*, isto é, a permutação que mapeia i em $n - i + 1$ para $1 \leq i \leq n$, satisfaz essa distância de transposição. Christie [17] e Meidanis, Walter e Dias [41] demonstraram que de fato a distância de transposição da permutação reversa é $\lfloor n/2 \rfloor + 1$ e conjecturaram que a distância de transposição da reversa seria o valor do diâmetro de transposição. Eriksson *et al.* [23] encontraram valores para o diâmetro de transposição (para $n = 13$ e $n = 15$) que falsificam a conjectura anterior; além disso, demonstraram que é possível ordenar uma permutação com no máximo $\lfloor (2n - 2)/3 \rfloor$ transposições para $n \geq 9$, ou seja, um limitante superior mais justo para o diâmetro de transposição. Elias e Hartman [22] demonstram que $\lfloor (n + 1)/2 \rfloor + 1$ é um limitante inferior mais justo para o diâmetro de transposição.

Proposição 20 (Diâmetro de Transposição). Para $Gen(E)$ e $n \geq 9$, temos:

$$\left\lfloor \frac{n+1}{2} \right\rfloor + 1 \leq d_\tau(n) \leq \left\lfloor \frac{2n-2}{3} \right\rfloor$$

2.7 Outros resultados em Rearranjos em Genomas

O problema de ordenação por reversões com sinais é o problema mais investigado e bem compreendido até o momento. O primeiro algoritmo exato de tempo de execução polinomial foi proposto por Hannenhalli e Pevzner [28]. Diversas melhorias e novos conceitos foram sugeridos e adicionados à teoria original dos autores [32, 8], culminando com o algoritmo exato de tempo de execução subquadrático de Tannier e Sagot [56]. Bader, Moret e Yan [4] ofereceram um algoritmo que calcula a distância de reversão com sinais em tempo de execução linear.

A pesquisa no problema de ordenação por transposições resultou em diversos algoritmos aproximativos [5, 17, 30]. O primeiro algoritmo aproximativo para o problema de ordenação por transposições possui complexidade de tempo de execução $O(n^2)$ e garante um fator de

aproximação de 1,5 [7]. No entanto, esse algoritmo é considerado muito difícil e por essa razão Christie [17] propôs um outro algoritmo mais simples com o mesmo fator de aproximação, mas com complexidade do tempo de execução de $O(n^4)$. Eriksson *et al.* [23] ofereceram um algoritmo que requer no máximo $\lfloor (2n-2)/3 \rfloor$ transposições para ordenar qualquer permutação. Recentemente, Hartman e Shamir [30] propuseram um novo algoritmo de fator de aproximação 1,5 e tempo de execução $O(n\sqrt{n \log n})$ para o problema de ordenação por transposições usando uma nova estrutura de dados desenvolvida por Kaplan e Verbin [33]. Elias e Hartman [22] projetaram um novo algoritmo com fator de aproximação 1,375. Para instâncias específicas existem algoritmos exatos que executam em tempo polinomial [23, 41, 17]. Não conhecemos nenhum algoritmo de tempo de execução polinomial que resolva o problema de ordenação por transposições nem uma demonstração de que o problema é *NP*-difícil.

Christie [16] propôs e resolveu o problema de ordenação por intercâmbio de blocos em genomas — uma generalização do problema de ordenação por transposições — por meio de um algoritmo de tempo polinomial.

2.8 Discussão

Nessa seção, iremos discutir algumas características do formalismo clássico para o problema de rearranjo em genomas.

As principais desvantagens do formalismo clássico são listados abaixo:

1. Ao se modelar um genoma como uma permutação com sinais no formalismo clássico, os blocos de genes têm de ser rotulados com inteiros consecutivos começando com 1, por causa da representação posicional de permutações com sinais. Isto dificulta a inserção e remoção de genes. Mesmo em problemas com operações conservativas (eventos que não mudam o conjunto de genes), esta limitação dificulta raciocínios recursivos sobre permutações menores.
2. A indexação posicional não se adequa a genomas com cromossomos circulares porque ela exige que algum gene seja considerado um gene inicial.
3. Há diferenças sutis entre os diagramas de ciclos dependendo do tipo de genoma (com cromossomos lineares ou circulares) e do tipo de evento de rearranjo a ser tratado (reversões com sinais ou transposições). Essas diferenças podem ser uma fonte de erros e também podem mascarar as diferenças reais entre os problemas de rearranjo.
4. O formalismo clássico faz uso freqüente de argumentos gráficos em suas demonstrações, o que pode levar a confusão. Uma analogia pode ser tomada da Geometria Euclidiana, cujos argumentos são principalmente gráficos, embora tenha sido enriquecida com o desenvolvimento da Geometria Analítica.

5. Embora o formalismo clássico não requeira que a permutação destino na qual a permutação de entrada será ordenada seja a identidade, na prática observamos que a grande maioria dos algoritmos baseados nesse formalismo parte dessa premissa. Tal convenção causa problemas para problemas de rearranjo envolvendo mais de dois genomas, porque não é possível encontrar uma rotulação dos genes que sirva para comparar todos os pares de genomas no problema.

As desvantagens anteriores nos motivaram a encontrar um modelo mais adequado para genomas e eventos de rearranjo. Meidanis e Dias [39] apresentaram um formalismo algébrico no qual o papel de permutações como modelos de genomas é mais profundamente explorado do que no formalismo clássico. Algumas vantagens do formalismo algébrico são:

1. Os rótulos originais dos genes podem ser usados como elementos da permutação representando um genoma.
2. No formalismo algébrico nenhum tipo de indexação é necessária: cada gene é mapeado no gene que aparece em seguida no genoma. Isso simplifica a especificação dos eventos de rearranjo.
3. Quando representados como permutações, os eventos de rearranjo de um mesmo tipo no formalismo clássico podem apresentar diferentes tamanhos de suporte, ao contrário dos eventos representados no formalismo algébrico que possuem o mesmo suporte. O suporte de um evento de rearranjo relaciona-se aos genes que sofrerão uma mudança de mapeamento no genoma. Apenas alguns genes sofrem a ação de mudança de mapeamento do ponto de vista biológico e esses genes são expressamente representados pelo suporte do evento de rearranjo no formalismo algébrico.
4. Permutações como um produto de ciclos representam mais adequadamente os genomas do que permutações em sua representação posicional.
5. Resultados bem conhecidos da teoria de grupos de permutações podem ser usados em argumentos no formalismo algébrico, ao invés de argumentos gráficos.
6. A maioria dos conceitos fundamentais do problema de rearranjo em genomas como genomas, eventos de rearranjo, o diagrama de ciclos, ciclos, pontos de quebra podem ser modelados por meio de permutações. Esse modelo expressa os principais conceitos de uma maneira uniforme, ao invés de utilizar diversas estruturas como conjuntos, grafos, pares, cadeias e etc.
7. A norma de um evento de rearranjo pode ser usada como um “peso” em problemas envolvendo mais de um tipo de evento. Esse peso baseado na norma mantém a mesma

proporção entre os pesos geralmente atribuídos a reversões com sinais, fusões, fissões e intercâmbios de blocos em estudos teóricos sobre rearranjos. Isto parece indicar que a norma esteja próxima do peso real que os eventos deveriam ter de acordo com sua frequência na natureza, embora não tenhamos evidências conclusivas neste sentido.

8. Uma vez que cada gene pode ser representado pelo seu próprio “nome” no formalismo algébrico, lidar com diversos genomas não envolve rotulações distintas de genes para diferentes pares de genomas.

Capítulo 3

Formalismo Algébrico

O formalismo algébrico é um modelo para genomas e eventos de rearranjo que tenciona tirar vantagens da representação por permutações para obter implementações mais eficientes de algoritmos para o problema de rearranjo em genomas. Nesse formalismo é possível representar genomas, cromossomos, a orientação de genes e eventos de rearranjo por meio de permutações em sua representação como um “produto de ciclos” e é possível tirar proveito dos resultados existentes para as permutações nessa representação para encontrar propriedades interessantes sobre o efeito de eventos de rearranjo em genomas. Essa maneira de representar genomas contrasta com a representação de genomas mais comumente utilizada, o formalismo clássico, na qual um genoma multicromossomal é modelado por meio de um conjunto de seqüências.

Nesse capítulo, discutimos os conceitos fundamentais de álgebra envolvendo permutações que serão utilizados pelo formalismo algébrico nas Seções 3.1, 3.2 e 3.3. Além disso, até onde sabemos, estendemos a teoria algébrica de permutações por meio da demonstração do comportamento de produtos envolvendo 2-ciclos e 3-ciclos e a apresentação de novos conceitos como a 3-norma e a 3-divisibilidade na Seção 3.4 e Seção 3.5.

Na Seção 3.6, apresentamos uma versão do formalismo algébrico apresentado por Meidanis e Dias [39] na qual fixamos algumas convenções e definimos novos conceitos para estender e melhorar o formalismo algébrico. Na Seção 3.7, modelamos alguns eventos de rearranjo por meio de permutações e discutimos para quais pares de genomas existe uma seqüência de eventos de rearranjo que transforma um genoma em outro.

Na Seção 3.8, discutimos uma estratégia gulosa para a resolução de problemas de rearranjo em genomas e analisamos em que situações essa estratégia é aplicável.

3.1 Permutações

Nessa seção apresentamos a definição de permutação e algumas de suas principais propriedades. A permutação é o conceito fundamental usado na representação de genomas e eventos de rear-

ranjo no formalismo algébrico. Há inúmeros trabalhos que tratam de permutações no contexto de estruturas algébricas abstratas [38, 47]. Nessa seção discutiremos apenas as propriedades fundamentais relevantes para a modelagem do problema de rearranjo em genomas. Os resultados enunciados nessa seção podem ser encontrados no trabalho de MacLane e Birkhoff [38].

Dado um conjunto finito E , uma *permutação* π sobre E é uma bijeção $\pi : E \rightarrow E$. A *permutação identidade* é a permutação ι tal que $\iota x = x$ para todo $x \in E$. Dizemos que a permutação π *mapeia* o elemento x no elemento y (ou equivalentemente que x é *mapeado em* y por π) quando $\pi x = y$ para $x, y \in E$. Dizemos ainda que y é a *imagem* de x em π quando $\pi x = y$.

Dada as permutações π e σ ambas sobre E , a permutação *composta* (também chamada de *produto*) de π e σ é a permutação $\theta : E \rightarrow E$ tal que $\theta x = \pi \sigma x$ para todo $x \in E$. Observe que aplicamos o mapeamento das permutações “pela direita” para obter a composta de duas permutações, ou seja, em $\pi \sigma$ obtemos primeiramente a imagem de x sob σ e então aplicamos π em σx . A operação de produto de permutações é *associativa*, ou seja, dadas as permutações π , σ e θ , então $\pi(\sigma \theta) = (\pi \sigma)\theta$. A *permutação inversa* de π é a permutação π^{-1} , tal que $\pi \pi^{-1} = \pi^{-1} \pi = \iota$. Além disso, uma vez que a permutação identidade mapeia cada elemento em si mesmo, então $\pi \iota = \iota \pi = \pi$.

Exemplo 21. O produto de duas permutações π e σ sobre o conjunto E é obtido da seguinte maneira: para cada elemento $x \in E$ usamos a fórmula $(\pi \sigma)x = \pi(\sigma x)$. Considere o conjunto $E = \{a, b, c, d, e, f, g\}$ e as permutações π e σ sobre E :

$$\pi a = b \quad \pi b = d \quad \pi c = f \quad \pi d = a \quad \pi e = c \quad \pi f = e \quad \pi g = g$$

e

$$\sigma a = g \quad \sigma b = e \quad \sigma c = c \quad \sigma d = a \quad \sigma e = f \quad \sigma f = d \quad \sigma g = b.$$

O produto $\theta = \pi \sigma$ é a seguinte permutação:

$$\theta a = g \quad \theta b = c \quad \theta c = f \quad \theta d = b \quad \theta e = e \quad \theta f = a \quad \theta g = d.$$

Por exemplo $\theta a = \pi \sigma a = \pi g = g$.

O conjunto de todas as permutações sobre o conjunto E e a operação de produto definem o *Grupo Simétrico* $S(E)$ [38, 47]. Esse fato decorre diretamente das propriedades do produto no conjunto de todas as permutações: associatividade do produto, existência de elemento neutro (a identidade) e existência da inversa para qualquer permutação.

Em um produto $\pi \sigma$ sobre E , quando $\pi = \sigma$, escreveremos $\pi \pi$ como π^2 . No caso geral, iremos escrever o produto $\pi \dots \pi$ (k vezes) como π^k para $k \geq 1$ e por convenção adotamos $\pi^0 = \iota$.

Seja π uma permutação sobre E e n um inteiro positivo. Definimos π^{-n} como $(\pi^{-1})^n$. Além disso, temos $\pi^{-n} = (\pi^{-1})^n = (\pi^n)^{-1}$. Dada π uma permutação sobre E , para quaisquer

inteiros não negativos r, s temos $\pi^r \pi^s = \pi^{r+s}$ e $(\pi^r)^s = \pi^{rs}$. Observe que $\pi^m \pi^n = \pi^n \pi^m$ para quaisquer inteiros não negativos m, n é uma consequência direta das propriedades anteriores.

Considere as sucessivas aplicações da permutação π sobre E em um elemento $x \in E$:

$$x, \pi x, \pi\pi x = \pi^2 x, \pi\pi\pi x = \pi^3 x, \dots, \pi^m x, \dots$$

O conjunto $\{\pi^m x \mid m \text{ é um inteiro}\}$ contém todas as imagens do elemento $x \in E$ nas potências de π . Um elemento y é *acessível por* x em π , denotado por $x \equiv_\pi y$, quando $y = \pi^m x$ para algum inteiro $m \geq 0$. Dada a permutação π sobre E , observe que a relação $x \equiv_\pi y$ é uma relação de equivalência para $x, y \in E$.

Seja π uma permutação sobre E e tome um elemento $x \in E$. A *órbita* de x sob a permutação π , denotada por $orb(\pi, x)$, é o conjunto $\{y \in E \mid y \equiv_\pi x\}$, ou seja, o conjunto dos elementos y tais que $y = \pi^k x$ para um inteiro $k \geq 0$. Uma órbita é chamada *não trivial* quando ela possui mais de um elemento, caso contrário a órbita é chamada de *trivial*. Denotamos por $Orb(\pi, E)$ o conjunto das órbitas da permutação π sobre E . As órbitas de uma permutação são mutuamente disjuntas devido a relação de acessibilidade ser uma relação de equivalência.

Discutiremos agora sobre como representar permutações. A maneira de representar uma permutação pode destacar algumas de suas propriedades que são mais adequadas à modelagem de genomas. As órbitas e a relação de acessibilidade de uma permutação têm um papel especial sobre a representação de permutações.

Considere a seguinte permutação σ sobre $E = \{a, b, c, d, e, f, g, h, i, j\}$:

$$\sigma = \begin{pmatrix} a & b & c & d & e & f & g & h & i & j \\ b & d & i & f & g & a & c & h & e & j \end{pmatrix}.$$

A permutação é representada em forma de uma *matriz*, onde cada elemento na primeira linha é mapeado para o elemento na mesma coluna na segunda linha em σ . Por exemplo, o elemento c é mapeado para o elemento i em σ . O conjunto de elementos acessíveis por a , ou seja $orb(\sigma, a)$ é $\{a, b, d, f\}$. Para a permutação σ anterior temos $orb(\sigma, a) = orb(\sigma, b) = orb(\sigma, d) = orb(\sigma, f)$. Além daqueles conjuntos há $orb(\sigma, c) = \{c, i, e, g\}$, $orb(\sigma, h) = \{h\}$ e $orb(\sigma, j) = \{j\}$. Os conjuntos $orb(\sigma, a)$, $orb(\sigma, c)$, $orb(\sigma, h)$ e $orb(\sigma, j)$ são classes de equivalência sobre os elementos de E . Como as classes de equivalências, ou seja, as órbitas são definidas exclusivamente pela relação de acessibilidade de uma permutação, podemos representar uma permutação por meio do conjunto das órbitas e da relação de acessibilidade dessa permutação. A representação de permutações por meio do conjunto de órbitas e da relação de acessibilidade destaca o relacionamento entre os elementos do conjunto base, mas não enfatiza as propriedades de produtos envolvendo permutações. Por esse motivo, iremos definir o conceito de ciclo que, de maneira intuitiva, corresponde a uma órbita e a um subconjunto da relação de equivalência de uma permutação.

Um *ciclo* é uma permutação $\pi \in S(E)$ tal que ele tem no máximo uma órbita não trivial. Um ciclo α é um k -*ciclo* quando sua órbita não trivial possui $k > 1$ elementos ou um 1-*ciclo*

quando $\alpha = \iota$. Dois ciclos são *disjuntos* quando as suas respectivas órbitas não triviais são disjuntas. Podemos representar um ciclo α diferente da identidade como uma *lista circular* dos elementos em sua órbita não trivial de tal forma que um elemento x é seguido pelo elemento αx .

Exemplo 22. Considere o conjunto de inteiros $E = \{1, 2, 3, 4, 5, 6\}$ e a permutação α abaixo sobre E . Uma vez que podemos escolher qualquer elemento de E para ser o “primeiro” elemento do ciclo, a mesma permutação pode ser escrita de maneiras diferentes. A permutação

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 3 & 5 & 6 & 1 & 4 & 2 \end{pmatrix}$$

é um 6-ciclo e ela pode ser escrita como $(2\ 5\ 4\ 1\ 3\ 6)$, $(5\ 4\ 1\ 3\ 6\ 2)$, $(1\ 3\ 6\ 2\ 5\ 4)$, etc. Apesar de seus diferentes formatos, todos representam a mesma permutação.

Iremos representar uma permutação qualquer como um *produto de ciclos*. Para facilitar a compreensão sobre esse tipo de representação, iremos primeiramente discutir um exemplo.

Exemplo 23. Considere a permutação π sobre o conjunto de inteiros $E = \{0, 1, \dots, 5\}$:

$$\pi 0 = 1, \pi 1 = 3, \pi 2 = 0, \pi 3 = 5, \pi 4 = 4, \pi 5 = 2$$

que pode ser representada como uma lista circular $(0\ 1\ 3\ 5\ 2)$. O ciclo π poderia ser obtido por meio do produto de outros dois ciclos, por exemplo o produto $\alpha\beta$ tal que $\alpha = (0\ 1\ 3)$ e $\beta = (3\ 5\ 2)$. De fato uma permutação pode ser escrita por meio de um número infinito de distintos produtos de ciclos. O ciclo π poderia também ser obtido por meio dos produtos:

$$\pi = (0\ 1)(1\ 3\ 5)(0\ 4)(5\ 2)(0\ 4)$$

e

$$\pi = (0\ 1\ 3)(3\ 5)(0\ 4\ 5)(4\ 5\ 2)(0\ 4)$$

Uma maneira de representar permutações que permitisse descrevê-las de infinitas maneiras não parece ser muito útil quando queremos, de fato, nos referir a uma forma “canônica” que apresente as propriedades intrínsecas de uma permutação que a distingue das demais. O problema dos produtos escolhidos no Exemplo 23 anterior é a falta, na própria representação, de evidência da relação de acessibilidade definida pela permutação. Dada uma função $f : X \rightarrow Y$, a *restrição de f ao conjunto $W \subseteq X$* , denotada por $f|_W$, é a função $f|_W$ de W para Y tal que $f|_W(x) = f(x)$ para todo $x \in W$. Seja π uma permutação sobre E , a *decomposição em ciclos disjuntos* de π é um conjunto de ciclos tal que um ciclo α sobre E pertence a essa decomposição quando $\alpha = \pi|_{orb(\alpha, x)}$ para algum x pertencente à órbita não trivial de α . A decomposição em ciclos da permutação identidade é vazia.

Exemplo 24. Se $E = \{1, 2, 3, 4, 5, 6\}$ e π sobre E é igual a

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 3 & 4 & 1 & 6 & 5 \end{pmatrix},$$

então uma decomposição em ciclos disjuntos dessa permutação possui os ciclos $(1\ 2\ 3\ 4)$ e $(5\ 6)$. Uma possível representação como produto de ciclos é $(1\ 2\ 3\ 4)(5\ 6)$. Os ciclos $(1\ 2\ 3\ 4)$ e $(5\ 6)$ são os ciclos cujas órbitas correspondem às órbitas $\text{orb}(\pi, 1)$ e $\text{orb}(\pi, 5)$ de π e suas relações de acessibilidade são os subconjuntos da relação de acessibilidade de π quando restrita respectivamente aos elementos da órbita $\text{orb}(\pi, 1)$ e $\text{orb}(\pi, 5)$. Observe que os produtos $(1\ 2\ 3\ 4)(5\ 6)$ e $(5\ 6)(1\ 2\ 3\ 4)$ são representações equivalentes de π .

Dada uma permutação π sobre E , um elemento x de E é chamado *fixo* quando $\pi x = x$. Elementos fixos não precisam aparecer na representação em produto de ciclos de π pois são intrinsecamente conhecidos por meio do conjunto E . Por exemplo, a permutação $\pi = (b\ c\ d)(e\ f)$ é a representação em produto de ciclos sobre $E = \{a, b, c, d, e, f\}$ na qual a não aparece por ser fixo. O conjunto dos elementos não fixos em uma permutação π sobre E é o *suporte* de π , denotado por $\text{Supp}(\pi)$; i.e. $\text{Supp}(\pi) = \{x \in E \mid \pi x \neq x\}$. O produto de duas permutações representadas como um produto de ciclos é mostrado abaixo no Exemplo 25.

Exemplo 25. Sejam

$$\pi = (a\ c)(e\ g\ b) \quad \text{e} \quad \sigma = (c\ g)(b\ d)$$

permutações sobre o conjunto $E = \{a, b, c, d, e, f, g\}$. O produto $\pi\sigma$ pode ser obtido do seguinte modo. Construa o produto ciclo a ciclo. Escolha um elemento em E que previamente não tenha sido incluído em algum ciclo de $\pi\sigma$, digamos a , comece um novo ciclo em $\pi\sigma$ e encontre a imagem de a em σ . Nesse caso, a é mapeado em si mesmo em σ , então $\sigma a = a$. Agora temos que encontrar σa em π e observamos que o elemento σa está em $(a\ c)$. O elemento a é mapeado para c sob π , então $\pi\sigma a = c$. O elemento c é mapeado para g sob σ e π mapeia g em b , logo $\pi\sigma c = b$. Sob σ , o elemento b é mapeado para d e d é fixo em π , logo $\pi\sigma b = d$. Continuamos com esse procedimento até que encontremos o primeiro elemento incluído no ciclo corrente (até o momento o elemento a nesse exemplo). Nesse exemplo, ao fim do procedimento temos

$$\pi\sigma = (a\ c\ b\ d\ e\ g).$$

O elemento g “fecha” o primeiro ciclo uma vez que $\pi\sigma g = a$. O elemento f é mapeado para si mesmo em ambas π e σ , então ele permanece fixo no produto das duas permutações.

Observe que em geral $\pi\sigma \neq \sigma\pi$, mas há situações nas quais a igualdade vale. Duas permutações são *disjuntas* quando seus suportes são disjuntos. Logo, permutações π e σ sobre E são disjuntas se e somente se para qualquer elemento $x \in E$ temos $\pi x = x$ ou $\sigma x = x$.

Proposição 26. Se π e σ são duas permutações disjuntas sobre E , então temos $\pi\sigma = \sigma\pi$.

Decorre da Proposição 26 que dadas as permutações mutuamente disjuntas $\sigma_1, \dots, \sigma_r \in S(E)$, o produto envolvendo essas permutações é comutativo.

Como vimos até agora, uma permutação pode ser representada como um produto dos ciclos em sua decomposição em ciclos disjuntos. A Proposição 27 informa que há apenas uma maneira de representar uma permutação como um produto de ciclos disjuntos desconsiderando apenas a ordem desses ciclos no produto.

Proposição 27. Qualquer permutação sobre E pode ser representada como um único produto de ciclos disjuntos, a menos da ordem dos ciclos no produto.

A Proposição 27 garante que qualquer permutação pode ser representada por meio de um único produto de ciclos disjuntos, a menos da ordem dos ciclos. Observe que um k -ciclo para $k \geq 3$ possui uma *ordem circular* dos elementos em sua órbita não trivial. Dizemos que os elementos a_1, a_2, \dots, a_m de E , para $3 \leq m \leq k$, *aparecem na ordem circular* $a_1 a_2 \dots a_m$ de um k -ciclo α com $k \geq 3$ quando $\alpha^{j_{i-1}} a_1 = a_i$ e $j_{i-1} < j_i$ para $2 \leq i \leq m \leq k$. Por exemplo, os elementos a, b, c, d e e aparecem na ordem circular $acbde$ no 5-ciclo $(a \ c \ b \ d \ e)$. Equivalentemente, podemos representar a ordem circular dos elementos do 5-ciclo anterior por: $acbde, cbdea, bdeac, deacb$ e $eachd$. A ordem circular de um ciclo permite uma maneira simples de obter a permutação inversa de uma dada permutação. A permutação inversa de um ciclo $\alpha = (a_1 \ a_2 \ \dots \ a_n)$ é $\alpha^{-1} = (a_n \ a_{n-1} \ \dots \ a_1)$, ou seja, a orientação do ciclo foi trocada. Para uma permutação qualquer $\pi = \alpha_1 \dots \alpha_r$ sobre E , a sua permutação inversa é simplesmente $\alpha_r^{-1} \dots \alpha_1^{-1}$. Observe que $\alpha_r^{-1} \dots \alpha_1^{-1} = \alpha_1^{-1} \dots \alpha_r^{-1}$ uma vez que ciclos disjuntos comutam. Logo, basta mudar a orientação de todos os ciclos em uma permutação representada como um produto de ciclos para obter a permutação inversa. Esse é apenas um exemplo de como a representação de uma permutação pode sugerir novas propriedades ou procedimentos para a manipulação de permutações. Discutiremos outros conceitos e propriedades que serão relevantes para a modelagem de genomas nas seções subsequentes.

3.2 Conjugação

Uma permutação π sobre E tem uma única representação como produto de ciclos disjuntos. Diversas permutações distintas podem compartilhar algumas propriedades tais como o número de ciclos em sua decomposição em ciclos disjuntos, os seus conjuntos de órbitas ou a cardinalidade de suas órbitas, por exemplo. Definimos a seguir uma operação entre permutações que intuitivamente “renomeia” os elementos de uma permutação. O conceito de conjugação pode ser encontrado no trabalho de MacLane e Birkhoff [38] e foi discutido no contexto de rearranjos em genomas por Meidanis e Dias [39]. Apresentamos uma definição formal para o conceito de

estrutura de ciclos que é uma propriedade de permutações relevante para o formalismo algébrico e que é preservada pela operação de conjugação.

Dadas as permutações β e α sobre E , a *conjugação de β por α* é a permutação $\alpha\beta\alpha^{-1}$. A operação de conjugação é denotada por $\alpha \cdot \beta = \alpha\beta\alpha^{-1}$ e α é chamado o *conjugador* enquanto β é chamado o *conjugado*.

O operador de conjugação preserva a estrutura da permutação na qual é aplicada, digamos β , no entanto ele modifica a relação de acessibilidade dos elementos de β de acordo com o conjugador α . Por exemplo, para $\beta = (a\ b\ c)(d\ e)$:

$$\alpha(a\ b\ c)(d\ e)\alpha^{-1} = (\alpha a\ \alpha b\ \alpha c)(\alpha d\ \alpha e)$$

Proposição 28. Dadas as permutações $\alpha, \beta, \gamma, \tau$ sobre E , então:

1. $\tau \cdot (\alpha\beta) = (\tau \cdot \alpha)(\tau \cdot \beta)$,
2. $\iota \cdot \alpha = \alpha$,
3. $\tau \cdot \alpha^{-1} = (\tau \cdot \alpha)^{-1}$,
4. $\gamma \cdot (\tau \cdot \alpha) = (\gamma\tau) \cdot \alpha$.
5. $(\tau \cdot \alpha)^k = \tau \cdot \alpha^k$ onde $k \geq 0$.

Meidanis e Dias [39] tratam de estruturas de ciclos de uma permutação sem definir formalmente esse conceito. Apresentamos uma definição simples e intuitiva para esse conceito e demonstramos que a operação de conjugação preserva a estrutura de ciclos de uma permutação. A *estrutura de ciclos* de uma permutação $\pi \in S(E)$ é o multiconjunto dos tamanhos de suas órbitas. Por exemplo, a estrutura de ciclos da permutação $(a\ b\ c)(d\ e)(f\ g)(h)(i)(j)$ sobre $E = \{a, b, c, d, e, f, g, h, i, j\}$ é $\{1, 1, 1, 2, 2, 3\}$. Demonstramos no Lema 29 que a operação de conjugação preserva a estrutura de ciclos de uma permutação.

Lema 29. Sejam α, β permutações sobre E , temos:

1. Se $\beta = \gamma \cdot \alpha$ onde γ é uma permutação sobre E então α e β têm a mesma estrutura de ciclos.
2. Se α e β possuem a mesma estrutura de ciclos então existe uma permutação γ sobre E tal que $\beta = \gamma \cdot \alpha$.

Prova:

1. Iremos mostrar que para cada órbita de α existe uma única órbita correspondente e de mesmo tamanho em $\gamma \cdot \alpha$.

Seja x um elemento qualquer de E e y um elemento na mesma órbita de x , isto é, temos $y = \alpha^k x$ para um inteiro k . Afirmamos que o elemento γy está na mesma órbita de γx em $\gamma \cdot \alpha$, como pode ser deduzido da seguinte igualdade:

$$\gamma y = \gamma \alpha^k x = \gamma \alpha^k \gamma^{-1} \gamma x = \gamma \cdot (\alpha^k) \gamma x = (\gamma \cdot \alpha)^k \gamma x.$$

Logo, temos $\gamma y \in orb(\gamma \cdot \alpha, \gamma x)$. Assim, como para cada elemento y em $orb(\alpha, x)$ existe um elemento γy em $orb(\gamma \cdot \alpha, \gamma x)$ então para cada órbita em α há uma órbita de mesmo tamanho em $\gamma \cdot \alpha$. Como α e $\beta = \gamma \cdot \alpha$ são definidos sobre o mesmo conjunto E , então o número de órbitas em ambas as permutações é o mesmo.

Portanto, as permutações α e $\beta = \gamma \cdot \alpha$ têm o mesmo número de órbitas e cada órbita em α corresponde a uma órbita de mesmo tamanho em $\gamma \cdot \alpha$, ou seja, ambas as permutações possuem a mesma estrutura de ciclos.

2. Dadas as permutações α e β com a mesma estrutura de ciclos, sejam o_1, \dots, o_k e p_1, \dots, p_k as órbitas de α e β , respectivamente, ordenadas pelo tamanho de cada órbita. Escolha elementos arbitrários $x_i \in o_i$ e $y_i \in p_i$ para $1 \leq i \leq k$. Definimos γ como a permutação tal que $\gamma \alpha^l x_i = \beta^l y_i$ para todo inteiro l e para $1 \leq i \leq k$. A permutação γ está bem definida pois o tamanho de o_i é igual ao de p_i para $1 \leq i \leq k$.

Qualquer $x \in E$ pode ser escrito como $x = \alpha^l x_i$ para um certo inteiro l e elemento x_i , então

$$\beta \gamma x = \beta \gamma \alpha^l x_i = \beta \beta^l y_i = \beta^{l+1} y_i = \gamma \alpha^{l+1} x_i = \gamma \alpha \alpha^l x_i = \gamma \alpha x.$$

Logo $\beta \gamma x = \gamma \alpha x$ para qualquer $x \in E$. Portanto $\beta = \gamma \cdot \alpha$.

□

3.3 Norma e Divisibilidade

Nessa seção, discutimos um parâmetro de permutações associado a um tipo particular de decomposição em ciclos. Os resultados e definições enunciados nessa seção encontram-se no trabalho de Meidanis e Dias [39], exceto pela introdução dos conceitos de “máximo divisor comum” entre permutações e “permutações primas” que são contribuições desse trabalho. Esses novos conceitos são prescindíveis no atual estágio de desenvolvimento do formalismo algébrico, mas acreditamos que possam colaborar para a simplificação de notação a medida que o formalismo seja estendido.

Uma *decomposição em k -ciclos* de uma permutação π sobre E é uma representação de π como um produto de k -ciclos, não necessariamente disjuntos. Decomposições em 2-ciclos e em 3-ciclos têm um papel importante na busca de limitantes inferiores para distâncias de rearranjo no formalismo algébrico.

Lema 30. Toda permutação em $S(E)$ possui uma decomposição em 2-ciclos.

Prova: Como qualquer permutação pode ser representada por um produto de ciclos disjuntos segundo a Proposição 27, iremos encontrar uma decomposição em 2-ciclos de um ciclo e uma permutação arbitrária pode ser composta pelas decomposições em 2-ciclos de seus ciclos. O ciclo $(a_1 a_2 \dots a_r)$ pode ser reescrito como

$$(a_1 a_2 \dots a_r) = (a_1 a_r)(a_1 a_{r-1}) \dots (a_1 a_3)(a_1 a_2).$$

□

Há diversas maneiras de escrever uma permutação como um produto de 2-ciclos (veja o Exemplo 31). Uma decomposição em 2-ciclos *mínima* de uma permutação π sobre E contém o número mínimo de 2-ciclos cujo produto é π . Esse número mínimo de 2-ciclos é chamado de *norma* de π e ele é denotado por $\|\pi\|$. Por exemplo, temos $\|\iota\| = 0$ e a norma de um 2-ciclo é 1.

Exemplo 31. A permutação $\pi = (7 \ 2 \ 8 \ 1)(3 \ 5 \ 10)(6 \ 4 \ 9)$ sobre o conjunto $E = \{x \in \mathbb{Z} \mid 1 \leq x \leq 10\}$ possui pelo menos duas representações distintas como produto de 2-ciclos:

$$(3 \ 2)(7 \ 2)(2 \ 8)(8 \ 1)(7 \ 5)(3 \ 7)(5 \ 10)(6 \ 4)(4 \ 9)$$

e

$$(7 \ 1)(7 \ 8)(7 \ 2)(3 \ 10)(3 \ 5)(6 \ 9)(6 \ 4).$$

Podemos deduzir dessas decomposições em 2-ciclos que $\|\pi\| \leq 7$.

Seguem algumas propriedades envolvendo a norma de permutações.

Proposição 32. Para quaisquer permutações α e β sobre E , temos:

1. $\|\alpha\| = 0$ se e somente se $\alpha = \iota$
2. $\|\alpha^{-1}\| = \|\alpha\|$
3. $\|\beta \cdot \alpha\| = \|\alpha\|$
4. $\|\alpha\beta\| \leq \|\alpha\| + \|\beta\|$
5. $\|\alpha\beta\| = \|\beta\alpha\|$

Seja $o(\pi, E)$ o número de órbitas da permutação π sobre E . Quando o conjunto E é conhecido pelo contexto, iremos chamá-lo apenas de $o(\pi)$. Do mesmo modo, denotamos por $no(\pi)$ o número de órbitas não triviais de π sobre E . O número de órbitas de tamanho par e o número de órbitas de tamanho ímpar de uma permutação π são respectivamente denotados por $o_{even}(\pi)$ e $o_{odd}(\pi)$. Observe que a cardinalidade da decomposição em ciclos disjuntos da permutação π é $no(\pi)$. Por exemplo, se $\pi = (7\ 2\ 8\ 1)(3\ 5\ 10)(6\ 4\ 9)$ e $E = \{x \in \mathbb{Z} \mid 1 \leq x \leq 12\}$, então $o(\pi) = 5$ e $no(\pi) = 3$.

O produto envolvendo um 2-ciclo $(a\ b)$ e uma permutação β “separa” em órbitas distintas os elementos a e b . Meidanis e Dias [39] apresentam esse resultado sem demonstrá-lo. Fornecemos uma demonstração para esse resultado no Lema 33.

Lema 33. Dado um 2-ciclo $\alpha = (a\ b)$ e uma permutação β ambos sobre E , temos $b \in orb(\beta, a)$ se e somente se

1. $b \notin orb(\alpha\beta, a)$ e
2. $b \notin orb(\beta\alpha, a)$.

Prova:

1. Se $b \in orb(\beta, a)$ então $b = \beta^k a$ para $0 < k < |orb(\beta, a)|$. Como α é um 2-ciclo então $\alpha x = x$ para $x \neq a$ e $x \neq b$ em E . De $b = \beta^k a$ obtemos $\beta^{-1}b = \beta^{k-1}a$. Fazendo as substituições adequadas

$$\alpha\beta\beta^{k-1}a = \alpha\beta\beta^{-1}b = \alpha b = a,$$

obtemos $\alpha\beta\beta^{k-1}a = a$. Como $\beta^{j+1}a \neq a$ e $\beta^{j+1}a \neq b$ para $0 \leq j \leq k-2$ então $\alpha\beta\beta^j a = \beta^{j+1}a$ para $0 \leq j \leq k-2$. Logo os elementos $\beta^j a$ para $0 \leq j \leq k-1$ e apenas esses elementos pertencem a uma mesma órbita em $\alpha\beta$. Portanto, se $b \in orb(\beta, a)$ então $b \notin orb(\alpha\beta, a)$.

Por outro lado, suponha que temos $b \notin orb(\alpha\beta, a)$. Temos $\alpha\alpha\beta = \beta$ pois α é um 2-ciclo. Além disso, substituindo adequadamente em

$$\alpha\alpha\beta(\alpha\beta)^{-1}a = \alpha a = b,$$

ou seja, colocando em outros termos, obtemos $\beta(\alpha\beta)^{-1}a = b$.

Como $\alpha\alpha\beta = \beta$ então $(\alpha\alpha\beta)^k a = \beta^k a$ para $0 \leq k \leq |orb(\alpha\beta, a)| - 1$. Como $(\alpha\beta)^{-1}a = (\alpha\beta)^m a$ para $m = |orb(\alpha\beta, a)| - 1$, então $(\alpha\beta)^{-1}a \in orb(\beta, a)$. Como $\beta(\alpha\beta)^{-1}a = b$ então $b \in orb(\beta, a)$. Portanto, se $b \notin orb(\alpha\beta, a)$ então $b \in orb(\beta, a)$.

2. Se $b \in \text{orb}(\beta, a)$ então $a \in \text{orb}(\beta, b)$ e $a = \beta^k b$ para $0 < k < |\text{orb}(\beta, b)|$. Como $\alpha a = b$, $\alpha b = a$ e $\alpha x = x$ para $x \neq a$ e $x \neq b$ então $\beta \alpha b = \beta a$, $\beta \alpha a = \beta b$ e $\beta \alpha x = \beta x$ para $x \neq a$ e $x \neq b$ em E . De $\beta^{j+1} b \neq a$ e $\beta^{j+1} b \neq b$ para $0 \leq j \leq k-2$ obtemos $\beta \alpha \beta^{j+1} b = \beta^{j+2} b$ para $0 \leq j \leq k-2$. Como $\beta \alpha a = \beta b$, $\beta^k b = a$ e $\beta \alpha \beta^{j+1} b \neq b$ para $0 \leq j \leq k-2$ então apenas os elementos $\beta^{j+1} b$ para $0 \leq j \leq k-2$ pertencem à órbita que contém a em $\beta \alpha$ e portanto $b \notin \text{orb}(\beta \alpha, a)$.

Por outro lado, se $b \notin \text{orb}(\beta \alpha, a)$, então de maneira semelhante ao item anterior, obtemos $b \in \text{orb}(\beta, a)$.

Portanto temos $b \in \text{orb}(\beta, a)$ se e somente se $b \notin \text{orb}(\alpha \beta, a)$ e $b \notin \text{orb}(\beta \alpha, a)$. \square

O valor da norma de uma permutação pode ser obtido por meio do tamanho do conjunto E e do número de órbitas da permutação. A Proposição 34 demonstrada por Meidanis e Dias [39] estabelece uma fórmula para se obter a norma de uma permutação sem ser necessário obter sua decomposição em 2-ciclos.

Proposição 34. Dada uma permutação π sobre E , temos $\|\pi\| = |E| - o(\pi)$.

A norma do produto envolvendo duas permutações disjuntas é igual a soma das normas das permutações.

Proposição 35. Dadas duas permutações α e β disjuntas sobre E temos $\|\alpha\beta\| = \|\alpha\| + \|\beta\|$.

Uma permutação α *divide* uma permutação β , o que é denotado por $\alpha|\beta$, quando $\|\beta\alpha^{-1}\| = \|\beta\| - \|\alpha\|$. Dizemos também que α é um *divisor* de β .

Exemplo 36. Considere as seguintes permutações sobre $E = \{x \in \mathbb{Z} \mid 0 \leq x \leq 7\}$:

$$\pi = (1\ 6\ 4)(3\ 5\ 2)(0\ 7)$$

$$\sigma = (1\ 5\ 3)(0\ 2\ 6\ 4\ 7)$$

Um exemplo de decomposições em 2-ciclos mínimas dessas permutações são:

$$\pi = (1\ 6\ 4)(3\ 5\ 2)(0\ 7) = (1\ 6)(6\ 4)(3\ 5)(5\ 2)(0\ 7)$$

$$\sigma = (1\ 5\ 3)(0\ 2\ 6\ 4\ 7) = (1\ 5)(5\ 3)(0\ 2)(2\ 6)(6\ 4)(4\ 7)$$

Além disso, dadas as permutações $\phi = (1\ 3)$ e $\theta = (6\ 7\ 0)$ sobre E , temos $\phi|\sigma$, mas $\phi \nmid \pi$. Temos ainda $\theta|\sigma$, mas $\theta \nmid \pi$.

Meidanis e Dias [39] que a relação de divisibilidade é uma relação de ordem, isto é, uma relação reflexiva, anti-simétrica e transitiva.

Proposição 37. A relação de divisibilidade é uma relação de ordem, i.e.:

1. $\alpha|\alpha$
2. se $\alpha|\beta$ e $\beta|\alpha$, então $\alpha = \beta$
3. se $\alpha|\beta$ e $\beta|\gamma$, então $\alpha|\gamma$

A divisibilidade entre permutações tem uma característica semelhante à divisibilidade no conjunto dos números inteiros. Em ambos os conceitos existe a idéia de que o divisor “compõe” de algum modo o dividendo. Enquanto no caso de números inteiros o divisor é um fator em um produto de inteiros; no caso das permutações o divisor é uma permutação que compartilha algumas propriedades com o dividendo como a distribuição e ordem dos elementos entre as órbitas em ambas as permutações. Por exemplo, dada a permutação $\beta = (a \ d \ b \ e \ c)(f \ g \ h)$ sobre $E = \{a, b, c, d, e, f, g, h\}$ a permutação $\alpha = (a \ b \ c)(f \ g)$ sobre E é um divisor de β . A Proposição 38 lista algumas propriedades ajudam a determinar se uma permutação é um divisor de outra permutação.

Proposição 38. Para quaisquer permutações α, β e γ sobre E , temos:

1. $\iota|\alpha$,
2. $\alpha|\beta$ se e somente se $\alpha^{-1}|\beta^{-1}$,
3. $\alpha|\beta$ se e somente se $\gamma \cdot \alpha|\gamma \cdot \beta$,

A divisibilidade entre duas permutações e o número de órbitas das duas se relacionam segundo a fórmula apresentada no Lema 39. O Lema 39 é importante pois oferece uma fórmula para a variação do número de órbitas do produto envolvente uma permutação e seu divisor em relação ao número de órbitas da permutação.

Lema 39. Dadas as permutações α e β sobre E , se $\alpha|\beta$ então $o(\beta\alpha^{-1}) = o(\beta) + \|\alpha\|$.

Prova: Se $\alpha|\beta$ então

$$o(\beta\alpha^{-1}) = |E| - \|\beta\alpha^{-1}\| = |E| - (\|\beta\| - \|\alpha\|) = |E| - (|E| - o(\beta)) + \|\alpha\| = o(\beta) + \|\alpha\|.$$

$$\text{Logo } o(\beta\alpha^{-1}) = o(\beta) + \|\alpha\|. \quad \square$$

O ciclo α é um ciclo da permutação β , para α e β sobre E , quando $\alpha|\beta$ e não existe um ciclo γ distinto de α e β tal que $\alpha|\gamma$ e $\gamma|\beta$. Por exemplo, a permutação $(3 \ 5 \ 2)$ é um ciclo de $(1 \ 6 \ 4)(3 \ 5 \ 2)(0 \ 7)$.

Dadas as permutações α e β sobre E , o máximo divisor comum de α e β , denotado por $\gcd(\alpha, \beta)$, é o conjunto das permutações de maior norma que dividam α e β .

Exemplo 40. Considere as permutações $\alpha = (1 \ 5 \ 3 \ 6 \ 4 \ 2)$ e $\beta = (1 \ 4 \ 5 \ 3 \ 6 \ 2)$ sobre $E = \{x \in \mathbb{Z} \mid 1 \leq x \leq 6\}$. O conjunto máximo divisor comum de α e β é $\gcd(\alpha, \beta) = \{(1 \ 5 \ 3 \ 6 \ 2)\}$.

Duas permutações α e β são primas entre si quando $\gcd(\alpha, \beta) = \{\iota\}$.

3.4 Produtos envolvendo 2-ciclos

O resultado do produto de duas permutações é influenciado pela distribuição e ordem dos elementos nos ciclos das decomposições em ciclos disjuntos de ambas as permutações. Os 2-ciclos afetam a relação de acessibilidade de modo particularmente interessante. Dado um ciclo α e uma permutação β sobre E , a distribuição dos elementos em $\text{Supp}(\alpha)$ nas órbitas de β determina se $\alpha|\beta$ ou não. Mostraremos a influência da distribuição de elementos entre as órbitas na divisibilidade quando α é um 2-ciclo. Considere primeiramente o seguinte exemplo de um produto envolvendo um 2-ciclo.

Exemplo 41. Dado o 2-ciclo $\alpha = (a\ b)$ e uma permutação $\pi = (c\ a\ d\ b)(e\ f\ g)$ ambos definidos sobre $E = \{a, b, c, d, e, f, g\}$. Os produtos envolvendo as duas permutações são:

$$(a\ b)(c\ a\ d\ b)(e\ f\ g) = (c\ b)(d\ a)(e\ f\ g)$$

e

$$(c\ a\ d\ b)(e\ f\ g)(a\ b) = (c\ a)(d\ b)(e\ f\ g).$$

Nos dois casos, temos $b \in \text{orb}(\pi, a)$, $b \notin \text{orb}(\alpha\pi, a)$ e $b \notin \text{orb}(\pi\alpha, a)$. Os elementos a e b do 2-ciclo que pertenciam à mesma órbita em π , pertencem a órbitas distintas nos produtos $\alpha\pi$ e $\pi\alpha$.

Considere a permutação $\sigma = (a\ c\ e)(b\ d\ f)$ sobre E . Os produtos $\alpha\sigma$ e $\sigma\alpha$ são respectivamente:

$$(a\ b)(a\ c\ e)(b\ d\ f) = (a\ c\ e\ b\ d\ f)$$

e

$$(a\ c\ e)(b\ d\ f)(a\ b) = (a\ d\ f\ b\ c\ e)$$

Nesse caso temos $b \notin \text{orb}(\sigma, a)$, $b \in \text{orb}(\alpha\sigma, a)$ e $b \in \text{orb}(\sigma\alpha, a)$. Os elementos a e b que pertenciam a órbitas distintas de σ , pertencem à mesma órbita nos produtos $\alpha\sigma$ e $\sigma\alpha$.

O produto de um 2-ciclo $(a\ b)$ e uma permutação β “separa” os elementos a e b em órbitas distintas em $(a\ b)\beta$ (ou $\beta(a\ b)$) quando a e b pertencem a mesma órbita de β . Meidanis e Dias [39] discutem esse comportamento do produto de uma permutação e um 2-ciclo mas não o demonstram. Uma consequência do Lema 33 é o Corolário 42 no qual explicitamos o comportamento do produto de uma permutação e um 2-ciclo.

Corolário 42. Dado um 2-ciclo $\alpha = (a\ b)$ e uma permutação β ambos sobre E , temos

1. Os elementos a e b pertencem a órbita o não trivial do mesmo ciclo γ de β se e somente se a órbita o é a união das órbitas que contém a e b , respectivamente, enquanto as demais órbitas distintas de o de β são preservadas em $\alpha\beta$.

2. Os elementos a e b pertencem às respectivas órbitas o_a e o_b de β se e somente se o_a e o_b são unidas em uma mesma órbita o , enquanto todos as demais órbitas de β são preservadas em $\alpha\beta$.

O Corolário 42 informa como se dá a mudança na estrutura de ciclos de um produto entre uma permutação β e um 2-ciclo α . Esse resultado é essencialmente o mesmo tanto para o produto $\alpha\beta$ quanto $\beta\alpha$ pois essas permutações têm a mesma estrutura de ciclos em decorrência de uma ser obtida da outra por meio de uma conjugação por α .

Nos próximos lemas apresentamos alguns relacionamentos entre os conceitos de norma, divisibilidade e o comportamento de produtos de permutações envolvendo 2-ciclos. Esses novos resultados estendem o conjunto de propriedades do produto envolvendo 2-ciclos oferecidos por Meidanis e Dias [39] e são úteis para compreender como a norma, o suporte e o número de órbitas de uma permutação são afetados pela operação de produto.

Lema 43. Sejam $\alpha = (a\ b)$ um 2-ciclo e β uma permutação, ambos sobre E . Os elementos a e b pertencem ao suporte de γ onde γ é um ciclo de β se e somente se $(a\ b)|\beta$.

Prova: Se a e b pertencem a $Supp(\gamma)$ para um ciclo γ de β então a órbita não trivial do ciclo γ de β é dividida em dois conjuntos $\{a, \beta a, \dots, \beta^{-1}b\}$ e $\{b, \beta b, \beta^2 b, \dots, \beta^{-1}a\}$ em $\alpha\beta$ segundo o Corolário 42. Logo $o(\alpha\beta) = o(\beta) + 1$. Além disso $\|\alpha\beta\| = \|\beta\alpha^{-1}\|$ pela Proposição 32, então

$$\|\beta\alpha^{-1}\| = |E| - o(\alpha\beta) = |E| - (o(\beta) + 1) = (|E| - o(\beta)) - 1 = \|\beta\| - \|\alpha\|;$$

ou seja, temos $\alpha|\beta$.

Inversamente, suponha que $\alpha|\beta$, onde $\alpha = (a\ b)$. Então, pelo Lema 39, temos $o(\alpha\beta) = o(\beta) + 1$, i.e. o número de órbitas é aumentado de um em $\alpha\beta$ em relação a β . Pelo Corolário 42, os elementos a e b pertencem a mesma órbita de β , que é a órbita não trivial de um ciclo γ de β . Logo a e b pertencem a $Supp(\gamma)$ para um ciclo γ de β .

Portanto os elementos a e b pertencem a $Supp(\gamma)$ para um ciclo γ de β se e somente se $(a\ b)|\beta$. \square

Segundo o Lema 43, a decomposição em ciclos de uma permutação β pode ser modificada pelo produto de um 2-ciclo α em apenas dois casos a serem considerados de acordo com β ser divisível ou não por α : respectivamente há um aumento no número de órbitas de $\alpha\beta$ de 1 ou uma diminuição no número de órbitas de $\alpha\beta$ de 1 (o número de órbitas é sempre comparado ao número de órbitas de β). Se um 2-ciclo $\alpha = (a\ b)$ divide uma permutação β então os elementos a e b aparecem na mesma órbita em β . Logo, informalmente, o ciclo $(a \dots b \dots)$ é “quebrado” em dois ciclos $(a \dots)(b \dots)$ em $\alpha\beta$. Por outro lado, se $(a\ b)$ não divide β então dois ciclos em β são “fundidos” em $\alpha\beta$. Cada um desses dois ciclos de β contém um elemento de α . Logo, outra maneira de verificar se há um aumento ou diminuição no número de órbitas é encontrar em quais ciclos de β os elementos no suporte de α aparecem.

Lema 44. Sejam α um 2-ciclo e β uma permutação, ambos sobre E . Então

$$\|\beta\alpha\| = \begin{cases} \|\beta\| - 1 & \text{se } \alpha|\beta, \\ \|\beta\| + 1 & \text{caso contrário.} \end{cases}$$

Prova: Se $\alpha|\beta$ então $\|\beta\alpha^{-1}\| = \|\beta\| - \|\alpha\|$. Como $\alpha^{-1} = \alpha$ e $\|\alpha\| = 1$ porque α é um 2-ciclo, então

$$\|\beta\alpha\| = \|\beta\| - \|\alpha\| = \|\beta\| - 1.$$

Por outro lado, se $\alpha \nmid \beta$ então os elementos a e b pertencem a ciclos distintos de β pelo Lema 43. Logo, pelo Corolário 42, os dois ciclos distintos são reunidos em um único ciclo em $\beta\alpha$ e temos $o(\beta\alpha) = o(\beta) - 1$. Então

$$\|\beta\alpha\| = |E| - o(\beta\alpha) = |E| - o(\beta) + 1 = \|\beta\| + 1.$$

□

Lema 45. Sejam α um 2-ciclo e β uma permutação, ambos sobre E . Então

1. se $\alpha|\beta$ então $o(\alpha\beta) = o(\beta) + 1$
2. se $\alpha \nmid \beta$ então $o(\alpha\beta) = o(\beta) - 1$

Prova:

1. Suponha que $\alpha|\beta$ então pelo Lema 39 temos $o(\beta\alpha^{-1}) = o(\beta) + \|\alpha\|$. Como $\alpha\beta = \alpha^{-1} \cdot (\beta\alpha^{-1})$, então $o(\alpha\beta) = o(\beta\alpha^{-1})$ pelo Lema 29. Logo $o(\alpha\beta) = o(\beta) + 1$.
2. Uma vez que $o(\alpha\beta) = o(\beta\alpha^{-1})$ e usando-se o Lema 43 então a e b pertencem a órbitas de diferentes ciclos de β . Essas órbitas diferentes são reunidas em uma única órbita em $\alpha\beta$ pelo Corolário 42. Logo $o(\alpha\beta) = o(\beta) - 1$.

□

As propriedades de produtos de permutações envolvendo 2-ciclos permitem relacionar a divisibilidade entre permutações e seus suportes.

Lema 46. Dadas as permutações α e β sobre E , então

1. se $\alpha|\beta$ então $\text{Supp}(\alpha) \subseteq \text{Supp}(\beta)$.
2. se α é um ciclo tal que $\alpha|\beta$ e $\alpha \neq \beta$ então $\text{Supp}(\alpha) \subset \text{Supp}(\beta)$.

Prova:

1. Suponha que existe um elemento $x \in \text{Supp}(\alpha)$ tal que $x \notin \text{Supp}(\beta)$. Temos $(x y)|\alpha$ para qualquer $y \in \text{orb}(\alpha, x)$ distinto de x pelo Lema 43. Então, pela Proposição 37, temos $(x y)|\beta$ de $(x y)|\alpha$ e $\alpha|\beta$. Há dois casos a ser considerados:

- (a) Se $y \in \text{Supp}(\beta)$ então $\|(x y)\beta\| = \|\beta\| + 1$ pelo Lema 43, como x e y pertencem a órbitas distintas de β . Logo $(x y) \nmid \beta$.
- (b) Se $y \notin \text{Supp}(\beta)$ então $(x y)$ e β são disjuntos. Logo $\|\beta(x y)\| = \|(x y)\| + \|\beta\|$ e então $(x y) \nmid \beta$.

De qualquer modo, temos $(x y) \nmid \beta$ o que contradiz o fato de $(x y)|\beta$. Logo não há um elemento $x \in \text{Supp}(\alpha)$ tal que $x \notin \text{Supp}(\beta)$, ou seja, temos $\text{Supp}(\alpha) \subseteq \text{Supp}(\beta)$.

2. Se α é um ciclo tal que $\alpha|\beta$ e $\alpha \neq \beta$ então há novamente dois casos a se considerar:

- (a) Se β é um ciclo e $\alpha|\beta$ então $\text{Supp}(\alpha) \subseteq \text{Supp}(\beta)$ pelo item anterior (mais precisamente, temos $\text{orb}(\alpha) \subseteq \text{orb}(\beta)$ porque ambos α e β são ciclos). Como $\alpha \neq \beta$ então existe um elemento $x \in \text{Supp}(\beta)$ tal que $x \notin \text{Supp}(\alpha)$. Logo $\text{Supp}(\alpha) \subset \text{Supp}(\beta)$.
- (b) Se β não é um ciclo e $\alpha|\beta$ então $\text{Supp}(\alpha) \subseteq \text{Supp}(\beta)$ e existem elementos $x, y \in \text{Supp}(\beta)$ tais que $\text{orb}(\beta, x) \neq \text{orb}(\beta, y)$. Suponha que $\text{Supp}(\alpha) = \text{Supp}(\beta)$, então $x, y \in \text{Supp}(\alpha)$ e então $(x y)|\alpha$ pelo Lema 43 e $(x y)|\beta$. Como $\text{orb}(\beta, x) \neq \text{orb}(\beta, y)$ então $(x y) \nmid \beta$ pelo Lema 43. Como encontramos uma contradição, deduzimos que $\text{Supp}(\alpha) \neq \text{Supp}(\beta)$. Portanto $\text{Supp}(\alpha) \subset \text{Supp}(\beta)$.

Em ambos os casos, temos $\text{Supp}(\alpha) \subset \text{Supp}(\beta)$.

□

3.5 Produtos envolvendo 3-ciclos

O efeito do produto de uma permutação por um 3-ciclo é distinto do efeito de um produto por um 2-ciclo. Ao contrário do caso envolvendo 2-ciclos, a ordem dos três elementos no suporte do 3-ciclo é relevante para determinar a modificação na decomposição em ciclos de uma permutação. No Exemplo 47 descrevemos informalmente os efeitos de um produto envolvendo um 3-ciclo.

Exemplo 47. Dado o 3-ciclo $\alpha = (a \ b \ c)$ e uma permutação $\pi = (a \ d \ b \ c)(e \ f \ g)$ ambos definidos sobre $E = \{a, b, c, d, e, f, g\}$. Os produtos envolvendo as duas permutações são:

$$(a \ b \ c)(a \ d \ b \ c)(e \ f \ g) = (b \ a \ d \ c)(e \ f \ g)$$

e

$$(a \ d \ b \ c)(e \ f \ g)(a \ b \ c) = (b \ a \ c \ d)(e \ f \ g).$$

As órbitas de π , $\alpha\pi$ e $\pi\alpha$ são as mesmas, no entanto os ciclos dessas permutações são diferentes. Os segmentos ad e b têm sua ordem trocada em $\alpha\pi$, o mesmo ocorre com os segmentos db e a em $\pi\alpha$.

Considere a permutação $\sigma = (a\ g\ c\ e\ b\ d\ f)$ sobre E . Os produtos $\alpha\sigma$ e $\sigma\alpha$ são respectivamente:

$$(a\ b\ c)(a\ g\ c\ e\ b\ d\ f) = (a\ g)(c\ e)(b\ d\ f)$$

e

$$(a\ g\ c\ e\ b\ d\ f)(a\ b\ c) = (a\ d\ f)(b\ e)(c\ g).$$

Nesse caso, as órbitas de σ , $\alpha\sigma$ e $\sigma\alpha$ são distintas entre si. A única órbita de σ foi dividida em três órbitas em $\alpha\sigma$ e $\sigma\alpha$. Os segmentos ag , ce e bdf são transformados em três ciclos em $\alpha\sigma$, enquanto os segmentos dfa , eb e gc transformam-se em três ciclos em $\sigma\alpha$.

Considere agora a permutação $\theta = (a\ d)(b\ g)(c\ f)$ sobre E . Os produtos $\alpha\theta$ e $\theta\alpha$ são:

$$(a\ b\ c)(a\ d)(b\ g)(c\ f) = (a\ d\ b\ g\ c\ f)$$

e

$$(a\ d)(b\ g)(c\ f)(a\ b\ c) = (a\ g\ b\ f\ c\ d).$$

As quatro órbitas de θ (os conjuntos $\{a, d\}$, $\{b, g\}$, $\{c, f\}$ e $\{e\}$) foram reduzidas para duas órbitas em $\alpha\theta$ e $\theta\alpha$ (os conjuntos $\{a, d, b, g, c, f\}$ e $\{e\}$). Em $\alpha\theta$ os três ciclos $(a\ d)$, $(b\ g)$ e $(c\ f)$ são transformados no ciclo $(a\ d\ b\ g\ c\ f)$ reunindo os segmentos ad , bg e cf nessa ordem (a mesma ordem circular dos elementos a, b e c em α). Em $\theta\alpha$ os três ciclos $(a\ d)$, $(b\ g)$ e $(c\ f)$ são transformados no ciclo $(a\ g\ b\ f\ c\ d)$ que reúne os segmentos da , gb e fc nessa ordem (os segmentos são reunidos na mesma ordem circular dos elementos a, b e c em α , mas cada segmento sofre uma rotação tal que a, b e c aparecem no fim de cada segmento).

Como último exemplo, considere a permutação $\beta = (a\ d\ b\ g)(c\ f\ e)$ sobre E . Os produtos $\alpha\beta$ e $\beta\alpha$ são:

$$(a\ b\ c)(a\ d\ b\ g)(c\ f\ e) = (a\ d\ c\ f\ e)(b\ g)$$

e

$$(a\ d\ b\ g)(c\ f\ e)(a\ b\ c) = (a\ g)(b\ f\ e\ c\ d).$$

As duas órbitas de β são transformadas em duas novas órbitas em $\alpha\beta$ e $\beta\alpha$. Podemos descrever o efeito do 3-ciclo α em β no produto $\alpha\beta$ como uma troca dos segmentos bg e cfe entre os seus respectivos ciclos. No produto $\beta\alpha$ os segmentos trocados são ga e fec .

Meidanis e Dias [39] enunciaram sem demonstração qual é o comportamento do produto envolvendo uma permutação qualquer e um 3-ciclo. Apresentamos no Teorema 48 uma prova desse resultado.

Teorema 48. *Dado um 3-ciclo $\alpha = (a\ b\ c)$ e uma permutação β ambos sobre um conjunto E , temos*

1. Os elementos a, b e c pertencem a órbitas de três ciclos distintos de β , se e somente se as órbitas desses três ciclos são reunidas em uma única órbita na qual a, b e c aparecem na ordem circular abc em um ciclo de $\alpha\beta$.
2. Dois elementos de a, b e c , digamos a e b , pertencem a mesma órbita, enquanto c pertence a uma órbita distinta de β se e somente se $c \in \text{orb}(\alpha\beta, a)$, $b \notin \text{orb}(\alpha\beta, a)$ e $o(\beta) = o(\alpha\beta)$.
3. Os elementos a, b e c pertencem a mesma órbita e aparecem na ordem circular abc em um ciclo de β se e somente se os elementos a, b e c pertencem a mesma órbita e aparecem na ordem circular acb em um ciclo de $\alpha\beta$.
4. Os elementos a, b e c pertencem a mesma órbita e aparecem na ordem circular acb em um ciclo de β , se e somente se os elementos a, b e c pertencem a órbitas distintas de $\alpha\beta$.

Prova: A demonstração de cada um dos casos do teorema utiliza-se da seguinte observação: para x em E distinto de $\beta^{-1}a$, $\beta^{-1}b$ e $\beta^{-1}c$ temos $\alpha\beta x = \beta x$ pois $\alpha y = y$ para y diferente de a, b e c .

1. Se os elementos a, b e c pertencem a órbitas de três ciclos distintos de β , então avaliaremos qual é o valor de $\alpha\beta x$ para $x \in E$. Se $x = \beta^{-1}a$ então temos $\beta\beta^{-1}a = a$ e $\alpha a = b$. Logo $\alpha\beta\beta^{-1}a = b$. De maneira semelhante obtemos $\alpha\beta\beta^{-1}b = c$ e $\alpha\beta\beta^{-1}c = a$. Como $\alpha\beta x = \beta x$ para x distinto de $\beta^{-1}a$, $\beta^{-1}b$ e $\beta^{-1}c$ então $\beta^{-1}a$ é acessível por a , $\beta^{-1}b$ é acessível por b e $\beta^{-1}c$ é acessível por c em $\alpha\beta$. Como $\beta^{-1}a$ é acessível por a e $\alpha\beta\beta^{-1}a = b$ então b é acessível por a em $\alpha\beta$. Utilizando a transitividade nos casos dos elementos b e c , obtemos que o elemento c é acessível por b e o elemento a é acessível por c . Logo a, b e c pertencem a mesma órbita de $\alpha\beta$. Além disso, o elemento c não é nenhum dos elementos $\alpha\beta^i a$ para $1 \leq i \leq k$, onde k é o menor inteiro não negativo tal que $\alpha\beta^k a = b$, devido a $\alpha\beta x = \beta x$ para x distinto de $\beta^{-1}a$ e $\alpha\beta\beta^{-1}a = b$. Logo, temos $k < m$ onde m é o menor inteiro não negativo tal que $\alpha\beta^m a = c$. Portanto as órbitas dos três ciclos são reunidas em uma única órbita na qual a, b e c aparecem na ordem circular abc em um ciclo de $\alpha\beta$.

Por outro lado, suponha que a, b e c aparecem na ordem circular abc em um ciclo de $\alpha\beta$. Para demonstrar que os elementos devem pertencer a três órbitas distintas em β avaliaremos a permutação $\alpha^{-1}\alpha\beta$. Temos $\alpha^{-1}\alpha\beta x = \alpha\beta x$ para $x \neq (\alpha\beta)^{-1}a$, $x \neq (\alpha\beta)^{-1}b$ e $x \neq (\alpha\beta)^{-1}c$ pois $\alpha x = x$. Como $\alpha\beta(\alpha\beta)^{-1}b = b$ e $\alpha^{-1}b = a$ então $\beta(\alpha\beta)^{-1}b = a$. De maneira semelhante, obtemos $\beta(\alpha\beta)^{-1}c = b$ e $\beta(\alpha\beta)^{-1}a = c$. Como $\alpha^{-1}\alpha\beta x = \alpha\beta x$ para $x \neq (\alpha\beta)^{-1}b$ e $\beta(\alpha\beta)^{-1}b = a$ então os elementos b e c não pertencem à órbita contendo a . Como $\alpha^{-1}\alpha\beta x = \alpha\beta x$ para $x \neq (\alpha\beta)^{-1}c$ e $\beta(\alpha\beta)^{-1}c = b$ então os elementos a e c não pertencem à órbita contendo b . Logo os elementos a, b e c pertencem a três órbitas distintas de β .

2. Suponha que $b \in \text{orb}(\beta, a)$ e $c \notin \text{orb}(\beta, a)$. Como $\alpha\beta x = \beta x$ para $x \neq \beta^{-1}a$, $x \neq \beta^{-1}b$ e $x \neq \beta^{-1}c$, então todas as órbitas que não contêm $\beta^{-1}a$, $\beta^{-1}b$ e $\beta^{-1}c$ são mantidas inalteradas em $\alpha\beta$. Como $\beta\beta^{-1}b = b$ e $\alpha b = c$ então $\alpha\beta\beta^{-1}b = c$. Logo $c \in \text{orb}(\alpha\beta, a)$ pois $\beta^{-1}b$ é acessível por a em $\alpha\beta$. Como $\beta\beta^{-1}c = c$ e $\alpha c = a$ então $\alpha\beta\beta^{-1}c = a$. Como $b \notin \text{orb}(\beta, c)$ e $\alpha\beta\beta^{-1}b = c$ então $b \notin \text{orb}(\alpha\beta, a)$. Como $\beta^{-1}a$ é acessível por b em β , $\beta\beta^{-1}a = a$ e $\alpha a = b$ então $\alpha\beta\beta^{-1}a = b$. Logo as duas órbitas contendo a e c em β são substituídas por duas órbitas contendo respectivamente a e b em $\alpha\beta$ e as demais órbitas de β são preservadas em $\alpha\beta$. Portanto, obtemos $c \in \text{orb}(\alpha\beta, a)$, $b \notin \text{orb}(\alpha\beta, a)$ e $o(\beta) = o(\alpha\beta)$.

Por outro lado, se $c \in \text{orb}(\alpha\beta, a)$, $b \notin \text{orb}(\alpha\beta, a)$ e $o(\beta) = o(\alpha\beta)$ então podemos provar que $b \in \text{orb}(\beta, a)$ e $c \notin \text{orb}(\beta, a)$ fazendo $\alpha' = \alpha^{-1}$ e $\beta' = \alpha\beta$ e aplicando a implicação que acabamos de demonstrar para α' e β' .

3. Suponha que a, b e c pertencem a mesma órbita e aparecem na ordem circular abc em um ciclo de β . Como a, b e c aparecem na ordem circular abc então $k < m$ onde k é o menor inteiro não negativo tal que $b = \beta^k a$ e m é o menor inteiro não negativo tal que $c = \beta^m a$.

Como $\beta\beta^{-1}b = b$ e $\alpha b = c$ então $\alpha\beta\beta^{-1}b = c$. Aplicando o produto $\alpha\beta$ de maneira semelhante para $\beta^{-1}a$ e $\beta^{-1}c$, obtemos $\alpha\beta\beta^{-1}a = b$ e $\alpha\beta\beta^{-1}c = a$. Os demais elementos de E distintos de $\beta^{-1}a$, $\beta^{-1}b$ e $\beta^{-1}c$ são mapeados em $\alpha\beta$ para os mesmos elementos aos quais eram mapeados em β . Logo a, b e c pertencem a mesma órbita em $\alpha\beta$ pois c é acessível por a e b é acessível por c . Além disso, como a relação de acessibilidade é preservada para os elementos distintos de $\beta^{-1}a$, $\beta^{-1}b$ e $\beta^{-1}c$, então $c = (\alpha\beta)^k a$ e $b = (\alpha\beta)^m a$. Portanto, os elementos a, b e c pertencem a mesma órbita e aparecem na ordem circular acb em um ciclo de $\alpha\beta$.

Por outro lado, se os elementos a, b e c pertencem a mesma órbita e aparecem na ordem circular acb em um ciclo de $\alpha\beta$ então podemos demonstrar que os elementos a, b e c pertencem a mesma órbita e aparecem na ordem circular abc em um ciclo de β fazendo $\alpha' = \alpha^{-1}$ e $\beta' = \alpha\beta$ e aplicando o resultado demonstrado acima neste item.

4. Se os elementos a, b e c pertencem a mesma órbita e aparecem na ordem circular acb em um ciclo de β , então $k < m$ onde k é o menor inteiro não negativo tal que $c = \beta^k a$ e m é o menor inteiro não negativo tal que $b = \beta^m a$.

Como $\beta\beta^{-1}a = a$ e $\alpha a = b$ então $\alpha\beta\beta^{-1}a = b$. Aplicando o produto $\alpha\beta$ de maneira semelhante para $\beta^{-1}b$ e $\beta^{-1}c$, obtemos $\alpha\beta\beta^{-1}b = c$ e $\alpha\beta\beta^{-1}c = a$. Como $\alpha\beta\beta^{-1}a = b$, $\alpha\beta x = \beta x$ para $x \in E$ distinto de $\beta^{-1}a$, $\beta^{-1}b$ e $\beta^{-1}c$ e os elementos a, b e c ocorrem na ordem acb em β , então $a \notin \text{orb}(\alpha\beta, b)$ e $c \notin \text{orb}(\alpha\beta, b)$ pois a órbita $\text{orb}(\alpha\beta, b)$ contém apenas os elementos $\beta^{m+i}a$ para $0 \leq i \leq n - m - 1$ e $n = |\text{orb}(\beta, a)|$ em β . Além disso, o elemento a não está na órbita $\text{orb}(\alpha\beta, c)$, pois a órbita $\text{orb}(\alpha\beta, c)$ é composta apenas

pelos elementos $\beta^{k+i}a$ para $0 \leq i \leq m - k - 1$ devido a $\alpha\beta\beta^{-1}b = c$, $\alpha\beta x = \beta x$ para $x \in E$ distinto de $\beta^{-1}a$, $\beta^{-1}b$ e $\beta^{-1}c$ e os elementos a , b e c ocorrem na ordem acb em β . Portanto, os elementos a , b e c pertencem a órbitas distintas de $\alpha\beta$.

Por outro lado, se os elementos a , b e c pertencem a órbitas distintas de $\alpha\beta$ então é possível demonstrar que os elementos a , b e c pertencem a mesma órbita e aparecem na ordem circular acb em um ciclo de β por meio da aplicação do item 1 demonstrado anteriormente para as permutações α' e β' onde $\alpha' = \alpha^{-1}$ e $\beta' = \alpha\beta$. Segundo o resultado do item 1, as órbitas contendo a , b e c são reunidas em uma única órbita em $\alpha'\beta' = \beta$ e esses elementos possuem a mesma ordem circular em $\alpha'\beta'$ e em α' , ou seja, a ordem acb .

□

Meidanis e Dias [39] utilizaram o Teorema 48 para analisar eventos de rearranjo representados por 3-ciclos, mas não desenvolveram o relacionamento entre o produto de três ciclos e os conceitos de norma e divisibilidade, por exemplo. De maneira semelhante ao caso do produto envolvendo 2-ciclos, apresentamos as propriedades que relacionam a norma, a divisibilidade e o número de órbitas de uma permutação. Esses novos resultados são contribuições de nossa pesquisa para a extensão do formalismo algébrico.

Lema 49. Sejam $\alpha = (a \ b \ c)$ um 3-ciclo e β uma permutação, ambos sobre E . Então os elementos a , b e c pertencem a $Supp(\gamma)$ para um ciclo γ de β e aparecem na ordem circular abc em γ se e somente se $(a \ b \ c)|\beta$.

Prova: Se os elementos a , b e c pertencem a $Supp(\gamma)$ para um ciclo γ de β e aparecem nessa ordem circular em α então pelo Teorema 48, temos $o(\beta\alpha^{-1}) = o(\beta) + 2$. Além disso, temos $\|\alpha\| = 2$ uma vez que α é um 3-ciclo. Daí, desenvolvendo-se a fórmula que relaciona a norma e o número de órbitas de uma permutação:

$$\|\beta\alpha^{-1}\| = |E| - o(\beta\alpha^{-1}) = |E| - (o(\beta) + 2) = |E| - (|E| - \|\beta\|) - 2 = \|\beta\| - \|\alpha\|.$$

Logo $\|\beta\alpha^{-1}\| = \|\beta\| - \|\alpha\|$ e $\alpha|\beta$.

Por outro lado, se $\alpha|\beta$ então $o(\beta\alpha^{-1}) = o(\beta) + \|\alpha\|$ pelo Lema 39, ou seja α^{-1} é um 3-ciclo tal que $o(\beta\alpha^{-1}) = o(\beta) + 2$. Logo, pelo Teorema 48, os elementos a , b e c pertencem a $Supp(\gamma)$ para um ciclo γ de β e aparecem na ordem circular abc em γ . □

Considere uma permutação π sobre E tal que existe um 3-ciclo $\alpha = (x \ y \ z)$ cujos elementos são todos contidos no suporte de um ciclo de π . Nessa situação, os elementos de α podem aparecer em duas ordens em π : xyz ou xzy . O papel dos 3-ciclos na variação no número de órbitas é determinado não apenas pela distribuição dos elementos do 3-ciclo entre os ciclos de uma permutação, mas também pela ordem em que os elementos aparecem na permutação.

Lema 50. Sejam $\alpha = (a \ b \ c)$ um 3-ciclo e β uma permutação, ambos sobre E . Então

1. $\alpha|\beta$ se e somente se $o(\beta\alpha^{-1}) = o(\beta) + 2$.
2. se $\alpha \nmid \beta$ e todos os 2-ciclos $(a\ b)$, $(a\ c)$ e $(b\ c)$ dividem β então $o(\beta\alpha^{-1}) = o(\beta)$.
3. se $\alpha \nmid \beta$ e somente um dos 2-ciclos $(a\ b)$, $(a\ c)$ e $(b\ c)$ dividem β então $o(\beta\alpha^{-1}) = o(\beta)$.
4. se $(a\ b)$, $(a\ c)$ e $(b\ c)$ não dividem β então $o(\beta\alpha^{-1}) = o(\beta) - 2$.

Prova:

1. Se $\alpha|\beta$ então $\|\beta\alpha^{-1}\| = \|\beta\| - \|\alpha\|$. Além disso, temos $\|\beta\alpha^{-1}\| = |E| - o(\beta\alpha^{-1})$ e $\|\beta\| = |E| - o(\beta)$, então usando essas igualdades e o fato de $\alpha|\beta$ temos $|E| - o(\beta\alpha^{-1}) = |E| - o(\beta) - 2$ que implica $o(\beta\alpha^{-1}) = o(\beta) + 2$. Inversamente, se $o(\beta\alpha^{-1}) = o(\beta) + 2$ então $\|\beta\alpha^{-1}\| = \|\beta\| - 2$. Como $\|\alpha\| = 2$ então $\|\beta\alpha^{-1}\| = \|\beta\| - \|\alpha\|$. Portanto $\alpha|\beta$.
2. Se $\alpha \nmid \beta$ e todos os 2-ciclos $(a\ b)$, $(a\ c)$ e $(b\ c)$ dividem β então, pelo Lema 43, temos $orb(\beta, a) = orb(\beta, b) = orb(\beta, c)$ e pelo Lema 49 temos $\alpha^{-1}|\beta$. Decompondo α^{-1} em 2-ciclos, temos $\alpha^{-1} = (c\ b\ a) = (c\ b)(b\ a)$. Como $(c\ b)|\beta$ então $o(\beta(c\ b)) = o(\beta) + 1$. Além disso, temos $a \in orb(\beta(c\ b), c)$ porque $\beta(c\ b)x = \beta x$ para $x \neq b$ e $x \neq c$ e os elementos c , b e a aparecem nessa ordem circular em β . Então $(b\ a) \nmid \beta(c\ b)$, e $o(\beta(c\ b)(b\ a)) = o(\beta(c\ b)) - 1 = o(\beta) + 1 - 1 = o(\beta)$. Portanto $o(\beta\alpha^{-1}) = o(\beta)$.
3. Suponha que $\alpha \nmid \beta$ e, sem perda de generalidade, temos $(a\ b)|\beta$, então $orb(\beta, a) = orb(\beta, b) \neq orb(\beta, c)$. Considere a decomposição em 2-ciclos $(b\ a)(a\ c)$ da permutação α^{-1} . Pelo Lema 45 e como $o(\beta(b\ a)) = o((b\ a)\beta)$ então $o(\beta(b\ a)) = o(\beta) + 1$. Além disso, temos $orb(\beta, c) \in Orb(\beta(b\ a))$ porque $\beta(b\ a)x = \beta x$ para $x \neq b$ e $x \neq a$. Logo a , b e c aparecem em órbitas distintas em $\beta(b\ a)$. Daí $(a\ c) \nmid \beta(b\ a)$ e, usando o Lema 45 e o fato de que $o(\beta\alpha) = o(\alpha\beta)$ quando α é um 2-ciclo, derivamos

$$o(\beta(b\ a)(a\ c)) = o(\beta(b\ a)) - 1 = o(\beta) + 1 - 1 = o(\beta).$$

Portanto $o(\beta\alpha^{-1}) = o(\beta)$.

4. Se $(a\ b)$, $(a\ c)$ e $(b\ c)$ não dividem β então $orb(\beta, a)$, $orb(\beta, b)$ e $orb(\beta, c)$ são órbitas disjuntas de β pelo Lema 43, ou seja, cada elemento no suporte de α aparece em um ciclo distinto em β . Mas podemos reescrever α^{-1} como $\alpha^{-1} = (c\ b\ a) = (c\ b)(b\ a)$. Logo $o(\beta(c\ b)) = o(\beta) - 1$ decorre de $(c\ b) \nmid \beta$ e do Lema 45. Como $\beta(c\ b)x = \beta x$ para $x \neq b$ e $x \neq c$ então $orb(\beta, a) \in Orb(\beta(c\ b))$. Logo $o(\beta(c\ b)(b\ a)) = o(\beta) - 2$ porque $(b\ a) \nmid \beta(c\ b)$. Portanto $o(\beta\alpha^{-1}) = o(\beta) - 2$.

□

Os conceitos de norma e divisibilidade são relacionados à representação de uma permutação como um produto de 2-ciclos. Podemos definir, com pequenas modificações, conceitos

semelhantes que são relacionados à decomposição de uma permutação em 3-ciclos. Diferentemente do caso do produto de 2-ciclos, nem sempre é possível encontrar uma decomposição em 3-ciclos para uma permutação. Dizemos que uma permutação é *decomponível* em 3-ciclos quando existe uma decomposição em 3-ciclos dessa permutação. Dada uma permutação π sobre E decomponível em 3-ciclos, definimos a *3-norma* de π , denotada por $\|\pi\|_3$, como o número mínimo de 3-ciclos em uma decomposição em 3-ciclos de π .

Exemplo 51. Sejam $\pi = (7\ 2\ 8\ 1)(3\ 5\ 10)(6\ 4\ 9)$ e $\gamma = (10\ 2\ 8\ 5\ 1)(3\ 6\ 4\ 7\ 9)$ permutações sobre $\{x \in \mathbb{Z} \mid 1 \leq x \leq 10\}$. A permutação γ pode ser representada por meio do seguinte produto de 3-ciclos:

$$\gamma = (10\ 2\ 8)(8\ 5\ 1)(3\ 6\ 4)(4\ 7\ 9).$$

Logo $\|\gamma\|_3 \leq 4$.

Por outro lado, a permutação π não possui uma decomposição em 3-ciclos. Se tentarmos encontrar um produto envolvendo o número máximo de 3-ciclos que resulte em π esse produto conterá sempre um 2-ciclo além dos 3-ciclos. Por exemplo:

$$\pi = (7\ 2\ 8)(8\ 1\ 3)(1\ 3\ 5)(5\ 10\ 6)(10\ 6\ 4)(4\ 9).$$

Seria possível determinar se uma permutação é decomponível em 3-ciclos sem procurar por uma decomposição em 3-ciclos? Para responder a essa pergunta, iremos demonstrar alguns fatos referentes a paridade da norma de uma permutação. A paridade da norma nos ajudará a caracterizar as permutações que são decomponíveis em 3-ciclos.

Lema 52. Dadas as permutações α e β sobre E , temos $\|\alpha\beta\| \equiv \|\alpha\| + \|\beta\| \pmod{2}$

Prova: Faremos uma demonstração por indução em $\|\beta\|$.

Considere como caso base $\|\beta\| = 0$ e $\|\beta\| = 1$. Para $\|\beta\| = 0$, então $\beta = \iota$ (pela Proposição 32) e logo $\|\alpha\beta\| \equiv \|\alpha\| \equiv \|\alpha\| + \|\beta\| \pmod{2}$. Se $\|\beta\| = 1$ então $\beta = (a\ b)$ e temos dois subcasos:

- se $(a\ b) \nmid \alpha$ então $\|\alpha\beta\| = \|\alpha\| + 1$ pelo Lema 44. Logo $\|\alpha\beta\| \equiv \|\alpha\| + 1 \equiv \|\alpha\| + \|\beta\| \pmod{2}$.
- se $(a\ b) \mid \alpha$ então $\|\alpha\beta\| = \|\alpha\| - 1$ pelo Lema 44. Logo $\|\alpha\beta\| = \|\alpha\| - 1 \equiv (\|\alpha\| \pmod{2} + (-1) \pmod{2}) \pmod{2} \equiv (\|\alpha\| + 1) \pmod{2} \equiv \|\alpha\| + \|\beta\| \pmod{2}$.

Tomaremos como hipótese de indução que o resultado segue para $\|\beta\| < k$.

Seja $\beta = \beta_1 \beta_2 \dots \beta_k$, onde β_i é um 2-ciclo e k é mínimo. Considere β_1 e $\beta' = \beta_2 \dots \beta_k$. Pela hipótese de indução e como $\|\beta'\| < \|\beta\|$ devido a minimalidade da decomposição em 2-ciclos, temos $\|\alpha\beta_1\beta'\| \equiv \|\alpha\beta_1\| + \|\beta'\| \pmod{2}$. Mas $\|\alpha\beta_1\| \equiv \|\alpha\| + 1 \pmod{2}$ como

demonstramos no caso base. Logo $\|\alpha\beta\| \equiv (\|\alpha\| + 1 \pmod{2}) + \|\beta'\| \pmod{2} \equiv \|\alpha\| + (1 + \|\beta'\|) \pmod{2}$. Como $\|\beta\| = 1 + \|\beta'\|$, então nós temos $\|\alpha\beta\| \equiv \|\alpha\| + \|\beta\| \pmod{2}$. \square

Uma permutação é uma *permutação ímpar* quando sua norma é ímpar, caso contrário ela é uma *permutação par*. Um r -ciclo é um *ciclo ímpar* quando r é ímpar, caso contrário ele é um *ciclo par*. Os conceitos de paridade (par ou ímpar) em permutações em geral e ciclos são opostos.

Lema 53. Ciclos ímpares são permutações pares e ciclos pares são permutações ímpares.

Prova: Pelo Lema 30 um r -ciclo $\pi = (a_1 a_2 \dots a_r)$ pode ser escrito como

$$\pi = (a_1 a_r)(a_1 a_{r-1}) \dots (a_1 a_3)(a_1 a_2),$$

ou seja, como um produto de $r - 1$ 2-ciclos. Esse produto de 2-ciclos $(a_1 a_r) \dots (a_1 a_2)$ é, de fato, uma decomposição em 2-ciclos de π . Mostraremos que essa decomposição é uma decomposição mínima em 2-ciclos. Pela Proposição 34 temos $\|\pi\| = |E| - o(\pi)$. Seja f o número de elementos fixos em π . Logo

$$|E| - o(\pi) = |Supp(\pi)| + f - (no(\pi) + f) = |Supp(\pi)| - no(\pi) = r - 1,$$

uma vez que π é um r -ciclo. Então o produto de 2-ciclos envolve $\|\pi\|$ 2-ciclos e ele é uma decomposição mínima em 2-ciclos de π . Logo a norma $\|\pi\|$ é ímpar (par) se e somente se r é par (ímpar). Portanto ciclos ímpares são permutações pares e ciclos pares são permutações ímpares. \square

Mostramos no Lema 54 uma caracterização das permutações que são decomponíveis em um produto de 3-ciclos:

Lema 54. Dada uma permutação π sobre E , existe uma decomposição em 3-ciclos de π se e somente se $\|\pi\|$ é par, ou seja, a 3-norma é definida para permutações pares apenas.

Prova: Seja π uma permutação tal que $\|\pi\|$ é par; i.e. $0 \equiv \|\pi\| \pmod{2}$. Para cada par de 2-ciclos consecutivos; digamos α e β , em uma decomposição mínima em 2-ciclos de π podemos obter 3-ciclos de acordo com dois casos:

- i) Se $|Supp(\alpha) \cap Supp(\beta)| = 1$; isto é, temos $\alpha = (a b)$ e $\beta = (b c)$, então substituímos os dois ciclos pelo 3-ciclo $(a b c)$.
- ii) Se $Supp(\alpha) \cap Supp(\beta) = \emptyset$; i.e. $\alpha = (a b)$, $\beta = (c d)$, e a, b, c e d são distintos entre si, então obtemos dois 3-ciclos $(a b c)$ e $(b c d)$.

O caso no qual $Supp(\alpha) = Supp(\beta)$; isto é, temos $\alpha = \beta = (a b)$ não ocorre em um produto mínimo de 2-ciclos pois teríamos $\alpha\beta = \beta\alpha = \iota$.

Como o norma de π é par então o número de 2-ciclos em uma decomposição em 2-ciclos de π é par. Logo, ao aplicarmos uma transformação dos pares de 2-ciclos em 3-ciclos seguindo os casos acima, o resultado será uma decomposição da permutação π em 3-ciclos. Agora, considere a permutação π tal que há uma decomposição mínima em 3-ciclos de π : $\pi = \alpha_1 \alpha_2 \dots \alpha_k$. Logo $\|\pi\| = \|\alpha_1 \alpha_2 \dots \alpha_k\|$ e pela generalização do Lema 52, temos:

$$\|\alpha_1 \alpha_2 \dots \alpha_k\| \equiv \|\alpha_1\| + \|\alpha_2\| + \dots + \|\alpha_k\| \pmod{2}$$

Como a norma de um 3-ciclo é 2 então $\|\alpha_1 \alpha_2 \dots \alpha_k\| \equiv 0 \pmod{2}$; ou seja, a norma de π é par.

Portanto $\|\pi\|_3$ é definido apenas para permutações pares. \square

De modo semelhante à norma de uma permutação, a 3-norma de uma permutação par possui algumas propriedades úteis:

Lema 55. Dadas as permutações pares α e β ambas sobre E , temos

1. $\|\alpha\|_3 = 0$ se e somente se $\alpha = \iota$
2. $\|\alpha^{-1}\|_3 = \|\alpha\|_3$
3. $\|\beta \cdot \alpha\|_3 = \|\alpha\|_3$
4. $\|\alpha\beta\|_3 \leq \|\alpha\|_3 + \|\beta\|_3$
5. $\|\alpha\beta\|_3 = \|\beta\alpha\|_3$.

Prova:

1. Se α pode ser representado como um produto de zero 3-ciclos então α é a identidade. Por outro lado, a 3-norma da identidade é zero.
2. Seja $\alpha_1 \dots \alpha_k$ um produto mínimo de 3-ciclos cujo resultado é α , então

$$\alpha^{-1} = (\alpha_1 \dots \alpha_k)^{-1} = \alpha_k^{-1} \dots \alpha_1^{-1}.$$

Logo, o produto $\alpha_k^{-1} \dots \alpha_1^{-1}$ é uma decomposição em 3-ciclos de α^{-1} e $\|\alpha^{-1}\|_3 \leq \|\alpha\|_3$. Por simetria, obtemos $\|\alpha^{-1}\|_3 \geq \|\alpha\|_3$. Portanto $\|\alpha^{-1}\|_3 = \|\alpha\|_3$.

3. Dado $\alpha = \alpha_1 \dots \alpha_k$, tal que $\alpha_1 \dots \alpha_k$ é uma decomposição em 3-ciclos de α , então

$$\beta \cdot \alpha = (\beta \cdot \alpha_1) \dots (\beta \cdot \alpha_k)$$

e $\beta \cdot \alpha_i$, para $1 \leq i \leq k$, é um 3-ciclo uma vez que a conjugação preserva a estrutura do ciclo. Logo $\|\beta \cdot \alpha\|_3 \leq \|\alpha\|_3$. Como esse resultado vale para permutações quaisquer, temos $\|\beta^{-1} \cdot (\beta \cdot \alpha)\|_3 \leq \|\beta \cdot \alpha\|_3$. Como $\beta^{-1} \cdot (\beta \cdot \alpha) = \alpha$, então $\|\alpha\|_3 \leq \|\beta \cdot \alpha\|_3$. Portanto $\|\alpha\|_3 = \|\beta \cdot \alpha\|_3$.

4. Seja $\alpha = \alpha_1 \dots \alpha_k$ e $\beta = \beta_1 \dots \beta_l$ respectivas decomposições mínimas em 3-ciclos das permutações α e β . Então

$$\alpha\beta = \alpha_1 \dots \alpha_k \beta_1 \dots \beta_l$$

$$\text{e } \|\alpha\beta\|_3 \leq k + l = \|\alpha\|_3 + \|\beta\|_3.$$

5. Segundo o item 3, temos $\|\beta\alpha\|_3 = \|\alpha \cdot (\beta\alpha)\|_3$ e $\alpha \cdot (\beta\alpha) = \alpha\beta$. Portanto $\|\beta\alpha\|_3 = \|\alpha\beta\|_3$.

□

Dadas as permutações pares α e β sobre E , dizemos que α 3-divide β , o que é denotado por $\alpha|_3\beta$, quando $\|\beta\alpha^{-1}\|_3 = \|\beta\|_3 - \|\alpha\|_3$. O Lema 56 reúne algumas propriedades básicas da 3-divisibilidade.

Lema 56. Dadas as permutações pares α , β e γ sobre E , temos

1. $\alpha|_3\alpha$
2. se $\alpha|_3\beta$ e $\beta|_3\alpha$ então $\alpha = \beta$
3. se $\alpha|_3\beta$ e $\beta|_3\gamma$ então $\alpha|_3\gamma$
4. $\alpha|_3\beta$ se e somente se $\alpha^{-1}|_3\beta^{-1}$.
5. se $\alpha|_3\beta$ então $\gamma \cdot \alpha|_3\gamma \cdot \beta$

Prova:

1. A propriedade reflexiva segue diretamente de $\|\alpha\alpha^{-1}\|_3 = \|\iota\|_3 = 0 = \|\alpha\|_3 - \|\alpha\|_3$.
2. Sejam α e β permutações pares sobre E tais que $\alpha|_3\beta$ e $\beta|_3\alpha$. Logo

$$\|\beta\alpha^{-1}\|_3 = \|\beta\|_3 - \|\alpha\|_3 \quad \text{e} \quad \|\alpha\beta^{-1}\|_3 = \|\alpha\|_3 - \|\beta\|_3.$$

Somando os membros de cada igualdade obtemos $\|\beta\alpha^{-1}\|_3 + \|\alpha\beta^{-1}\|_3 = 0$.

Como $\alpha\beta^{-1} = (\beta\alpha^{-1})^{-1}$, então $\|\beta\alpha^{-1}\|_3 = \|\alpha\beta^{-1}\|_3$ pelo Lema 55. A partir das igualdades $\|\beta\alpha^{-1}\|_3 + \|\alpha\beta^{-1}\|_3 = 0$ e $\|\beta\alpha^{-1}\|_3 - \|\alpha\beta^{-1}\|_3 = 0$ obtemos $\|\alpha\beta^{-1}\|_3 = 0$, i.e. $\alpha\beta^{-1} = \iota$. Logo $\alpha = \beta$ e a propriedade anti-simétrica segue-se.

3. Sejam α , β e γ permutações pares sobre E tais que $\alpha|_3\beta$ e $\beta|_3\gamma$. Então temos

$$\|\beta\alpha^{-1}\|_3 = \|\beta\|_3 - \|\alpha\|_3 \quad \text{e} \quad \|\gamma\beta^{-1}\|_3 = \|\gamma\|_3 - \|\beta\|_3$$

e somando essas equações obtemos:

$$\|\beta\alpha^{-1}\|_3 + \|\gamma\beta^{-1}\|_3 = \|\gamma\|_3 - \|\alpha\|_3.$$

Verificaremos o relacionamento entre $\|\gamma\alpha^{-1}\|_3$ e $\|\gamma\|_3 - \|\alpha\|_3$. As desigualdades

$$\|\gamma\alpha^{-1}\|_3 = \|\gamma\beta^{-1}\beta\alpha^{-1}\|_3 \leq \|\gamma\beta^{-1}\|_3 + \|\beta\alpha^{-1}\|_3 \leq \|\gamma\|_3 - \|\alpha\|_3,$$

e

$$\|\gamma\|_3 = \|\gamma\alpha^{-1}\alpha\|_3 \leq \|\gamma\alpha^{-1}\|_3 + \|\alpha\|_3$$

implicam $\|\gamma\alpha^{-1}\|_3 \geq \|\gamma\|_3 - \|\alpha\|_3$. Logo $\|\gamma\alpha^{-1}\|_3 = \|\gamma\|_3 - \|\alpha\|_3$ e $\alpha|_3\gamma$.

4. Se $\alpha|_3\beta$ e como $\|\gamma\|_3 = \|\gamma^{-1}\|_3$ então temos $\|(\beta\alpha^{-1})^1\|_3 = \|\beta^{-1}\|_3 - \|\alpha^{-1}\|_3$. Manipulando o primeiro termo da igualdade:

$$\|(\beta\alpha^{-1})^1\|_3 = \|\alpha\beta^{-1}\|_3 = \|\beta^{-1}\alpha\|_3.$$

Conseqüentemente, temos $\|\beta^{-1}\alpha\|_3 = \|\beta^{-1}\|_3 - \|\alpha^{-1}\|_3$, ou de modo equivalente temos $\alpha^{-1}|_3\beta^{-1}$.

Se $\alpha^{-1}|_3\beta^{-1}$, então $\|\beta^{-1}\alpha\|_3 = \|\beta^{-1}\|_3 - \|\alpha^{-1}\|_3$. Logo $\|\beta^{-1}\|_3 - \|\alpha^{-1}\|_3 = \|\beta\|_3 - \|\alpha\|_3$. Além disso, como mostrado anteriormente, temos $\|\beta^{-1}\alpha\|_3 = \|(\beta\alpha^{-1})^1\|_3$ e $\|(\beta\alpha^{-1})^1\|_3 = \|\beta\alpha^{-1}\|_3$. Portanto $\alpha|_3\beta$.

5. Se $\alpha|_3\beta$ então $\|\beta\alpha^{-1}\|_3 = \|\beta\|_3 - \|\gamma\|_3$. Mas $\|\beta\|_3 = \|\gamma \cdot \beta\|_3$. Logo

$$\|\gamma \cdot \beta\|_3 - \|\gamma \cdot \alpha\|_3 = \|\beta\|_3 - \|\alpha\|_3 = \|\beta\alpha^{-1}\|_3 = \|\gamma \cdot (\beta\alpha^{-1})\|_3 = \|(\gamma \cdot \beta)(\gamma \cdot \alpha^{-1})\|_3.$$

Mas temos $\|(\gamma \cdot \beta)(\gamma \cdot \alpha^{-1})\|_3 = \|(\gamma \cdot \beta)(\gamma \cdot \alpha)^{-1}\|_3$. Portanto $\gamma \cdot \alpha|_3\gamma \cdot \beta$. E por simetria da igualdade, obtemos que $\gamma \cdot \alpha|_3\gamma \cdot \beta$ implica $\alpha|_3\beta$.

□

O relacionamento entre a norma e a 3-norma de uma permutação par é dado pelo Lema 57 e Teorema 58.

Lema 57. Dada a permutação par π sobre E , temos $\|\pi\|_3 \geq \|\pi\|/2$.

Prova: Dada a permutação par π sobre E , seja $\tau_1 \dots \tau_k$ uma decomposição mínima em 3-ciclos de π e $k = \|\pi\|_3$. Cada 3-ciclo τ_i para $1 \leq i \leq k$ é da forma $(a \ b \ c)$ e pode ser escrito como o produto de dois 2-ciclos $(a \ b)(b \ c)$. Logo uma decomposição em 2-ciclos de π obtida pela substituição de um 3-ciclo por dois 2-ciclos possui $2\|\pi\|_3$ 2-ciclos e pela definição da norma temos $2\|\pi\|_3 \geq \|\pi\|$. Portanto temos $\|\pi\|_3 \geq \|\pi\|/2$. □

Teorema 58. Se uma permutação par π sobre E possui apenas ciclos ímpares em sua decomposição em ciclos disjuntos então $\|\pi\|_3 = \|\pi\|/2$.

Prova: Para qualquer permutação par π temos $\|\pi\|_3 \geq \|\pi\|/2$ pelo Lema 57.

Para mostrarmos que $\|\pi\|_3 \leq \|\pi\|/2$ dada uma permutação π cuja decomposição em ciclos disjuntos é formada apenas por ciclos ímpares, iremos obter uma decomposição em 3-ciclos de π a partir de uma decomposição mínima em 2-ciclos de π .

Dada a permutação π cuja decomposição em ciclos disjuntos é composta por ciclos ímpares apenas, considere uma decomposição mínima em 2-ciclos de π obtida do seguinte modo: substitua cada ciclo $\alpha = (a_1 \ a_2 \ \dots \ a_r)$ para $r > 2$ da decomposição em ciclos de π pelo produto de 2-ciclos

$$(a_1 \ a_2)(a_2 \ a_3)(a_3 \ a_4) \dots (a_{r-2} \ a_{r-1})(a_{r-1} \ a_r).$$

Essa decomposição em 2-ciclos é mínima pois possui $r - 1$ 2-ciclos de acordo com a fórmula da norma segundo a Proposição 34. Então é possível agrupar os 2-ciclos dessa decomposição de π em pares da forma $(a \ b)(b \ c)$ e para cada par $(a \ b)(b \ c)$ há um 3-ciclo correspondente $(a \ b \ c)$. Logo $\|\pi\|_3 \leq \|\pi\|/2$.

Portanto, se uma permutação par π sobre E possui apenas ciclos ímpares em sua decomposição em ciclos disjuntos então $\|\pi\|_3 = \|\pi\|/2$. \square

Este resultado não é válido em geral. Por exemplo, para a permutação $\alpha = (a \ b)(c \ d)$, temos que $\|\alpha\| = 2$ e $\|\alpha\|_3 = 2$.

3.6 Modelagem de Genomas

Nessa seção iremos apresentar uma versão do formalismo algébrico definido originalmente por Meidanis e Dias [39] para a modelagem do problema de rearranjo em genomas. A versão apresentada nessa seção difere do trabalho original de Meidanis e Dias, pois adotamos a convenção de aplicar eventos de rearranjo em genomas pela esquerda. Além disso, introduzimos o conceito de sistema de genes, analisamos os tipos de genomas que podem servir de entrada para um problema de rearranjo em genomas por meio do conceito de “compatibilidade” entre genomas e relacionamos a definição de um genoma ao conceito de divisibilidade de permutações. Esses novos conceitos estendem a versão do formalismo algébrico de Meidanis e Dias.

Iremos modelar genomas como permutações que obedecem a certas propriedades relacionadas ao conjunto de genes modelados e sua relação de complementariedade.

O par formado por um conjunto E e uma permutação Γ sobre E é chamado de um *sistema de genes*, denotado por (E, Γ) , quando $Supp(\Gamma) = E$ e $\Gamma\Gamma = \iota$. A permutação Γ associa cada elemento de E ao seu elemento complementar. Por exemplo, dado o conjunto $E = \{-4, -3, -2, -1, 1, 2, 3, 4\}$ e a permutação $\Gamma = (1 \ -1)(2 \ -2)(3 \ -3)(4 \ -4)$, o par (E, Γ) é um sistema de genes uma vez que o suporte de Γ é E e $\Gamma\Gamma = \iota$. Uma partição de E em dois conjuntos E_+ e E_- tal que $E_- = \{\Gamma x \mid x \in E_+\}$ é chamada de uma *partição válida*.

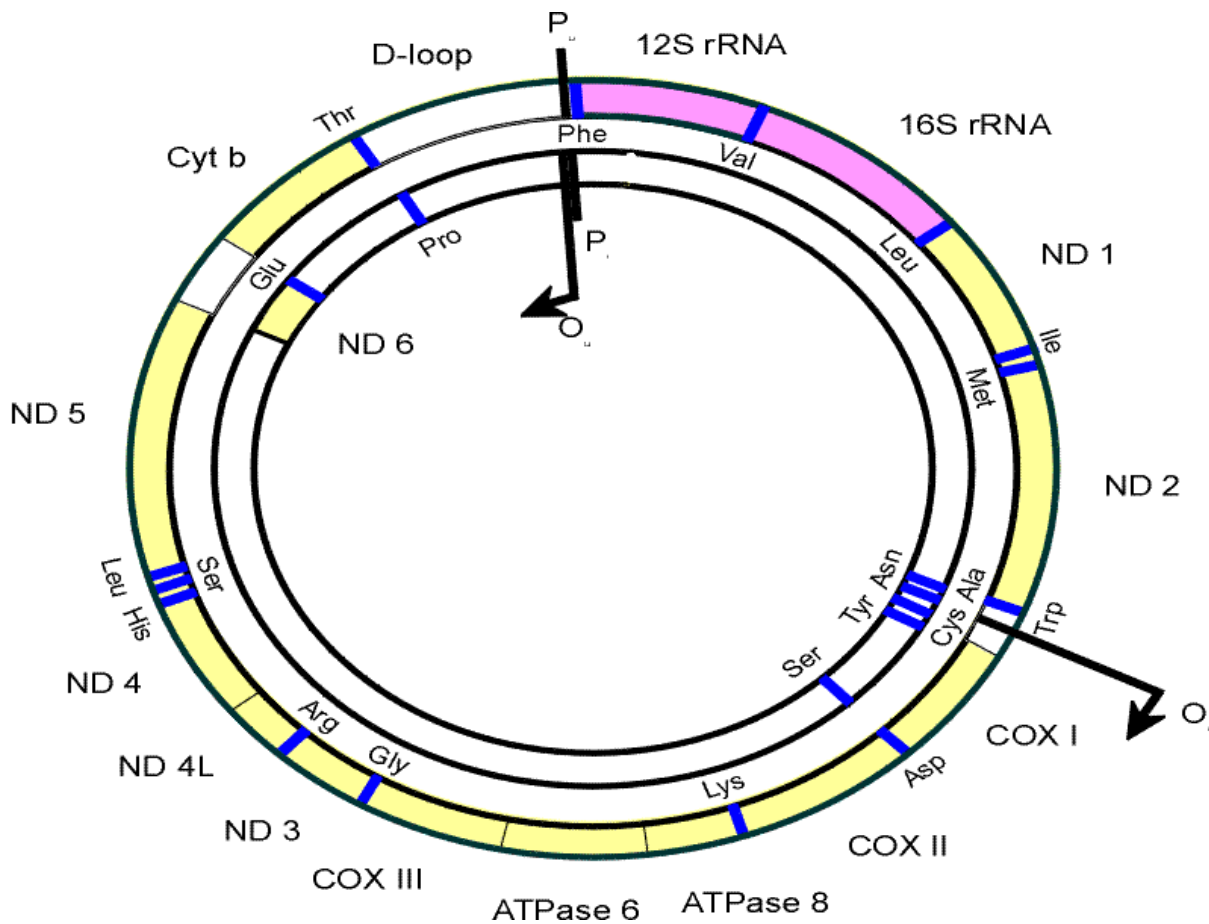


Figura 3.1: Genoma mitocondrial humano. O genoma mitocondrial possui duas fitas: uma fita H rica em guanina e a fita L rica em citosina. A fita H contém 12 dos 13 genes que codificam polipeptídios, 14 dos 22 genes que codificam RNA de transporte (tRNA) e ambos os genes que codificam RNA ribossômico (rRNA). Os genes ND1, ND2, COX1, COX2, ATP8, ATP6, COX3, ND3, ND4L, ND4, ND5 e CytB pertencem à fita H enquanto o gene ND6 aparece na fita L com orientação contrária aos genes anteriores (sequência complementar e reversa a sequência correspondente na fita H). Figura obtida da dissertação de Finnilä [24].

A Figura 3.1 mostra um diagrama representando o genoma mitocondrial humano. O genoma mitocondrial humano é uma molécula circular e fechada de DNA (mtDNA) composta por 16,569 pares de base (bp) [3], localizada dentro da matriz mitocondrial em milhares de cópias por célula [24, 34]. O DNA mitocondrial possui duas fitas: uma fita rica em guanina (chamada de fita H) e a outra rica em citosina (chamada de fita L). A fita H contém 12 dos 13 genes que codificam polipeptídios, 14 dos 22 genes que codificam RNA de transporte (tRNA) e ambos os genes que codificam RNA ribossômico (rRNA). As seqüências codificadoras de proteínas são contíguas no genoma [3]. Os genes ND1, ND2, COX1, COX2, ATP8, ATP6, COX3, ND3, ND4L, ND4, ND5 e CytB pertencem à fita H enquanto o gene ND6 aparece na fita L com orientação contrária aos genes anteriores (seqüência complementar e reversa à seqüência correspondente na fita H).

A cada gene de um genoma que aparece em uma das fitas, há uma seqüência de bases complementar e reversa ao gene. Nem todas essas seqüências complementares são genes, mas sua representação em um formalismo para rearranjo em genomas é relevante quando são envolvidas reversões com sinais.

O conjunto de seqüências de bases (genes ou não) pode ser modelado pelo sistema de genes (E, Γ) onde $E = E_+ \cup E_-$ tal que

$$E_+ = \{COX1, COX2, COX3, ATP6, ATP8, ND1, ND2, ND3, ND4L, ND4, ND5, ND6, CytB\}$$

e

$$E_- = \{-COX1, -COX2, -COX3, -ATP6, -ATP8, -ND1, -ND2, -ND3, -ND4L, -ND4, -ND5, -ND6, -CytB\}.$$

Usando a complementariedade das seqüências, define-se $\Gamma x = -x$ para $x \in E$. Os elementos de um sistema de genes podem ser não codificadores, ou seja, não serem de fato genes; por exemplo, a seqüência -ND4 não codifica nenhuma proteína. Para evitarmos que a nomenclatura seja pedante, iremos nos referir aos elementos de E simplesmente como “genes”, embora ressaltemos que as seqüências que são de fato genes dependam dos genomas modelados.

3.6.1 Genomas Circulares

Dado um sistema de genes (E, Γ) , uma permutação π sobre E é *admissível* quando a única permutação que divide a ambos π e Γ é a identidade. Os ciclos de uma permutação admissível são chamados de *fitas*. Uma permutação π sobre E é um *genoma* quando for admissível e $\pi^{-1} = \Gamma \cdot \pi$.

Exemplo 59. Considere o conjunto de genes $E = \{-5, -4, -3, -2, -1, 1, 2, 3, 4, 5\}$ e

$$\Gamma = (-5\ 5)(-4\ 4)(-3\ 3)(-2\ 2)(-1\ 1).$$

O sinais dos elementos indicam a orientação dos genes. Um exemplo de genoma para esse sistema de genes (E, Γ) é:

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

Os ciclos $(2\ -1\ 3)$, $(-3\ 1\ -2)$, $(5\ -4)$ e $(4\ -5)$ são fitas do genoma. Observe que em uma permutação π sobre E admissível com base no sistema de genes (E, Γ) temos $\Gamma x \notin \text{orb}(\pi, x)$ para $x \in E$. Em outros termos, uma fita não pode conter um gene e sua correspondente sequência complementar reversa. Outro aspecto a ser observado sobre os genomas é a possibilidade de escrever um genoma de diversas maneiras distintas pois é possível comutar as suas fitas uma vez que são ciclos disjuntos.

O genoma mitocondrial humano mostrado na Figura 3.1 pode ser modelado pela permutação

$$\begin{aligned} \theta = & (ND1\ ND2\ COX1\ COX2\ ATP8\ ATP6 \\ & COX3\ ND3\ ND4L\ ND4\ ND5\ -ND6\ CytB) \\ & (-CytB\ ND6\ -ND5\ -ND4\ -ND4L\ -ND3 \\ & -COX3\ -ATP6\ -ATP8\ -COX2\ -COX1\ -ND2\ -ND1). \end{aligned}$$

Dado um genoma π em um sistema de genes (E, Γ) , para cada fita α na decomposição em ciclos de π existe uma *fita complementar* $\Gamma \cdot \alpha^{-1}$. Um *cromossomo* de um genoma no sistema de genes (E, Γ) é o produto de duas fitas α e $\Gamma \cdot \alpha^{-1}$. Dois cromossomos são *disjuntos* quando os seus suportes são disjuntos. Um genoma é um produto de cromossomos disjuntos.

Exemplo 60. Considere o genoma

$$\pi = (-5\ -2\ 10)(-10\ 2\ 5)(1\ 7\ -9)(9\ -7\ -1)(3\ -4)(4\ -3)(-6)(6)(-8)(8).$$

no sistema de genes (E, Γ) para $E = \{x \in \mathbb{Z} \mid 1 \leq |x| \leq 10\}$ e Γ é a permutação que mapeia x em $-x$ para todo $x \in E$.

As permutações $(-5\ -2\ 10)(-10\ 2\ 5)$, $(1\ 7\ -9)(9\ -7\ -1)$, $(3\ -4)(4\ -3)$, $(-6)(6)$ e $(-8)(8)$ são cromossomos de π . Cada cromossomo é o produto de fitas complementares entre si.

Um sistema de genes (E, Γ) define o conjunto de genomas sobre o mesmo E e cujas propriedades fundamentais (admissibilidade e inversão por meio da conjugação por Γ) dependem da mesma permutação Γ . Iremos escrever que π é um genoma em (E, Γ) quando quisermos dizer que π pertence ao conjunto de genomas definido pelo sistema de genes (E, Γ) . Quando necessário, deixaremos claro no texto quando quisermos distinguir o par (E, Γ) do conjunto de genomas definido por esse par. Discutiremos na Seção 3.7 sobre como os diversos problemas de rearranjo em genomas são definidos para subconjuntos do conjunto de genomas definido por um sistema de genes.

3.6.2 Genomas Lineares

Nessa seção discutimos como representar genomas compostos por cromossomos lineares por meio do formalismo algébrico. O formalismo algébrico modela de maneira direta genomas compostos por fitas circulares. Por outro lado, na natureza é possível encontrar genomas compostos também por cromossomos lineares. É necessário realizar um mapeamento de um genoma contendo cromossomos lineares em um genoma contendo apenas cromossomos circulares para ser manipulado por meio da aplicação de eventos de rearranjo. Ao final da manipulação do genoma em seu formato com cromossomos circulares, aplica-se um mapeamento inverso para obter um genoma com cromossomos lineares novamente.

Para modelar um genoma que possua cromossomos lineares, temos de incluir no conjunto de genes um par distinto de pseudo-genes complementares $+p$ e $-p$ para cada cromossomo do genoma. Os pseudo-genes $+p$ e $-p$ são usados para unir as extremidades das fitas complementares do cromossomos lineares. Para obter-se novamente um genoma com cromossomos lineares basta remover os pseudo-genes do conjunto de genes e definir as extremidades das fitas como os genes adjacentes a esses pseudo-genes [40, 42, 58].

Exemplo 61. Considere um genoma contendo apenas um cromossomo linear:

$$[a, b, c, d],$$

tal que as suas extremidades sejam a e d .

Podemos modelar o genoma anterior no formalismo algébrico por meio da inclusão de dois genes, digamos $+0$ e -0 . O genoma no formalismo algébrico tem a seguinte forma:

$$(+0 \ a \ b \ c \ d)(\Gamma d \ \Gamma c \ \Gamma b \ \Gamma a \ -0).$$

3.7 Eventos de Rerranjo em Genomas

Nessa seção, modelaremos os eventos de rearranjo em genomas como permutações. Apresentamos alguns eventos de rearranjo que não haviam sido modelados anteriormente pelo formalismo algébrico, como as fusões, fissões e transposições generalizadas. Além disso, uma contribuição original dessa seção em relação ao trabalho de Meidanis e Dias [39] é a discussão sobre a compatibilidade entre genomas com respeito a eventos de rearranjo.

Dado um genoma π em um sistema de genes (E, Γ) , uma permutação ρ é um *evento de rearranjo aplicável* a π quando $\rho\pi$ é um genoma em (E, Γ) . Dizemos que *aplicamos* ρ em π quando obtemos o produto $\rho\pi$.

A ordem em que aplicamos o evento de rearranjo ρ é relevante para determinar o formato de permutação que esse evento irá assumir. Meidanis e Dias [39] adotam a convenção de aplicar um evento de rearranjo “pela direita”, ou seja, o resultado da aplicação de um evento ρ em

um genoma π é $\pi\rho$. Conseqüentemente, se $\pi\rho = \sigma$ então $\rho = \pi^{-1}\sigma$ ou de modo equivalente $\rho = \Gamma\pi\Gamma\sigma$.

Em nossa versão do formalismo algébrico, adotamos a convenção alternativa de aplicar um evento de rearranjo pela esquerda. Segundo essa convenção, se $\rho\pi = \sigma$ então $\rho = \sigma\pi^{-1}$ ou em outros termos $\rho = \sigma\Gamma\pi\Gamma$.

Ambas as convenções podem ser utilizadas de maneira independente (não se pode misturar as convenções) e permitem obter os mesmos resultados mostrados nesse trabalho. Adotamos a convenção de aplicar um evento de rearranjo pela esquerda devido ser mais natural a utilização da notação pré-fixa para operadores.

3.7.1 Reversões com Sinais

Um evento de rearranjo em genomas que parece ser dos mais relevantes do ponto de vista biológico é a reversão com troca de orientações de genes [28, 29]. Dado o genoma π em um sistema de genes (E, Γ) , uma *reversão com sinais* $\rho = \rho(u, v, \pi)$, onde u e v são genes distintos em E , é um evento de rearranjo aplicável a π que transforma o genoma π no genoma $\rho\pi$ cuja seqüência de genes entre u e v , incluindo u , mas excluindo v em um cromossomo θ de π é trocada pela seqüência inversa e complementar (em ambas as fitas de θ em π). A permutação que define a reversão com sinais $\rho(u, v, \pi)$ aplicável ao genoma π é $(u \pi \Gamma v)(v \pi \Gamma u)$ tal que u e v pertençam a mesma fita de π . Por exemplo, dado o genoma $(a \Gamma b c)(\Gamma c b \Gamma a)(d \Gamma e)(e \Gamma d)$ no sistema de genes (E, Γ) tal que $E = \{a, b, c, d, e, \Gamma a, \Gamma b, \Gamma c, \Gamma d, \Gamma e\}$, a permutação $(\Gamma b b)(c \Gamma a)$ é uma reversão com sinais aplicável a π .

Dado um genoma π em (E, Γ) e uma reversão com sinais $\rho(u, v, \pi)$ aplicável a π , seguem-se algumas propriedades fundamentais de reversões com sinais:

1. $\rho = \rho^{-1}$;
2. $(u v)|\pi$ e $(u v) \nmid \rho\pi$;
3. $\|\rho\pi\| = \|\pi\rho\|$ e $\|\rho\pi\| = \|\pi\|$.

Meidanis e Dias [39] demonstraram que a mesma reversão com sinal pode ser definida por diferentes pares de elementos do conjunto de genes.

Proposição 62. Dado o genoma π no sistema de genes (E, Γ) e seja $\rho(u, v, \pi)$ uma reversão com sinais aplicável a π , temos:

$$\rho(u, v, \pi) = \rho(v, u, \pi) = \rho(\pi\Gamma u, \pi\Gamma v, \pi) = \rho(\pi\Gamma v, \pi\Gamma u, \pi)$$

O efeito de uma reversão com sinais ρ aplicável a um genoma π é ilustrado na Figura 3.2.

Dois genomas π e σ são *compatíveis com relação a reversões com sinais* quando existe uma seqüência de reversões com sinais ρ_1, \dots, ρ_k tal que ρ_i é uma reversão com sinal aplicável a

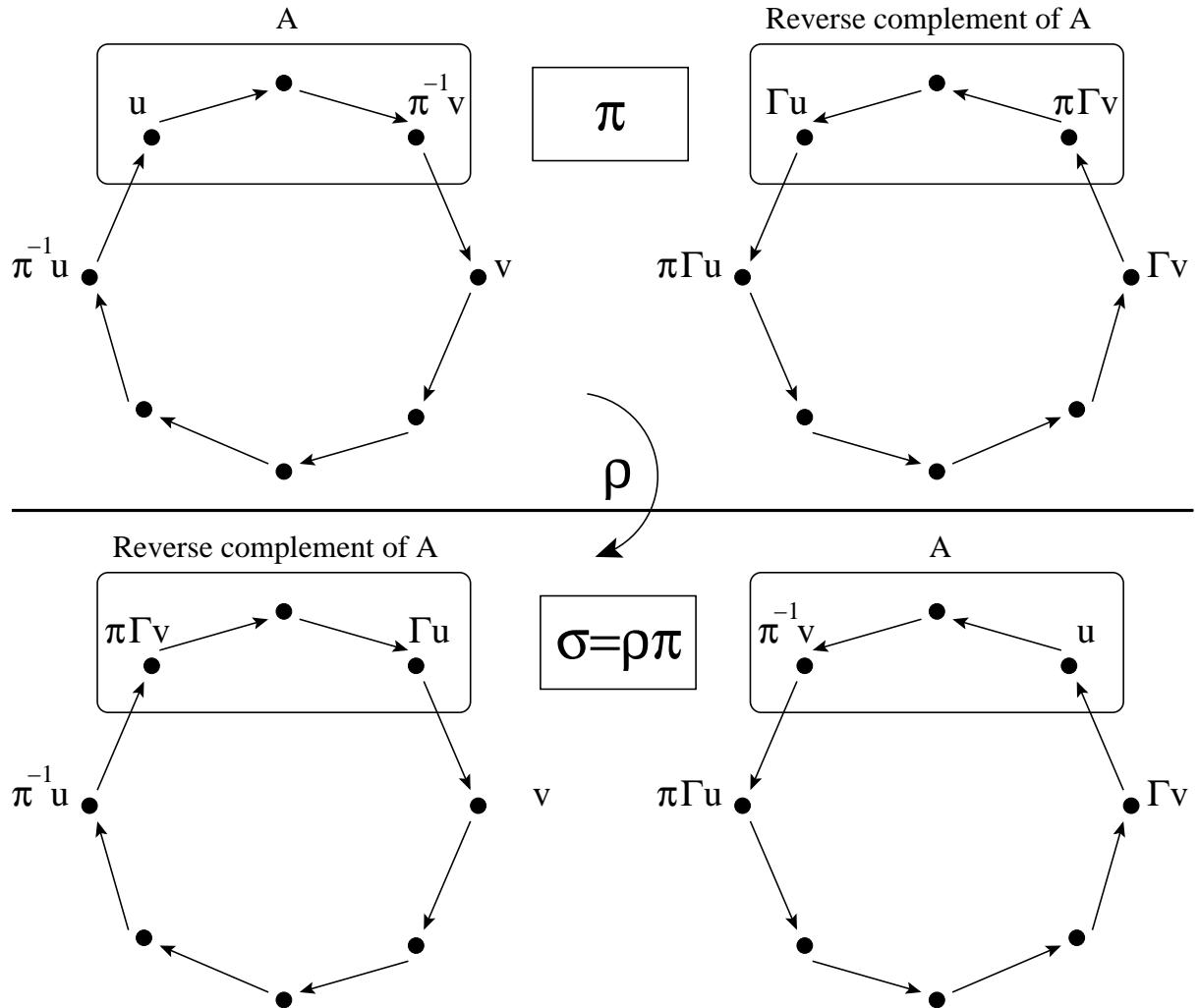


Figura 3.2: O diagrama mostra o resultado da aplicação de uma reversão com sinais ρ aplicável a um genoma unicromossomal π . Observe as modificações nos segmentos de genes $u, \dots, \pi^{-1}v$ e $\pi\Gamma v, \dots, \Gamma u$. Em uma reversão com sinais uma sequência de genes é invertida e a orientação dos genes trocada em ambas as fitas do mesmo cromossomo do genoma (Figura reproduzida do artigo de Meidanis e Dias [39]).

$\rho_{i-1} \dots \rho_1 \pi$ para $1 \leq i \leq k$ e $\rho_i \dots \rho_1 \pi = \sigma$. Nem sempre um par de genomas é compatível com relação a reversões com sinais. Por exemplo, genomas com o número distinto de cromossomos não são compatíveis.

Dois genomas π e σ em (E, Γ) são *equicromossomais* se para cada cromossomo de π existe um cromossomo em σ cuja a união das órbitas em cada cromossomo é a mesma. Os cromossomos de π e σ cuja união de suas duas órbitas são iguais são chamados de *cromossomos correspondentes*. Por exemplo, os genomas

$$(a \ b \ c)(\Gamma c \ \Gamma b \ \Gamma a)(c \ d)(\Gamma d \ \Gamma c)$$

e

$$(b \ a \ c)(\Gamma c \ \Gamma a \ \Gamma b)(c \ \Gamma d)(d \ \Gamma c)$$

são equicromossomais. Os cromossomos $(c \ d)(\Gamma d \ \Gamma c)$ e $(c \ \Gamma d)(d \ \Gamma c)$ são cromossomos correspondentes.

Lema 63. Dado um genoma π em (E, Γ) e uma reversão com sinais ρ aplicável a π então os genomas π e $\rho\pi$ são equicromossomais.

Prova: Uma reversão $\rho = \rho(u, v, \pi)$ é caracterizada por trocar os segmentos $u \dots \pi^{-1}v$ e $\pi\Gamma v \dots \Gamma u$ entre as fitas de um cromossomo de π . Logo, o cromossomo em $\rho\pi$ que resultou das alterações nas fitas causadas pela reversão mantém o mesmo conjunto de elementos, embora esses elementos possam pertencer a órbitas diferentes do cromossomo correspondente em π . Portanto os genomas π e $\rho\pi$ são equicromossomais. \square

Uma reversão aplicável a um genoma π não modifica os suportes dos cromossomos em $\rho\pi$, no entanto as órbitas de $\rho\pi$ que contém u e v são alteradas. Uma consequência direta do Lema 63 é que se existe uma sequência de reversões com sinais que transforme um genoma π em um genoma distinto σ então π e σ são equicromossomais. Em outros termos, genomas compatíveis com relação a reversões com sinais são equicromossomais. Por outro lado, se dois genomas distintos são equicromossomais então esses genomas são compatíveis com relação a reversões com sinais. Não demonstramos essa última implicação pois não tratamos nesse trabalho com transformações entre genomas envolvendo exclusivamente reversões com sinais. Genomas equicromossomais serão estudados no contexto do problema de rearranjo envolvendo reversões com sinais e intercâmbio de blocos.

Exemplo 64. Considere os genomas:

- $\pi = (a \ b \ c)(\Gamma c \ \Gamma b \ \Gamma a)(e \ d)(\Gamma d \ \Gamma e),$
- $\sigma = (a \ \Gamma c \ b)(\Gamma b \ c \ \Gamma a)(e \ \Gamma d)(d \ \Gamma e) \text{ e}$
- $\theta = (a \ b)(\Gamma b \ \Gamma a)(c \ e \ d)(\Gamma d \ \Gamma e \ \Gamma c).$

todos no sistema de genes (E, Γ) para o qual $E = \{a, b, c, d, e, \Gamma a, \Gamma b, \Gamma c, \Gamma d, \Gamma e\}$. Os genomas π e σ são compatíveis, mas os pares π, θ e σ, θ são incompatíveis.

Como π e σ são genomas distintos e eqüicromossomais, existe uma seqüência de reversões com sinais que transforma π em σ . A seqüência $\rho_1 = \rho(b, a, \pi)$, $\rho_2 = \rho(\Gamma b, a, \rho_1 \pi)$ e $\rho_3 = \rho(d, e, \rho_2 \rho_1 \pi)$ transforma π em σ , isto é, temos $\rho_3 \rho_2 \rho_1 \pi = \sigma$. Por outro lado, como os pares (π, θ) e (σ, θ) não são eqüicromossomais não há seqüências de reversões com sinais que transformem π ou σ em θ .

Dados os genomas compatíveis π e σ em um sistema de genes (E, Γ) , o *problema de rearranjo em genomas por reversões com sinais* consiste em encontrar uma seqüência com o número mínimo de reversões com sinais que transformem um genoma em outro. Esse número mínimo de reversões com sinais é chamado de *distância de reversões com sinais*, denotado por $d_r(\pi, \sigma)$, entre os genomas π e σ . Em outros termos, queremos encontrar uma seqüência de operações de reversão com sinais $\rho_1, \rho_2, \dots, \rho_k$, tal que:

$$\sigma = \rho_k \rho_{k-1} \dots \rho_1 \pi$$

onde cada ρ_i é uma reversão com sinais, para $1 \leq i \leq k$, e ρ_{i+1} é aplicável a $\rho_i \rho_{i-1} \dots \rho_1 \pi$, e k é mínimo.

3.7.2 Transposições

Dado um genoma π no sistema de genes (E, Γ) , uma *transposição* aplicável a π é a permutação $\tau = (u \ v \ w)(\pi \Gamma w \ \pi \Gamma v \ \pi \Gamma u)$, onde u, v e w são elementos distintos em E , tal que $\tau|_{\pi}$. Uma transposição $\tau = \tau(u, v, w, \pi)$ troca os segmentos adjacentes $u \dots \pi^{-1}v$ e $v \dots \pi^{-1}w$ em uma fita (e os segmentos $\pi \Gamma w \dots \Gamma v$ e $\pi \Gamma v \dots \Gamma u$ na fita complementar) de um cromossomo de π . Dado o genoma $\pi = (a \ b \ c)(\Gamma c \ \Gamma b \ \Gamma a)$ no sistema de genes (E, Γ) para o qual $E = \{a, b, c, \Gamma a, \Gamma b, \Gamma c\}$, a permutação $(a \ b \ c)(\Gamma b \ \Gamma a \ \Gamma c)$ é uma transposição aplicável a π .

Observe que definimos uma transposição utilizando uma forma distinta à utilizada por Meidanis e Dias [39]. Meidanis e Dias [39] definem uma transposição τ aplicável a um genoma π (em um sistema de genes (E, Γ)) como a permutação $(\pi u \ \pi v \ \pi w)(\Gamma w \ \Gamma v \ \Gamma u)$ tal que u, v, w apareçam nessa ordem circular em π . As duas definições são equivalentes, ou seja representam a mesma transposição, embora o formato das permutações que modelam uma transposição sejam distintos.

Dois genomas π e σ em (E, Γ) são *eqüiorbitais* quando $Orb(\pi, E) = Orb(\sigma, E)$.

Lema 65. Dado um genoma π em (E, Γ) e uma transposição τ aplicável a π então os genomas π e $\tau\pi$ são eqüiorbitais.

Prova: Uma transposição $\tau = \tau(u, v, w, \pi)$ é caracterizada por trocar os segmentos adjacentes $u \dots \pi^{-1}v$ e $v \dots \pi^{-1}w$ em uma fita (e os segmentos $\pi \Gamma w \dots \Gamma v$ e $\pi \Gamma v \dots \Gamma u$ na fita complementar) de um cromossomo de π . Logo, o cromossomo em $\tau\pi$ que resultou das alterações

nas fitas causadas pela transposição mantêm as mesmas órbitas ainda que as fitas tenham sido alteradas. Portanto os genomas π e $\tau\pi$ são equi-orbitais. \square

Uma transposição τ aplicável a um genoma π não modifica as órbitas do genoma $\tau\pi$, no entanto, a transposição τ altera a ordem dos elementos nas fitas de $\tau\pi$, ou seja, a relação de acessibilidade dos genes é modificada. Do Lema 65 resulta que se existe uma seqüência de transposições τ_1, \dots, τ_k para a qual $\tau_k \dots \tau_1 \pi = \sigma$, tal que τ_i é uma transposição aplicável a $\tau_{i-1} \dots \tau_1 \pi$ para $1 \leq i \leq k$ então π e σ são genomas equi-orbitais. Além disso, dados dois genomas distintos e equi-orbitais em (E, Γ) então existe uma seqüência de transposições que transforma um genoma em outro. Um exemplo de tal seqüência de transposições que transforma um genoma de origem no genoma destino é a formada por transposições que “movem” cada elemento $x \in E$ e $\Gamma x \in E$ para a sua posição adequada no genoma destino ao mover o segmento contendo x e o segmento consecutivo formado por elementos ainda não posicionados adequadamente. Por exemplo, dados os genomas equi-orbitais $\pi = (a \ b \ c)(\Gamma c \ \Gamma b \ \Gamma a)(e \ d)(\Gamma d \ \Gamma e)$ e $\sigma = (a \ c \ b)(\Gamma b \ \Gamma c \ \Gamma a)(e \ d)(\Gamma d \ \Gamma e)$, a transposição τ que troca os segmentos b e c de π é tal que $\tau\pi = \sigma$.

Exemplo 66. Considere os genomas:

- $\pi = (a \ b \ c)(\Gamma c \ \Gamma b \ \Gamma a)(e \ d)(\Gamma d \ \Gamma e)$,
- $\sigma = (a \ c \ b)(\Gamma b \ \Gamma c \ \Gamma a)(e \ d)(\Gamma d \ \Gamma e)$ e
- $\theta = (a \ b)(\Gamma b \ \Gamma a)(c \ e \ d)(\Gamma d \ \Gamma e \ \Gamma c)$.

todos no sistema de genes (E, Γ) para o qual $E = \{a, b, c, d, e, \Gamma a, \Gamma b, \Gamma c, \Gamma d, \Gamma e\}$. Os genomas π e σ são equi-orbitais, enquanto os demais pares de genomas nos quais θ participa não são equi-orbitais.

Como π e σ são genomas equi-orbitais, existe uma seqüência de transposições que transforma π em σ . A seqüência $\tau = \tau(b, c, a, \pi)$ transforma π em σ , isto é, temos $\tau\pi = \sigma$. Por outro lado, não há seqüências de transposições que transformem π ou σ em θ .

Dados os genomas equi-orbitais π e σ em (E, Γ) , o *problema de rearranjo em genomas por transposições* consiste em encontrar uma seqüência mínima de transposições que transforme o genoma π em σ . Em outras palavras, queremos encontrar uma seqüência de transposições τ_1, \dots, τ_k tal que

$$\tau_k \dots \tau_1 \pi = \sigma$$

e τ_i é aplicável a $\tau_{i-1} \dots \tau_1 \pi$, para $1 \leq i \leq k$ e k é mínimo. O número k de transposições na seqüência mínima é chamado de *distância de transposição* entre π e σ e esse número é denotado por $d_t(\pi, \sigma)$.

Como a partir de um genoma π é possível apenas obter genomas que sejam equi-orbitais a π por meio de seqüências de transposições, o problema de rearranjo em genomas por transposições é limitado a pares de genomas equi-orbitais em um sistema de genes.

Exemplo 67. Dados os genomas

$$\pi = (a\ d\ c\ b\ e)(\Gamma e\ \Gamma b\ \Gamma c\ \Gamma d\ \Gamma a)$$

e

$$\sigma = (a\ b\ c\ d\ e)(\Gamma e\ \Gamma d\ \Gamma c\ \Gamma b\ \Gamma a)$$

em (E, Γ) para $E = \{\Gamma e, \Gamma d, \Gamma c, \Gamma b, \Gamma a, a, b, c, d, e\}$, apresentamos a seguinte seqüência de transposições que transforma π em σ .

$$(d\ b\ e)(\Gamma b\ \Gamma c\ \Gamma a)(a\ d\ c\ b\ e)(\Gamma e\ \Gamma b\ \Gamma c\ \Gamma d\ \Gamma a) = (a\ b\ d\ c\ e)(\Gamma e\ \Gamma c\ \Gamma d\ \Gamma b\ \Gamma a)$$

$$(d\ c\ e)(\Gamma c\ \Gamma d\ \Gamma b)(a\ b\ d\ c\ e)(\Gamma e\ \Gamma c\ \Gamma d\ \Gamma b\ \Gamma a) = (a\ b\ c\ d\ e)(\Gamma e\ \Gamma d\ \Gamma c\ \Gamma b\ \Gamma a)$$

Logo a distância de transposição $d_t(\pi, \sigma)$ entre π e σ é menor ou igual a 2.

3.7.3 Transposições Generalizadas

O problema de rearranjo em genomas por transposições tem se mostrado de difícil análise devido à estrutura complexa inerente à sua modelagem [7, 22]. Visando uma compreensão melhor dos efeitos de uma transposição, apresentamos em um pôster [43] a definição do problema de rearranjo em genomas que envolve os eventos de rearranjo cujo formato é um único 3-ciclo e os genomas de origem e destino na entrada do problema são unicromossomais (cromossomo circular) e formados por uma única fita. Para esse problema de rearranjo, um genoma é uma permutação sobre um conjunto de genes E e qualquer 3-ciclo sobre E é um evento de rearranjo aplicável a esse genoma. A aplicação de um evento a um genoma neste contexto obedece ao comportamento de um produto de uma permutação e um 3-ciclo como indicado pelo Teorema 48, que sumarizamos a seguir. Dado um genoma π sobre E , o 3-ciclo que divide π possui o comportamento de uma transposição, ou seja troca dois segmentos consecutivos do genoma π . Se um 3-ciclo $(x\ y\ z)$ não divide o genoma π então três situações distintas podem ocorrer: três cromossomos do genoma π são “fundidos” em apenas um cromossomo quando x, y e z pertencem a cromossomos distintos de π ; um cromossomo é “dividido” em três cromossomos quando z, y, x aparecem nessa ordem circular em um cromossomo de π ; ou um segmento de um cromossomo é movido pra outro cromossomo distinto quando dois dos elementos em $\{x, y, z\}$ pertencem a um desses cromossomos e o terceiro elemento pertence ao outro cromossomo. Observe que nesse problema [43] os genomas “intermediários” obtidos pela aplicação de uma seqüência de 3-ciclos podem ser multicromossomais (permutações contendo mais de uma órbita), embora os genomas na entrada do problema sejam unicromossomais. Essa exigência de que os genomas na entrada sejam unicromossomais caracteriza o problema como uma versão relaxada do problema de rearranjo em genomas por transposições, no qual os tipos de rearranjo representados por 3-ciclos e não apenas as transposições são permitidos.

Baseando-se neste problema de rearranjo envolvendo 3-ciclos, desenvolvemos um problema correspondente para o caso geral de genomas (multicromossomais e contendo fitas complementares reversas) em um sistema de genes. Dados os genomas π e σ no sistema de genes (E, Γ) tais que σ possui ao menos uma fita cujo tamanho do suporte é pelo menos três, um evento de rearranjo $(a \ b \ c)(\pi\Gamma c \ \pi\Gamma b \ \pi\Gamma a)$ é uma *transposição generalizada* aplicável a π quando $b \in orb(\sigma, a)$ e $c \in orb(\sigma, a)$ para $a, b, c \in E$. Como no problema envolvendo 3-ciclos, um evento de rearranjo $(a \ b \ c)(\pi\Gamma c \ \pi\Gamma b \ \pi\Gamma a)$ que divide o genoma π possui o comportamento de uma transposição, ou seja, a troca de segmentos consecutivos do genoma, mas no caso de transposições generalizadas, as duas fitas de um cromossomo de π sofrem a troca de segmentos consecutivos devido ao produto envolvendo os 3-ciclos $(a \ b \ c)$ e $(\pi\Gamma c \ \pi\Gamma b \ \pi\Gamma a)$. A condição de que os elementos a, b e c devem pertencer a mesma órbita de σ é importante para o caso em que $(a \ b \ c)(\pi\Gamma c \ \pi\Gamma b \ \pi\Gamma a)$ não divide o genoma π . Essa condição garante que, além das transposições, apenas os 3-ciclos $(a \ b \ c)(\pi\Gamma c \ \pi\Gamma b \ \pi\Gamma a)$ tais que c, b, a apareçam nessa ordem em um ciclo de π , os 3-ciclos tais que a, b e c apareçam em cromossomos distintos, ou os 3-ciclos $(a \ b \ c)(\pi\Gamma c \ \pi\Gamma b \ \pi\Gamma a)$ tais que dois dos elementos em $\{a, b, c\}$ pertencem a cromossomo e o terceiro elemento pertence a um cromossomo distinto sejam transposições generalizadas. Os 3-ciclos $(a \ b \ c)(\pi\Gamma c \ \pi\Gamma b \ \pi\Gamma a)$ tais que a e b pertençam a uma fita enquanto c pertence à fita complementar reversa do mesmo cromossomo de π não são transposições generalizadas pois não atende a condição de a, b e c pertencer a mesma órbita de σ .

Duas considerações devem ser destacadas a respeito das transposições generalizadas. Primeiramente, nem sempre há uma transposição generalizada aplicável a π para qualquer par de genomas π e σ em um sistema de genes. Um exemplo de par de genomas π e σ para o qual não há transposições generalizadas aplicáveis a π é $\pi = (x \ y \ z)(\Gamma z \ \Gamma y \ \Gamma z)$ e $\sigma = (x)(y)(z)(\Gamma z)(\Gamma y)(\Gamma z)$. Em segundo lugar, uma transposição generalizada depende de ambos os genomas π e σ em sua definição ao invés de depender apenas do genoma π como no caso de reversões com sinais ou transposições.

Os genomas π e σ em (E, Γ) são *compatíveis com relação a transposições generalizadas* quando existe uma seqüência de transposições generalizadas τ_1, \dots, τ_k tal que $\tau_k \dots \tau_1 \pi = \sigma$ e τ_i é aplicável a $\tau_{i-1} \dots \tau_1 \pi$ para $1 \leq i \leq k$.

Exemplo 68. Dado os genomas

$$\pi = (a \ d \ c \ e \ b)(\Gamma b \ \Gamma e \ \Gamma c \ \Gamma d \ \Gamma a)(f \ h \ g)(\Gamma g \ \Gamma h \ \Gamma f);$$

$$\theta = (a \ b)(\Gamma b \ \Gamma a)(c)(\Gamma c)(d \ e)(\Gamma e \ \Gamma d)(f \ g \ h)(\Gamma h \ \Gamma g \ \Gamma f)$$

e

$$\sigma = (a \ b \ c \ d \ e)(\Gamma e \ \Gamma d \ \Gamma c \ \Gamma b \ \Gamma a)(f \ g \ h)(\Gamma h \ \Gamma g \ \Gamma f)$$

em (E, Γ) onde $E = \{a, b, c, d, e, f, g, h, \Gamma a, \Gamma b, \Gamma c, \Gamma d, \Gamma e, \Gamma f, \Gamma g, \Gamma h\}$.

Tomando o par de genomas π e σ , as permutações $\alpha = (a \ b \ c)(\Gamma d \ \Gamma e \ \Gamma b)$ e $\beta = (f \ h \ g)(\Gamma g \ \Gamma h \ \Gamma f)$ são transposições generalizadas aplicáveis a π pois a, b e c no primeiro caso, e f, h e g no segundo, pertencem a mesma órbita de σ . Como $(a \ b \ c)(\Gamma d \ \Gamma e \ \Gamma b) \nmid \pi$ e c, b, a aparecem nessa ordem circular em π então o cromossomo cujo suporte contém esses elementos é “quebrado” em três cromossomos:

$$\alpha\pi = (a \ d)(\Gamma d \ \Gamma a)(c \ e)(\Gamma e \ \Gamma c)(b)(\Gamma b)(f \ h \ g)(\Gamma g \ \Gamma h \ \Gamma f).$$

Por outro lado, a permutação $\beta = (f \ h \ g)(\Gamma g \ \Gamma h \ \Gamma f)$ divide π e por isso o efeito do produto $\beta\pi$ é a troca dos genes consecutivos h e g , o que equivale a uma transposição:

$$\beta\pi = (a \ d \ c \ e \ b)(\Gamma b \ \Gamma e \ \Gamma c \ \Gamma d \ \Gamma a)(f \ g \ h)(\Gamma h \ \Gamma g \ \Gamma f).$$

Considere agora o par de genomas θ e σ . A permutação $\tau = (a \ c \ d)(\Gamma e \ \Gamma c \ \Gamma b)$ é uma transposição generalizada aplicável em θ pois a, c e d pertencem a mesma fita de σ . Como a, c e d pertencem ao suporte de cromossomos distintos em θ então esses cromossomos são “reunidos” em um único cromossomo no produto $\tau\theta$:

$$\tau\theta = (a \ b \ c \ d \ e)(\Gamma e \ \Gamma d \ \Gamma c \ \Gamma b \ \Gamma a)(f \ g \ h)(\Gamma h \ \Gamma g \ \Gamma f).$$

A seqüência $(e \ d \ b)(\Gamma e \ \Gamma a \ \Gamma c)$, $(a \ c \ e)(\Gamma e \ \Gamma d \ \Gamma b)$, $(f \ h \ g)(\Gamma h \ \Gamma f \ \Gamma g)$ de transposições generalizadas transforma π em σ , logo π e σ são compatíveis com relação a transposições generalizadas. Por outro lado, o evento $\tau = (a \ c \ d)(\Gamma e \ \Gamma c \ \Gamma b)$ transforma θ em σ .

Observe que o efeito de transposições generalizadas baseia-se no efeito do produto de 3-ciclos (veja o Exemplo 47).

Os genomas π e σ em (E, Γ) são chamados *conorbitais* quando cada órbita de π é um subconjunto de uma órbita de σ .

Lema 69. Dados os genomas π e σ conorbitais em (E, Γ) , se uma transposição generalizada $\tau = (a \ b \ c)(\pi\Gamma c \ \pi\Gamma b \ \pi\Gamma a)$ aplicável a π é tal que ou a, b e c pertencem a uma mesma fita de π ou a três fitas de cromossomos distintos então os genomas $\tau\pi$ e σ são conorbitais.

Prova: Dada a transposição generalizada $\tau = (a \ b \ c)(\pi\Gamma c \ \pi\Gamma b \ \pi\Gamma a)$ para $a, b, c \in E$, há três casos para serem analisados:

- Uma transposição generalizada troca dois segmentos consecutivos de genes em uma mesma fita e outros dois segmentos consecutivos na fita complementar de um cromossomo de π quando $(a \ b \ c) \mid \pi$. Nesse caso as órbitas de $\tau\pi$ são as mesmas de π segundo o Teorema 48, embora a relação de acessibilidade dos elementos tenha sido alterada. Logo cada órbita de $\tau\pi$ é um subconjunto de uma órbita de σ .

- Se a transposição generalizada é tal que a, b e c pertençam a mesma fita de um cromossomo de π mas $(a \ b \ c) \nmid \pi$ então o cromossomo que contém a, b e c é dividido em três cromossomos sem haver troca de elementos entre as suas fitas segundo o Teorema 48. Como a, b e c pertencem a mesma órbita de σ por definição então as novas fitas de $\tau\pi$ cada uma contendo um dos elementos de $\{a, b, c\}$ estão contidas na órbita que contém a, b e c em σ (resultado equivalente vale para os elementos $\pi\gamma a, \pi\Gamma b$ e $\pi\Gamma c$). Logo as órbitas de $\tau\pi$ são contidas nas órbitas de σ .
- Se os elementos da mesma órbita de τ pertencem a fitas de três cromossomos distintos de π então esses três cromossomos são “reunidos” em um único cromossomo cuja uma de suas fitas contém os elementos a, b e c de acordo com o Teorema 48. Como π e σ são conorbitais então cada uma das fitas contendo a, b e c estão contidas em uma fita de σ que contém esses três elementos. A única fita em $\tau\pi$ que resulta da reunião das três fitas de π é contida na fita de σ contendo a, b e c pois nenhum outro elemento é adicionado a qualquer das três fitas de π . Logo as órbitas de $\tau\pi$ são contidas nas órbitas de σ .

Em todos os casos o genoma $\tau\pi$ que resultou das alterações nas fitas e cromossomos causadas pela transposição generalizada mantém a propriedade de que cada órbita sua esteja contida em uma órbita de σ . Portanto os genomas π e $\tau\pi$ são conorbitais. \square

Segundo o Lema 69, se existe uma sequência de transposições generalizadas τ_1, \dots, τ_k para a qual $\tau_k \dots \tau_1 \pi = \sigma$, tal que τ_i é uma transposição generalizada aplicável a $\tau_{i-1} \dots \tau_1 \pi$ para $1 \leq i \leq k$ então π e σ são conorbitais. Por ora, assumiremos que dados dois genomas distintos conorbitais existe uma sequência de transposições generalizadas que transforma um no outro.

Dados os genomas conorbitais π e σ em (E, Γ) , o *problema de rearranjo em genomas por transposições generalizadas* consiste em encontrar uma sequência mínima de transposições generalizadas que transforme o genoma π em σ . Em outras palavras, queremos encontrar uma sequência de transposições generalizadas τ_1, \dots, τ_k tal que

$$\tau_k \dots \tau_1 \pi = \sigma$$

e τ_i é aplicável a $\tau_{i-1} \dots \tau_1 \pi$, para $1 \leq i \leq k$ e k é mínimo. O número k de transposições generalizadas na sequência mínima é chamado de *distância de transposições generalizadas* entre π e σ e esse número é denotado por $d_{tg}(\pi, \sigma)$.

Quando discutirmos o problema de rearranjo em genomas por transposições generalizadas no Capítulo 5, iremos mostrar uma sequência particular de transposições generalizadas que é solução para o problema e depende da hipótese de que os genomas são conorbitais.

3.7.4 Intercâmbio de Blocos

Dado o genoma π no sistema de genes (E, Γ) um evento de rearranjo

$$\rho = (u\ x)(\pi\Gamma x\ \pi\Gamma u)(v\ y)(\pi\Gamma y\ \pi\Gamma v)$$

para $u \neq y$, $u \neq v$, $x \neq y$ e $u, v, x, y \in E$ é um *intercâmbio de blocos* aplicável a π quando $(u\ v\ x\ y)|\pi$, ou seja quando u, v, x e y pertencem a uma mesma órbita de π e aparecem nessa ordem circular em uma fita. O evento de intercâmbio de blocos $\rho = \rho(u, v, x, y, \pi)$ troca os segmentos $u \dots \pi^{-1}v$ e $x \dots \pi^{-1}y$ em uma fita (e os segmentos $\pi\Gamma v \dots \Gamma u$ e $\pi\Gamma y \dots \Gamma x$ na fita complementar) de um cromossomo de π .

Lema 70. Dado um genoma π em (E, Γ) e um intercâmbio de blocos ρ aplicável a π então os genomas π e $\rho\pi$ são eqüiorbitais.

Prova: Uma operação de intercâmbio de blocos $\rho = \rho(u, v, x, y, \pi)$ é caracterizada por trocar dois segmentos de genes em uma mesma fita e outros dois segmentos na fita complementar de um cromossomo de π . Logo, o cromossomo em $\tau\pi$ que resultou das alterações nas fitas causadas pelo intercâmbio de blocos mantém as mesmas órbitas ainda que as fitas tenham sido alteradas. Portanto os genomas π e $\rho\pi$ são eqüiorbitais. \square

Como no caso da transposição, um evento de intercâmbio de blocos aplicável a um genoma π não modifica as órbitas do genoma $\rho\pi$, embora a ordem dos elementos nas fitas de $\rho\pi$ seja modificada. Segundo o Lema 70, se houver uma seqüência de intercâmbios de blocos ρ_1, \dots, ρ_k para a qual $\rho_k \dots \rho_1\pi = \sigma$, tal que ρ_i é um intercâmbio de blocos aplicável a $\rho_{i-1} \dots \rho_1\pi$ para $1 \leq i \leq k$ então π e σ são eqüiorbitais. De maneira semelhante ao caso de transposições, se dois genomas são eqüiorbitais então existe uma seqüência de intercâmbios de bloco que transforma um genoma em outro: basta encontrar o mesmo tipo de seqüência de transposições como na Seção 3.7.2 uma vez que uma transposição é um tipo particular de intercâmbio de bloco.

Dados os genomas eqüiorbitais π e σ em (E, Γ) , o *problema de rearranjo em genomas por intercâmbio de blocos* consiste em encontrar uma seqüência mínima de intercâmbio de blocos que transforme o genoma π em σ . Em outras palavras, queremos encontrar uma seqüência de intercâmbio de blocos ρ_1, \dots, ρ_k tal que

$$\rho_k \dots \rho_1\pi = \sigma$$

e ρ_i é aplicável a $\rho_{i-1} \dots \rho_1\pi$, para $1 \leq i \leq k$ e k é mínimo. O número k de intercâmbio de blocos na seqüência mínima é chamado de *distância de intercâmbio de blocos* entre π e σ e esse número é denotado por $d_{ib}(\pi, \sigma)$.

3.7.5 Fusões e Fissões

Introduzimos nessa seção a modelagem dos eventos de rearranjo “Fusão” e “Fissão” por meio do formalismo algébrico. Esses dois eventos de rearranjo não haviam ainda sido modelados no

trabalho de Meidanis e Dias [39].

Dado o genoma π no sistema de genes (E, Γ) um evento de rearranjo $(u \ v)(\pi\Gamma v \ \pi\Gamma u)$ é uma *fissão* aplicável a π quando $(u \ v)|\pi$, ou seja u e v pertencem a uma mesma órbita de π . Um evento de rearranjo $(u \ v)(\pi\Gamma v \ \pi\Gamma u)$ é uma *fusão* quando u e v pertencem a órbitas de cromossomos distintos em π .

De modo diferente ao de reversões com sinais, transposições, intercâmbios de bloco e transposições generalizadas, para quaisquer pares de genomas distintos em um sistemas de genes existe uma seqüência de fusões e fissões que transforme um genoma em outro.

Dados os genomas π e σ em (E, Γ) , o *problema de rearranjo em genomas por fusões e fissões* consiste em encontrar uma seqüência mínima de fusões e fissões que transforme o genoma π em σ . Em outras palavras, queremos encontrar uma seqüência de fusões e fissões ρ_1, \dots, ρ_k tal que

$$\rho_k \dots \rho_1 \pi = \sigma$$

e ρ_i é aplicável a $\rho_{i-1} \dots \rho_1 \pi$, para $1 \leq i \leq k$ e k é mínimo. O número k de fusões e fissões na seqüência mínima é chamado de *distância de fusões e fissões* entre π e σ e esse número é denotado por $d_{ff}(\pi, \sigma)$. O problema de rearranjo em genomas envolvendo fissões e fusões pode conter em sua entrada um par de quaisquer genomas de um mesmo sistema de genes. Isso se deve ao fato de uma fissão ou fusão aplicável a um genoma modificar o conjunto de órbitas de um genoma.

3.8 Rearranjos em Genomas no Formalismo Algébrico

O desenvolvimento do estudo de problemas de rearranjo em genomas revelou diversas semelhanças entre alguns eventos de rearranjo [61]. Por meio dessas similaridades entre eventos de rearranjo é possível desenvolver uma estratégia geral para resolver problemas de rearranjo em genomas. Apresentamos nessa seção uma estratégia de resolução de problemas de rearranjo (ponderados) modelados por meio do formalismo algébrico.

Iremos generalizar o problema de rearranjos em genomas para lidar com reversões com sinais, fusões, fissões e intercâmbios de blocos. Além disso, introduzimos uma versão ponderada do problema de rearranjo em genomas que se baseia na atribuição de pesos aos eventos de rearranjo segundo as suas normas. Mostraremos como obter um limitante inferior para a distância genômica ponderada e também que é possível definir uma estratégia para encontrar uma seqüência de eventos de rearranjo que transforma um genoma em outro caso o problema de rearranjo obedecer a certas condições.

Acreditamos que o desenvolvimento de uma estratégia de resolução de problemas de rearranjo é uma contribuição de nossa pesquisa que permitirá que se avalie mais rapidamente se um determinado problema de rearranjo pode ser resolvido facilmente.

3.8.1 Distância Genômica

Sejam π e σ dois genomas em (E, Γ) e R um conjunto de tipos de eventos de rearranjo em genomas. Os genomas π e σ são chamados *compatíveis em R* quando existe uma sequência de eventos de rearranjos ρ_1, \dots, ρ_l dos tipos encontrados em R e

$$\rho_l \dots \rho_1 \pi = \sigma,$$

onde ρ_i é aplicável a $\rho_{i-1} \dots \rho_1 \pi$ para $1 \leq i \leq l$.

Dados um conjunto de tipos de eventos de rearranjo em genomas R e dois genomas π e σ em um sistema de genes (E, Γ) compatíveis em R , o *problema de rearranjo em genomas* $P(\pi, \sigma, R, E, \Gamma)$ consiste em encontrar uma sequência de eventos de rearranjo ρ_1, \dots, ρ_k dos tipos encontrados em R , tal que

$$\rho_k \dots \rho_1 \pi = \sigma,$$

onde ρ_i é aplicável a $\rho_{i-1} \dots \rho_1 \pi$ para $1 \leq i \leq k$ e k é mínimo. O valor k é chamado de *distância genômica* e é denotado por $d(\pi, \sigma)$.

Como para quaisquer pares de genomas π e σ que servem de entrada a um problema de rearranjo em genomas $P(\pi, \sigma, R, E, \Gamma)$ existe uma sequência de eventos de rearranjo ρ_1, \dots, ρ_k tal que $\rho_k \dots \rho_1 \pi = \sigma$ e ρ_i é um evento de rearranjo aplicável em $\rho_{i-1} \dots \rho_1 \pi$ para $1 \leq i \leq k$, temos $\rho_k \dots \rho_1 = \sigma \pi^{-1}$. Logo a permutação $\sigma \pi^{-1}$, a qual chamamos de *quociente* de π e σ , pode ser decomposta em um produto dos eventos de rearranjo ρ_1, \dots, ρ_k . Em outros termos, o problema de rearranjo em genomas $P(\pi, \sigma, R, E, \Gamma)$ é equivalente a encontrar um produto $\rho_k \dots \rho_1$ que seja igual a $\sigma \pi^{-1}$ e cada ρ_i é um evento de rearranjo cujo tipo é encontrado em R seja aplicável a $\rho_{i-1} \dots \rho_1 \pi$ para $1 \leq i \leq k$ e k é mínimo. Nós iremos definir alguns conceitos e mostrar algumas propriedades da permutação $\sigma \pi^{-1}$.

Dados os genomas π e σ em (E, Γ) , um *par* de $\sigma \pi^{-1}$ é uma dupla de ciclos α e $(\pi \Gamma) \cdot \alpha^{-1}$ de $\sigma \pi^{-1}$. Seja $c(\pi, \sigma)$ o número de pares de $\sigma \pi^{-1}$. Meidanis e Dias [39] mostram que o número de pares de $\sigma \pi^{-1}$ é $c(\pi, \sigma) = (|E| - \|\sigma \pi^{-1}\|)/2$, o que equivale a metade do número de órbitas de $\sigma \pi^{-1}$. Denotamos a diferença $c(\rho \pi, \sigma) - c(\pi, \sigma)$ por $\Delta c(\rho, \pi, \sigma)$ onde ρ é um evento de rearranjo aplicável ao genoma π . O conceito de um par de $\sigma \pi^{-1}$ corresponde ao conceito de ciclos alternados do diagrama de ciclos (ou grafo de pontos de quebra) de um genoma [28, 29, 32] (ver Capítulo 2).

Exemplo 71. Dado o conjunto de genes

$$E = \{a, b, c, d, e, f, g, h, i, -a, -b, -c, -d, -e, -f, -g, -h, -i\}$$

e a permutação $\Gamma = (a -a)(b -b) \dots (i -i)$. Considere os genomas

$$\pi = (a -c -d b -i f g -h -e)(e h -g -f i -b d c -a)$$

e

$$\sigma = (a\ b\ c\ d\ e\ f\ g\ h\ i)(-i\ -h\ -g\ -f\ -e\ -d\ -c\ -b\ -a)$$

no sistema de genes (E, Γ) . A permutação $\sigma\pi^{-1}$ é

$$(a\ -d\ -b)(-i\ c\ e)(b\ -c)(-a\ d)(-e\ -g\ i)(f\ -h\ h)(-f)(g).$$

A permutação $(b\ -c)(-a\ d)$ é um exemplo de par de $\sigma\pi^{-1}$. O quociente $\sigma\pi^{-1}$ possui $c(\pi, \sigma) = 4$ pares.

A maneira que um evento de rearranjo afeta $\sigma\pi^{-1}$ é importante, pois se $\rho_1 \dots \rho_k$ é uma sequência de eventos de rearranjo tal que $\rho_k \dots \rho_1 \pi = \sigma$ e ρ_i é aplicável a $\rho_{i-1} \dots \rho_1 \pi$ para $1 \leq i \leq k$ então $\sigma\pi^{-1}\rho_1^{-1} \dots \rho_k^{-1} = \sigma\sigma^{-1} = \iota$. Em outros termos, cada evento ρ_i ao ser aplicado a π altera o número de pares (preferencialmente aumentando) em $\sigma\pi^{-1}\rho_1^{-1} \dots \rho_i^{-1}$ em relação a $\sigma\pi^{-1}\rho_1^{-1} \dots \rho_{i-1}^{-1}$ para $1 \leq i \leq k$ de modo que ao final da aplicação da sequência de eventos o quociente de $\rho_k \dots \rho_1 \pi$ e σ seja a identidade.

Exemplo 72. Considere novamente o genomas π e σ no sistema de genes (E, Γ) do Exemplo 71.

O quociente

$$\sigma\pi^{-1} = (a\ -d\ -b)(-i\ c\ e)(b\ -c)(-a\ d)(-e\ -g\ i)(f\ -h\ h)(-f)(g)$$

pode ser decomposto, por exemplo, no produto de reversões com sinais:

$$\rho_5\rho_4\rho_3\rho_2\rho_1 = (a\ -b)(-i\ e)(-d\ -b)(e\ c)(-a\ d)(b\ -c)(-e\ -g)(-h\ h)(-e\ i)(f\ -h),$$

tal que ρ_i é aplicável a $\rho_{i-1} \dots \rho_1 \pi$ para $1 \leq i \leq 5$.

Uma estratégia para encontrar uma sequência de eventos de rearranjo que seja mínima e transforme π em σ é a de determinar qual o evento de rearranjo ρ aplicável a π que provoca o máximo aumento possível no número de pares em $\sigma\pi^{-1}\rho^{-1}$ em relação a $\sigma\pi^{-1}$. Se sempre existir tal evento de rearranjo para quaisquer pares de genomas na entrada do problema e pudermos identificá-lo de maneira rápida é possível projetar um algoritmo simples e eficiente para a solução de um problema de rearranjo em genomas. Infelizmente, nem sempre existe um evento de rearranjo que provoque o aumento máximo no número de pares do quociente para dois genomas π e σ quaisquer. Mesmo que exista um evento de rearranjo ρ que aumente o número de pares de $\sigma\pi^{-1}$, esse fato não implica que para $\rho\pi$ e σ existirá um evento de rearranjo desse tipo em $\sigma\pi^{-1}\rho^{-1}$ (veja o Exemplo 73).

Exemplo 73. Dado o conjunto de genes

$$E = \{a, b, c, d, e, f, g, h, -a, -b, -c, -d, -e, -f, -g, -h\}$$

e a permutação $\Gamma = (a - a)(b - b) \dots (h - h)$. Considere os genomas

$$\pi = (a - b \ c \ e - f \ d \ g - h)(h - g - d \ f - e - c \ b - a)$$

e

$$\sigma = (a \ b \ c \ d \ e \ f \ g \ h)(-h - g - f - e - d - c - b - a)$$

no sistema de genes (E, Γ) . O quociente $\sigma\pi^{-1}$ é

$$(a - g)(h - h)(b - b)(-a \ c)(e \ d - e \ g)(-d - f \ f - c).$$

Se quisermos encontrar uma seqüência mínima de reversões com sinais que transforme π em σ utilizando a estratégia de selecionar apenas as reversões com sinais que aumentem o número de pares no quociente, obteremos um par de genomas para o qual não há tal tipo de reversão com sinais. A seqüência de reversões com sinais $(a - g)(h - h)(b - b)(-a \ c)(d - e)(-f \ f)$ (de fato todas as seqüências envolvendo esses três eventos de rearranjo) leva ao par de genomas

$$(a - g)(h - h)(b - b)(-a \ c)(d - e)(-f \ f)\pi,$$

ou seja $(a \ b \ c \ e \ f \ d \ g \ h)(-h - g - d - f - e - c - b - a)$ e

$$\sigma = (a \ b \ c \ d \ e \ f \ g \ h)(-h - g - f - e - d - c - b - a),$$

cujo quociente é $(e \ d \ g)(-d - f - g)$ e não há nenhuma reversão com sinais ρ aplicável ao genoma $(a - g)(h - h)(b - b)(-a \ c)(d - e)(-f \ f)\pi$ que aumente o número de pares em $(e \ d \ g)(-d - f - g)\rho$ em relação a $(e \ d \ g)(-d - f - g)$. Portanto a estratégia de encontrar seqüências de eventos de rearranjo que maximizem a variação no número de pares no quociente falha para o problema envolvendo reversões com sinais.

A validade da estratégia depende dos tipos de eventos de rearranjo considerados no problema de rearranjo em questão. O problema de rearranjo em genomas por reversões com sinais, por exemplo, não pode ser solucionado por meio da estratégia gulosa e simples de procurar as reversões com sinais que aumentem o número de pares do quociente dos dois genomas na entrada. O Exemplo 73 ilustra uma instância do problema de rearranjo em genomas por reversões com sinais para a qual a estratégia falha (o Exemplo 5 apresenta uma situação análoga no formalismo clássico). Demonstraremos que a estratégia gulosa acima funciona para problemas de rearranjo ponderados que envolvam uma determinada combinação de eventos de rearranjo.

3.8.2 Distância Genômica Ponderada

Em problemas de rearranjo em genomas que envolvem diferentes tipos de eventos de rearranjo, é comum adotarmos um peso para cada tipo de evento de rearranjo que tente refletir a frequência em que tais eventos ocorram em um cenário evolutivo parcimonioso [20, 28, 61]. Os pesos

dos eventos de rearranjo que pertencem a uma seqüência de eventos que transforma um genoma em outro são geralmente somados para serem usados como um parâmetro alternativo à distância genômica. Propomos a norma do evento de rearranjo como um parâmetro para calcular o peso de um evento. A norma fornece um parâmetro sistemático e formal aplicável a qualquer evento de rearranjo que possa ser representado pelo formalismo algébrico. A norma também é um parâmetro que está de acordo com atribuições de pesos anteriores [11]. Por outro lado, um problema na atribuição de pesos baseados na norma é a impossibilidade de distinguir pelo peso alguns eventos de rearranjo distintos como transposições e transposições generalizadas por exemplo e, além disso, o fato de a norma coincidir com os pesos (ou proporções entre pesos) atribuídos a eventos de rearranjo em trabalhos anteriores não garante que seja um parâmetro que possua relevância biológica. Dados os genomas π e σ em (E, Γ) , o *peso* de um evento de rearranjo qualquer ρ aplicável a π , denotado por $w(\rho)$, é $w(\rho) = \|\rho\|/2$. O peso de um evento de rearranjo ρ é relacionado a sua capacidade de “quebrar” ou “juntar” pares em $\sigma\pi^{-1}\rho^{-1}$ em comparação a $\sigma\pi^{-1}$. Além disso, dependendo da estrutura do evento de rearranjo, por exemplo se tivermos um produto de 2-ciclos ou um produto de 3-ciclos, o evento de rearranjo afeta as paridades dos ciclos de $\sigma\pi^{-1}$.

Uma vez que atribuímos pesos aos diferentes eventos de rearranjo, podemos definir uma versão “ponderada” do problema de rearranjo em genomas. Dados o conjunto de tipos de eventos de rearranjo em genomas R , os genomas π e σ em um sistema de genes (E, Γ) compatíveis em R e uma função de peso w , o *problema de rearranjo em genomas ponderado* $P(\pi, \sigma, R, E, \Gamma, w)$ consiste em encontrar uma seqüência de eventos de rearranjo ρ_1, \dots, ρ_k dos tipos encontrados em R , tal que

$$\rho_k \dots \rho_1 \pi = \sigma,$$

onde ρ_i é aplicável a $\rho_{i-1} \dots \rho_1 \pi$ para $1 \leq i \leq k$ e $\sum_{i=1}^k w(\rho_i)$ é mínimo. O valor $\sum_{i=1}^k w(\rho_i)$ é chamado de *distância genômica ponderada* e é denotado por $W(\pi, \sigma)$.

Mostraremos algumas propriedades que relacionam o peso de um evento de rearranjo à divisibilidade do quociente por esse evento que permitem aplicarmos naturalmente uma estratégia gulosa para resolver o problema de rearranjo em genomas ponderado. Na Seção 5.1, demonstraremos que propriedades equivalentes envolvendo a 3-divisibilidade do quociente permitem a aplicação da mesma estratégia gulosa para a solução do problema ponderado envolvendo transposições generalizadas.

Um *evento 2-bom* para (π, σ) é um evento de rearranjo ρ aplicável a π tal que $\rho|\sigma\pi^{-1}$. Caracterizaremos um evento 2-bom de acordo com a diferença entre o número de pares do quociente após a aplicação do evento de rearranjo e o quociente original de dois genomas. O Lema 74 permite que utilizemos $\Delta c(\rho, \pi, \sigma) = w(\rho)$ como propriedade fundamental do genoma.

Lema 74. Dados os genomas π e σ em (E, Γ) , um evento de rearranjo ρ aplicável a π é um evento 2-bom para (π, σ) se e somente se $\Delta c(\rho, \pi, \sigma) = w(\rho)$.

Prova: Considere os genomas π e σ em (E, Γ) . Se ρ é um evento 2-bom para (π, σ) então $\rho|\sigma\pi^{-1}$, ou equivalentemente $\|\sigma\pi^{-1}\rho^{-1}\| = \|\sigma\pi^{-1}\| - \|\rho\|$. Manipulando essa formula:

$$\frac{\|\rho\|}{2} = \frac{\|\sigma\pi^{-1}\| - \|\sigma\pi^{-1}\rho^{-1}\|}{2} = \frac{|E| - \|\sigma\pi^{-1}\rho^{-1}\| - |E| + \|\sigma\pi^{-1}\|}{2} = c(\rho\pi, \sigma) - c(\pi, \sigma)$$

e como $c(\rho\pi, \sigma) - c(\pi, \sigma) = \Delta c(\rho, \pi, \sigma)$ então $\Delta c(\rho, \pi, \sigma) = w(\rho)$.

Inversamente, se $\Delta c(\rho, \pi, \sigma) = w(\rho)$, ou seja, temos $c(\rho\pi, \sigma) - c(\pi, \sigma) = \|\rho\|/2$, então pela definição de $c(\pi, \sigma)$ temos

$$\frac{\|\sigma\pi^{-1}\| - \|\sigma\pi^{-1}\rho^{-1}\|}{2} = \frac{\|\rho\|}{2}.$$

Ou de modo equivalente, temos $\|\sigma\pi^{-1}\rho^{-1}\| = \|\sigma\pi^{-1}\| - \|\rho\|$ e $\rho|\sigma\pi^{-1}$. Logo ρ é um evento 2-bom para (π, σ) .

Portanto, um evento de rearranjo ρ aplicável a π é um evento 2-bom para (π, σ) se e somente se $\Delta c(\rho, \pi, \sigma) = w(\rho)$. \square

O Lema 75 oferece um limitante inferior para a soma dos pesos dos eventos de rearranjo em uma seqüência que transforma um genoma em outro. Além disso, o resultado afirma que uma seqüência formada apenas por eventos 2-bons possui peso total mínimo.

Lema 75. Dados os genomas π e σ compatíveis em R no sistema de genes (E, Γ) , para qualquer seqüência de eventos de rearranjo ρ_1, \dots, ρ_k , tal que $\rho_k \dots \rho_1\pi = \sigma$ e ρ_i é aplicável ao genoma $\rho_{i-1} \dots \rho_1\pi$, temos:

1. $\sum_{j=1}^k w(\rho_j) \geq \frac{\|\sigma\pi^{-1}\|}{2}$;
2. $\sum_{j=1}^k w(\rho_j) = \frac{\|\sigma\pi^{-1}\|}{2}$ se e somente se cada evento de rearranjo ρ_i é um evento 2-bom para $(\rho_{i-1} \dots \rho_1\pi, \sigma)$ para $1 \leq i \leq k$.

Prova:

1. Seja ρ_1, \dots, ρ_k uma seqüência de eventos de rearranjo tais que $\rho_k \dots \rho_1\pi = \sigma$ e ρ_i é aplicável a $\rho_{i-1} \dots \rho_1\pi$ para $1 \leq i \leq k$. Logo $\rho_k \dots \rho_1 = \sigma\pi^{-1}$ e temos o seguinte limitante superior para $\|\sigma\pi^{-1}\|$.

$$\begin{aligned} \|\sigma\pi^{-1}\| &= \|\rho_k \dots \rho_1\| \\ &\leq \|\rho_k\| + \dots + \|\rho_1\| \\ &= 2 \sum_{j=1}^k \frac{\|\rho_j\|}{2} \\ &= 2 \sum_{j=1}^k w(\rho_j) \end{aligned}$$

Logo temos $\sum_{j=1}^k w(\rho_j) \geq \frac{\|\sigma\pi^{-1}\|}{2}$.

2. Assumimos que cada evento de rearranjo ρ_i é um evento 2-bom para $(\rho_{i-1} \dots \rho_1 \pi, \sigma)$ para $1 \leq i \leq k$. Pela definição de peso, temos:

$$\sum_{j=1}^k w(\rho_j) = \frac{\|\rho_1\| + \dots + \|\rho_k\|}{2}. \quad (3.1)$$

Uma vez que o evento de rearranjo ρ_i é um evento 2-bom para $(\rho_{i-1} \dots \rho_1 \pi, \sigma)$ para $1 \leq i \leq k$ e pelo Lema 74 temos

$$\frac{\|\rho_i\|}{2} = c(\rho_i \dots \rho_1 \pi, \sigma) - c(\rho_{i-1} \dots \rho_1 \pi, \sigma).$$

Pela definição do número de pares temos

$$\frac{\|\rho_i\|}{2} = \frac{\|\sigma \pi^{-1} \rho_1^{-1} \dots \rho_{i-1}^{-1}\|}{2} - \frac{\|\sigma \pi^{-1} \rho_1^{-1} \dots \rho_i^{-1}\|}{2}. \quad (3.2)$$

Usando a Equação 3.1 e a Equação 3.2 e manipulando como segue:

$$\begin{aligned} \sum_{j=1}^k w(\rho_j) &= \frac{\|\sigma \pi^{-1}\| - \|\sigma \pi^{-1} \rho_1^{-1} \dots \rho_k^{-1}\|}{2} \\ &= \frac{\|\sigma \pi^{-1}\| - \|\sigma \sigma^{-1}\|}{2} \\ &= \frac{\|\sigma \pi^{-1}\|}{2}. \end{aligned}$$

Logo $\sum_{j=1}^k w(\rho_j) = \frac{\|\sigma \pi^{-1}\|}{2}$.

Por outro lado, se $\sum_{j=1}^k w(\rho_j) = \|\sigma \pi^{-1}\|/2$ então $\|\sigma \pi^{-1}\| = \sum_{j=1}^k \|\rho_j\|$.

Pela propriedade da desigualdade triangular, temos

$$\|\sigma \pi^{-1} \rho_1^{-1} \dots \rho_{i-1}^{-1}\| \leq \|\sigma \pi^{-1} \rho_1^{-1} \dots \rho_i^{-1}\| + \|\rho_i\|,$$

para $1 \leq i \leq k$, em outras palavras temos

$$0 \leq \|\sigma \pi^{-1} \rho_1^{-1} \dots \rho_i^{-1}\| - \|\sigma \pi^{-1} \rho_1^{-1} \dots \rho_{i-1}^{-1}\| + \|\rho_i\|,$$

para $1 \leq i \leq k$. Como cada termo $\|\sigma \pi^{-1} \rho_1^{-1} \dots \rho_i^{-1}\| - \|\sigma \pi^{-1} \rho_1^{-1} \dots \rho_{i-1}^{-1}\| + \|\rho_i\|$ é não negativo e manipulando a soma expandida, obtemos

$$\begin{aligned} 0 &\leq \sum_{i=1}^k \left(\|\sigma \pi^{-1} \rho_1^{-1} \dots \rho_i^{-1}\| - \|\sigma \pi^{-1} \rho_1^{-1} \dots \rho_{i-1}^{-1}\| + \|\rho_i\| \right) \\ &= \|\sigma \pi^{-1} \rho_1^{-1} \dots \rho_k^{-1}\| - \|\sigma \pi^{-1}\| + \sum_{i=1}^k \|\rho_i\|. \end{aligned}$$

Como $\sigma\pi^{-1}\rho_1^{-1} \dots \rho_k^{-1} = \iota$ e $\sum_{i=1}^k \|\rho_i\| = \|\sigma\pi^{-1}\|$ então

$$\sum_{i=1}^k \left(\|\sigma\pi^{-1}\rho_1^{-1} \dots \rho_i^{-1}\| - \|\sigma\pi^{-1}\rho_1^{-1} \dots \rho_{i-1}^{-1}\| + \|\rho_i\| \right) = 0.$$

Então $\|\sigma\pi^{-1}\rho_1^{-1} \dots \rho_i^{-1}\| - \|\sigma\pi^{-1}\rho_1^{-1} \dots \rho_{i-1}^{-1}\| + \|\rho_i\| = 0$ para $1 \leq i \leq k$; i.e. temos

$$\|\sigma\pi^{-1}\rho_1^{-1} \dots \rho_i^{-1}\| = \|\sigma\pi^{-1}\rho_1^{-1} \dots \rho_{i-1}^{-1}\| - \|\rho_i\|$$

para $1 \leq i \leq k$. Logo, temos $\rho_i|\sigma\pi^{-1}\rho_1^{-1} \dots \rho_{i-1}^{-1}$ para $1 \leq i \leq k$, ou seja, cada ρ_i aplicável a $\rho_{i-1} \dots \rho_1\pi$ é um evento 2-bom para $(\rho_{i-1} \dots \rho_1\pi, \sigma)$.

□

O Lema 74 e o Lema 75 podem ser utilizados para justificar a estratégia de buscar por eventos 2-bons para encontrar uma seqüência de eventos de rearranjo que transforme um genoma em outro com distância genômica ponderada mínima. O Teorema 76 justifica a utilização da estratégia para o caso de eventos de rearranjo de peso 1 (fusões, fissões e reversões com sinais).

Teorema 76. *Se para quaisquer genomas distintos π e σ compatíveis em R , conjunto dos tipos fusão, fissão e reversão com sinais, no sistema de genes (E, Γ) , houver um evento de rearranjo ρ aplicável a π tal que ρ é um evento 2-bom para (π, σ) então existe uma seqüência de eventos de rearranjo ρ_1, \dots, ρ_k onde ρ_i é um evento 2-bom para $(\rho_{i-1} \dots \rho_1\pi, \sigma)$ e $\rho_k \dots \rho_1\pi = \sigma$ para $1 \leq i \leq k$.*

Prova: Dados os genomas distintos π e σ compatíveis em R no sistema de genes (E, Γ) e como a norma de $\sigma\pi^{-1}$ é par pelo Lema 52, demonstramos por indução em k para a norma de $\|\sigma\pi^{-1}\| = 2k$ a existência de uma seqüência de eventos de rearranjo ρ_1, \dots, ρ_k onde ρ_i é um evento 2-bom para $(\rho_{i-1} \dots \rho_1\pi, \sigma)$ para $1 \leq i \leq k$.

Para $k = 1$ temos $\|\sigma\pi^{-1}\| = 2$ e como existe uma fusão, fissão ou reversão com sinal ρ que é um evento 2-bom para (π, σ) então $\|\sigma\pi^{-1}\rho^{-1}\| = \|\sigma\pi^{-1}\| - \|\rho\|$. Como $\|\rho\| = 2$ então $\|\sigma\pi^{-1}\rho^{-1}\| = 0$ e logo $\sigma\pi^{-1}\rho^{-1} = \iota$. Portanto temos $\rho\pi = \sigma$ e ρ é um evento 2-bom para (π, σ) .

Para $k > 1$ temos $\|\sigma\pi^{-1}\rho^{-1}\| = \|\sigma\pi^{-1}\| - \|\rho\|$ pois existe um evento 2-bom ρ para (π, σ) . Como $\|\sigma\pi^{-1}\| = 2k$ e $\|\rho\| = 2$ então $\|\sigma\pi^{-1}\rho^{-1}\| = 2(k-1)$. Além disso, temos $\|\sigma\pi^{-1}\rho^{-1}\| = \|\sigma(\rho\pi)^{-1}\|$. Pela hipótese de indução, no caso em que temos $k-1$, ou seja, para $\|\sigma(\rho\pi)^{-1}\| = 2(k-1)$, existe uma seqüência ρ_2, \dots, ρ_k de eventos de rearranjo tal que ρ_i é um evento 2-bom para $(\rho_{i-1} \dots \rho_2\rho\pi, \sigma)$ para $2 \leq i \leq k$. Portanto, como $\rho_1 = \rho$ é um evento 2-bom para (π, σ) então existe uma seqüência de eventos de rearranjo ρ_1, \dots, ρ_k onde ρ_i é um evento 2-bom para $(\rho_{i-1} \dots \rho_1\pi, \sigma)$ e $\rho_k \dots \rho_1\pi = \sigma$ para $1 \leq i \leq k$.

□

Deduzimos a partir do Teorema 76 que se houver um evento 2-bom para (π, σ) de um tipo em R aplicável a um genoma π distinto de σ , então é possível encontrar uma sequência de eventos de rearranjo que seja solução para um problema $P(\pi, \sigma, R, E, \Gamma, w)$ e $W(\pi, \sigma)$ é mínimo.

3.9 Considerações Finais

Nesse capítulo apresentamos uma versão modificada do formalismo algébrico para problemas de rearranjo em genomas. Apresentamos o conceito de permutação, órbitas de uma permutação, ciclos, produtos entre permutações e a representação de uma permutação como um produto de ciclos disjuntos. Discutimos os efeitos do produto por 2-ciclos e 3-ciclos nas órbitas e na ordem dos elementos nos ciclos de uma permutação. Apresentamos a operação de conjugação que “renomeia” os elementos de uma permutação sem modificar a sua estrutura. Discutimos a relação da divisibilidade e a decomposição de permutações em 2-ciclos ou 3-ciclos. Em especial, contribuímos com a demonstração de diversas propriedades envolvendo 3-ciclos que não se encontravam em trabalhos anteriores sobre o formalismo algébrico. Além disso, mostramos como modelar genomas por meio de permutações representadas como produto de ciclos, o conceito de um sistema de genes e as propriedades fundamentais de genomas. O conceito de sistema de genes e a definição de genomas baseada na divisibilidade pela permutação Γ são contribuições novas para o formalismo. Apresentamos uma maneira de representar eventos de rearranjo aplicáveis a um certo genoma como permutações. Introduzimos um novo evento de rearranjo (transposição generalizada) que pode contribuir para uma melhor compreensão de transposições. Juntamente com a definição dos eventos de rearranjo, definimos os respectivos problemas de rearranjo em genomas que envolvem esses eventos. Por último, discutimos como um problema de rearranjo em genomas é tratado no formalismo algébrico. Após uma sistematização de como um problema é abordado pelo formalismo algébrico, apresentamos uma estratégia geral para resolver determinados problemas de rearranjo que satisfazem ao critério de existência de eventos de rearranjo “adequados” (os 2-bons) dentre os eventos permitidos no problema.

Capítulo 4

Nova Estrutura de Dados para Genomas

Diversas estruturas de dados têm sido utilizadas para a representação de genomas [33, 32, 8]. Nesse capítulo descrevemos uma nova estrutura de dados que representa os genomas π e σ em um sistema de genes (E, Γ) e a permutação $\sigma\pi^{-1}$ sobre E . A estrutura de dados é uma combinação de vetores que modelam as permutações π , π^{-1} e σ ; além de uma tabela de espalhamento (hash table) e uma lista de listas circulares representando a permutação $\sigma\pi^{-1}$.

Mostraremos no Capítulo 5 que essa estrutura de dados nos permite aplicar certos eventos de rearranjo, tais como fissões, fusões, transposições generalizadas e reversões com sinais, em tempo de execução constante.

4.1 Genes rotulados como inteiros

Trataremos primeiramente do caso em que π e σ são genomas no sistema de genes (E, Γ) onde $E = \{-n, \dots, -1, 1, \dots, n\}$ e $\Gamma = (1 \ -1)(2 \ -2) \dots (n \ -n)$. Por tratarmos de números inteiros, não é necessário utilizarmos uma tabela de espalhamento. Na Seção 4.2, estenderemos a estrutura de dados para lidarmos com sistemas de genes cujo conjunto E contenha nomes de genes ao invés de inteiros, o que requer buscas em uma tabela de espalhamento. Os genomas π e σ são representados por vetores P e S onde $P[x] = \pi x$ e $S[x] = \sigma x$ para $x \in E$. Similarmente nós definimos o vetor $invP$ onde $invP[x] = \pi^{-1}x$ para $x \in E$. O gene complementar Γx é simplesmente $-x$. Além disso, descrevemos a permutação $\sigma\pi^{-1}$ como uma lista duplamente ligada Q cujos nós que chamamos de *pares* contém apontadores para duas listas circulares. Cada nó na lista Q modela um par de ciclos α e β de $\sigma\pi^{-1}$ onde $\beta = (\pi\Gamma) \cdot \alpha^{-1}$ e esses ciclos são as listas circulares apontadas pelo nó. Não representamos 1-ciclos na lista, embora modificações na lista devido a aplicação de eventos de rearranjo podem levar ao aparecimento de pares contendo dois 1-ciclos que devem ser eliminados (chamamos a um par com dois 1-ciclos de *par trivial*). Finalmente, usamos um vetor T cuja posição $T[x].node$ aponta para o nó em Q que contém x e $T[x].pair$ aponta para o par (nó com duas listas circulares) de x , para

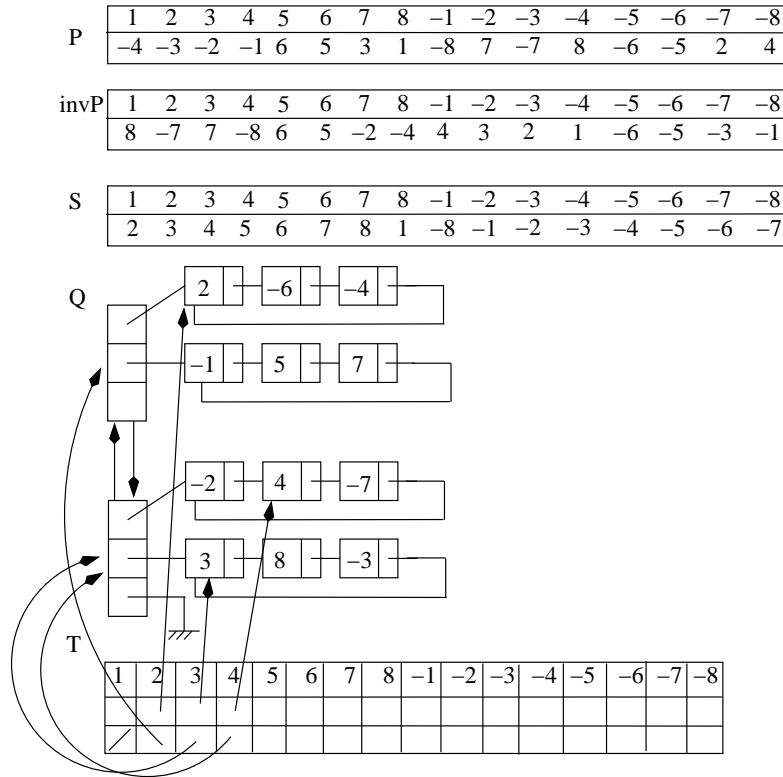


Figura 4.1: Genomas $\pi = (3 \ -2 \ 7)(-7 \ 2 \ -3)(-5 \ -6)(6 \ 5)(1 \ -4 \ 8)(-8 \ 4 \ -1)$ e $\sigma = (1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8)(-8 \ -7 \ -6 \ -5 \ -4 \ -3 \ -2 \ -1)$ no sistema de genes (E, Γ) onde $E = \{x \in \mathbb{Z} \mid 1 \leq |x| \leq 8\}$ e $\Gamma x = -x$ são representados pelos vetores P e S . A permutação $\sigma\pi^{-1}$ é representada pela lista Q . O vetor T tem em cada entrada indexada por um gene x um apontador para o nó contendo x e para o par de listas circulares em Q . Para deixar a figura mais clara, mostramos apenas os apontadores dos quatro primeiros genes de T .

Procedimento initializePairsList(Lista de Pares Q , vetor S , vetor $invP$, vetor T)

```

1: for all  $x \in E$  do
2:    $T[x].node \leftarrow T[x].pair \leftarrow NIL$ 
3: end for
4: for all  $x \in E$  do
5:   if  $S[invP[x]] \neq x$  e  $T[x].node = NIL$  then
6:      $buildPair(x)$ 
7:   end if
8: end for
9: return Pair List  $Q$ 

```

Figura 4.2: Procedimento para inicializar a Lista de Pares Q .

todo $x \in E$ (usamos por convenção $T[x].node = T[x].pair = NIL$ quando não há x em Q). A Figura 4.1 ilustra a estrutura de dados para dois genomas em um sistema de genes com oito genes.

O vetor P é inicializado ao se atribuir $P[x] = \pi x$, para cada gene x em E . O mesmo procedimento pode ser usado para inicializar o vetor S . Podemos inicializar $invP$ ao mesmo tempo que inicializamos P ao atribuir-se $invP[\pi x] = x$ para cada atribuição de um valor a $P[x]$ em P . A inicialização de Q é baseada nos vetores previamente inicializados S e $invP$. No início do procedimento, todos os apontadores em T são inicializados para NIL . Para cada gene $x \in E$, se $S[invP[x]] \neq x$ e $T[x].node = NIL$, então o procedimento $buildPair(x)$ é chamado. O procedimento $buildPair(x)$ é responsável por criar um novo par com x e incluí-lo em Q . O procedimento começa criando um par N , armazenando o valor de x em uma variável auxiliar y e entrando em um laço que cria nós para y e $\pi\Gamma y$, atualiza os apontadores em $T[y]$ e $T[P[-y]]$ para apontá-los para o par N e os nós contendo y e $\pi\Gamma y$ respectivamente, adiciona os nós às listas circulares que estão sendo construídas e atribui $S[invP[y]]$ a y . O laço termina quando $y = x$. Suponha que o par N em Q modela a permutação $\alpha((\pi\Gamma) \cdot \alpha^{-1})$ e $y \in Supp(\alpha)$, então y é mapeado em $\sigma\pi^{-1}y$ em α enquanto $\pi\Gamma\sigma\pi^{-1}y$ é mapeado em $\pi\Gamma y$ em $(\pi\Gamma) \cdot \alpha^{-1}$. Logo o nó contendo y tem de ser adicionado à cauda da lista circular modelando α e o nó contendo $\pi\Gamma y$ tem de ser adicionado à cabeça da lista circular modelando $(\pi\Gamma) \cdot \alpha^{-1}$. Após o fim do laço, ambas as listas circulares são fechadas. O procedimento que inicializa Q , chamado $initializePairList(Q)$ e o procedimento que constrói um novo par e o insere em Q são ilustrados pela Figura 4.2 e pela Figura 4.3 respectivamente.

Os vetores P , $invP$, S , T , e a lista Q provém as seguintes operações: pesquisa, troca, remoção, teste de pares triviais e deleção de par trivial. Os eventos de rearranjo são uma combinação das operações anteriores. Uma *pesquisa* por um elemento $x \in E$ em um genoma π (ou σ)

Procedimento buildPair(gene x)

- 1: Criar novo par N e inserir N em Q
- 2: $y \leftarrow x$
- 3: **repeat**
- 4: Criar novo nó A com o gene y
- 5: Inserir A à cauda da lista circular 1 de N
- 6: $T[y].node \leftarrow A$
- 7: $T[y].pair \leftarrow N$
- 8: Criar novo nó B com gene $P[-y]$
- 9: Inserir B à cabeça da lista circular 2 de N
- 10: $T[P[-y]].node \leftarrow B$
- 11: $T[P[-y]].pair \leftarrow N$
- 12: $y \leftarrow S[invP[y]]$
- 13: **until** $y = x$
- 14: Fechar a lista circular contendo x
- 15: Fechar a lista circular contendo $P[-x]$

Figura 4.3: Procedimento usado para construir um par N e incluí-lo em Q .

consiste em encontrar πx (ou σx). A pesquisa de um elemento $x \in E$ no vetor A (o vetor A pode ser P , S ou $invP$) é implementada retornando-se o valor em $A[x]$. Uma *troca* de duas entradas $A[x]$ e $A[y]$ (o vetor A pode ser P , S ou $invP$) consiste em trocar os conteúdos de $A[x]$ e $A[y]$. A operação de *remoção*, denotada por $remove(Q, x, y)$, remove os nós que contêm os genes $x, y \in E$. Como as listas circulares podem se encurtar com a aplicação de um evento de rearranjo, elas podem se tornar nós de um par trivial e, neste caso, precisam ser removidas de Q . A operação *teste de pares triviais*, denotado pelo procedimento $trivialPair(Q, x)$, é responsável por detectar se um par contendo x é um par trivial. O teste de pares triviais é implementado verificando-se se o conteúdo do primeiro nó possui o mesmo conteúdo do próximo nó em uma das listas circulares no par. A operação de *deleção de um par trivial* deleta um par trivial de Q que contém os genes $x, \pi\Gamma x \in E$. A operação de deleção de um par trivial, chamada $deletePair(pair)$, é implementada usando-se um apontador $T[x].pair$ e a remoção padrão de nós de uma lista duplamente ligada. Essa operação envolve também uma modificação dos apontadores $node$ e $pair$ em $T[x]$ e $T[\pi\Gamma x]$ para NIL , onde x e $\pi\Gamma x$ são genes em um par trivial.

A atualização da estrutura de dados depende de características específicas do evento de rearranjo (ou conjunto de eventos) levado em consideração. No Capítulo 5, discutiremos como a estrutura é modificada de acordo com o problema de rearranjo em genomas.

Procedimento initializeHash(Tabela de Espalhamento H , Lista de Genes L)

```

1: for all  $i \in \{j \mid 1 \leq |j| \leq |E_+|\}$  do
2:    $H[L[i]].index \leftarrow i$ 
3: end for
4: return Vetor  $H$ 

```

Figura 4.4: Procedimento usado para inicializar a Tabela de Espalhamento a partir do vetor de genes L .

4.2 Genes rotulados com seus próprios nomes

Assumimos anteriormente que o sistema de genes fosse baseado em um conjunto de inteiros, mas genomas são geralmente representados por seqüências de nomes dos genes (cadeias de caracteres) na entrada de problemas de rearranjo em genomas. As seqüências de cadeias de caracteres devem ser convertidas a uma representação adequada (seqüências de inteiros) para poderem ser utilizadas eficientemente pelos vetores P e $invP$ e a lista Q . Para esclarecer a discussão, usamos a seguinte notação: genes são denotados pelas letras x, y, z, w , enquanto os índices associados aos genes são denotados por i, j, k, l . Assumimos que o sistema de genes na entrada do problema é representado por um vetor de genes L que contém todos os nomes de genes. O vetor L é uma concatenação dos conjuntos E_+ e E_- nessa ordem em uma partição E_+ e E_- de (E, Γ) tal que $L[|E_+| + i] = \Gamma L[i]$ para $1 \leq i \leq |E_+|$. Incluímos uma tabela de espalhamento (*hash table*) H para armazenar os índices dos genes em L . Cada nome de gene $L[i] = x$ é uma chave e o dado armazenado na tabela de espalhamento para a chave x é o índice $H[x].index = i$. Os índices de L são usados no mesmo papel que o de genes para os vetores P e $invP$ no caso especial no qual E é um conjunto de inteiros.

Se o conjunto de nomes de genes é o mesmo para quaisquer pares de genomas na entrada de um problema de rearranjo então uma tabela de espalhamento perfeita [19] permite tomar um nome de um gene como uma chave e obter o seu índice nos vetores em tempo constante. A Figura 4.4 ilustra o procedimento de inicialização da tabela de espalhamento H .

Para o caso geral a inicialização dos vetores P e S é um pouco diferente do que no caso de genes representados por inteiros. Assumimos que os genomas π e σ na entrada do procedimento *initialize* são representados por uma lista de cromossomos e cada cromossomo é uma lista de genes na ordem circular em que eles aparecem nas fitas do cromossomo. Para cada nome de gene x na lista de cromossomos de π , o índice i de x e o índice j de πx em H são obtidos e j é atribuído a $P[i]$. O mesmo procedimento é usado para a inicialização de S . A Figura 4.5 mostra o procedimento de inicialização de P .

A pesquisa de um elemento $x \in E$ no genoma π (ou σ) em (E, Γ) consiste em encontrar πx

Procedimento initialize(Genoma π , Tabela T , Lista de Genes L)
 {O vetor A corresponde ao genoma π .}
 1: **for all** Gene x na lista de cromossomos de π **do**
 2: $index \leftarrow H[x].index$
 3: $A[index] \leftarrow H[\pi x].index$
 4: **end for**
 5: **return** Vetor A

Figura 4.5: Procedimento usado para inicializar os vetores P e S .

(ou σx) na tabela de espalhamento H utilizando-se a chave x e retornando $L[P[i]]$ (ou $L[S[i]]$). A operação de pesquisa leva tempo de execução constante $O(h)$ (onde h é o tempo gasto para aplicar a função de *hash* e acessar a tabela de espalhamento). As operações que modificam P , $invP$ e Q ; como a remoção de um gene de Q por exemplo, não requerem mudanças nos índices de H e por esse motivo essas operações levam um tempo de execução constante para serem efetuadas.

Lema 77. A inicialização dos vetores P , S , $invP$ e T ; da tabela de espalhamento H e da lista de pares Q pode ser realizada em tempo de execução linear no número de genes $n = |E|$.

Prova: A tabela de espalhamento H leva o tempo $O(n)$ para ser construída, uma vez que cada gene é lido da entrada, seu nome tomado como uma chave pela função de *hash* e o seu índice armazenado em H (ver Figura 4.4). A inicialização dos vetores P e S leva o tempo de execução de $O(n)$ pois para cada elemento x em E , na ordem em que aparecem em π e σ , tem de ser atribuído os índices de πx e σx a P e S , o que envolve um acesso a H para πx e σx . Como assumimos que π e σ são representados por uma lista dos cromossomos e cada cromossomo é uma lista em sua ordem circular, percorremos as listas que representam os cromossomos para obter x e πx (ou x e σx) em tempo constante. O vetor $invP$ pode ser inicializado com a seguinte atribuição $invP[P[i]] = i$ para $i \in \{k \in \mathbb{Z} \mid 1 \leq |k| \leq |E_+|\}$, o que leva o tempo de execução $O(n)$.

Construir a lista Q leva tempo de execução $O(n)$ porque encontrar $S[invP[i]]$ para $i \in \{k \in \mathbb{Z} \mid 1 \leq |k| \leq |E_+|\}$ envolve uma pesquisa em $invP$ e S que leva $O(1)$ e a atribuição dos apontadores na posição $T[i]$ que leva tempo constante. \square

4.3 Considerações Finais

Projetamos uma estrutura de dados que implementa o modelo de genomas no formalismo algébrico. Primeiramente apresentamos a versão da estrutura de dados que trata do caso especial

em que os genes do sistema de genes são rotulados por números inteiros. Posteriormente estendemos a estrutura com uma tabela de espalhamento (hash table) para lidarmos com o caso em que os genes são rotulados com os seus próprios nomes. A estrutura de dados permite realizar as operações de pesquisa de gene, troca de genes, remoção de um gene de um par, teste de pares triviais e deleção de par trivial em tempo constante. O resultado da aplicação de um evento de rearranjo em um genoma pode ser implementado por meio de uma combinação das operações anteriores, conforme será mostrado no Capítulo 5.

Usaremos essa estrutura de dados no Capítulo 5 com algumas modificações para resolver três problemas distintos de rearranjo em genomas: o problema envolvendo fusões, fissões e reversões com sinais; o problema envolvendo transposições generalizadas e o problema envolvendo intercâmbio de blocos e reversões com sinais.

Capítulo 5

Algoritmos para Novos Problemas de Rearranjo em Genomas

Apresentamos nesse capítulo algumas soluções e avanços na área de rearranjo em genomas trazidos pelo formalismo algébrico.

Na Seção 5.1 mostramos que o limitante inferior para a distância de transposição obtido por meio do formalismo algébrico é equivalente ao obtido pelo formalismo clássico. Discutimos em quais condições o limitante inferior para a distância de transposição baseada em número de pontos de quebra é satisfeito. Apresentamos também uma solução para o problema de rearranjo em genomas ponderado por transposições generalizadas que pode ser obtida em tempo de execução linear por um algoritmo que utiliza uma versão modificada da estrutura de dados apresentada no Capítulo 3.6. Na Seção 5.2 discutimos o problema de rearranjo em genomas ponderado por fissões, fusões e reversões com sinais. Mostramos que o problema de rearranjo em genomas envolvendo fusões, fissões e reversões com sinais pode ser definido para quaisquer pares de genomas em um sistema de genes. Além disso, caracterizamos quais são as propriedades de um evento de rearranjo que pertence a solução desse problema e demonstramos que uma sequência de fusões, fissões e reversões com sinais pode ser obtida em tempo de execução linear por meio de uma estrutura de dados que representa de maneira eficiente um par de genomas e o quociente entre eles. Na Seção 5.3 apresentamos o problema de rearranjo em genomas ponderado por intercâmbio de blocos e reversões com sinais e projetamos um algoritmo para a sua solução. A Tabela 5 resume os dados de distância genômica e tempo de execução de cada algoritmo apresentado nesse capítulo.

Problema	Distância Ponderada	Tempo
Transposições Generalizadas	$\ \sigma\pi^{-1}\ _3$	$O(n)$
Fusões, Fissões e Reversões com Sinais	$\ \sigma\pi^{-1}\ /2$	$O(n)$
Intercâmbio de Blocos e Reversões com Sinais	$\ \sigma\pi^{-1}\ /2$	$O(n^2)$

Tabela 5.1: Distância e tempo de execução para problemas de rearranjo em genomas ponderados.

5.1 Distância de Transposição e Transposições Generalizadas

O problema de rearranjo em genomas por transposições consiste em encontrar uma sequência mínima de transposições que transforme um genoma outro. Como dito no Capítulo 1, diversos algoritmos aproximativos para esse problema foram propostos [5, 17]. O bem conhecido algoritmo aproximativo de Bafna e Pevzner [5] possui complexidade de tempo de execução $O(n^2)$ e um fator de aproximação de 1,5. Esse algoritmo de Bafna e Pevzner [5] é considerado muito complicado, então Christie [17] propôs um algoritmo diferente com o mesmo fator de aproximação, embora de complexidade de tempo de $O(n^4)$. Dois algoritmos mais simples, de complexidade de tempo subquadrática e fator de aproximação de 1,5 e 1,375 foram respectivamente propostos por Hartman e Shamir [30] e por Elias e Hartman [22]. Christie e Irving [16, 17] propuseram e resolveram o problema de rearranjo em genomas por intercâmbio de blocos, que é uma generalização de um evento de transposição no qual dois blocos não adjacentes de genes são trocados dentro de um cromossomo. Lin *et al.* [37] ofereceram um outro algoritmo para o problema de rearranjo em genomas envolvendo intercâmbio de blocos baseado numa versão do formalismo algébrico e aplicaram esse algoritmo no estudo comparativo de vibriões. Apesar de todo o sucesso na obtenção de soluções para problemas de rearranjo em genomas envolvendo outros eventos de rearranjo como por exemplo reversões com sinais [28, 32], a complexidade do problema de rearranjo em genomas por transposições é desconhecida até o momento, ou seja, não sabemos se há um algoritmo com tempo de execução polinomial que resolva o problema ou se o problema é *NP*-difícil.

Nessa seção apresentamos uma equivalência entre o limitante inferior para a distância de transposição do formalismo clássico e o do formalismo algébrico. Embora a demonstração da equivalência entre os limitantes inferiores seja extensa, a demonstração do limitante inferior oferecida pelo formalismo algébrico é simples e intuitiva. Além disso, apresentamos uma generalização do problema de rearranjo em genomas por transposições na qual permitimos além das transposições, outros eventos de rearranjo que são representados por 3-ciclos no formalismo algébrico: transposições generalizadas. Mostraremos como encontrar uma sequência mínima de transposições generalizadas que transforma um genoma em outro.

5.1.1 Limitantes para a Distância de Transposição

Genomas equi-orbitais π e σ no sistema de genes (E, Γ) possuem a mesma paridade uma vez que ambos possuem o mesmo número de órbitas o que implica em $\|\pi\| = \|\sigma\|$. Conseqüentemente $\|\sigma\pi^{-1}\|_3$ é bem definido pelo Lema 52. Usamos o Lema 52 para provar o seguinte limitante inferior para a distância de transposição baseada na 3-norma de $\sigma\pi^{-1}$.

Teorema 78 (Limitante Inferior). *Dados os genomas equi-orbitais π e σ em (E, Γ) , temos*

$$d_t(\pi, \sigma) \geq \frac{\|\sigma\pi^{-1}\|_3}{2}.$$

Prova: Seja $\tau_1, \tau_2, \dots, \tau_k$ uma seqüência de transposições que transforma a permutação π na permutação σ , ou seja, temos $\tau_k \dots \tau_1 \pi = \sigma$ e τ_i é aplicável a $\tau_{i-1} \dots \tau_1 \pi$ para $1 \leq i \leq k$. Reescrevendo a equação, obtemos $\tau_k \tau_{k-1} \dots \tau_1 = \sigma\pi^{-1}$. Logo o produto $\tau_k \dots \tau_1$ é uma decomposição em 3-ciclos de $\sigma\pi^{-1}$. Uma vez que π e σ têm a mesma paridade pelo fato de serem equi-orbitais então $\sigma\pi^{-1}$ é uma permutação par segundo o Lema 52. Então existe uma decomposição mínima em 3-ciclos de $\sigma\pi^{-1}$ tal que $\|\sigma\pi^{-1}\|_3 \leq 2k$ pois cada transposição é composta por dois 3-ciclos. Logo $d_\tau(\pi, \sigma) \geq (\|\sigma\pi^{-1}\|_3)/2$ é válido inclusive para uma seqüência mínima de transposições que transforme π em σ . \square

Considere $n = |E|$ nos próximos lemas. Mostraremos que o limitante inferior do Teorema 78 é equivalente ao limitante inferior para a distância de transposição proposto por Bafna e Pevzner [5], ou seja, temos $d_t(\pi, \sigma) \geq \frac{n - o_{\text{odd}}(\sigma\pi^{-1})}{4}$ usando os termos do formalismo algébrico. Destacamos duas observações sobre a equivalência do limitante inferior. O problema de rearranjo em genomas por transposições no formalismo clássico é um problema de ordenação de uma permutação na permutação identidade ι , enquanto o problema de rearranjo em genomas por transposições no formalismo algébrico é definido para um par de genomas quaisquer (compatíveis para transposições). Essa diferença na entrada de dados dos dois problemas é resolvida ao fixarmos σ . Os genomas na entrada do problema no formalismo clássico são unicromossomais, enquanto os genomas no formalismo algébrico são multicromossomais. Afirmamos que os limitantes sejam iguais para o caso de genomas unicromossomais, no entanto é possível estender esse resultado para genomas multicromossomais por meio da concatenação de cromossomos de um genoma no formalismo clássico [28].

Lema 79. Dado o 3-ciclo α e a permutação π sobre E , temos $o_{\text{odd}}(\pi\alpha^{-1}) \geq o_{\text{odd}}(\pi) - 2$.

Prova: Segundo o Lema 50, temos $o(\pi) - 2 \leq o(\pi\alpha^{-1}) \leq o(\pi) + 2$. Logo $o(\pi) - 2 \leq o(\pi\alpha^{-1})$. De forma análoga aos casos em que $o(\pi) - 2 < o(\pi\alpha^{-1})$, a variação no número de órbitas é maior que -2 , a variação no número de órbitas ímpares poderá ser negativa apenas nos casos em que o número de órbitas não varia ou o número de órbitas é reduzido de dois em $\pi\alpha^{-1}$ em relação a π . Considere $\alpha = (a \ b \ c)$. Os casos em que o número de órbitas de $\pi\alpha^{-1}$ pode

diminuir em relação a π são quando dois elementos no suporte de α pertencem a uma órbita distinta à do terceiro elemento na permutação π e quando os três elementos no suporte de α pertencem a órbitas distintas de π .

Suponhamos que a e b pertencem a mesma órbita, enquanto $c \notin \text{orb}(\pi, a)$. Se o menor inteiro não negativo k tal que $a = \pi^k b$ é ímpar, o menor inteiro não negativo j tal que $b = \pi^j a$ é par; ou seja, o tamanho de $\text{orb}(\pi, a)$ é ímpar; e o tamanho da órbita que contém c é ímpar, então pelo Teorema 48 as órbitas $\text{orb}(\pi\alpha^{-1}, a)$ e $\text{orb}(\pi\alpha^{-1}, b)$ são pares, enquanto as demais não são alteradas. Logo $o_{\text{odd}}(\pi\alpha^{-1}) = o_{\text{odd}}(\pi) - 2$. Em quaisquer outros casos, temos $o_{\text{odd}}(\pi\alpha^{-1}) \geq o_{\text{odd}}(\pi)$.

Se a, b e c pertencem a três órbitas distintas de π há quatro casos para análise:

1. Se as três órbitas são ímpares então a órbita em $\pi\alpha^{-1}$ que reúne essas três órbitas é ímpar e $o_{\text{odd}}(\pi\alpha^{-1}) = o_{\text{odd}}(\pi) - 2$.
2. Se duas das três órbitas são ímpares e a terceira é par então a órbita em $\pi\alpha^{-1}$ que reúne essas três órbitas é par e $o_{\text{odd}}(\pi\alpha^{-1}) = o_{\text{odd}}(\pi) - 2$.
3. Se duas das três órbitas são pares e a terceira é ímpar então a órbita resultante da união das três em $\pi\alpha^{-1}$ é ímpar e $o_{\text{odd}}(\pi\alpha^{-1}) = o_{\text{odd}}(\pi)$.
4. Se as três órbitas são pares então a união das três órbitas é par em $\pi\alpha^{-1}$ e $o_{\text{odd}}(\pi\alpha^{-1}) = o_{\text{odd}}(\pi)$.

Em todos os casos temos $o_{\text{odd}}(\pi\alpha^{-1}) \geq o_{\text{odd}}(\pi) - 2$. □

Lema 80. Seja π uma permutação sobre E , tal que $\|\pi\|$ é par e

$$\pi = \pi_1 \pi_2 \dots \pi_k$$

é a decomposição em 3-ciclos de π , na qual π_i é um 3-ciclo para $1 \leq i \leq k$, então

$$k \geq \frac{n - o_{\text{odd}}(\pi)}{2}$$

Prova: Iremos demonstrar a desigualdade por indução em k .

Para $k = 0$. Então $\pi = \iota$ e $o_{\text{odd}}(\pi) = n$. Logo $0 \geq \frac{n-n}{2} = 0$.

A hipótese de indução é para um k tal que $0 \leq k' < k$ e $\pi = \pi_1 \pi_2 \dots \pi_{k'}$, então $k' \geq \frac{n - o_{\text{odd}}(\pi)}{2}$.

Pela hipótese de indução, temos $k - 1 \geq \frac{n - o_{\text{odd}}(\pi')}{2}$ para $\pi' = \pi_1 \dots \pi_{k-1}$.

Temos também $o_{\text{odd}}(\pi) = o_{\text{odd}}(\pi' \pi_k) \geq o_{\text{odd}}(\pi') - 2$ segundo o Lema 79. Manipulando essas duas desigualdades:

$$\begin{aligned} k = (k-1) + 1 &\geq \frac{n - o_{\text{odd}}(\pi')}{2} + 1 \\ &= \frac{n - (o_{\text{odd}}(\pi') - 2)}{2} \\ &\geq \frac{n - o_{\text{odd}}(\pi)}{2}; \end{aligned}$$

obtemos

$$k \geq \frac{n - o_{\text{odd}}(\pi)}{2}.$$

□

Lema 81. Seja π uma permutação sobre E tal que $\|\pi\|$ é par, então $\|\pi\|_3 \leq \frac{n - o_{\text{odd}}(\pi)}{2}$, onde $n = |E|$.

Prova: Considere uma ordem dos ciclos na decomposição em ciclos disjuntos de π , tal que os ciclos de comprimento ímpar são agrupados no início do produto de ciclos; ou seja, temos $\pi = \alpha' \alpha''$, onde α' é um produto de ciclos ímpares de π e α'' é o produto de ciclos pares de π . Observe que essa ordenação é possível devido a comutatividade de ciclos disjuntos de uma permutação:

$$\pi = \underbrace{\alpha_1 \alpha_2 \dots \alpha_l}_{\alpha'} \underbrace{\alpha_{l+1} \alpha_{l+2} \dots \alpha_m}_{\alpha''}$$

Para mostrarmos que $\|\pi\|_3 \leq \frac{n - o_{\text{odd}}(\pi)}{2}$, construiremos uma decomposição em 3-ciclos de π por meio das decomposições em 2-ciclos de α' e α'' .

Seja $\alpha' = \alpha_1 \alpha_2 \dots \alpha_l$, onde α_i é um ciclo ímpar de π para $1 \leq i \leq l$. Denote $n' = |\text{Supp}(\alpha')|$ e $n'' = |\text{Supp}(\alpha'')|$. Observe que $\|\alpha'\| = n' - o_{\text{odd}}(\pi)$ e que $\|\alpha'\|$ é par, porque a norma de um ciclo ímpar é par, os ciclos em questão são disjuntos e pela Proposição 35. Pelo Teorema 58, temos $\|\alpha'\|_3 = \|\alpha'\|/2$. Logo $\|\alpha'\|_3 = \frac{n' - o_{\text{odd}}(\pi)}{2}$.

Considere agora uma decomposição em 2-ciclos de α'' utilizando-se o seguinte procedimento: substitua cada ciclo $(a_1 \dots a_r)$ para $r > 2$ de α'' pelo produto de 2-ciclos:

$$(a_1 a_2)(a_2 a_3)(a_3 a_4) \dots (a_{r-2} a_{r-1})(a_{r-1} a_r).$$

Como a norma de um ciclo par é ímpar e $\|\pi\|$ é par, então o número de ciclos de α'' ($o_{\text{even}}(\pi)$) é par. Seja $L(\alpha_i)$ o comprimento do ciclo α_i , para $l+1 \leq i \leq m$. Então a norma de α'' pode ser obtida do seguinte modo:

$$\|\alpha''\| = \sum_{i=l+1}^m (L(\alpha_i) - 1) = \left(\sum_{i=l+1}^m L(\alpha_i) \right) - (m - l) = n'' - o_{\text{even}}(\pi)$$

A análise da decomposição em 2-ciclos de α'' merece uma atenção especial. Como α'' é um produto de ciclos pares é possível que já existissem 2-ciclos nesse produto antes de obtermos a decomposição em 2-ciclos. Além disso, como a norma de um ciclo par é ímpar, o número de 2-ciclos obtidos a partir de um ciclo será ímpar e portanto ao parear-se 2-ciclos consecutivos no produto, existirão pares contendo 2-ciclos obtidos de ciclos distintos em α'' . Logo há pares de 2-ciclos consecutivos cuja interseção de seus suportes é vazia e pares cujos suportes possuem um elemento em comum. Chamaremos o primeiro tipo de par de 2-ciclos de *par disjunto*, enquanto o segundo tipo de par é chamado de *par não-disjunto*. Podemos obter um 3-ciclo de um par não-disjunto de maneira semelhante a utilizada para obter um 3-ciclo de um par de 2-ciclos $(a\ b)(b\ c)$ encontrados em α' . Por outro lado, um par disjunto $(a\ b)(c\ d)$ gera dois 3-ciclos $(a\ b\ c)(b\ c\ d)$. Logo cada par de 2-ciclos contribui com um 3-ciclo para a decomposição em 3-ciclos de α'' mais um número x de 3-ciclos, onde x é o número de pares disjuntos na decomposição em 2-ciclos de π . Então:

$$\|\alpha''\|_3 = \frac{n'' - o_{\text{even}}(\pi)}{2} + x$$

Como os 2-ciclos disjuntos são tomados aos pares, o valor de x pode estar no seguinte intervalo:

$$\left\lceil \frac{o_{\text{even}}(\alpha'') - 1}{2} \right\rceil \leq x \leq o_{\text{even}}(\alpha'') - 1$$

De fato, temos $x = \lceil \frac{o_{\text{even}}(\alpha'') - 1}{2} \rceil$, porque para cada par disjunto combinado para construir dois 3-ciclos, um par disjunto distinto não será combinado para construir dois 3-ciclos devido a paridade (ímpar) do número de 2-ciclos de α'' .

Logo, podemos obter a 3-norma de α'' do seguinte modo:

$$\begin{aligned} \|\alpha''\|_3 &= \frac{n'' - o_{\text{even}}(\pi)}{2} + \left\lceil \frac{o_{\text{even}}(\alpha'') - 1}{2} \right\rceil \\ &= \frac{n'' - o_{\text{even}}(\pi)}{2} + \left(\frac{o_{\text{even}}(\pi) - 2}{2} + 1 \right) \\ &= \frac{n''}{2} \end{aligned}$$

Como α' e α'' são permutações pares suas 3-normas são bem definidas e usando o Lema 55, tem-se:

$$\begin{aligned} \|\pi\|_3 &\leq \|\alpha'\|_3 + \|\alpha''\|_3 \\ &\leq \frac{n' - o_{\text{odd}}(\pi)}{2} + \frac{n''}{2} \\ &\leq \frac{n - o_{\text{odd}}(\pi)}{2} \end{aligned}$$

Portanto temos:

$$\|\pi\|_3 \leq \frac{n - o_{\text{odd}}(\pi)}{2}.$$

□

Usando os dois lemas anteriores (Lema 80 e Lema 81), obtemos uma fórmula para a 3-norma de uma permutação par π em função do número de órbitas de tamanho ímpar em π :

Teorema 82. *Para a permutação par π sobre E e $n = |E|$, temos*

$$\|\pi\|_3 = \frac{(n - o_{\text{odd}}(\pi))}{2}.$$

Um corolário imediato do Teorema 82 é uma fórmula equivalente ao limitante inferior proposto por Bafna e Pevzner [7] para a distância de transposição. É necessário fazer uma ressalva a respeito do que queremos dizer por equivalência entre os limitantes inferiores, pois a equivalência entre os conceitos usados no formalismo algébrico e no formalismo clássico não é obtida de maneira imediata.

Dados π e σ genomas unicromossomais e equi-orbitais em (E, Γ) onde $n = |E|$, escolhemos um elemento arbitrário x em uma das órbitas de σ . Definimos a *função de mapeamento de genes em inteiros* $f : E \rightarrow \mathbb{N}$ tal que $f(\sigma^l x) = l + 1$ para $0 \leq l \leq |\text{orb}(\sigma, x)| - 1$. A função f é bem definida pois l é menor do que o tamanho da órbita que contém x . A sequência $f(\sigma^0 x) \dots f(\sigma^{|\text{orb}(\sigma, x)|-1} x)$ é uma permutação com sinais que mapeia $\{1, \dots, |\text{orb}(\sigma, x)|\}$ em $\{z \in \mathbb{N} \mid z = l + 1 \text{ para } 0 \leq l \leq |\text{orb}(\sigma, x)| - 1\}$.

Definimos os genomas Σ e Π no formalismo clássico obtidos de uma *transformação de órbitas em permutações com sinais* como os conjuntos das permutações com sinais obtidas das transformações aplicadas a órbita de σ e π que contém o elemento x usando-se a mesma função f de mapeamento de genes em inteiros para (E, Γ) para ambos os genomas.

Por exemplo, considere os genomas $\pi = (x \ z \ y \ s \ r \ t)(-t \ -r \ -s \ -y \ -z \ -x)$ e $\sigma = (x \ y \ z \ r \ s \ t)(-t \ -s \ -r \ -z \ -y \ -x)$ no sistema de genes (E, Γ) , onde $E = \{x, y, z, r, s, t\}$ e $\Gamma w = -w$ para $w \in E$ no formalismo algébrico. Escolhemos o elemento x para definir f . A função f possui os seguintes mapeamentos:

$$f(x) = 1, f(y) = 2, f(z) = 3, f(r) = 4, f(s) = 5, f(t) = 6.$$

Os genomas equivalentes no formalismo clássico são:

$$\Pi = \{[1, 3, 2, 5, 4, 6]\}$$

e

$$\Sigma = \{[1, 2, 3, 4, 5, 6]\}.$$

Por meio da transformação de genomas no formalismo algébrico em genomas no formalismo clássico é possível demonstrar a igualdade dos limitantes inferiores para a distância de transposição em ambos os formalismos.

Teorema 83. *Dados os genomas π e σ em (E, Γ) e os genomas Π e Σ com n' genes obtidos por meio de uma transformação de π e σ usando uma função f , temos*

$$\frac{n' + 1 - c_{\text{odd}}(\Pi)}{2} = \frac{(|E| - o_{\text{odd}}(\sigma\pi^{-1}))}{4}.$$

Prova: Como os genomas no formalismo algébrico representam os genes e suas seqüências complementares reversas correspondentes de um cromossomo circular o número de elementos de $|E|$ é o dobro de $n' + 1$, o número de genes em um genoma no formalismo clássico adicionado de um elemento devido ao cromossomo representado no formalismo clássico ser linear. Além disso, cada par no quociente $\sigma\pi^{-1}$ corresponde a um ciclo no diagrama de ciclos de Π . Como cada par é um produto de dois ciclos, o número de órbitas de $\sigma\pi^{-1}$ é o dobro do número de ciclos no diagrama de ciclos e temos $o_{\text{odd}}(\sigma\pi^{-1}) = 2c_{\text{odd}}(\Pi)$. Logo $|E| - o_{\text{odd}}(\sigma\pi^{-1}) = 2(n' + 1 - c_{\text{odd}}(\Pi))$. Portanto, dados os genomas π e σ em (E, Γ) e os genomas Π e Σ com n' genes obtidos por meio de uma transformação de π e σ usando uma função f , temos

$$\frac{n' + 1 - c_{\text{odd}}(\Pi)}{2} = \frac{(|E| - o_{\text{odd}}(\sigma\pi^{-1}))}{4}.$$

□

Limitante baseado em Pontos de Quebra

Dados os genomas π e σ no sistema de genes (E, Γ) , um elemento $x \in E$ é chamado de um *ponto de quebra* para (π, σ) quando $x \in \text{Supp}(\sigma\pi^{-1})$. Os pontos de quebra aparecem em pares $x, \pi\Gamma x$ em $\sigma\pi^{-1}$. O número de pares de pontos de quebra em $\sigma\pi^{-1}$ é denotado por $b(\pi, \sigma)$. A variação no número de pares de pontos de quebra $b(\tau\pi, \sigma) - b(\pi, \sigma)$ é denotada por $\Delta b(\tau, \pi, \sigma)$. Uma transposição τ aplicável a π pode criar ou eliminar no máximo 3 pares de pontos de quebra, ou seja, temos $-3 \leq \Delta b(\tau, \pi, \sigma) \leq 3$. Bafna e Pevzner [7] modelam genomas por meio de apenas uma fita e por isso os pontos de quebra não aparecem aos pares em sua teoria. Como nos referimos ao número de pares de pontos de quebra ao escrevermos $b(\pi, \sigma)$, esse parâmetro possui o mesmo valor numérico em ambos os formalismos. Bafna e Pevzner demonstraram que o número de pares de pontos de quebra $b(\pi, \sigma)$ pode ser usado como um parâmetro para um limitante inferior da distância de transposição entre π e σ . O limitante inferior para a distância de transposição baseada no número de ciclos ímpares (ou, equivalentemente, na 3-norma do quociente no formalismo algébrico) apresenta um valor mais justo do que o limitante baseado em pontos de quebra [7]. Por esse motivo, parece ter havido pouca investigação sobre quais pares de genomas satisfazem o limitante baseado em pontos de quebra. Para melhor compreendermos o comportamento de transposições que afetam o número de pontos de quebra de um par de genomas, investigamos o limitante inferior baseado em pontos de quebra e conseguimos uma caracterização dos genomas que o satisfazem.

Proposição 84 (Bafna e Pevzner). Dados os genomas π e σ , temos

$$d_t(\pi, \sigma) \geq \frac{b(\pi, \sigma)}{3}.$$

No Lema 85, caracterizamos quais são os pares de genomas em um sistema de genes cuja distância de transposição entre eles satisfaz o limitante inferior baseado em pontos de quebra.

Lema 85. Dados os genomas π e σ em (E, Γ) , temos que $d_t(\pi, \sigma) = \frac{b(\pi, \sigma)}{3}$ se e somente se $d_t(\pi, \sigma) = (\|\sigma\pi^{-1}\|_3)/2$ e $\sigma\pi^{-1}$ é um produto de 3-ciclos disjuntos.

Prova: Considere primeiramente os genomas π e σ tais que $\sigma\pi^{-1}$ é um produto de $2k$ 3-ciclos disjuntos e $d_t(\pi, \sigma) = (\|\sigma\pi^{-1}\|_3)/2$. Logo $\|\sigma\pi^{-1}\|_3 = 2k$ e portanto $d_t(\pi, \sigma) = (\|\sigma\pi^{-1}\|_3)/2 = k$.

Como os $2k$ ciclos da decomposição em ciclos de $\sigma\pi^{-1}$ são disjuntos, então o tamanho do suporte de $\sigma\pi^{-1}$ é $6k$. O número de pontos de quebra é $b(\pi, \sigma) = |Supp(\sigma\pi^{-1})|/2 = 3k$. Logo:

$$d_t(\pi, \sigma) = k = \frac{3k}{3} = \frac{b(\pi, \sigma)}{3}.$$

Portanto, se $d_t(\pi, \sigma) = (\|\sigma\pi^{-1}\|_3)/2$ e $\sigma\pi^{-1}$ é um produto de 3-ciclos disjuntos, então $d_t(\pi, \sigma) = \frac{b(\pi, \sigma)}{3}$.

Considere agora que a distância de transposição de π a σ seja $d_t(\pi, \sigma) = \frac{b(\pi, \sigma)}{3}$. Logo existe uma seqüência mínima de transposições τ_1, \dots, τ_k tal que $\tau_k \tau_{k-1} \dots \tau_1 \pi = \sigma$ e $k = \frac{b(\pi, \sigma)}{3}$; ou seja, cada τ_i , para $1 \leq i \leq k$, elimina 3 pares de pontos de quebra. Além disso, como $b(\pi, \sigma) = |Supp(\sigma\pi^{-1})|/2$, então o suporte de $\sigma\pi^{-1} \tau_1^{-1} \dots \tau_i^{-1}$ tem seu tamanho reduzido de 6 em relação a $\sigma\pi^{-1} \tau_1^{-1} \dots \tau_{i-1}^{-1}$, ou, em outros termos, seis novos elementos fixos são criados em $\sigma\pi^{-1} \tau_1^{-1} \dots \tau_{i-1}^{-1} \tau_i^{-1}$. Pelo Teorema 48, o resultado de um produto de uma permutação qualquer θ por um 3-ciclo α é uma permutação com 3 novos elementos fixos apenas quando α^{-1} é um dos ciclos de θ . Logo para cada transposição τ_i para $1 \leq i \leq k$ em uma seqüência mínima de transposições que transforma π em σ existe um produto de dois 3-ciclos igual a τ_i em $\sigma\pi^{-1}$ e portanto $\sigma\pi^{-1}$ é um produto de $2k$ 3-ciclos. Como cada elemento que se torna fixo não é mais transformado em um ponto de quebra após a aplicação das transposições, então todos os ciclos de $\sigma\pi^{-1}$ são disjuntos entre si. Portanto $\sigma\pi^{-1} = \tau_k \tau_{k-1} \dots \tau_1$ e pelo Teorema 82 deduzimos

$$\begin{aligned} \|\sigma\pi^{-1}\|_3 &= \frac{|E| - o_{odd}(\sigma\pi^{-1})}{2} \\ &= \frac{|Supp(\sigma\pi^{-1})| - |E/Supp(\sigma\pi^{-1})| - (2k + |E/Supp(\sigma\pi^{-1})|)}{2} \\ &= \frac{6k - 2k}{2} \\ &= 2k \\ &= 2 \frac{b(\pi, \sigma)}{3} \\ &= 2d_t(\pi, \sigma). \end{aligned}$$

Portanto, se $d_t(\pi, \sigma) = \frac{b(\pi, \sigma)}{3}$, então $d_t(\pi, \sigma) = (\|\sigma\pi^{-1}\|_3)/2$ e $\sigma\pi^{-1}$ é um produto de 3-ciclos disjuntos. \square

O Lema 85 caracteriza os pares de genomas π e σ cuja distância de transposição satisfaz o limitante inferior de pontos de quebra. O Exemplo 86 mostra alguns pares de genomas que satisfazem as condições do Lema 85.

Exemplo 86. Considere os genomas $\pi = (1\ 3\ 5\ 4\ 6\ 2)(-2\ -6\ -4\ -5\ -3\ -1)$ e $\sigma = (1\ 2\ 3\ 4\ 5\ 6)(-6\ -5\ -4\ -3\ -2\ -1)$ no sistema de genes (E, Γ) , onde $E = \{x \in \mathbb{Z} \mid 1 \leq |x| \leq 6\}$ e $\Gamma x = -x$ para $x \in E$. Observe que:

$$\sigma\pi^{-1} = (1\ 3\ 2)(-6\ -1\ -2)(4\ 6\ 5)(-3\ -4\ -5).$$

Todos os 3-ciclos que compõem $\sigma\pi^{-1}$ são transposições e além disso a aplicação de uma dessas transposições reduz de 3 o número de pares de pontos de quebra. Naturalmente $\|\sigma\pi^{-1}\|_3 = 4$ e os 3-ciclos de $\sigma\pi^{-1}$ são disjuntos. Logo $d_t(\pi, \sigma) = (\|\sigma\pi^{-1}\|_3)/2$ e portanto $d_t(\pi, \sigma) = (b(\pi, \sigma))/3$.

O genoma unicromossomal π no sistema de genes (E, Γ) , onde $E = \{x \in \mathbb{Z} \mid 1 \leq |x| \leq n\}$ tal que n é múltiplo de 3 e $\Gamma x = -x$ para $x \in E$, que obedece ao formato

$$\begin{aligned} \pi &= (2\ 1\ 3\ 5\ 4\ 6\ \dots\ k+1\ k\ k+2\ \dots\ n-1\ n-2\ n) \\ &\quad (-n\ -n+2\ -n+1\ \dots\ -k-2\ -k\ -k-1\ \dots\ -6\ -4\ -5\ -3\ -1\ -2) \end{aligned}$$

para $1 \leq k \leq n/3$ é um genoma cuja distância de transposição ao genoma

$$\sigma = (1\ \dots\ n-1\ n)(-n\ -n+1\ \dots\ -1)$$

é $d_t(\pi, \sigma) = b(\pi, \sigma)/3$, pois o quociente

$$\sigma\pi^{-1} = (2\ 1\ 3)(-1\ -2\ -n) \dots (n-1\ n-2\ n)(-n+2\ -n+1\ -n+3)$$

é um produto de 3-ciclos disjuntos e a sequência

$$(2\ 1\ 3)(-1\ -2\ -n)(5\ 4\ 6)(-4\ -5\ -3) \dots (n-1\ n-2\ n)(-n+2\ -n+1\ -n+3)$$

de transposições transforma π em σ .

Por outro lado, observe que dados os genomas π e σ o fato do quociente $\sigma\pi^{-1}$ ser composto por um produto de 3-ciclos disjuntos apenas, não implica que $d_t(\pi, \sigma) = \frac{b(\pi, \sigma)}{3}$. Para $\pi = (1\ 6\ 5\ 4\ 3\ 2)(-2\ -3\ -4\ -5\ -6\ -1)$ e $\sigma = (1\ 2\ 3\ 4\ 5\ 6)(-6\ -5\ -4\ -3\ -2\ -1)$ temos:

$$\sigma\pi^{-1} = (1\ 3\ 5)(-6\ -4\ -2)(2\ 4\ 6)(-1\ -5\ -3),$$

porém a distância de π a σ é $d_t(\pi, \sigma) = 3$.

5.1.2 Propriedades do Grafo de Transposições Boas

Nessa seção discutiremos a relação entre genomas equijorbitais pertencentes a um sistema de genes segundo a existência de seqüências de determinados tipos de transposições que transformam um genoma em outro.

Dados os genomas π e σ no sistema de genes (E, Γ) , o genoma π é chamado de *facilmente transformável* em σ quando a sua distância de transposição $d_t(\pi, \sigma)$ é igual a um limitante inferior que possa ser calculado em tempo de execução polinomial. Omitiremos a menção a σ ao discutirmos genomas facilmente transformáveis quando σ puder ser deduzido do contexto. Focamos a atenção em genomas facilmente transformáveis que satisfaçam ao limitante inferior de pontos de quebra e a o limitante inferior de decomposição em 3-ciclos discutidos na Seção 5.1.1. Para alguns problemas de rearranjo em genomas relacionados ao problema de rearranjo em genomas por transposições, tais como os problemas envolvendo reversões sem sinais e transposições de prefixo, que são respectivamente *NP*-difícil e de complexidade desconhecida, as permutações facilmente transformáveis foram caracterizadas [57, 21] para o caso especial em que os genomas são unicromossomais.

Para caracterizar os genomas que são facilmente transformáveis, definimos um grafo cujos vértices são gerados a partir do conjunto de genomas de um sistema de genes e as arestas são representadas pelas transposições aplicáveis a esses genomas que atendam a certos critérios de divisibilidade. Apresentamos algumas das principais propriedades desse grafo. Por exemplo, demonstramos que esse grafo é bipartido e apresentamos um isomorfismo entre grafos definidos para diferentes genomas σ . Mostraremos que o conjunto de genomas facilmente transformáveis é um subconjunto dos vértices de um dos componentes desse grafo.

Dados os genomas π e σ no sistema de genes (E, Γ) , um evento de rearranjo τ aplicável a π é um *evento 3-bom* quando $\tau|_3\sigma\pi^{-1}$. Para o caso especial no qual os genomas são unicromossomais e equicromossomais, uma transposição τ aplicável a π é uma *transposição boa* quando τ é um evento 3-bom.

Exemplo 87. Considere os genomas do Exemplo 86:

$$\pi = (1\ 3\ 5\ 4\ 6\ 2)(-2\ -6\ -4\ -5\ -3\ -1)$$

e

$$\sigma = (1\ 2\ 3\ 4\ 5\ 6)(-6\ -5\ -4\ -3\ -2\ -1)$$

no sistema de genes (E, Γ) , onde $E = \{x \in \mathbb{Z} \mid 1 \leq |x| \leq 6\}$ e $\Gamma x = -x$ para $x \in E$. O quociente de π e σ é:

$$\sigma\pi^{-1} = (1\ 3\ 2)(-6\ -1\ -2)(4\ 6\ 5)(-3\ -4\ -5).$$

A transposição $\tau = (1\ 3\ 2)(-6\ -1\ -2)$ aplicável a π é uma transposição boa pois

$$\sigma\pi^{-1}\tau^{-1} = (4\ 6\ 5)(-3\ -4\ -5)$$

e, logo $\|\sigma\pi^{-1}\tau^{-1}\|_3 = \|\sigma\pi^{-1}\|_3 - \|\tau\|_3$

Definição 88 (Grafo de Transposições Boas). Seja $U(\sigma)$ o conjunto de genomas eqüicromossomais ao genoma unicromossomal σ em (E, Γ) . O *grafo de transposições boas* de σ , denotado por $G(\sigma)$, consiste no grafo cujo conjunto de vértices é $U(\sigma)$ e dados dois vértices π_1 e π_2 há uma aresta orientada de π_1 para π_2 quando houver uma transposição τ aplicável a π_1 tal que $\tau|_3\sigma\pi_1^{-1}$ e $\pi_2 = \tau\pi_1$.

A Figura 5.1 mostra um exemplo de grafo de transposições boas para $\sigma = (1\ 2\ 3\ 4)(-4\ -3\ -2\ -1)$ no sistema de genes (E, Γ) , onde $E = \{x \in \mathbb{Z} \mid 1 \leq x \leq 4\}$ e $\Gamma x = -x$ para $x \in E$.

Lema 89. Dados os genomas eqüiorbitais σ , α e β em (E, Γ) , se houver uma aresta de α a β em $G(\sigma)$, então $\|\sigma\beta^{-1}\|_3 = \|\sigma\alpha^{-1}\|_3 - 2$.

Prova: Se houver uma aresta de α para β , então $\beta = \tau\alpha$ e $\tau|_3\sigma\alpha^{-1}$. Logo, substituindo $\tau\alpha$ em β

$$\|\sigma\beta^{-1}\|_3 = \|\sigma\alpha^{-1}\tau^{-1}\|_3 = \|\sigma\alpha^{-1}\|_3 - \|\tau\|_3 = \|\sigma\alpha^{-1}\|_3 - 2,$$

obtemos $\|\sigma\beta^{-1}\|_3 = \|\sigma\alpha^{-1}\|_3 - 2$. □

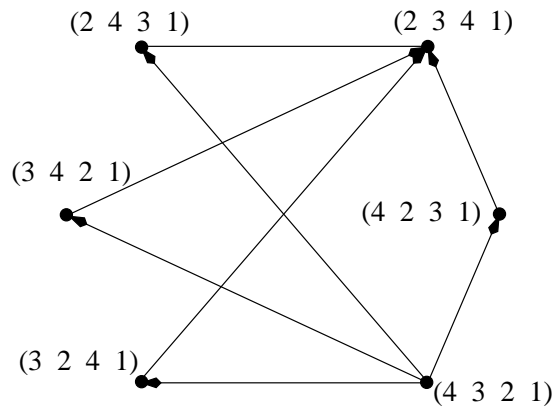


Figura 5.1: Grafo de Transposições Boas do genoma $\sigma = (1\ 2\ 3\ 4)(-4\ -3\ -2\ -1)$ onde $E = \{x \in \mathbb{Z} \mid 1 \leq x \leq 4\}$ e $\Gamma x = -x$ para $x \in E$. O conjunto de vértices de $G(\sigma)$ é formado pelos genomas eqüiorbitais a σ no sistema de genes (E, Γ) . Para facilitar a compreensão da figura, mostramos apenas uma das fitas de cada cromossomo.

Lema 90. Dados os genomas eqüiorbitais π e σ em (E, Γ) , qualquer caminho de π a σ em $G(\sigma)$ tem comprimento $\frac{\|\sigma\pi^{-1}\|_3}{2}$.

Prova: Considere o caminho $P = \alpha_0, \alpha_1, \dots, \alpha_k$ tal que $\alpha_1 = \pi$ e $\alpha_k = \sigma$ cujo comprimento é k . Pelo Lema 89, temos $\|\sigma\alpha_{i+1}^{-1}\|_3 = \|\sigma\alpha_i^{-1}\|_3 - 2$ para $0 \leq i \leq k-1$. Então aplicando o Lema 89 para cada vértice no caminho P , obtemos

$$\|\sigma\alpha_k^{-1}\|_3 = \|\sigma\alpha_{k-1}^{-1}\|_3 - 2 = \|\sigma\alpha_0^{-1}\|_3 - 2k.$$

Como $\|\sigma\alpha_k^{-1}\|_3 = \|\sigma\sigma^{-1}\|_3 = 0$ e $\|\sigma\alpha_0^{-1}\|_3 = \|\sigma\pi^{-1}\|_3$. Então, temos $\|\sigma\sigma^{-1}\|_3 = \|\sigma\pi^{-1}\|_3 - 2k$, e logo $k = (\|\sigma\pi^{-1}\|_3)/2$. Portanto, o comprimento do caminho de π a σ é $(\|\sigma\pi^{-1}\|_3)/2$.

□

Lema 91. Dados os genomas equi-orbitais π e σ no sistema de genes (E, Γ) , há um caminho de π a σ em $G(\sigma)$ se e somente se $d_t(\pi, \sigma) = \frac{\|\sigma\pi^{-1}\|_3}{2}$.

Prova: Suponha que há um caminho $\alpha_0, \dots, \alpha_k$ em $G(\sigma)$ tal que $\alpha_0 = \pi$ e $\alpha_k = \sigma$. Pelo Lema 90, temos $k = (\|\sigma\pi^{-1}\|_3)/2$. Pela definição de $G(\sigma)$, temos $\alpha_{i+1} = \tau_i\alpha_i$ e τ_i é uma transposição boa aplicável a α_i , para $0 \leq i \leq k-1$. Logo, temos $\tau_{k-1} \dots \tau_0\pi = \sigma$. Pelo Teorema 78, temos $d_t(\pi, \sigma) \geq (\|\sigma\pi^{-1}\|_3)/2$ e como $\tau_{k-1}, \dots, \tau_0$ é uma seqüência de transposições que transforma π em σ , então $d_t(\pi, \sigma) = (\|\sigma\pi^{-1}\|_3)/2$.

Inversamente, suponha que $d_t(\pi, \sigma) = (\|\sigma\pi^{-1}\|_3)/2$. Seja τ_1, \dots, τ_k uma seqüência mínima de transposições tal que $\sigma = \tau_k \tau_{k-1} \dots \tau_1\pi$ e τ_i é aplicável a $\tau_{i-1} \dots \tau_1\pi$ para $1 \leq i \leq k$, onde $k = (\|\sigma\pi^{-1}\|_3)/2$. Como as transposições são aplicáveis e π é um genoma, então cada $\tau_i \dots \tau_1\pi$ é um genoma para $1 \leq i \leq k$ e logo um vértice de $G(\sigma)$. Devemos mostrar agora que $\tau_i|_3\sigma\pi^{-1}\tau_1^{-1} \dots \tau_{i-1}^{-1}$ para $1 \leq i \leq k$ para demonstrar que há uma aresta de $\tau_{i-1} \dots \tau_1\pi$ a $\tau_i \dots \tau_1\pi$. Para $1 \leq i \leq k$, deduzimos

$$\begin{aligned} \|\sigma\pi^{-1}\tau_1^{-1} \dots \tau_i^{-1}\|_3 &= \|\tau_k \dots \tau_1\tau_1^{-1} \dots \tau_i^{-1}\|_3 \\ &= \|\tau_k \tau_{k-1} \dots \tau_{i+1}\|_3 \\ &= 2(k-i). \end{aligned}$$

Por outro lado, temos:

$$\|\sigma\pi^{-1}\tau_1^{-1} \dots \tau_{i-1}^{-1}\|_3 - \|\tau_i\|_3 = 2(k - (i-1)) - 2 = 2(k-i)$$

Logo, para $1 \leq i \leq k$, temos $\tau_i|_3\sigma\pi^{-1}\tau_1^{-1} \dots \tau_{i-1}^{-1}$. Portanto, a seqüência de genomas

$$\pi, \tau_1\pi, \tau_2\tau_1\pi, \dots, \sigma$$

é um caminho de π a σ .

□

Lema 92. Dado o genoma σ em (E, Γ) , o grafo $G(\sigma)$ não possui ciclos orientados.

Prova: Suponha que exista um ciclo orientado em $G(\sigma)$. Seja π um vértice que pertence ao ciclo.

Pela definição de $G(\sigma)$, há um caminho $\alpha_0, \dots, \alpha_k$, para $\alpha_0 = \alpha_k = \pi$. Além disso, também pela definição de $G(\sigma)$, existe uma seqüência de transposições tal que $\alpha_i = \tau_i \alpha_{i-1}$ para $1 \leq i \leq k$. Logo as transposições $\tau_1, \tau_2, \dots, \tau_k$, para $k > 0$, são tais que

$$\tau_k \tau_{k-1} \dots \tau_1 \pi = \pi$$

Pelo Lema 89, se há uma aresta de um vértice α para o vértice β , então há uma transposição τ aplicável a α tal que

$$\|\sigma\beta^{-1}\|_3 = \|\sigma\alpha^{-1}\|_3 - 2.$$

Logo $\|\sigma(\tau_i \dots \tau_1 \pi)^{-1}\|_3 - \|\sigma(\tau_{i-1} \dots \tau_1 \pi)^{-1}\|_3 = -2$ para $1 \leq i \leq k$. Conseqüentemente, temos

$$\|\sigma(\tau_k \tau_{k-1} \dots \tau_1 \pi)^{-1}\|_3 - 2k = \|\sigma\pi^{-1}\|_3,$$

ou equivalentemente $\|\sigma\pi^{-1}\|_3 - 2k = \|\sigma\pi^{-1}\|_3$, ou seja, temos $k = 0$. Mas essa igualdade contradiz a hipótese de $k > 0$. Portanto o grafo $G(\sigma)$ não possui ciclos orientados. \square

Teorema 93. Dado σ em (E, Γ) , o grafo de transposições boas $G(\sigma)$ é bipartido.

Prova: Definimos a seguinte bipartição de $U(\sigma)$:

$$V_1 = \{\pi \in U(\sigma) \mid \|\sigma\pi^{-1}\|_3 \equiv 0 \pmod{2}\}$$

e

$$V_2 = \{\pi \in U(\sigma) \mid \|\sigma\pi^{-1}\|_3 \equiv 1 \pmod{2}\}.$$

Observe que o conjunto V_1 não é vazio pois possui o vértice σ (lembrar que $\|\sigma\sigma^{-1}\|_3 = 0$).

Mostraremos que quaisquer dois vértices que pertençam a uma das partições V_1 ou V_2 não são ligados por uma aresta.

Sejam α e β dois vértices que pertençam a uma partição, digamos V_1 . Pela definição anterior da bipartição, afirmamos que $\|\sigma\alpha^{-1}\|_3 \equiv \|\sigma\beta^{-1}\|_3 \pmod{2}$. Logo $\|\sigma\alpha^{-1}\|_3 \neq \|\sigma\beta^{-1}\|_3 \pm 2$ e usando a contrapositiva do Lema 89 não há nenhuma aresta entre α e β . \square

Dado o genoma σ em $U(\sigma)$, o grafo de transposições boas $G(\sigma)$ é bipartido. Além disso, não é difícil verificar que se existe uma aresta de um genoma α para um genoma β em $G(\sigma)$, então $\|\sigma\beta^{-1}\|_3 = \|\sigma\alpha^{-1}\|_3 - 2$. Esse fato básico a respeito de grafos de transposições boas é fundamental para deduzir as propriedades que seguem abaixo.

Lema 94. Dados os genomas equiorbitais σ e π em (E, Γ) , os grafos $G(\sigma)$ e $G(\pi)$ são isomorfos.

Prova: Dados os genomas equiorbitais π e $\sigma \in U(\sigma)$ (como π e σ são equiorbitais então $U(\sigma) = U(\pi)$), tome uma permutação α tal que $\alpha \cdot \sigma = \pi$. Observe que a permutação α existe devido a σ e π possuírem ambos a mesma estrutura de ciclos (são equiorbitais) e os mesmos elementos, ainda que em ordem diversa. Defina a função $f : U(\sigma) \rightarrow U(\sigma)$ como $f(\theta) = \alpha \cdot \theta$ para $\theta \in U(\sigma)$. Como a conjugação sobre uma permutação mantém a sua estrutura de ciclos, então $f(\theta) \in U(\sigma)$.

Suponha que (θ_1, θ_2) seja uma aresta de $G(\sigma)$. Logo por definição de grafo de transposições boas $\theta_2 = \tau\theta_1$ e $\tau|_3\sigma\theta_1^{-1}$, onde τ é uma transposição aplicável em θ_1 . Considere os genomas $f(\theta_1) = \alpha\theta_1\alpha^{-1}$ e $\alpha\theta_2\alpha^{-1}$. Iremos mostrar que existe uma transposição τ^* aplicável a $f(\theta_1)$ tal que $f(\theta_2) = \tau^*f(\theta_1)$ e $\tau^*|_3\pi(f(\theta_1))^{-1}$. Tomemos

$$\tau^* = f(\theta_2)(f(\theta_1))^{-1} = \alpha\theta_2\alpha^{-1}\alpha\theta_1^{-1}\alpha^{-1} = \alpha\theta_2\theta_1^{-1}\alpha^{-1} = \alpha \cdot \tau.$$

Logo τ^* é uma transposição e como $f(\theta_2) = \tau^*f(\theta_1)$ e ambas as permutações $f(\theta_2)$ e $f(\theta_1)$ são genomas, então τ^* é uma transposição aplicável a $f(\theta_1)$. Basta provar que $\tau^*|_3\pi(f(\theta_1))^{-1}$. Pelo Lema 56 temos

$$\tau|_3\sigma\theta_1^{-1} \Leftrightarrow \alpha \cdot \tau|_3\alpha \cdot (\sigma\theta_1^{-1}) \Leftrightarrow \tau^*|_3(\alpha \cdot \sigma)(\alpha \cdot \theta_1^{-1}) \Leftrightarrow \tau^*|_3\pi(\alpha \cdot \theta_1)^{-1} \Leftrightarrow \tau^*|_3\pi(f(\theta_1))^{-1}.$$

Logo (θ_1, θ_2) é uma aresta de $G(\sigma)$ se e somente se $(f(\theta_1), f(\theta_2))$ é uma aresta de $G(\pi)$. Portanto f é um isomorfismo entre $G(\pi)$ e $G(\sigma)$, i.e. os grafos $G(\pi)$ e $G(\sigma)$ são isomorfos. \square

Chamamos de σ -componente ao componente de $G(\sigma)$ que possui σ entre os seus vértices. Pelo Lema 91, todo genoma facilmente transformável pertence ao σ -componente de $G(\sigma)$. No entanto, nem todos os genomas unicromossomais no σ -componente satisfazem ao limitante inferior da decomposição em 3-ciclos. Observe que o grafo $G(\sigma)$ não possui ciclos orientados e que σ é um sumidouro em $G(\sigma)$. No entanto, observe que um componente de $G(\sigma)$ pode ter mais do que um um vértice sumidouro. Considere, por exemplo, os genomas

$$\pi = (a \ i \ h \ g \ f \ e \ d \ c \ b)(\Gamma b \ \Gamma c \ \Gamma d \ \Gamma e \ \Gamma f \ \Gamma g \ \Gamma h \ \Gamma i \ \Gamma a)$$

e

$$\sigma = (a \ b \ c \ d \ e \ f \ g \ h \ i)(\Gamma i \ \Gamma h \ \Gamma g \ \Gamma f \ \Gamma e \ \Gamma d \ \Gamma c \ \Gamma b \ \Gamma a);$$

então

$$\sigma\pi^{-1} = (a \ c \ e \ g \ i \ b \ d \ f \ h)(\Gamma h \ \Gamma f \ \Gamma d \ \Gamma b \ \Gamma i \ \Gamma g \ \Gamma e \ \Gamma c \ \Gamma a).$$

Há apenas três transposições aplicáveis a π tais que $\tau|_3\sigma\pi^{-1}$:

$$(a \ g \ d)(\Gamma e \ \Gamma h \ \Gamma b), \ (c \ i \ f)(\Gamma g \ \Gamma a \ \Gamma d) \text{ e } (e \ b \ h)(\Gamma i \ \Gamma c \ \Gamma d).$$

Seja τ qualquer uma das transposições anteriores aplicáveis a π , o genoma $\tau\pi$ é um sumidouro em $G(\sigma)$.

n	Número de Sumidouros	$ U(\sigma) $	Proporção
3	1	2	50,00%
4	1	6	16,67%
5	2	24	8,33%
6	8	120	6,67%
7	30	720	4,58%
8	109	5040	2,16%

Tabela 5.2: Proporção de sumidouros no grafo de transposições boas $G(\sigma)$.

O σ -componente pode também ter mais do que um sumidouro o que implica poder haver genomas para os quais não há um caminho que os ligue a σ no grafo $G(\sigma)$.

A Tabela 5.2 resume a análise da proporção de vértices sumidouros em $G(\sigma)$, para $\sigma = (1\ 2\ \dots\ n)(-n\ \dots\ -2\ -1)$, em comparação com o número total de vértices no grafo.

O experimento foi conduzido por meio de uma implementação das principais funções referentes a manipulação de permutações e do grafo de 2-movimentos válidos na ferramenta *Mathematica* [60]. Utilizou-se um processador Pentium III 800MHz, memória de cache de 256Kb e memória principal de 64Mb. A partir de $n \geq 9$, o tempo de execução necessário para encontrar todos os sumidouros do grafo é demasiadamente grande para a configuração do computador utilizada e por isso não obtemos dados para esses valores.

Pode-se observar a partir dos experimentos realizados que a proporção de sumidouros parece tender a zero à medida que a ordem do grafo $G(\sigma)$ é aumentada. Esse comportamento pode ser útil para a identificação das características comuns de genomas para os quais não há uma transposição boa aplicável.

Genomas Facilmente Transformáveis

Uma maneira de caracterizar os genomas facilmente transformáveis baseia-se em utilizar a propriedade de “atingibilidade” de σ a partir desses genomas. Dado o grafo $G(\sigma)$ e dois genomas π_1 e π_2 de $U(\sigma)$, dizemos que π_2 é *atingível* a partir de π_1 (ou também que π_1 *atinge* π_2) quando houver um caminho de π_1 a π_2 em $G(\sigma)$. O conjunto de genomas atingíveis a partir de π em $G(\sigma)$ é denotado por

$$R(\pi) = \{\pi' \in U(\sigma) \mid \pi' \text{ é atingível a partir de } \pi\}.$$

Lema 95. Dados os genomas equiorbitais e unicromossomais π e σ em (E, Γ) , o genoma π é facilmente transformável em σ se e somente se $\sigma \in R(\pi)$ para o grafo $G(\sigma)$.

Prova: Se $\sigma \in R(\pi)$ então existe um caminho orientado de π a σ em $G(\sigma)$. Logo, pelo Lema 91, temos $d_t = \|\sigma\pi^{-1}\|_3/2$. O valor $\|\sigma\pi^{-1}\|_3$ pode ser obtido em tempo polinomial

no número de elementos de E por meio de uma contagem do número de órbitas ímpares de $\sigma\pi^{-1}$. A contagem do número de órbitas pode ser realizada construindo-se cada órbita de $\sigma\pi^{-1}$; por exemplo, utilizando o procedimento de construção da lista Q da estrutura de dados do Capítulo 4. Portanto, o genoma π é facilmente transformável em σ .

Por outro lado, se π é facilmente transformável em σ então existe um caminho orientado de π a σ pelo Lema 91. Logo σ é atingível a partir de π e $\sigma \in R(\pi)$ para o grafo $G(\sigma)$.

Portanto o genoma π é facilmente transformável em σ se e somente se $\sigma \in R(\pi)$ para o grafo $G(\sigma)$. \square

Para decidirmos se um genoma π no σ -componente de $G(\sigma)$ é facilmente transformável é suficiente resolver o problema de encontrar todos os genomas pertencentes a $R(\pi)$. Observe que qualquer genoma π de um σ -componente de $G(\sigma)$ é um genoma facilmente transformável, caso σ seja o único vértice sumidouro do σ -componente.

Lema 96. Dado o grafo de transposições boas $G(\sigma)$ para $\sigma \in (E, \Gamma)$, existe um genoma $\pi \in (E, \Gamma)$ tal que a cardinalidade de $R(\pi)$ é $2^{\lfloor |E|/6 \rfloor}$ dado que $|E|$ seja um múltiplo de 3.

Prova: Para σ em um sistema de genes (E, Γ) , renomeamos os elementos de E com inteiros de tal modo que

$$\sigma = (1 \ 2 \ \dots \ n)(-n \ \dots \ -2 \ -1).$$

Considere o genoma unicromossomal

$$\pi = (2 \ 1 \ 3 \ \dots \ k+1 \ k \ k+2 \ \dots \ n-1 \ n-2 \ n)(-n \ -(n-2) \ \dots \ -3 \ -1 \ -2).$$

Observe que

$$\sigma\pi^{-1} = (2 \ 1 \ 3)(-1 \ -2 \ -n) \dots (n-1 \ n-2 \ n)(-(n-2) \ -(n-1) \ -(n-3))$$

A decomposição em ciclos disjuntos de $\sigma\pi^{-1}$ é composta por $b(\pi, \sigma)/3$ transposições disjuntas aplicáveis a π pois cada um dos pares de $\sigma\pi^{-1}$ divide o genoma π . Além disso, a aplicação de quaisquer dessas transposições $\tau = (i+1 \ i \ i+2)(-i \ -(i+1) \ -(i-1))$ onde $i = 3k+1$ para $1 \leq k \leq (n-3)/3$ (e $\tau = (2 \ 1 \ 3)(-1 \ -2 \ -n)$ para $k = 0$) de $\sigma\pi^{-1}$ em π envolve apenas elementos do suporte da transposição e não modificam a ordem circular dos demais elementos em $\tau\pi$.

$$\tau\pi = (2 \ 1 \ 3 \ \dots \ i \ i+1 \ i+2 \ \dots \ n-1 \ n-2 \ n)(-n \ -(n-2) \ -(n-1) \ \dots \ -3 \ -1 \ -2).$$

Como os elementos que não estão no suporte de uma transposição não são afetados, então todos os pares de $\sigma\pi^{-1}$ permanecem transposições, a menos que já tenham sido aplicadas, e após a aplicação de todas as transposições em π em qualquer ordem, obtemos σ . Logo é possível transformar π em σ utilizando-se de $d_t(\pi, \sigma) = (\|\sigma\pi^{-1}\|_3)/2$ transposições.

Segundo o Lema 85, os genomas π e σ satisfazem $d_t(\pi, \sigma) = b(\pi, \sigma)/3$ e para qualquer transposição τ escolhida em $\sigma\pi^{-1}$, temos $\tau|_3\sigma\pi^{-1}$. Logo, há $m = b(\pi, \sigma)/3$ genomas atingíveis que podem ser atingidos diretamente por π . Cada um desses m genomas possui um quociente distinto $\sigma\pi^{-1}\tau^{-1}$ que difere de $\sigma\pi^{-1}$ apenas pela transformação de um dos seus pares de 3-ciclos em seis elementos fixos. Além disso, para qualquer par τ' de $\sigma\pi^{-1}\tau^{-1}$ temos $\tau'|_3\sigma\pi^{-1}\tau^{-1}$. Conseqüentemente há $m - 1$ genomas atingíveis por $\tau\pi$. Logo, após a aplicação de uma transposição, o número de genomas atingíveis pelo genoma corrente é reduzido de 1. Se repetirmos essas aplicações de transposições até não restar nenhum par original de $\sigma\pi^{-1}$, verificamos que há $m!$ genomas atingíveis por π . Como $m = |Supp(\sigma\pi^{-1})|/6 = |E|/6$, então existe um genoma $\pi \in (E, \Gamma)$ tal que a cardinalidade de $R(\pi)$ é $2^{|E|/6}$ dado que $|E|$ seja um múltiplo de 3. \square

O Lema 96 mostra que o conjunto $R(\pi)$ não pode ser construído em tempo de execução polinomial. Esse resultado nos impede de aplicar uma busca enumerativa por todos os genomas unicromossomais atingíveis a partir do genoma π para encontrar um caminho de π a σ . Por outro lado, o Lema 96 não implica que não exista nenhum algoritmo de tempo de execução polinomial para encontrar um caminho cuja origem é π . É possível caracterizar os genomas facilmente transformáveis desde que se ache tal algoritmo.

Se um genoma unicromossomal π atinge σ em $G(\sigma)$, então o comprimento do caminho de π a σ é maior do que o comprimento de qualquer outro caminho cuja origem seja π . Logo, encontrar um caminho de comprimento máximo de um genoma do σ -componente é um modo alternativo de determinar os genomas facilmente transformáveis.

5.1.3 Transposições Generalizadas

Dados os genomas conorbitais π e σ no sistema de genes (E, Γ) , denotamos por $c_{odd}(\pi, \sigma)$ o número de pares de $\sigma\pi^{-1}$ cujos ciclos são ímpares. Observe que $2c_{odd}(\pi, \sigma) = o_{odd}(\sigma\pi^{-1})$. Usamos a notação $\Delta c_{odd}(\tau, \pi, \sigma) = c_{odd}(\tau\pi, \sigma) - c_{odd}(\pi, \sigma)$ para a variação no número de pares cujos ciclos são ímpares onde τ é uma transposição generalizada aplicável a π . Uma transposição generalizada τ aplicável a um genoma π é um evento 3-bom para (π, σ) quando $\tau|_3\sigma\pi^{-1}$. Observe que o conceito de evento 3-bom para transposições generalizadas distingue-se do conceito semelhante de evento 2-bom para outros eventos de rearranjo devido àquele tratar da 3-divisibilidade, ao invés da divisibilidade. O Lema 97, a seguir, caracteriza um evento 3-bom por meio da variação do número de pares cujos ciclos são ímpares. Essa variação na paridade do comprimento de ciclos de $\sigma\pi^{-1}$ decorre do efeito do produto de permutações envolvendo 3-ciclos que pode alterar a paridade do comprimento de ciclos.

Lema 97. Dados os genomas conorbitais π e σ no sistema de genes (E, Γ) , uma transposição generalizada τ aplicável a π é um evento 3-bom para (π, σ) se e somente se $\Delta c_{odd}(\tau, \pi, \sigma) = w(\tau)$.

Prova: Dados os genomas conorbitais π e σ no sistema de genes (E, Γ) , se τ é uma transposição generalizada aplicável a π e um evento 3-bom para (π, σ) então $\tau|_3\sigma\pi^{-1}$, ou seja, temos $\|\sigma\pi^{-1}\tau^{-1}\|_3 = \|\sigma\pi^{-1}\|_3 - \|\tau\|_3$. Como uma transposição generalizada τ é uma permutação par composta por um produto de ciclos ímpares então $\|\tau\|_3 = \|\tau\|/2$. Expandindo a fórmula do peso por meio do Teorema 58 e usando o Teorema 82:

$$\begin{aligned}
 w(\tau) &= \frac{\|\tau\|}{2} \\
 &= \|\tau\|_3 \\
 &= \|\sigma\pi^{-1}\|_3 - \|\sigma\pi^{-1}\tau^{-1}\|_3 \\
 &= \frac{|E| - 2\|\sigma\pi^{-1}\tau^{-1}\|_3 - |E| + 2\|\sigma\pi^{-1}\|_3}{2} \\
 &= \frac{o_{\text{odd}}(\sigma\pi^{-1}\tau^{-1}) - o_{\text{odd}}(\sigma\pi^{-1})}{2} \\
 &= c_{\text{odd}}(\tau\pi, \sigma) - c_{\text{odd}}(\pi, \sigma)
 \end{aligned}$$

e como $c_{\text{odd}}(\tau\pi, \sigma) - c_{\text{odd}}(\pi, \sigma) = \Delta c_{\text{odd}}(\tau, \pi, \sigma)$ então $\Delta c_{\text{odd}}(\tau, \pi, \sigma) = w(\tau)$.

Inversamente, se $\Delta c_{\text{odd}}(\tau, \pi, \sigma) = w(\tau)$, ou seja, temos $c_{\text{odd}}(\tau\pi, \sigma) - c_{\text{odd}}(\pi, \sigma) = \|\tau\|/2$, então, pela definição de $c_{\text{odd}}(\pi, \sigma)$ e como $2c_{\text{odd}}(\pi, \sigma) = o_{\text{odd}}(\sigma\pi^{-1})$, temos

$$\|\sigma\pi^{-1}\|_3 - \|\sigma\pi^{-1}\tau^{-1}\|_3 = \frac{\|\tau\|}{2}.$$

Como $\frac{\|\tau\|}{2} = \|\tau\|_3$, então $\|\sigma\pi^{-1}\tau^{-1}\|_3 = \|\sigma\pi^{-1}\|_3 - \|\tau\|_3$ e portanto $\tau|_3\sigma\pi^{-1}$. Logo τ é uma transposição generalizada aplicável a π e um evento 3-bom para (π, σ) .

Portanto, dados os genomas conorbitais π e σ no sistema de genes (E, Γ) , uma transposição generalizada τ aplicável a π é um evento 3-bom para (π, σ) se e somente se $\Delta c_{\text{odd}}(\tau, \pi, \sigma) = w(\tau)$. \square

Dados dois genomas π e σ em (E, Γ) , a permutação $\sigma\pi^{-1}$ que possui o número máximo de ciclos ímpares é a identidade. Uma transposição generalizada τ que seja aplicável a π e aumente de $w(\tau)$ o número de pares cujos os ciclos são ímpares em $\sigma\pi^{-1}\tau^{-1}$ em relação a $\sigma\pi^{-1}$ pertence a uma sequência mínima de eventos de rearranjo que transforma π em σ como mostraremos no Lema 98.

Lema 98. Dados os genomas conorbitais π e σ no sistema de genes (E, Γ) , para qualquer sequência de transposições generalizadas τ_1, \dots, τ_k , tal que $\tau_k \dots \tau_1\pi = \sigma$ e τ_i é aplicável ao genoma $\tau_{i-1} \dots \tau_1\pi$, temos:

1. $\sum_{j=1}^k w(\tau_j) \geq \|\sigma\pi^{-1}\|_3$;
2. $\sum_{j=1}^k w(\tau_j) = \|\sigma\pi^{-1}\|_3$ se e somente se cada transposição generalizada τ_i é um evento 3-bom para $(\tau_{i-1} \dots \tau_1\pi, \sigma)$ para $1 \leq i \leq k$.

Prova:

1. Dados os genomas conorbitais π e σ no sistema de genes (E, Γ) , seja τ_1, \dots, τ_k uma seqüência de eventos de rearranjo tais que $\tau_k \dots \tau_1 \pi = \sigma$ e τ_i é aplicável a $\tau_{i-1} \dots \tau_1 \pi$ para $1 \leq i \leq k$. Logo $\tau_k \dots \tau_1 = \sigma \pi^{-1}$. Como τ_i para $1 \leq i \leq k$ é uma transposição generalizada então $\|\tau_i\|_3 = \|\tau_i\|/2$ para $1 \leq i \leq k$ pelo Teorema 58. Usando o quarto item do Lema 55 temos o seguinte limitante superior para $\|\sigma \pi^{-1}\|_3$.

$$\begin{aligned}
 \|\sigma \pi^{-1}\|_3 &= \|\tau_k \dots \tau_1\|_3 \\
 &\leq \|\tau_k\|_3 + \dots + \|\tau_1\|_3 \\
 &= \sum_{j=1}^k \frac{\|\tau_j\|}{2} \\
 &= \sum_{j=1}^k w(\tau_j)
 \end{aligned}$$

Logo $\sum_{j=1}^k w(\tau_j) \geq \|\sigma \pi^{-1}\|_3$.

2. Assumimos que cada evento de rearranjo τ_i é um evento 3-bom para $(\tau_{i-1} \dots \tau_1 \pi, \sigma)$ para $1 \leq i \leq k$. Pela definição de peso, temos:

$$\sum_{j=1}^k w(\tau_j) = \frac{\|\tau_1\| + \dots + \|\tau_k\|}{2}. \quad (5.1)$$

Uma vez que o evento de rearranjo τ_i é um evento 3-bom para $(\tau_{i-1} \dots \tau_1 \pi, \sigma)$ para $1 \leq i \leq k$ então pelo Lema 97 temos

$$\frac{\|\tau_i\|}{2} = c_{\text{odd}}(\tau_i \dots \tau_1 \pi, \sigma) - c_{\text{odd}}(\tau_{i-1} \dots \tau_1 \pi, \sigma).$$

Pela definição do número de pares cujos ciclos são ímpares, temos

$$\frac{\|\tau_i\|}{2} = \|\sigma \pi^{-1} \tau_1^{-1} \dots \tau_{i-1}^{-1}\|_3 - \|\sigma \pi^{-1} \tau_1^{-1} \dots \tau_i^{-1}\|_3. \quad (5.2)$$

Usando a Equação 5.1 e a Equação 5.2 e manipulando como segue:

$$\begin{aligned}
 \sum_{j=1}^k w(\tau_j) &= \|\sigma \pi^{-1}\|_3 - \|\sigma \pi^{-1} \tau_1^{-1} \dots \tau_k^{-1}\|_3 \\
 &= \|\sigma \pi^{-1}\|_3 - \|\sigma \sigma^{-1}\|_3 \\
 &= \|\sigma \pi^{-1}\|_3.
 \end{aligned}$$

Portanto $\sum_{j=1}^k w(\tau_j) = \|\sigma\pi^{-1}\|_3$.

Por outro lado, se $\sum_{j=1}^k w(\tau_j) = \|\sigma\pi^{-1}\|_3$ então $2\|\sigma\pi^{-1}\|_3 = \sum_{j=1}^k \|\tau_j\|_3$.

Pela propriedade da desigualdade triangular (item 4) do Lema 55, temos

$$\|\sigma\pi^{-1}\tau_1^{-1} \dots \tau_{i-1}^{-1}\|_3 \leq \|\sigma\pi^{-1}\tau_1^{-1} \dots \tau_i^{-1}\|_3 + \|\tau_i\|_3,$$

para $1 \leq i \leq k$, em outras palavras, temos

$$0 \leq \|\sigma\pi^{-1}\tau_1^{-1} \dots \tau_i^{-1}\|_3 - \|\sigma\pi^{-1}\tau_1^{-1} \dots \tau_{i-1}^{-1}\|_3 + \|\tau_i\|_3,$$

para $1 \leq i \leq k$. Como cada termo $\|\sigma\pi^{-1}\tau_1^{-1} \dots \tau_i^{-1}\|_3 - \|\sigma\pi^{-1}\tau_1^{-1} \dots \tau_{i-1}^{-1}\|_3 + \|\tau_i\|_3$ é não negativo e manipulando a soma expandida, obtemos

$$\begin{aligned} 0 &\leq \sum_{i=1}^k \left(\|\sigma\pi^{-1}\tau_1^{-1} \dots \tau_i^{-1}\|_3 - \|\sigma\pi^{-1}\tau_1^{-1} \dots \tau_{i-1}^{-1}\|_3 + \|\tau_i\|_3 \right) \\ &= \|\sigma\pi^{-1}\tau_1^{-1} \dots \tau_k^{-1}\|_3 - \|\sigma\pi^{-1}\|_3 + \sum_{i=1}^k \|\tau_i\|_3. \end{aligned}$$

Como $\sigma\pi^{-1}\tau_1^{-1} \dots \tau_k^{-1} = \iota$ e $\sum_{i=1}^k (\|\tau_i\|_3/2) = \|\sigma\pi^{-1}\|_3$ então

$$\sum_{i=1}^k \left(\|\sigma\pi^{-1}\tau_1^{-1} \dots \tau_i^{-1}\|_3 - \|\sigma\pi^{-1}\tau_1^{-1} \dots \tau_{i-1}^{-1}\|_3 + \|\tau_i\|_3 \right) = 0.$$

Então $\|\sigma\pi^{-1}\tau_1^{-1} \dots \tau_i^{-1}\|_3 - \|\sigma\pi^{-1}\tau_1^{-1} \dots \tau_{i-1}^{-1}\|_3 + \|\tau_i\|_3 = 0$ para $1 \leq i \leq k$; i.e. temos

$$\|\sigma\pi^{-1}\tau_1^{-1} \dots \tau_i^{-1}\|_3 = \|\sigma\pi^{-1}\tau_1^{-1} \dots \tau_{i-1}^{-1}\|_3 - \|\tau_i\|_3$$

para $1 \leq i \leq k$. Logo $\tau_i|_3\sigma\pi^{-1}\tau_1^{-1} \dots \tau_{i-1}^{-1}$ para $1 \leq i \leq k$, ou seja, cada τ_i aplicável a $\tau_{i-1} \dots \tau_1\pi$ é um evento 3-bom para $(\tau_{i-1} \dots \tau_1\pi, \sigma)$.

□

Os próximos lemas (Lema 99 e Lema 100) serão usados para demonstrar que para qualquer par de genomas compatíveis em relação a transposições generalizadas existe uma transposição generalizada que é um evento 3-bom para o par de genomas.

Lema 99. Dados os genomas conorbitais π e σ em (E, Γ) , temos $\sigma\pi^{-1}x \in orb(\sigma, x)$.

Prova: Como os genomas π e σ em (E, Γ) são compatíveis com relação a transposições generalizadas então $orb(\pi, x) \subseteq orb(\sigma, x)$. Logo $\pi^{-1}x \in orb(\sigma, x)$, pois $\pi^{-1}x \in orb(\pi, x)$. Portanto temos $\sigma\pi^{-1}x \in orb(\sigma, x)$. □

Lema 100. Dados os genomas conorbitais π e σ em (E, Γ) , para toda órbita $o \in Orb(\sigma\pi^{-1}, E)$ existe uma órbita $p \in Orb(\sigma, E)$ tal que $o \subseteq p$.

Prova: Como os genomas π e σ em (E, Γ) são compatíveis com relação a transposições generalizadas então $orb(\pi, x) \subseteq orb(\sigma, x)$ para qualquer $x \in E$. Pelo Lema 99, temos $\sigma\pi^{-1}x \in orb(\sigma, x)$. Logo $\sigma\pi^{-1}x = \sigma^j x$ para $j \geq 0$, por definição de $orb(\sigma, x)$. Seja $(\sigma\pi^{-1})^k x$ um elemento da órbita $orb(\sigma\pi^{-1}, x)$ para $0 \leq k \leq |orb(\sigma\pi^{-1}, x)| - 1$. Como $\sigma\pi^{-1}x = \sigma^j x$ então $(\sigma\pi^{-1}x)^k = (\sigma^j x)^k = \sigma^{jk} x$ para $0 \leq k \leq |orb(\sigma\pi^{-1}, x)| - 1$. Como $j \geq 0$ e $k \geq 0$ então $jk \geq 0$. Logo $(\sigma\pi^{-1})^k x \in orb(\sigma, x)$ pela definição de órbita. Portanto, temos $orb(\sigma\pi^{-1}, x) \subseteq orb(\sigma, x)$ para todo $x \in E$, em outros termos, para toda órbita $o = orb(\sigma\pi^{-1}, x) \in Orb(\sigma\pi^{-1}, E)$ existe uma órbita $p = orb(\sigma, x) \in Orb(\sigma, E)$ tal que $o \subseteq p$. \square

Lema 101. Dados os genomas conorbitais π e σ distintos em (E, Γ) , há uma transposição generalizada τ aplicável a π tal que τ é um evento 3-bom para (π, σ) .

Prova: Dados os genomas π e σ em (E, Γ) , iremos considerar dois casos:

1. Se não houver pares em $\sigma\pi^{-1}$ com comprimento maior que dois, então existem pelo menos dois pares cujos comprimentos dos ciclos é dois, pelo fato de π e σ serem genomas distintos e conorbitais. Como cada órbita de σ possui pelo menos 3 genes, então existem dois pares $(a \ \sigma\pi^{-1}a)(\pi\Gamma\sigma\pi^{-1}a \ \pi\Gamma a)$ e $(b \ \sigma\pi^{-1}b)(\pi\Gamma\sigma\pi^{-1}b \ \pi\Gamma b)$ tais que a e b pertencem a mesma órbita em σ . Iremos mostrar que o evento

$$\tau = (a \ \sigma\pi^{-1}a \ b)(\pi\Gamma b \ \pi\Gamma\sigma\pi^{-1}a \ \pi\Gamma a)$$

é um evento de rearranjo aplicável a π e τ é um evento 3-bom para (π, σ) . Os três elementos a , $\sigma\pi^{-1}a$ e b pertencentes a E são diferentes entre si pois foram escolhidos entre os elementos de pares com comprimento igual a dois.

Além disso, pelo Lema 99 temos $\sigma\pi^{-1}a \in orb(\sigma, a)$, então τ é aplicável em π . Pelo Teorema 48, o produto $\sigma\pi^{-1}\tau^{-1}$ resulta em

$$\beta(a \ \sigma\pi^{-1}a \ b)(\pi\Gamma b \ \pi\Gamma\sigma\pi^{-1}a \ \pi\Gamma\sigma\pi^{-1}b)(\sigma\pi^{-1}a)(\pi\Gamma a),$$

onde β é o produto dos demais 2-ciclos que não foram modificados pelo produto. Como os 2-ciclos em β são disjuntos entre si e não são afetados pelo produto τ^{-1} , então temos $o_{odd}(\sigma\pi^{-1}\tau^{-1}) = o_{odd}(\sigma\pi^{-1}) + 4$.

2. Se houver um par com comprimento maior que dois em $\sigma\pi^{-1}$ então definimos a transposição generalizada

$$\tau = (a \ \sigma\pi^{-1}a \ \sigma\pi^{-1}\sigma\pi^{-1}a)(\pi\Gamma\sigma\pi^{-1}\sigma\pi^{-1}a \ \pi\Gamma\sigma\pi^{-1}a \ \pi\Gamma a).$$

Como a , $\sigma\pi^{-1}a$, $\sigma\pi^{-1}\sigma\pi^{-1}a$ pertencem a mesma órbita em $\sigma\pi^{-1}$, então esses três elementos pertencem a mesma órbita em σ pelo Lema 100. Logo, a transposição generalizada τ é aplicável em π . Pelo Teorema 48, o produto $\sigma\pi^{-1}\tau^{-1}$ é

$$\beta(a (\sigma\pi^{-1})^3 a \dots (\sigma\pi^{-1})^{-1} a)(\pi\Gamma(\sigma\pi^{-1})^{-1} a \dots \pi\Gamma a)(\sigma\pi^{-1} a)(\pi\Gamma a),$$

onde β é o produto dos demais ciclos que não foram modificados pelo produto. Como os ciclos em β são disjuntos entre si e dois novos pares compostos por ciclos ímpares são criados em $\sigma\pi^{-1}\tau^{-1}$ então $o_{\text{odd}}(\sigma\pi^{-1}\tau^{-1}) = o_{\text{odd}}(\sigma\pi^{-1}) + 4$.

Em ambos os casos, encontramos uma variação no número de órbitas ímpares de 4. Pelo Teorema 82 e manipulando a expressão $o_{\text{odd}}(\sigma\pi^{-1}\tau^{-1}) = o_{\text{odd}}(\sigma\pi^{-1}) + 4$ como se segue

$$\begin{aligned} o_{\text{odd}}(\sigma\pi^{-1}\tau^{-1}) &= o_{\text{odd}}(\sigma\pi^{-1}) + 4 \\ 2\|\sigma\pi^{-1}\tau^{-1}\|_3 - |E| &= 2\|\sigma\pi^{-1}\|_3 - |E| + 2\|\tau\|_3 \\ \|\sigma\pi^{-1}\tau^{-1}\|_3 &= \|\sigma\pi^{-1}\|_3 - \|\tau\|_3, \end{aligned}$$

obtemos $\|\sigma\pi^{-1}\tau^{-1}\|_3 = \|\sigma\pi^{-1}\|_3 - \|\tau\|_3$. Logo $\tau|_3\sigma\pi^{-1}$ e portanto τ é um evento 3-bom para (π, σ) . \square

Algoritmo

Dados os genomas conorbitais π e σ em um sistema de genes (E, Γ) , o *problema de rearranjo em genomas ponderado envolvendo transposições generalizadas* consiste em encontrar uma seqüência de transposições generalizadas τ_1, \dots, τ_k , tal que

$$\tau_k \dots \tau_1 \pi = \sigma,$$

onde τ_i é aplicável a $\tau_{i-1} \dots \tau_1 \pi$ para $1 \leq i \leq k$ e $\sum_{i=1}^k w(\tau_i)$ é mínimo. O valor $\sum_{i=1}^k w(\tau_i)$ é denotado por $W_{tg}(\pi, \sigma)$.

Apresentamos agora um algoritmo que resolve o problema de rearranjos em genomas ponderado por transposições generalizadas em tempo de execução linear. Para obter tal desempenho de tempo de execução, modificamos a estrutura de dados apresentada no Capítulo 4. Como discutimos no Capítulo 4, a identificação de eventos de rearranjo bons e a atualização da estrutura de dados que reflete a aplicação do evento de rearranjo são os procedimentos que determinam qual será o tempo de execução total de um algoritmo para um problema de rearranjo em genomas.

Dado o problema $P(\pi, \sigma, E, \Gamma)$ para π e σ conorbitais, consideremos primeiramente a questão de como identificar uma transposição generalizada que seja um evento 3-bom para (π, σ) . Pelo Lema 101, podemos encontrar um evento 3-bom para (π, σ) escolhendo três genes consecutivos na ordem circular de um ciclo de $\sigma\pi^{-1}$ caso exista algum par em $\sigma\pi^{-1}$ cujos ciclos têm

suporte maior ou igual a três. Essa escolha de três genes pode ser trivialmente implementada pela escolha dos elementos nos três primeiros nós do primeiro par em Q cujas listas circulares contêm mais de três elementos. Um problema na estrutura de dados que já se torna claro é como encontrar pares de Q cujo tamanho de seus ciclos sejam maiores do que três de maneira eficiente (idealmente em tempo constante). Para simplificar a terminologia, iremos nos referir aos pares que contêm ciclos de tamanho maior ou igual a três por *pares longos*, enquanto os pares cujos ciclos são 2-ciclos chamaremos de *pares curtos* tanto em $\sigma\pi^{-1}$ como na estrutura Q . Para garantir que encontraremos rapidamente pares longos, modificamos a inicialização da lista Q de tal modo que ao final da inicialização todos os pares longos apareçam primeiramente na lista enquanto os pares curtos apareçam no final da lista. Além disso, como havíamos implementado a estrutura Q como uma lista duplamente ligada, a modificação anterior de manter os pares longos como os primeiros na lista demandaria um tempo de execução quadrático no número de pares, seja na inicialização ou atualização da estrutura caso não a modificássemos para mover pares na lista de maneira eficiente. Observe que o tamanho dos pares de Q são reduzidos à medida que as transposições generalizadas vão sendo aplicadas e por isso pode ser necessário mover pares para a sua posição adequada na lista para manter a propriedade anterior. Há duas maneiras de projetar Q para que a sua inicialização e atualização sejam rápidas. A primeira solução consiste em implementar Q como uma lista duplamente ligada como anteriormente e além disso mantermos dois apontadores: um para o início da lista (o início do trecho contendo os pares longos) e outro para o primeiro par curto na lista. No momento em que um par fosse identificado como curto (na inicialização ou na atualização de Q) ele seria inserido no início do trecho contendo pares curtos apontado pelo segundo apontador por meio de uma inserção em listas duplamente ligadas. Uma segunda solução possível é implementar Q como uma lista circular duplamente ligada na qual temos apenas um apontador para o primeiro par longo N e o primeiro par no sentido anti-horário a partir de N seria o início dos pares curtos na lista circular. A Figura 5.1.3 mostra as duas diferentes implementações de Q .

Utilizaremos a solução indicada no item (a) da Figura 5.1.3, ou seja, definiremos dois novos apontadores $pShort$ e $pLong$ que apontam, respectivamente, para o primeiro par curto e para o primeiro par longo em Q . Essa implementação permite que o procedimento de inicialização da lista Q mostrado na Figura 4.2 possa ser mantido o mesmo. Para inicializar a nova lista Q , basta modificar o procedimento $buildPair(gene\ x)$ conforme mostrado na Figura 5.3

Caso todos os pares de Q sejam pares curtos então os genes, provenientes de diferentes pares, ao serem escolhidos para compor a transposição generalizada devem pertencer a mesma órbita do genoma σ . Para descobrir se dois genes pertencem a uma mesma órbita de σ , é necessário verificar todos os genes pertencentes à órbita em σ definida por um dos dois genes. Essa verificação levaria um tempo de execução linear no número de genes e portanto o tempo de execução para aplicar um evento seria linear. Podemos reduzir o tempo de execução da escolha de genes nesse caso específico em que todos os pares de Q são pares curtos por meio da

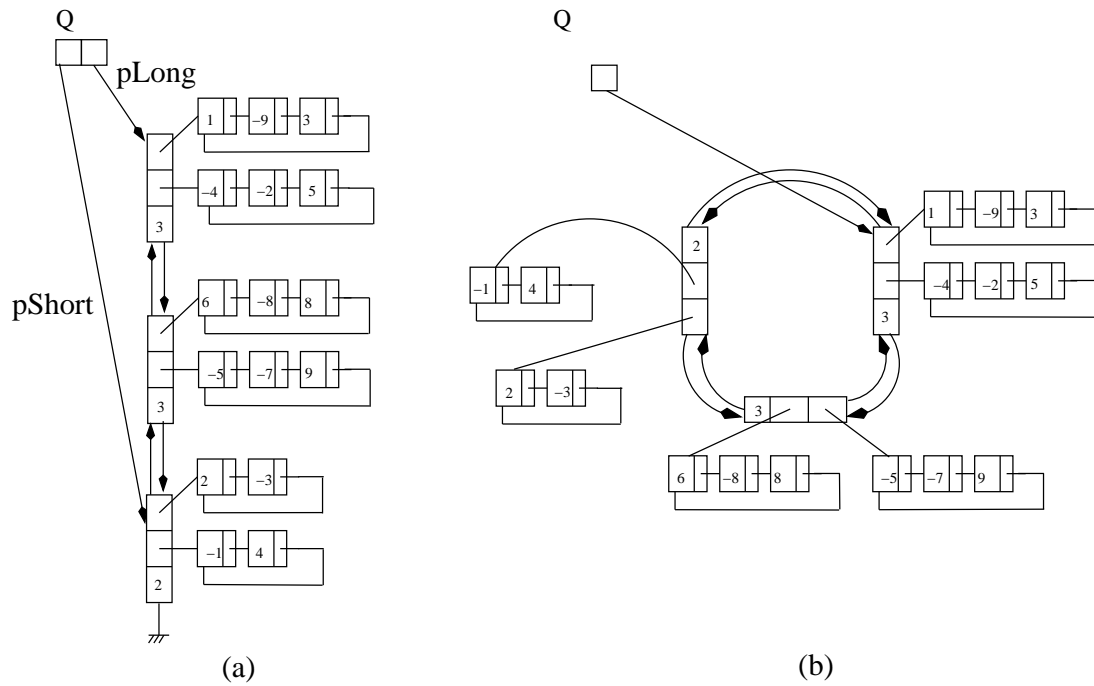


Figura 5.2: Lista Q modificada para separar pares longos de pares curtos. O exemplo da figura utiliza os genomas $\pi = (1 \ -3 \ -4 \ 2 \ -9 \ 6 \ 7 \ -8 \ -5)(5 \ 8 \ -7 \ -6 \ 9 \ -2 \ 4 \ 3 \ -1)$ e $\sigma = (1 \ 2 \ 3 \ 4)(-4 \ -3 \ -2 \ -1)(5 \ 6 \ 7 \ 8 \ 9)(-9 \ -8 \ -7 \ -6 \ -5)$ em (E, Γ) , onde $E = \{x \in \mathbb{Z} \mid 1 \leq |x| \leq 9\}$ e $\Gamma x = -x$ para $x \in E$. Há duas maneiras possíveis de implementar a separação entre pares longos e curtos na lista. (a) A lista Q foi modificada com a inclusão de dois apontadores $pLong$ e $pShort$. (b) A lista Q é modificada para tornar-se uma lista duplamente ligada circular e o ponteiro Q aponta para o primeiro par longo que aparece na lista circular no sentido horário.

Procedimento buildPair(gene x)

```

1: Criar par  $N$ 
2:  $count \leftarrow 0$ 
3:  $y \leftarrow x$ 
4: repeat
5:   Criar novo nó  $A$  com o gene  $y$ 
6:   Inserir  $A$  na cauda da lista circular 1 de  $N$ 
7:    $T[y].node \leftarrow A$ 
8:    $T[y].pair \leftarrow N$ 
9:   Criar novo nó  $B$  com o gene  $P[-y]$ 
10:  Inserir  $B$  a cabeça da lista circular 2 de  $N$ 
11:   $T[P[-y]].node \leftarrow B$ 
12:   $T[P[-y]].pair \leftarrow N$ 
13:   $y \leftarrow S[invP[y]]$ 
14:   $count++$ 
15: until  $y = x$ 
16: Fechar a lista circular contendo  $x$ 
17: Fechar a lista circular contendo  $P[-x]$ 
18: if  $count \geq 3$  then
19:   Inserir  $N$  em  $Q.pLong$ 
20: else
21:   Inserir  $N$  em  $Q.pShort$ 
22: end if

```

Figura 5.3: Procedimento usado para construir um novo par N e incluí-lo na lista Q modificada.

inclusão de uma nova estrutura para representar as órbitas de σ . Seja SL uma lista duplamente ligada contendo os genes das órbitas não triviais de σ que não são fixos em $\sigma\pi^{-1}$. Os genes são distribuídos na lista SL de maneira que qualquer sublista de SL que tenha como primeiro e último elementos os genes da mesma órbita não contenha sublistas menores de genes de órbitas distintas da órbita daqueles dois. Por exemplo, dado o genoma

$$\sigma = (a \ b \ c \ d)(\Gamma d \ \Gamma c \ \Gamma b \ \Gamma a)(e \ f \ g)(\Gamma g \ \Gamma f \ \Gamma e)(h)(\Gamma h)$$

a lista correspondente a σ é $SL = a, b, c, d, \Gamma d, \Gamma c, \Gamma b, \Gamma a, e, f, g, \Gamma g, \Gamma f, \Gamma e$. Como os genomas π e σ são conorbitais então cada órbita não trivial de σ tem tamanho pelo menos 3 e $orb(\pi, x) \subseteq orb(\sigma, x)$ para qualquer $x \in E$. Logo podemos escolher os primeiros 3 elementos da lista SL , digamos a, b, c , e a transposição generalizada $(a \ b \ c)(\pi\Gamma c \ \pi\Gamma b \ \pi\Gamma a)$ é aplicável a π e os pares afetados em Q serão aqueles cujas órbitas contenham esses três elementos. Pelo Lema 100, o fato de haver apenas pares curtos em Q e como SL não contém os elementos fixos em $\sigma\pi^{-1}$, então apenas dois pares (e.g. $(a \ \sigma\pi^{-1}a)(\pi\Gamma\sigma\pi^{-1}a \ \pi\Gamma a)$ e $(b \ \sigma\pi^{-1}b)(\pi\Gamma\sigma\pi^{-1}b \ \pi\Gamma b)$ onde $c = \sigma\pi^{-1}b$) serão afetados. A inicialização de SL se dá ao visitar todos os genes no vetor S verificando quais dos genes são fixos em $\sigma\pi^{-1}$, ou seja verifica-se se $S[invP[x]] = x$ para cada gene $x \in E$ e neste caso ele não é adicionado a lista SL . A cada atualização da lista Q , a lista SL deve ser atualizada caso novos genes se tornarem fixos. Para realizar a atualização de SL em tempo constante, nós adicionamos um campo de apontador a cada elemento do vetor T que aponta para o nó em SL que contém aquele elemento ou para NIL caso o elemento não esteja na lista SL . Os genes que se tornaram fixos são removidos de SL excluindo-se os nós apontados por aqueles elementos em T , o que pode ser realizado em tempo constante pois são operações de remoção em uma lista duplamente ligada.

As demais operações da estrutura de dados não são modificadas pelo fato de lidarmos com transposições generalizadas. A Figura 5.1.3 mostra a estrutura de dados utilizada para implementar o algoritmo para o problema de rearranjo por transposições generalizadas.

Dados π e σ em (E, Γ) , seja $\tau = (a \ \sigma\pi^{-1}a \ b)(\pi\Gamma b \ \pi\Gamma\sigma\pi^{-1}a \ \pi\Gamma a)$ uma transposição generalizada aplicável a π que foi escolhida a partir de SL e Q por ser um evento 3-bom para (π, σ) . A implementação de como a estrutura de dados deve ser atualizada devido aplicação da transposição generalizada τ é mostrada a seguir:

1. Os genes $a, \sigma\pi^{-1}a, b, \pi\Gamma b, \pi\Gamma\sigma\pi^{-1}a$ e $\pi\Gamma a$ são pesquisados para encontrar os elementos $\pi^{-1}a, \pi^{-1}\sigma\pi^{-1}a, \pi^{-1}b, \Gamma b, \Gamma\sigma\pi^{-1}a$ e Γa em $invP$.
2. Atualização de P . Os genes $\pi^{-1}a, \pi^{-1}\sigma\pi^{-1}a, \pi^{-1}b, \Gamma b, \Gamma\sigma\pi^{-1}a$ e Γa são mapeados em $\sigma\pi^{-1}a, b, a, \pi\Gamma\sigma\pi^{-1}a, \pi\Gamma a$ e $\pi\Gamma b$ respectivamente em $\tau\pi$, enquanto os demais genes em P não são alterados pois são mapeados para genes que são fixos em τ .

Então, os genes $\pi^{-1}a, \pi^{-1}\sigma\pi^{-1}a, \pi^{-1}b, \Gamma b, \Gamma\sigma\pi^{-1}a$ e Γa em P são atualizados para $\sigma\pi^{-1}a, b, a, \pi\Gamma\sigma\pi^{-1}a, \pi\Gamma a$ e $\pi\Gamma b$ respectivamente em P' , onde P' representa $\tau\pi$.

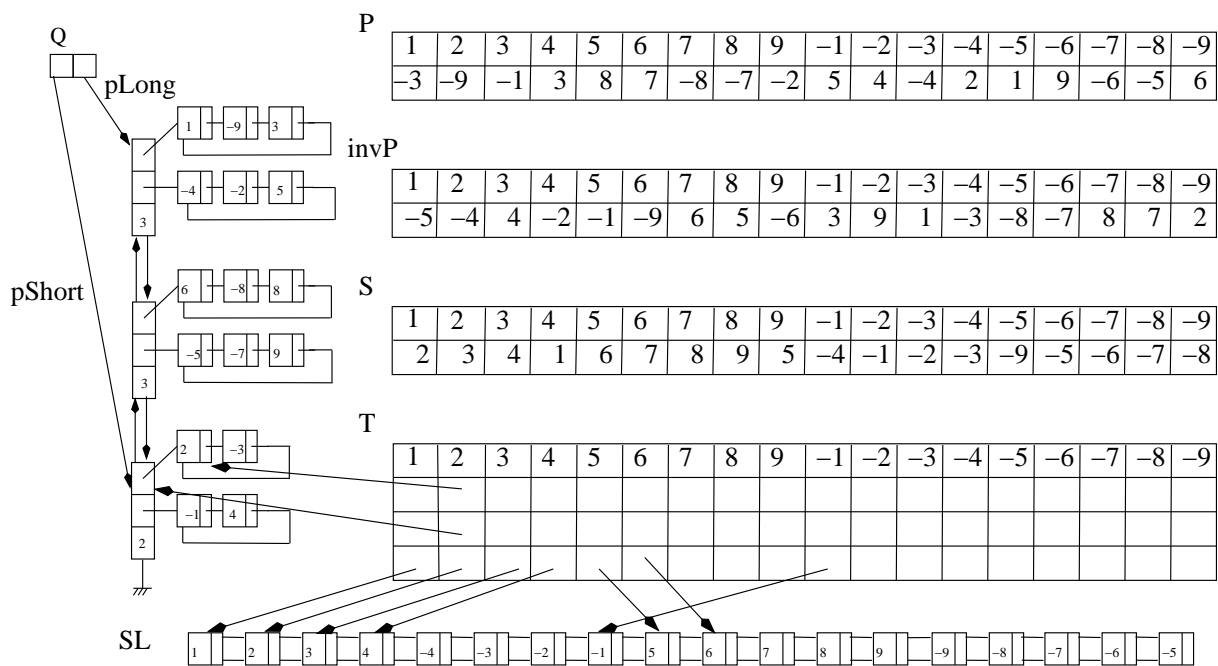


Figura 5.4: Estrutura de dados modificada para tratar do problema de rearranjo em genomas envolvendo transposições generalizadas. É mostrada uma versão estendida da estrutura de dados apresentada na Figura 4.1. Para manter a clareza da figura, representamos apenas alguns dos apontadores originados no vetor T .

3. Atualização de $invP$. Os elementos a , $\sigma\pi^{-1}a$, b , $\pi\Gamma b$, $\pi\Gamma\sigma\pi^{-1}a$ e $\pi\Gamma a$ são mapeados nos elementos $\pi^{-1}b$, $\pi^{-1}a$, $\pi^{-1}\sigma\pi^{-1}a$, Γa , Γb e $\Gamma\sigma\pi^{-1}a$ respectivamente em $\pi^{-1}\tau^{-1}$ (o inverso de $\tau\pi$) enquanto os genes restantes não são alterados pelos mesmos motivos do item anterior.
4. A lista Q é atualizada de dois modos diferentes dependendo se Q possui algum par longo ou se possui apenas pares curtos.
 - Caso Q possua um par longo então os genes a , $\sigma\pi^{-1}a$ e b da transposição generalizada τ pertencem à mesma órbita do par longo N apontado por $pLong$. Nesse caso, os genes b , $\sigma\pi^{-1}a$, $\pi\Gamma a$ e $\pi\Gamma\sigma\pi^{-1}a$ são fixos em $\sigma\pi^{-1}\tau^{-1}$ e a lista Q deve ser atualizada por meio da remoção desses elementos de suas listas circulares correspondentes. Remover b , $\sigma\pi^{-1}a$, $\pi\Gamma\sigma\pi^{-1}a$ e $\pi\Gamma a$ pode ser implementado em tempo constante, uma vez que esses elementos estão nos nós seguintes de $\sigma\pi^{-1}a$, a , $\pi\Gamma b$ e $\pi\Gamma\sigma\pi^{-1}a$ respectivamente nas listas circulares e a e $\pi\Gamma b$ são diretamente acessados usando-se $T[a].node$ e $T[P[-b]].node$. O campo tamanho do par N (i.e. $N.size$) deve ser reduzido de dois. Caso o tamanho do par tenha sido reduzido a 2, então o par N deve ser removido da lista Q e reinserido no início da lista apontada por $pShort$. Se restar apenas um elemento em cada lista circular apontado por N , então N é um par trivial e ele é removido de Q tomando-se a como argumento do procedimento de deleção.
 - Caso Q possua apenas pares curtos, considere os pares curtos N que contém os genes a e $\sigma\pi^{-1}a$ e M que contém o gene b . Nesse caso, os genes $\sigma\pi^{-1}a$ e $\pi\Gamma a$ são fixos em $\sigma\pi^{-1}\tau^{-1}$ e portanto devem ser removidos da lista Q . A remoção de $\sigma\pi^{-1}a$ e $\pi\Gamma a$ pode ser implementada usando-se os apontadores $T[a].node$ e $T[P[-[S[invP[a]]]]].node$ para os genes a e $\pi\Gamma\sigma\pi^{-1}a$ que são os nós anteriores a $\sigma\pi^{-1}a$ e $\pi\Gamma a$ respectivamente em Q . Além disso, o nó contendo a é removido do par N e inserido no par M após o nó b . Do mesmo modo, o nó contendo $\pi\Gamma\sigma\pi^{-1}a$ é removido de N e inserido em M após o nó $\pi\Gamma b$. O par N é deletado de Q ; o tamanho do par M é acrescido de 1 e M passa a ser apontado por $pLong$.

Chamaremos o procedimento que realiza a pesquisa pelos genes $\pi^{-1}a$, $\pi^{-1}\sigma\pi^{-1}a$, $\pi^{-1}b$, Γb , $\Gamma\sigma\pi^{-1}a$ e Γa em $invP$ e a atualização dos vetores P e $invP$ de `updateArrays(array P, array invP, array S, gene x, gene y, gene z)`. Usaremos como argumentos desse procedimento os genes a , $\sigma\pi^{-1}a$ e b . A Figura 5.5 mostra a implementação do procedimento de atualização dos vetores da estrutura de dados.

O procedimento que implementa a atualização da lista de pares Q , chamado `updateQ(Pairs List Q, gene x, gene y, gene z)`, é mostrado na Figura 5.8. O procedimento `updateQ(Pairs List Q, gene x, gene y, gene z)` depende das subrotinas mostradas na Figura 5.6 e Figura 5.7 que

Procedimento updateArrays(array P , array $invP$, gene x , gene y , gene z)

```

1:  $pgX \leftarrow P[-x]$ 
2:  $pgY \leftarrow P[-y]$ 
3:  $pgZ \leftarrow P[-z]$  {  $x = a, y = \sigma\pi^{-1}a, z = b$  }
   { variáveis auxiliares para  $P$  }
4:  $tpX \leftarrow y$ 
5:  $tpY \leftarrow z$ 
6:  $tpZ \leftarrow x$ 
7:  $tpgZ \leftarrow P[-y]$ 
8:  $tpgY \leftarrow P[-x]$ 
9:  $tpgX \leftarrow P[-z]$ 
   { variáveis auxiliares para  $invP$  }
10:  $ptX \leftarrow invP[z]$ 
11:  $ptY \leftarrow invP[x]$ 
12:  $ptZ \leftarrow invP[y]$ 
13:  $ptgZ \leftarrow -x$ 
14:  $ptgY \leftarrow -z$ 
15:  $ptgX \leftarrow -y$ 
   { atualiza  $P$  }
16:  $P[invP[x]] \leftarrow tpX$ 
17:  $P[invP[y]] \leftarrow tpY$ 
18:  $P[invP[z]] \leftarrow tpZ$ 
19:  $P[invP[pgZ]] \leftarrow tpgZ$ 
20:  $P[invP[pgY]] \leftarrow tpgY$ 
21:  $P[invP[pgX]] \leftarrow tpgX$ 
   { atualiza  $invP$  }
22:  $invP[x] \leftarrow ptX$ 
23:  $invP[y] \leftarrow ptY$ 
24:  $invP[z] \leftarrow ptZ$ 
25:  $invP[pgZ] \leftarrow ptgZ$ 
26:  $invP[pgY] \leftarrow ptgY$ 
27:  $invP[pgX] \leftarrow ptgX$ 

```

Figura 5.5: Procedimento de atualização dos vetores P e $invP$ em decorrência da aplicação de uma transposição generalizada. Assumimos que a ordem dos genes passados ao procedimento modifica os valores dos argumentos em $x = a, y = \sigma\pi^{-1}a$ e $z = b$. Utilizamos t (de τ), p (de π) e g (de Γ) para nomear as variáveis auxiliares usadas no procedimento. Por exemplo, a variável $tpgX$ refere-se ao gene $\tau\pi\Gamma x$. No vetor P , os elementos indexados por $\pi^{-1}x, \pi^{-1}y$ e $\pi^{-1}z$ são modificados para os elementos $\tau\pi x, \tau\pi y$ e $\tau\pi z$ respectivamente. No vetor $invP$, os elementos indexados por x, y e z são modificados para os elementos $\pi^{-1}z, \pi^{-1}x$ e $\pi^{-1}y$ respectivamente. Os elementos indexados por $\Gamma z, \Gamma y$ e Γx são modificados para $\pi\Gamma y, \pi\Gamma x$ e $\pi\Gamma z$ respectivamente em P . Os elementos indexados por $\pi\Gamma z, \pi\Gamma y$ e $\pi\Gamma x$ são modificados para os elementos $\Gamma x, \Gamma z$ e Γy em $invP$.

implementam os dois casos possíveis de atualização da lista de pares (existência ou ausência de pares longos em Q).

A Figura 5.9 mostra o algoritmo para encontrar uma seqüência de transposições generalizadas que transforma o genoma π no genoma σ em (E, Γ) .

Lema 102. Dados os genomas π, σ conorbitais no sistema de genes (E, Γ) , o algoritmo *TGsort* retorna uma seqüência τ_1, \dots, τ_r de transposições generalizadas tal que $W_{tg}(\pi, \sigma) = \|\sigma\pi^{-1}\|_3$ e a seqüência de transposições generalizadas transforma o genoma π em σ em tempo de execução $O(n)$, onde $n = |E|$.

Prova: Mostraremos que o algoritmo *TGsort* é correto por meio de uma invariante de laço baseada no parâmetro r : a igualdade $\theta = \tau_r \dots \tau_1 \pi$ e o evento de rearranjo τ_i é um evento 3-bom para $(\tau_{i-1} \dots \tau_1 \pi, \sigma)$ aplicável a $\tau_{i-1} \dots \tau_1 \pi$ são ambos verdadeiros para $1 \leq i \leq r$.

Para $r = 0$, antes do laço na linha 12, temos $\theta = \pi$ e a invariante é trivialmente válida.

Suponha que a invariante é válida para $r = p$, ou seja, temos $\theta = \tau_p \dots \tau_1 \pi$ e τ_m é um evento 3-bom para $(\tau_{m-1} \dots \tau_1 \pi, \sigma)$ e é aplicável a $\tau_{m-1} \dots \tau_1 \pi$ para $1 \leq m \leq p$. Na próxima iteração do laço na linha 21, após a escolha dos genes i, j e k , definimos a permutação $\tau_{p+1} = (x \ \sigma\theta^{-1}x \ y)(\theta\Gamma y \ \theta\Gamma\sigma\theta^{-1}x \ \theta\Gamma x)$ pois $x = L[i]$, $\sigma\theta^{-1}x = L[j]$ e $y = L[k]$. O evento de rearranjo τ_{p+1} é uma transposição generalizada pois $x, \sigma\pi^{-1}x$ e y pertencem a mesma órbita de σ pela escolha de i, j e k feita na linha 15 ou na linha 18. Além disso, a transposição generalizada τ_{p+1} é um evento 3-bom para (θ, σ) pelo Lema 101. Logo a invariante é satisfeita antes da próxima iteração na linha 12.

Ao fim de $r = \|\sigma\pi^{-1}\|_3/2$ iterações do laço, temos $\theta = \tau_r \dots \tau_1 \pi$. Como τ_m é um evento 3-bom para $(\tau_{m-1} \dots \tau_1 \pi, \sigma)$ e é aplicável a $\tau_{m-1} \dots \tau_1 \pi$ para $1 \leq m \leq r$ então $r = W_{tg}(\pi, \sigma)/2$ pelo Lema 98. Logo $\theta = \sigma$, ou, em outros termos, temos $\sigma\theta^{-1} = \iota$. Se $\sigma\theta^{-1} = \iota$ então a lista de pares Q é vazia porque ela não contém pares triviais. Logo a condição do laço na linha 12 é falsa e a linha 25 é executada terminando o algoritmo. Portanto temos $\sigma = \tau_r \dots \tau_1 \pi$ e τ_1, \dots, τ_r é uma seqüência de transposições generalizadas, tal que τ_m é um evento 3-bom para $(\tau_{m-1} \dots \tau_1 \pi, \sigma)$ e τ_m é aplicável a $\tau_{m-1} \dots \tau_1 \pi$, para $1 \leq m \leq r$, transformando o genoma π no genoma σ .

O tempo de execução total do algoritmo *TGSort* é a somatória dos tempos de inicialização da estrutura de dados e o produto do número de iterações do laço *while* na linha 12 pelo tempo gasto para escolher um evento de rearranjo, atualizar a lista de pares Q e atualizar os vetores P e $invP$. Pelo Lema 77, a inicialização dos vetores $P, S, invP$ e T ; além da tabela de espalhamento H pode ser realizada em tempo de execução linear no número de genes $n = |E|$. Como a estrutura de dados é modificada para atender às características específicas do problema de rearranjo por transposições generalizadas então é preciso avaliar qual é o tempo de execução da construção de Q e da lista SL . A inicialização de Q toma tempo de execução $O(n)$ para criar cada nó de cada par incluído na lista duplamente ligada, pois encontrar $S[invP[i]]$ para

Procedimento updateLongPairQ(Pairs List Q , gene x , gene y , gene z)

```

    {  $x \leftarrow a, y \leftarrow \sigma\pi^{-1}a, z \leftarrow b$  }
1:  $pgX \leftarrow P[-x]$ 
2:  $pgY \leftarrow P[-y]$ 
3:  $pgZ \leftarrow P[-z]$ 
4:  $remove(Q, y, z)$ 
5:  $T[z].node \leftarrow NIL$  { remove  $z$  }
6:  $remove(Q, x, y)$ 
7:  $T[y].node \leftarrow NIL$  { remove  $y$  }
8:  $remove(Q, pgY, pgX)$ 
9:  $T[pgX].node \leftarrow NIL$  { remove  $pgX$  }
10:  $remove(Q, P[-z], pgY)$ 
11:  $T[pgY].node \leftarrow NIL$  { remove  $pgY$  }
12:  $N = T[x].pair;$ 
13:  $N.size \leftarrow N.size - 2$ 
14: if  $N.size = 2$  then
15:    $Q.pLong \leftarrow N.next$ 
16:   if  $Q.pLong \rightarrow size < 3$  then
17:      $Q.pLong \leftarrow NIL$ 
18:   end if
19:   Remove o par  $N$  da lista de pares longos;
20:   Insere o par  $N$  na lista de pares curtos;
21:    $pShort \leftarrow N$ 
22: end if
23: if  $N.size = 1$  then
24:    $remove(Q, x, x)$  { deleção do par  $N$  }
25:    $T[x].node \leftarrow NIL$ 
26:    $remove(Q, P[-z], P[-z])$ 
27:    $T[P[-z]].node \leftarrow NIL$ 
28:    $delete(N)$ 
29: end if
```

Figura 5.6: Procedimento de atualização da lista de pares Q quando há um par longo. O procedimento remove os elementos $z, y, \pi\Gamma x$ e $\pi\Gamma y$ do par longo N que os contém, decrementa de 2 o campo tamanho do par e avalia alguns possíveis casos. Se o tamanho do par foi reduzido para 2 então esse par é movido para o início do trecho da lista composto por pares curtos, enquanto $pLong$ aponta para o próximo no trecho de pares longos caso ainda houver algum par longo, caso contrário $pLong$ aponta para NIL . Se o tamanho do par N foi decrescido até atingir 1 então seus elementos são removidos e o par é deletado de Q .

Procedimento updateShortPairQ(Pairs List Q , gene x , gene y , gene z)

$\{x = a, y = \sigma\pi^{-1}a, z = b\}$

- 1: $pgX \leftarrow P[-x]$
- 2: $pgY \leftarrow P[-y]$
- 3: $pgZ \leftarrow P[-z]$
- 4: $N \leftarrow T[x].pair$
- 5: $M \leftarrow T[z].pair$ $\{y = \sigma\pi^{-1}a$ e $pgX = \pi\Gamma a$ são removidos de $N\}$
- 6: $remove(Q, x, y)$
- 7: $T[y].node \leftarrow NIL$ $\{\text{remove } y\}$
- 8: $remove(Q, pgY, pgX)$
- 9: $T[pgX].node \leftarrow NIL$ $\{\text{remove } pgX\}$
- 10: Criar um novo nó A com o gene x
- 11: Inserir A após o nó contendo z em M
- 12: $T[x].node \leftarrow A$
- 13: $T[x].pair \leftarrow M$
- 14: Criar um novo nó B com o gene pgY
- 15: Inserir B após o nó contendo pgZ em M
- 16: $T[pgY].node \leftarrow B$
- 17: $T[pgY].pair \leftarrow M$ $\{\text{deleção do par } N\}$
- 18: $remove(Q, x, x)$
- 19: $T[x].node \leftarrow NIL$
- 20: $remove(Q, pgY, pgY)$
- 21: $T[pgY].node \leftarrow NIL$
- 22: $delete(N)$
- 23: $M.size \leftarrow M.size + 1$
- 24: $Q.pLong \leftarrow M$

Figura 5.7: Procedimento de atualização da lista de pares Q quando há apenas pares curtos. Os elementos y e $\pi\Gamma x$ são removidos do par N que os contém, um novo nó A contendo z é criado e inserido após o nó contendo z no par M e a entrada $T[x]$ é modificada adequadamente. Um novo nó B contendo $\pi\Gamma y$ é inserido após o nó contendo $\pi\Gamma z$ no par M e a entrada $T[\pi\Gamma y]$ é modificada adequadamente. Os elementos remanescentes no par N são removidos e o par N é deletado da lista Q . Finalmente o campo tamanho do par M é incrementado de 1 e o apontador $pLong$ aponta para o novo par longo M .

Procedimento updateQ(Pairs List Q , gene x , gene y , gene z)
 $\{ x = a, y = \sigma\pi^{-1}a, z = b \}$
1: $pgX \leftarrow P[-x]$
2: $pgY \leftarrow P[-y]$
3: $pgZ \leftarrow P[-z]$
4: **if** $Q.pLong! = NIL$ **then**
5: updateLongPairQ(Pairs List Q , gene x , gene y , gene z)
6: **else**
7: updateShortPairQ(Pairs List Q , gene x , gene y , gene z)
8: **end if**

Figura 5.8: Procedimento de atualização da lista de pares Q . O procedimento avalia se existe algum par longo em Q e dependendo desse resultado chama os procedimentos auxiliares *updateLongPairQ* ou *updateShortPairQ* respectivamente quando houver algum par longo na lista ou quando não houver nenhum par longo em Q .

$i \in \{x \in \mathbb{Z} \mid 1 \leq |x| \leq |E_+|\}$ envolve uma pesquisa em $invP$ e S que leva $O(1)$ e a atribuição dos apontadores na posição $T[i]$ que leva tempo constante. Além disso, o fato de mantermos os campos com os tamanhos dos pares e os apontadores $pLong$ e $pShort$ não altera o tempo de execução assintótico final da inicialização de Q , pois o campo de tamanho é incrementado a cada inserção de um gene em um par e cada par é inserido na lista duplamente ligada Q no início do trecho dos pares longos ou curtos de acordo com o seu tipo. A inicialização de SL consiste em para um dado nó criado com o gene i (índice do gene $L[i]$) incluir os genes $S[i]$, $S[S[i]]$ e assim sucessivamente, desde que cada gene pertença ao suporte de $\sigma\pi^{-1}$. A verificação de que um gene pertence ao suporte do quociente pode ser implementada identificando-se a presença do gene em Q , o que pode ser implementado em tempo constante usando-se os apontadores no vetor T .

Como o número de eventos de rearranjos encontrados pelo algoritmo é mínimo, então o número de iterações do laço *while* no algoritmo *TGSort* é $d_{tg}(\pi, \sigma)$. No pior caso, apenas um elemento de cada fita do genoma será colocado em sua posição adequada em relação ao outro genoma por iteração do laço na linha 12, i.e. o bloco de controle *while* será executado $O(n)$ vezes. Para cada passo no laço *while* verifica-se se os genomas θ e σ são iguais na linha 12. Essa verificação pode ser efetuada em tempo constante ao constatar se a lista Q está vazia ou não. Se Q está vazia então $\theta = \sigma$ pois $\sigma\theta^{-1} = \iota$. Escolher os elementos i, j e k para os quais a transposição generalizada $(L[i] \ L[j] \ L[k])(\theta\Gamma L[k] \ \theta\Gamma L[j] \ \theta\Gamma L[i])$ seja um evento 3-bom para (θ, σ) é executado em tempo constante. Verificamos primeiramente se há pares longos em Q , ou seja, temos $pLong \neq NIL$ e nesse caso escolhemos os primeiros três elementos em uma das listas circulares do par apontado por $pLong$ como os elementos i, j e k respectivamente. Caso contrário, se houver apenas pares curtos, então escolhemos três elementos consecutivos na lista

Algoritmo TGSORT(Genoma π , genoma σ , Genes L)

```

1:  $r \leftarrow 0$ 
2: for all  $i \in \{j \mid 1 \leq |j| \leq |E_+|\}$  do
3:    $T[L[i]].index \leftarrow i$ 
4: end for
5:  $P \leftarrow initialize(\pi, T, L)$ 
6: for all  $i \in \{j \mid 1 \leq |j| \leq |E_+|\}$  do
7:    $invP[P[i]] \leftarrow i$ 
8: end for
9:  $S \leftarrow initialize(\sigma, T, L)$ 
10:  $initialize(\sigma, SL)$ 
11:  $initializePairsList(Q, S, invP)$  {Considere  $\theta = \pi$ }
12: while  $notEmpty(Q)$  do
13:    $r++$  { Escolha dos genes  $x, \sigma\theta^{-1}x$  e  $y$  }
14:   if  $pLong \neq NIL$  then
15:     Tome os três primeiros genes  $i, j$  e  $k = S[invP[i]]$  em  $pLong$ .
16:     Remova  $j$  e  $k$  de  $SL$ .
17:   else
18:     Tome os 3 primeiros genes  $i, j$  e  $k$  em  $SL$ .
19:     Remova  $i, j$  e  $k$  de  $SL$ .
20:   end if
21:   Faça  $\tau_r = (L[i] \ L[j] \ L[k])(L[P[-k]] \ L[P[-j]] \ L[P[-i]])$  { faça  $\theta = \tau_r\pi$  }
22:    $updateQ(Q, i, j, k)$ 
23:    $updateArrays(P, invP, S, i, j, k)$ 
24: end while
25: return  $\tau_1, \dots, \tau_r$ 

```

Figura 5.9: Dados os genomas π, σ em (E, Γ) , o algoritmo TGSORT retorna uma seqüência de transposições generalizadas com distância genômica ponderada mínima $W_{tg}(\pi, \sigma)$ que transforma o genoma π em σ .

SL. Em ambos os casos um número constante de apontadores é verificado. Além disso, a lista duplamente ligada *SL* é modificada em tempo constante pois no máximo seis de seus nós são removidos e esses nós são acessados rapidamente a partir de apontadores em *T*. Pelo Lema 101, a transposição generalizada $(L[i] \ L[j] \ L[k])(\theta\Gamma L[k] \ \theta\Gamma L[j] \ \theta\Gamma L[i])$ aplicável a θ é um evento 3-bom para (θ, σ) onde θ é um genoma representado pelo vetor *P* na iteração corrente do laço. A atualização dos vetores *P* e *invP* é realizada em tempo constante pois envolve apenas um número constante de atribuições que devem ser realizadas como mostrado na Figura 5.5. Atualizar a lista de pares *Q* envolve a remoção de um número constante de genes que são acessados rapidamente utilizando-se os apontadores em *T*; a eliminação e criação de pares novos; e a atualização dos apontadores em *T*, *pLong* e *pShort*. Remover $\sigma\pi^{-1}i$ e $\pi\Gamma i$ pode ser implementado em tempo constante usando-se os apontadores $T[i].node$ e $T[P[-[S[invP[i]]]]].node$ para os genes *i* e $\pi\Gamma\sigma\pi^{-1}i$ que são os nós anteriores a $\sigma\pi^{-1}i$ e $\pi\Gamma i$ respectivamente em *Q*. Do mesmo modo, a reinserção de *i* após o nó contendo *k* e do nó contendo $\pi\Gamma\sigma\pi^{-1}i$ após o nó $\pi\Gamma k$ no caso em que há apenas pares curtos em *Q* é feita em tempo constante pois são apenas operações regulares sobre listas ligadas. Os apontadores em *T* e os de *Q* (*pLong* e *pShort*) são atualizados convenientemente de acordo com a Figura 5.8 e como não envolvem nenhum tipo de percurso nas listas ou vetores tomam tempo $O(1)$.

Portanto o tempo de execução total do algoritmo *TGSort* é $O(n)$. \square

Teorema 103. *Dados os genomas conorbitais π e σ em (E, Γ) temos*

$$W_{tg}(\pi, \sigma) = \|\sigma\pi^{-1}\|_3.$$

Prova: Dados os genomas conorbitais π e σ em (E, Γ) , o Lema 102 garante a existência de uma seqüência de transposições generalizadas τ_1, \dots, τ_k tais que $\tau_k \dots \tau_1 \pi = \sigma$ e τ_i é aplicável a $\tau_{i-1} \dots \tau_1 \pi$ e τ_i é um evento 3-bom para $(\tau_{i-1} \dots \tau_1 \pi, \sigma)$ para $1 \leq i \leq k$. Além disso, pelo Lema 98, temos

$$\sum_{i=1}^k w(\tau_i) \geq \|\sigma\pi^{-1}\|_3.$$

Portanto $W_{tg}(\pi, \sigma) = \|\sigma\pi^{-1}\|_3$. \square

5.2 Fissão, Fusão e Reversão com Sinais

Dados os genomas π e σ em um sistema de genes (E, Γ) , o *problema de rearranjo em genomas ponderado envolvendo fissões, fusões e reversões com sinais* consiste em encontrar uma seqüência de fissões, fusões e reversões com sinais ρ_1, \dots, ρ_k , tal que

$$\rho_k \dots \rho_1 \pi = \sigma,$$

onde ρ_i é aplicável a $\rho_{i-1} \dots \rho_1 \pi$ para $1 \leq i \leq k$ e $\sum_{i=1}^k w(\rho_i)$ é mínimo. O valor $\sum_{i=1}^k w(\rho_i)$ é chamado de *distância genômica ponderada de rearranjo baseada em fissões, fusões e reversões com sinais* entre os genomas π e σ e é denotado por $W_{fssr}(\pi, \sigma)$.

Nessa seção, propomos um algoritmo para encontrar uma sequência de peso mínimo de fissões, fusões e reversões com sinais que transforma um genoma em outro [45]. O algoritmo baseia-se na representação dos genomas por meio da estrutura de dados discutida no Capítulo 4. A complexidade de tempo de execução total do algoritmo é $O(n)$, onde n é o número de genes (assumimos que o tempo gasto para acessar a tabela de espalhamento da estrutura de dados é constante).

Também apresentamos uma fórmula para o distância genômica ponderada baseada em fissões, fusões e reversões com sinais que pode ser calculado em tempo de execução $O(n)$.

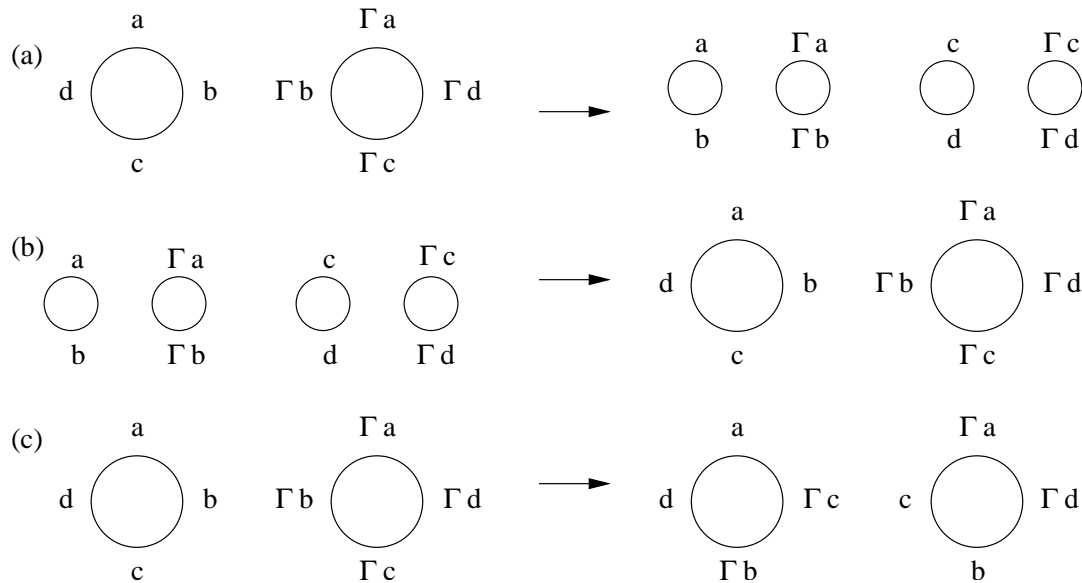


Figura 5.10: Eventos de rearranjo em genomas vistos como operações de quebra-junção em ciclos. Todos os ciclos são orientados no sentido horário. (a) O genoma $\pi = (a \ b \ c \ d)(\Gamma d \ \Gamma c \ \Gamma b \ \Gamma a)$ é transformado no genoma $\rho\pi = (a \ b)(\Gamma b \ \Gamma a)(c \ d)(\Gamma d \ \Gamma c)$ pela fissão $\rho = (a \ c)(\Gamma b \ \Gamma d)$. (b) O genoma $\pi = (a \ b)(\Gamma b \ \Gamma a)(c \ d)(\Gamma d \ \Gamma c)$ é transformado no genoma $\rho\pi = (a \ b \ c \ d)(\Gamma d \ \Gamma c \ \Gamma b \ \Gamma a)$ pela fusão $\rho = (a \ c)(\Gamma d \ \Gamma b)$. (c) O genoma $\pi = (a \ b \ c \ d)(\Gamma d \ \Gamma c \ \Gamma b \ \Gamma a)$ é transformado no genoma $\rho\pi = (a \ \Gamma c \ \Gamma b \ d)(\Gamma d \ b \ c \ \Gamma a)$ pela reversão com sinais $\rho = (b \ \Gamma b)(d \ \Gamma a)$.

Os eventos de rearranjo fusão, fissão e reversão com sinais podem ser vistos como operações de “quebra” e “junção” em ciclos como ilustrado na Figura 5.10. Essas operações de quebra e junção são definidas de acordo com a distribuição dos elementos no suporte do evento de rearranjo entre as órbitas do genoma no qual é aplicável. Como fusões, fissões e reversões com sinais possuem a mesma estrutura de ciclos, é possível tirar vantagem dessa propriedade e do

modo que os elementos do suporte de um evento de rearranjo são distribuídos nas órbitas de um genoma para identificar um evento 2-bom para um par de genomas.

Lema 104. Dados quaisquer genomas distintos π, σ no sistema de genes (E, Γ) , há uma fusão, fissão ou reversão com sinais ρ aplicável a π tal que ρ é um evento 2-bom para (π, σ) .

Prova:

Considere a permutação $\rho = (x \sigma \pi^{-1} x)(\pi \Gamma \sigma \pi^{-1} x \pi \Gamma x)$ onde x e $\sigma \pi^{-1} x$ são elementos de E . Há três casos a serem considerados dependendo de em quais órbitas de π os elementos $x, \sigma \pi^{-1} x, \pi \Gamma \sigma \pi^{-1} x, \pi \Gamma x$ pertencem:

- Se $\sigma \pi^{-1} x \in orb(\pi, x)$ então $\rho = (x \sigma \pi^{-1} x)(\pi \Gamma \sigma \pi^{-1} x \pi \Gamma x)$ é uma fissão aplicável a π pois $(x \sigma \pi^{-1} x)|\pi$ pelo Lema 43.
- Se $\sigma \pi^{-1} x \notin orb(\pi, x)$ e x e $\sigma \pi^{-1} x$ pertencem a órbitas do mesmo cromossomo então vamos mostrar que $\rho = (x \sigma \pi^{-1} x)(\pi \Gamma \sigma \pi^{-1} x \pi \Gamma x)$ é uma reversão com sinais aplicável a π e ela é um evento 2-bom para (π, σ) . Os elementos x e $\sigma \pi^{-1} x$ pertencem a fitas distintas do genoma π e temos $\Gamma \pi^{-1} \sigma \pi^{-1} x = \pi \Gamma \sigma \pi^{-1} x$ pela propriedade fundamental de genomas $\Gamma \pi \Gamma = \pi^{-1}$, então $\pi \Gamma \sigma \pi^{-1} x \in orb(\pi, x)$ e logo $(x \pi \Gamma \sigma \pi^{-1} x)|\pi$ pelo Lema 43. Além disso, temos $x \neq \pi \Gamma \sigma \pi^{-1} x$ porque caso contrário $x = \pi \Gamma \sigma \pi^{-1} x$ implica $\pi \Gamma x = \sigma \Gamma \pi \Gamma x$ e então existe um elemento $z \in E$ tal que $\sigma \Gamma z = z$, ou seja, dois genes complementares pertencem à mesma fita de σ , o que contradiz a definição de um genoma no mesmo sistema de genes (E, Γ) . Portanto, o evento de rearranjo $(x \sigma \pi^{-1} x)(\pi \Gamma \sigma \pi^{-1} x \pi \Gamma x)$ é uma reversão com sinais aplicável a π .
- Se x e $\sigma \pi^{-1} x$ pertencem a órbitas de cromossomos distintos de π então o evento de rearranjo $\rho = (x \sigma \pi^{-1} x)(\pi \Gamma \sigma \pi^{-1} x \pi \Gamma x)$ é uma fusão aplicável a π pois $(x \sigma \pi^{-1} x) \nmid \pi$ pelo Lema 43.

Como $\sigma \pi^{-1}$ é um produto de pares $\alpha \pi \Gamma \alpha^{-1} \pi \Gamma$ então para qualquer ciclo $(a_1 \dots a_m)$ de $\sigma \pi^{-1}$ existe um ciclo $(\pi \Gamma a_m \dots \pi \Gamma a_1)$ de $\sigma \pi^{-1}$ e logo $\pi \Gamma x \notin orb(\sigma \pi^{-1}, x)$. Além disso, como $\sigma \pi^{-1} x \in orb(\sigma \pi^{-1}, x)$, $\pi \Gamma \sigma \pi^{-1} x \in orb(\sigma \pi^{-1}, \pi \Gamma x)$ e $\pi \Gamma x \notin orb(\sigma \pi^{-1}, x)$ então $\rho|\sigma \pi^{-1}$. Logo o evento de rearranjo ρ é um evento 2-bom para (π, σ) . \square

Discutimos agora como implementar fissões, fusões e reversões com sinais usando operações da estrutura de dados. Dados π e σ em (E, Γ) , restringimos a implementação a eventos de rearranjo da forma $(a \sigma \pi^{-1} a)(\pi \Gamma \sigma \pi^{-1} a \pi \Gamma a)$. Todos esses eventos de rearranjo são eventos bons para (π, σ) . Fazemos tal restrição para garantir que cada evento de rearranjo pode ser realizado em tempo de execução constante. O evento de rearranjo $\rho = (a \sigma \pi^{-1} a)(\pi \Gamma \sigma \pi^{-1} a \pi \Gamma a)$ aplicável ao genoma π é implementado conforme explicado abaixo:

1. Os elementos $a, \sigma \pi^{-1} a, \pi \Gamma \sigma \pi^{-1} a$ e $\pi \Gamma a$ são pesquisados para se encontrar os elementos $\pi^{-1} a, \pi^{-1} \sigma \pi^{-1} a, \pi^{-1} \pi \Gamma \sigma \pi^{-1} a$ e $\pi^{-1} \pi \Gamma a$ em $invP$.

2. Atualização de P . Os genes $\pi^{-1}a$, $\pi^{-1}\sigma\pi^{-1}a$, $\pi^{-1}\pi\Gamma\sigma\pi^{-1}a$ e $\pi^{-1}\pi\Gamma a$ são mapeados em $\sigma\pi^{-1}a$, a , $\pi\Gamma a$ e $\pi\Gamma\sigma\pi^{-1}a$ respectivamente em $\rho\pi$, enquanto os genes restantes não são alterados, porque no produto $\alpha\beta$ o elemento $\beta^{-1}x$ é mapeado no elemento y diferente de x quando x pertence ao suporte de α , caso contrário $\alpha x = x$ e $\alpha\beta\beta^{-1}x = x$. Então, os genes $\pi^{-1}a$, $\pi^{-1}\sigma\pi^{-1}a$, $\pi^{-1}\pi\Gamma\sigma\pi^{-1}a$ e $\pi^{-1}\pi\Gamma a$ em P são atualizados para $\sigma\pi^{-1}a$, a , $\pi\Gamma a$ e $\pi\Gamma\sigma\pi^{-1}a$ respectivamente em P' , onde P' representa $\rho\pi$. Como $P[\pi^{-1}a] = a$ e $P[\pi^{-1}\sigma\pi^{-1}a] = \sigma\pi^{-1}a$ então podemos trocar os conteúdos de $P[\pi^{-1}a]$ e $P[\pi^{-1}\sigma\pi^{-1}a]$ para obter as imagens corretas de $\pi^{-1}a$ e $\pi^{-1}\sigma\pi^{-1}a$ em P' . Similarmente, temos $P[\pi^{-1}\pi\Gamma\sigma\pi^{-1}a] = \pi\Gamma\sigma\pi^{-1}a$ e $P[\pi^{-1}\pi\Gamma a] = \pi\Gamma a$, então trocando $P[\pi^{-1}\pi\Gamma\sigma\pi^{-1}a]$ e $P[\pi^{-1}\pi\Gamma a]$ atribui as imagens de $\Gamma\sigma\pi^{-1}a$ e Γa em P' . A Figura 5.13 ilustra a troca dos elementos em P para atualizá-lo em um dado exemplo.
3. Atualização de $invP$. Os genes a , $\sigma\pi^{-1}a$, $\pi\Gamma\sigma\pi^{-1}a$ e $\pi\Gamma a$ são mapeados nos elementos $\pi^{-1}\sigma\pi^{-1}a$, $\pi^{-1}a$, $\pi^{-1}\pi\Gamma a$ e $\pi^{-1}\pi\Gamma\sigma\pi^{-1}a$ respectivamente em $\pi^{-1}\rho^{-1}$ (o inverso de $\rho\pi$) enquanto os genes restantes não são alterados pelos mesmos motivos do item anterior. Os genes em $invP[a]$, $invP[\sigma\pi^{-1}a]$, $invP[\pi\Gamma\sigma\pi^{-1}a]$ e $invP[\pi\Gamma a]$ tem de ser atualizados para $\pi^{-1}\sigma\pi^{-1}a$, $\pi^{-1}a$, $\pi^{-1}\pi\Gamma a$ e $\pi^{-1}\pi\Gamma\sigma\pi^{-1}a$ respectivamente em $invP'$, onde $invP'$ representa o genoma $\pi^{-1}\rho^{-1}$. Como $invP[a] = \pi^{-1}a$ e $invP[\sigma\pi^{-1}a] = \pi^{-1}\sigma\pi^{-1}a$ então $invP$ é atualizado em $invP'$ trocando-se os genes $invP[a]$ e $invP[\sigma\pi^{-1}a]$. Similarmente, temos $invP[\pi\Gamma\sigma\pi^{-1}a] = \pi^{-1}\pi\Gamma\pi^{-1}a$ e $invP[\pi\Gamma a] = \pi^{-1}\pi\Gamma a$, então trocando $invP[\pi\Gamma a]$ e $invP[\pi\Gamma\sigma\pi^{-1}a]$ atribui as imagens dos genes $\pi\Gamma a$ e $\pi\Gamma\sigma\pi^{-1}a$ em $invP'$. A Figura 5.13 mostra os elementos que devem ser trocados em $invP$ em um exemplo particular.
4. A lista Q é atualizada por meio de uma remoção dos elementos $\sigma\pi^{-1}a$ e $\pi\Gamma a$ das suas listas circulares correspondentes. Remover $\sigma\pi^{-1}a$ e $\pi\Gamma a$ pode ser implementado em tempo constante, uma vez que ambos elementos estão nos nós seguintes de a e $\pi\Gamma\sigma\pi^{-1}a$ respectivamente nas listas circulares e a e $\pi\Gamma\sigma\pi^{-1}a$ são diretamente acessados usando $T[a].node$ e $T[\pi\Gamma\sigma\pi^{-1}a].node$. Se restar apenas um elemento em cada lista circular apontado por um nó de Q então esse par é um par trivial e ele é removido de Q tomando-se a como um parâmetro do procedimento de deleção.

O procedimento de atualização dos vetores P e $invP$ é mostrado na Figura 5.11. Utilizamos um procedimento auxiliar chamado $exchange(A, x, y)$ que troca o conteúdo das posições x e y do vetor A .

A Figura 5.13 mostra a ação de um evento de rearranjo como uma fissão, fusão ou reversão com sinais nos vetores P e $invP$. No exemplo ilustrado pela figura, os vetores P e $invP$ representam o genoma $\pi = (-3 \ 2 \ -6 \ -9 \ 5)(-5 \ 9 \ 6 \ -2 \ 3)(-7 \ -4 \ -8 \ 1)(-1 \ 8 \ 4 \ 7)$ em um sistema de genes (E, Γ) , onde por simplicidade, trocamos os nomes dos genes pelos seus

Procedimento `updateArrays(Array P , Array $invP$, gene i , gene j)`

- 1: `exchange(P , $P[invP[i]]$, $P[invP[j]]$)`
- 2: `exchange(P , $P[-j]$, $P[-i]$)`
- 3: `exchange($invP$, $invP[i]$, $invP[j]$)`
- 4: `exchange($invP$, $invP[P[-j]]$, $invP[P[-i]]$)`

Figura 5.11: Procedimento de atualização dos vetores P e $invP$. Os elementos indexados por $invP[i]$ e $invP[j]$ são trocados no vetor P , bem como os elementos indexados por $P[-j]$ e $P[-i]$. No vetor $invP$, os elementos indexados por $invP[i]$ e $invP[j]$ e os indexados por $P[-j]$ e $P[-i]$ são trocados.

Procedimento `updateQ(Pair List Q , Array T , gene i , gene j)`

- 1: `remove(Q , j)`
- 2: `remove(Q , $P[-i]$)`
- 3: `$T[f(L[j])].node = NIL$`
- 4: `$T[f(L[P[-i]])].node = NIL$`
- 5: **if** `trivialPair(Q , i)` **then**
- 6: `pair = $T[i].pair$`
- 7: `remove(Q , i)`
- 8: `remove(Q , $P[-j]$)`
- 9: `delete(pair)`
- 10: **end if**

Figura 5.12: Procedimento de atualização da lista de pares Q . Os genes j e $\pi\Gamma i$ são removidos de Q e suas entradas em T são modificadas para apontar para NIL . Se o par ao qual i pertence é trivial então esse par é deletado da lista de pares.

índices. Os vetores P' e $invP'$ representam o genoma $\rho\pi$ onde $\rho = (1 \ -7)(-1 \ 8)$ é uma fissão aplicável a π . No item (a) da figura os genes $P[7]$ e $P[-1]$ são trocados, bem como o conteúdo dos elementos $P[1]$ e $P[-8]$ no vetor P ; enquanto no item (b) os genes $invP[1]$ e $invP[-7]$ são trocados, bem como o conteúdo dos elementos $invP[8]$ e $invP[-1]$.

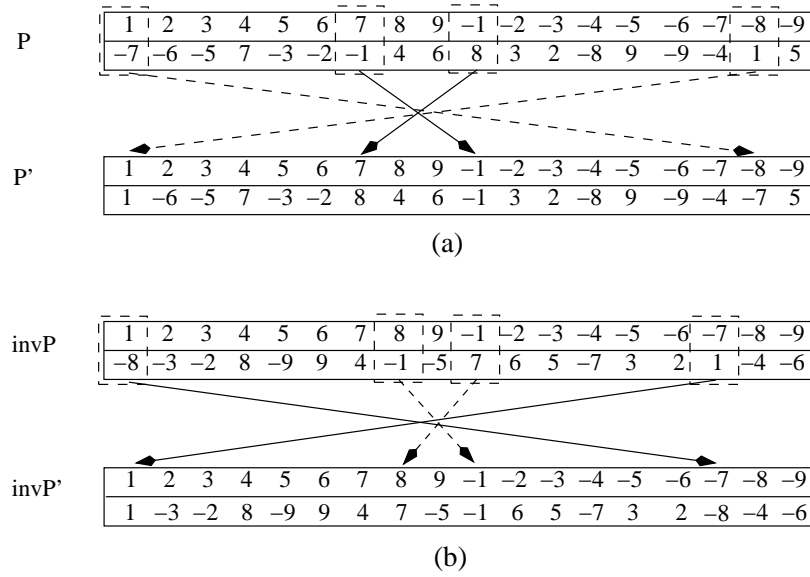


Figura 5.13: Exemplo da ação de um evento de rearranjo de dois 2-ciclos (fissões, fusões e reversões com sinais) sobre os vetores P e $invP$. Os vetores P e $invP$ são baseados no genoma $\pi = (-3 \ 2 \ -6 \ -9 \ 5)(-5 \ 9 \ 6 \ -2 \ 3)(-7 \ -4 \ -8 \ 1)(-1 \ 8 \ 4 \ 7)$ em um sistema de genes (E, Γ) . Os vetores P' e $invP'$ representam o genoma $\rho\pi$ onde $\rho = (1 \ -7)(-1 \ 8)$ é uma fissão aplicável a π . (a) Os genes $P[7]$ e $P[-1]$ são trocados, bem como o conteúdo dos elementos $P[1]$ e $P[-8]$ no vetor P . (b) No vetor $invP$, os genes $invP[1]$ e $invP[-7]$ são trocados, bem como o conteúdo dos elementos $invP[8]$ e $invP[-1]$.

O algoritmo *FFSRSort* que encontra uma sequência de peso mínimo de fusões, fissões e reversões com sinais que transforma o genoma π no genoma σ ambos no sistema de genes (E, Γ) é mostrado na Figura 5.14.

Lema 105. Dados os genomas π, σ em (E, Γ) , o algoritmo *FFSRSort* retorna uma sequência de fissões, fusões e reversões com sinais que são eventos bons para (π, σ) com distância genômica ponderada mínima $W_{ffsr}(\pi, \sigma)$ que transforma o genoma π em σ em tempo de execução $O(n)$, onde $n = |E|$.

Prova: Iremos mostrar que o algoritmo *FFSRSort* é correto por meio da seguinte definição de uma invariante de laço no parâmetro r : tem-se $\theta = \rho_r \dots \rho_1 \pi$ e o evento de rearranjo ρ_i é um evento 2-bom para $(\rho_{i-1} \dots \rho_1 \pi, \sigma)$ aplicável a $\rho_{i-1} \dots \rho_1 \pi$, para $1 \leq i \leq r$.

Para $r = 0$, antes do laço na linha 11, temos $\theta = \pi$ e a invariante é trivialmente válida.

Procedimento Algorithm FFSRSort(Genoma π , Genoma σ , Genes L)

```

1:  $r = 0$ 
2: for all  $i \in \{j \mid 1 \leq |j| \leq |E_+|\}$  do
3:    $T[L[i]].index \leftarrow i$ 
4: end for
5:  $P \leftarrow initialize(\pi, T, L)$ 
6: for all  $i \in \{j \mid 1 \leq |j| \leq |E_+|\}$  do
7:    $invP[P[i]] \leftarrow i$ 
8: end for
9:  $S \leftarrow initialize(\sigma, T, L)$ 
10:  $initializePairsList(Q, S, invP)$ 
11: while  $notEmpty(Q)$  do
12:    $r++$  {Escolha os genes  $x$  e  $\sigma\theta^{-1}x$  no primeiro par apontado por  $Q$ }
13:   Tome  $i$  e  $j$  em  $Q$ 
14:   Faça  $\rho_r \leftarrow (L[i] \ L[j])(L[P[-j]] \ L[P[-i]])$ 
15:    $updateQ(Q, T, i, j) \ \{\theta = \rho_r\theta\}$ 
16:    $updateArrays(P, invP, i, j)$ 
17: end while
18: return  $\rho_1, \dots, \rho_r$ 

```

Figura 5.14: Dados os genomas π, σ em (E, Γ) , o algoritmo FFSRsort retorna uma seqüência de fissões, fusões e reversões com sinais com distância genômica ponderada mínima $W_{ffsr}(\pi, \sigma)$ que transforma o genoma π em σ .

Suponha que a invariante é válida para $r = k$, ou seja, temos $\theta = \rho_k \dots \rho_1 \pi$ e ρ_i é um evento 2-bom para $(\rho_{i-1} \dots \rho_1 \pi, \sigma)$ e é aplicável a $\rho_{i-1} \dots \rho_1 \pi$ para $1 \leq i \leq k$. Na próxima iteração do laço na linha 14 temos $\rho_{k+1} = (x \sigma \theta^{-1} x)(\theta \Gamma \sigma \theta^{-1} x \theta \Gamma x)$ pois $x = L[i]$ e $\sigma \theta^{-1} x = L[j]$. O evento de rearranjo ρ_{k+1} é um evento 2-bom para (θ, σ) pelo Lema 104. Logo a invariante permanece válida antes da próxima iteração do laço na linha 11.

Após $r = \|\sigma \pi^{-1}\|/2$ iterações do laço, temos $\theta = \rho_r \dots \rho_1 \pi$. Como ρ_m é um evento 2-bom para $(\rho_m \dots \rho_1 \pi, \sigma)$ e é um evento aplicável a $\rho_{m-1} \dots \rho_1 \pi$ para $1 \leq m \leq r$, então $r = W_{ffsr}(\pi, \sigma)$ pelo Lema 75. Logo $\theta = \sigma$, i.e. $\sigma \theta^{-1} = \iota$. Se $\sigma \theta^{-1} = \iota$ então a lista Q é vazia porque ela não contém pares triviais. Logo a condição na linha 11 é falsa e o algoritmo executa o código na linha 18. Portanto o algoritmo encontra uma seqüência de fusões, fissões e reversões com sinais ρ_1, \dots, ρ_r atendendo a invariante que transforma o genoma π no genoma σ .

Agora dicutimos qual é o tempo de execução do algoritmo *FFSRsort*. Pelo Lema 77, a inicialização da tabela de espalhamento H , dos vetores P , S , $invP$ e T leva o tempo $O(n)$. Após ter sido construída, a tabela de espalhamento não é mais acessada durante a execução do algoritmo pois as operações de rearranjo são escolhidas de acordo com as posições dos inteiros armazenados na lista Q .

Contruir a lista Q leva tempo de execução $O(n)$ porque encontrar $S[invP[i]]$ para $i \in \{x \in \mathbb{Z} \mid 1 \leq |x| \leq |E_+|\}$ envolve uma pesquisa em $invP$ e S que leva $O(1)$ e a atribuição dos apontadores na posição $T[i]$ que leva tempo constante.

O número de iterações do laço *while* é igual a distância $d_{ffsr}(\pi, \sigma)$ pelo Lema 75, pois a cada iteração é escolhido um evento 2-bom para (θ, σ) . No pior caso, apenas um elemento de cada fita do genoma será colocado em sua posição adequada em relação ao outro genoma por iteração do laço na linha 11, i.e. o bloco de controle *while* será executado $O(n)$ vezes. Para cada passo no laço *while* verifica-se se os genomas θ e σ são iguais na linha 11. Essa verificação pode ser efetuada em tempo constante ao constarmos se a lista Q está vazia ou não. Se Q está vazia então $\theta = \sigma$ pois $\sigma \pi^{-1} = \iota$. Encontrar uma fissão, fusão ou reversão com sinais que seja um evento 2-bom para (θ, σ) é executado em tempo $O(1)$ por meio da escolha de elementos i e j , correspondentes a x e $\sigma \theta^{-1} x$, onde i é o primeiro elemento no primeiro par de Q . Pelo Lema 104, o evento de rearranjo $(x \sigma \theta^{-1} x)(\theta \Gamma \sigma \theta^{-1} x \theta \Gamma x)$ é um evento 2-bom para (θ, σ) onde θ é um genoma representado pelo vetor P na iteração corrente do laço. O processo de atualização da estrutura de dados pode ser feito em tempo constante pois as operações de troca nos vetores P e $invP$ levam tempo constante. Remover j e $P[-i]$ de Q leva tempo de execução constante pois é possível remover o nó de cada uma das listas circulares em tempo $O(1)$ quando há apontadores para aqueles nós em $T[j]$ e $T[P[-i]]$. Portanto o tempo de execução total é $O(n)$. \square

Teorema 106. *Dados os genomas π e σ em (E, Γ) temos*

$$W_{f_{sr}}(\pi, \sigma) = \frac{\|\sigma\pi^{-1}\|}{2}.$$

Prova: Dados os genomas π e σ em (E, Γ) , o Lema 105 garante a existência de uma sequência de eventos de rearranjo ρ_1, \dots, ρ_k tais que $\rho_k \dots \rho_1 \pi = \sigma$ e ρ_i é aplicável a $\rho_{i-1} \dots \rho_1 \pi$ e ρ_i é um evento 2-bom para $(\rho_{i-1} \dots \rho_1 \pi, \sigma)$ para $1 \leq i \leq k$. Além disso, pelo Lema 75, temos $W_{f_{sr}}(\pi, \sigma) \geq \|\sigma\pi^{-1}\|/2$. Portanto $W_{f_{sr}}(\pi, \sigma) = \|\sigma\pi^{-1}\|/2$. \square

5.3 Intercâmbio de Blocos e Reversão com Sinais

Apresentamos um algoritmo para encontrar a distância genômica ponderada entre dois genomas baseada em reversões com sinais e intercâmbio de blocos [44]. Como definido na Seção 3.7, um intercâmbio de blocos é um evento de rearranjo em genomas que troca dois segmentos não necessariamente consecutivos de uma fita (e também em sua fita complementar) em um genoma, mantendo as orientações originais dos genes. Reversões com sinais são eventos que invertem um segmento e trocam as orientações dos genes em duas fitas complementares em um genoma. Reversões com sinais [29] são importantes para a análise comparativa de genomas. Por outro lado, o estudo de intercâmbio de blocos é motivado mais por suas propriedades matemáticas [17] do que por alguma relevância biológica. O estudo da combinação desses dois eventos de rearranjo revela propriedades interessantes sobre pares de genomas. Por exemplo, o comportamento do produto de um genoma e intercâmbios de blocos e reversões com sinais é influenciado pela distribuição dos genes nas fitas dos cromossomos dos dois genomas. Considere o conjunto R contendo apenas os tipos de eventos de rearranjo intercâmbio de blocos e reversões com sinais. O algoritmo tira proveito do fato de haver para quaisquer pares de genomas equicromossomais um intercâmbio de blocos ou reversão com sinais que é um evento 2-bom para o par de genomas. Como visto na Seção 3.8, a existência de tais eventos bons para quaisquer pares de genoma em um sistema de genes permite que uma estratégia gulosa possa ser utilizada para resolver o problema de rearranjo em genomas.

Lema 107. *Se dois genomas π e σ de um sistema de genes (E, Γ) são compatíveis com relação a intercâmbio de blocos e reversão com sinais então π e σ possuem a mesma estrutura de ciclos e são equicromossomais.*

Prova: Se os genomas π e σ de um sistema de genes (E, Γ) são compatíveis com relação a intercâmbio de blocos e reversão com sinais então existe uma sequência de reversões com sinais e intercâmbios de blocos que transforma π em σ . Pelo Lema 70, um intercâmbio de blocos não altera as órbitas de um genoma. Logo um intercâmbio de blocos não altera a estrutura de ciclos de um genoma e a união das órbitas de um cromossomo em π será igual a união

das órbitas de um cromossomo em σ . Pela definição de reversões com sinais, os segmentos complementares reversos de ambas as fitas de um cromossomo no qual a reversão com sinais atua são invertidos e seus elementos têm suas orientações trocadas. Logo as órbitas são alteradas pela reversão com sinais, mas o tamanho das órbitas é mantido o mesmo. Além disso, como houve apenas uma troca de elementos das duas órbitas do mesmo cromossomo a união das órbitas mantém-se a mesma. Portanto, os genomas π e σ possuem a mesma estrutura de ciclos e são eqüicromossomais. \square

Dados os genomas eqüicromossomais π e σ , o *problema de rearranjo em genomas ponderado por intercâmbios de blocos e reversões com sinais* $P(\pi, \sigma, R, E, \Gamma, w)$ consiste em encontrar uma seqüência de intercâmbios de blocos e reversões com sinais com peso mínimo que transforme o genoma π em σ . Em outras palavras, queremos encontrar uma seqüência de eventos de rearranjo $\rho_1 \rho_2 \dots \rho_k$, tal que:

$$\sigma = \rho_k \rho_{k-1} \dots \rho_1 \pi$$

onde cada ρ_i é um intercâmbio de blocos ou uma reversão com sinais e esse evento de rearranjo ρ_i é aplicável a $\rho_{i-1} \dots \rho_1 \pi$ para $1 \leq i \leq k$ e $\sum_{i=1}^k w(\rho_i)$ é mínimo. Denotamos a esse valor mínimo por $W_{bisr}(\pi, \sigma)$.

Lema 108. Dados os genomas distintos e eqüiorbitais π e σ em (E, Γ) , há um intercâmbio de blocos ρ aplicável ao genoma π tal que ρ é um evento 2-bom para (π, σ) e $\rho\pi$ e σ são genomas eqüiorbitais.

Prova: Como os genomas π e σ são distintos, considere os elementos $x, y, \sigma\pi^{-1}x$ e $\sigma\pi^{-1}y$ onde $\sigma\pi^{-1}x \neq x, \sigma\pi^{-1}x \neq \sigma\pi^{-1}y, x \neq y$, e $x \neq \sigma\pi^{-1}y$, tais que:

1. $y \in orb(\pi, x)$;
2. $\pi\sigma\pi^{-1}x \neq x$;
3. $\sigma\pi^{-1}y \neq y$ e $\pi\sigma\pi^{-1}y \neq y$;
4. $(y \sigma\pi^{-1}y)(x \sigma\pi^{-1}x) \nmid \pi$.

Na Seção 3 de Lin *et al.* [37] existe uma demonstração de que os elementos x e y existem para quaisquer pares de genomas distintos e eqüiorbitais.

Como $\pi\sigma\pi^{-1}x \neq x, \pi\sigma\pi^{-1}y \neq y$, e $(x \sigma\pi^{-1}x)(y \sigma\pi^{-1}y) \nmid \pi$ então $x, \sigma\pi^{-1}y, \sigma\pi^{-1}x$ e y aparecem nessa ordem circular ou na ordem circular $\sigma\pi^{-1}x, \sigma\pi^{-1}y, x$ e y em uma fita do genoma π . A permutação $(x \sigma\pi^{-1}y)(y \sigma\pi^{-1}x)$ divide π devido a ordem circular dos quatro elementos e pelo Corolário 42 (ver propriedades de produtos envolvendo 2-ciclos na Seção 3.4). Logo a permutação

$$\rho = (x \sigma\pi^{-1}x)(\pi\Gamma\sigma\pi^{-1}x \pi\Gamma x)(y \sigma\pi^{-1}y)(\pi\Gamma\sigma\pi^{-1}y \pi\Gamma y)$$

é um intercâmbio de blocos aplicável a π pois $(x \sigma \pi^{-1} y)(y \sigma \pi^{-1} x)|\pi$. Além disso, o intercâmbio de blocos ρ é um evento 2-bom para (π, σ) porque $\rho|\sigma\pi^{-1}$ pela escolha de x e y . Como ρ é um intercâmbio de blocos então $\rho\pi$ e σ são genomas equi-orbitais segundo o Lema 70. Portanto ρ é um intercâmbio de blocos aplicável ao genoma π tal que ρ é um evento 2-bom para (π, σ) e $\rho\pi$ e σ são genomas equi-orbitais. \square

Lema 109. Dados os genomas equicromossomais π e σ , distintos e não equi-orbitais no sistema de genes (E, Γ) , há uma reversão com sinais ρ aplicável a π tal que ρ é um evento 2-bom para (π, σ) .

Prova: Como os genomas π e σ são equicromossomais, são distintos e não são equi-orbitais então $\sigma\pi^{-1} \neq \iota$ e existe um elemento $x \in E$ tal que $\sigma\pi^{-1}x \notin orb(\pi, x)$, caso contrário $\sigma\pi^{-1}x \in orb(\pi, x)$ para qualquer $x \in E$ implica $orb(\sigma, x) \subseteq orb(\pi, x)$ e como π e σ são equicromossomais, então π e σ teriam a mesma estrutura de ciclos segundo o Lema 107 e seriam equi-orbitais o que contradiz a hipótese inicial dos genomas não serem equi-orbitais. Mostraremos que a reversão com sinais $\rho = (x \sigma \pi^{-1} x)(\pi \Gamma \sigma \pi^{-1} x \pi \Gamma x)$ é aplicável a π e ela é um evento 2-bom para (π, σ) .

Os elementos σy e y para qualquer $y \in E$ pertencem a mesma órbita de σ e como π e σ são equicromossomais então σy e y pertencem a união das órbitas do mesmo cromossomo de π . Logo $\sigma\pi^{-1}x$ e $\pi^{-1}x$ pertencem a união das órbitas do mesmo cromossomo e como $\pi^{-1}x \in orb(\pi, x)$ então x e $\sigma\pi^{-1}x$ pertencem a fitas distintas do mesmo cromossomo de π . Além disso, temos $\Gamma\pi^{-1}\sigma\pi^{-1}x = \pi\Gamma\sigma\pi^{-1}x$ pela propriedade $\Gamma\pi\Gamma = \pi^{-1}$. Logo $\pi\Gamma\sigma\pi^{-1}x \in orb(\pi, x)$ e pelo Lema 43 temos $(x \pi\Gamma\sigma\pi^{-1}x)|\pi$. Além disso, temos $x \neq \pi\Gamma\sigma\pi^{-1}x$ porque, caso contrário, a igualdade $x = \pi\Gamma\sigma\pi^{-1}x$ implica em $\pi\Gamma x = \sigma\Gamma\pi\Gamma x$ e então há um elemento $z = \pi\Gamma x$ tal que $\sigma\Gamma z = z$, o que contradiz a definição de um genoma em um sistema de genes. Logo, a reversão com sinais $(x \sigma \pi^{-1} x)(\pi \Gamma \sigma \pi^{-1} x \pi \Gamma x)$ é aplicável a π . Além disso, como $\sigma\pi^{-1}$ é um produto de pares de ciclos da forma $\alpha\pi\Gamma\alpha^{-1}(\pi\Gamma)^{-1}$ então $\pi\Gamma x \notin orb(\sigma\pi^{-1}, x)$. Como $\sigma\pi^{-1}x \in orb(\sigma\pi^{-1}, x)$, $\pi\Gamma\sigma\pi^{-1}x \in orb(\sigma\pi^{-1}, \pi\Gamma x)$ e $\pi\Gamma x \notin orb(\sigma\pi^{-1}, x)$ então $\rho|\sigma\pi^{-1}$ segundo o Corolário 42. Portanto a reversão com sinais ρ é um evento 2-bom para (π, σ) . \square

O algoritmo *SRBISort* ilustrado na Figura 5.15 encontra uma sequência de eventos bons que transforme um genoma em outro e a distância genômica ponderada entre dois genomas em tempo de execução $O(n^2)$.

Lema 110. Dados os genomas π e σ em (E, Γ) , o algoritmo *SRBISort* apresenta uma sequência de eventos bons para (π, σ) com um peso mínimo $W_{bistr}(\pi, \sigma)$ que transforma o genoma π em σ em tempo de execução $O(n^2)$, onde $n = |E|$.

Prova: Mostramos que o algoritmo *SRBISort* é correto ao definir uma invariante de laço sobre o parâmetro r : temos $\theta = \rho_r \dots \rho_1 \pi$ e ρ_i é um evento 2-bom para $(\rho_{i-1} \dots \rho_1 \pi, \sigma)$ aplicável a $\rho_{i-1} \dots \rho_1 \pi$, para $1 \leq i \leq r$.

Algoritmo SRBISort(Genoma π , Genoma σ)

```

1:  $r \leftarrow 0$ 
2:  $\theta \leftarrow \pi$ 
3: initializeLabels( $\theta$ ,  $\pi$ ,  $E$ )
4:  $W \leftarrow 0$ 
5: while  $\theta \neq \sigma$  do
6:    $r \leftarrow r + 1$ 
7:   if isEqual(Orb( $\theta$ ,  $E$ ), Orb( $\sigma$ ,  $E$ )) then
8:     encontre um evento  $\rho_r$  tal que
9:      $\rho_r$  é um intercâmbio de blocos
10:     $\rho_r$  é aplicável a  $\theta$ 
11:     $\rho_r$  é um evento 2-bom para  $(\theta, \sigma)$ .
12:   else
13:     Encontre um evento  $\rho_r$  tal que
14:      $\rho_r$  é uma reversão com sinais
15:      $\rho_r$  é aplicável a  $\theta$ 
16:      $\rho_r$  é um evento 2-bom para  $(\theta, \sigma)$ .
17:   end if
18:   updateLabels( $\rho_r$ ,  $\theta$ ,  $E$ )
19:    $\theta \leftarrow \rho_r \theta$ 
20:    $W \leftarrow W + w(\rho_r)$ 
21: end while
22: return  $\rho_1, \dots, \rho_r$  e  $W$ 

```

Figura 5.15: Dados os genomas π e σ em (E, Γ) , o algoritmo *SRBISort* retorna uma sequência mínima de intercâmbios de blocos e reversões com sinais com um peso mínimo $W = W_{bisr}(\pi, \sigma)$ que transforma o genoma π no σ .

Para $r = 0$, antes do laço principal na linha 5, temos $\theta = \pi$ e a invariante é válida.

Suponha que a invariante seja válida para $r = k$, ou seja, temos $\theta = \rho_k \dots \rho_1 \pi$ e ρ_i é um evento 2-bom para $(\rho_{i-1} \dots \rho_1 \pi, \sigma)$ aplicável a $\rho_{i-1} \dots \rho_1 \pi$ para $1 \leq i \leq k$. Na próxima iteração do laço na linha 7 temos dois casos: θ e σ são genomas equiorbitais ou não. Em ambos os casos há um evento 2-bom para (θ, σ) pelo Lema 108 e Lema 109. Logo a invariante permanece válida antes da próxima iteração do laço na linha 5.

Se $\theta = \sigma$ então a condição na linha 5 é falsa e o algoritmo executa o código na linha 22. Nesse ponto da execução do algoritmo, temos $\theta = \rho_r \dots \rho_1 \pi$ tal que cada ρ_i é um evento 2-bom para $(\rho_{i-1} \dots \rho_1 \pi, \sigma)$ aplicável a $\rho_{i-1} \dots \rho_1 \pi$, para $1 \leq i \leq r$ uma vez que a invariante de laço é válida. Logo $\sigma = \rho_r \dots \rho_1 \pi$ e ρ_1, \dots, ρ_r é uma sequência de eventos bons, tal que ρ_i é um evento 2-bom para $(\rho_{i-1} \dots \rho_1 \pi, \sigma)$ e ρ_i é aplicável a $\rho_{i-1} \dots \rho_1 \pi$, para $1 \leq i \leq r$, transformando o genoma π no genoma σ .

Iremos mostrar agora qual é a complexidade assintótica de tempo de execução do algoritmo *SRBISort*. No pior caso, apenas um elemento em cada fita do genoma será colocado em sua posição correta por iteração do laço na linha 5, i.e. o bloco de instruções *while* será executado $O(n)$ vezes. Uma sequência de reversões com sinais é um exemplo de uma instância de pior caso na complexidade de tempo. Para cada passo no laço *while* verifica-se se os genomas θ e σ são equiorbitais na linha 7 por meio da função $isEqual(Orb(\theta, E), Orb(\sigma, E))$. Essa função pode ser implementada em tempo de execução $O(n)$ do seguinte modo. Antes do laço *while*, escolhe-se um elemento x_i para $1 \leq i \leq q$ no suporte de cada cromossomo de π , onde q é o número de cromossomos de π . Para cada elemento no suporte do cromossomo que contém x_i mantemos dois rótulos lógicos que indicam se o elemento pertence ou não a mesma órbita que contém o x_i escolhido em π e σ , respectivamente. A inicialização dos rótulos é feita pelo procedimento $initilizeLabels(\theta, \sigma, E)$. Os genomas π e σ são equiorbitais quando os dois rótulos de cada elemento de E têm o mesmo valor. Durante a execução do laço *while*, os rótulos precisam ser atualizados para o par de genomas $\rho_r \theta$ e σ quando ρ_r é uma reversão com sinais (intercâmbios de blocos não alteram as órbitas). O processo de atualização, implementado pelo procedimento $updateLabels(\rho_r, \theta, E)$, consiste em inverter o valor de rótulo correspondente ao genoma θ de cada elemento no segmento invertido pela reversão ρ_r . A atualização dos rótulos pode ser realizado em tempo de execução $O(n)$ pois uma reversão com sinais pode alterar a orientação, e consequentemente o valor do rótulo de até $(n - 1)/2$ pares de elementos de θ (um elemento em uma fita e o outro na fita complementar reversa). Encontrar um intercâmbio de blocos que seja um evento 2-bom para (θ, σ) pode ser executado em tempo $O(n)$ ao se representar os genomas por meio de um simples vetor e usando-se uma rotação de elementos como sugerido por Lin *et al.* [37]. Reversões com sinais que sejam eventos bons podem ser encontradas em tempo de execução $O(n)$ usando-se a mesma estrutura de dados empregada na verificação de que dois genomas são equiorbitais. Dado um elemento $x \in E$ é necessário identificar se x e $\sigma \theta^{-1} x$ pertencem a fitas distintas de θ ; e, nesse caso, a reversão com sinais

$\rho_r = (x \sigma \theta^{-1} x)(\theta \Gamma \sigma \theta^{-1} x \theta \Gamma x)$ é um evento 2-bom para (θ, σ) . Como há no máximo $n/2$ pares x e $\sigma \theta^{-1} x$, então tal verificação leva tempo de execução $O(n)$. Logo a complexidade de tempo total é quadrática em n . \square

O algoritmo *SRBISort* é apresentado em um pseudo-código sem detalhar como cada passo é implementado. Decidimos apresentar o algoritmo nesse formato pois, ao contrário dos eventos de rearranjo usados nos algoritmos da Seção 5.1 e da Seção 5.2, ainda não sabemos se é possível implementar a atualização da estrutura de dados provocada por um intercâmbio de blocos em tempo de execução constante. Logo qualquer estrutura de dados que possa implementar a aplicação das operações em tempo $O(n)$, como por exemplo vetores, pode ser utilizada (a estrutura de dados do Capítulo 4 permite implementar a atualização causada por reversões com sinais em tempo constante e por isso pode ser mais rápida que outras implementações para instâncias cuja solução possua um número de reversões com sinais muito superior ao número de intercâmbio de blocos).

Teorema 111. *Dados os genomas π e σ em (E, Γ) , temos*

$$W_{bistr}(\pi, \sigma) = \frac{\|\sigma \pi^{-1}\|}{2}.$$

Prova: Dados os genomas π e σ em (E, Γ) , o Lema 110 garante a existência de uma sequência de eventos de rearranjo ρ_1, \dots, ρ_k tal que $\rho_k \dots \rho_1 \pi = \sigma$ e ρ_i é aplicável a $\rho_{i-1} \dots \rho_1 \pi$ e ρ_i é um evento 2-bom para $(\rho_{i-1} \dots \rho_1 \pi, \sigma)$ para $1 \leq i \leq k$. Além disso, pelo Lema 75, temos $W_{bistr}(\pi, \sigma) \geq \|\sigma \pi^{-1}\|/2$ e $\sum_{i=1}^k w(\rho_i) = \|\sigma \pi^{-1}\|/2$. Portanto $W(\pi, \sigma) = \|\sigma \pi^{-1}\|/2$. \square

Dados os genomas π e σ em (E, Γ) , o Teorema 111 oferece uma fórmula simples para calcular $W_{bistr}(\pi, \sigma)$. O valor de $W_{bistr}(\pi, \sigma)$ pode ser obtido em $O(n)$, que é a complexidade de tempo de execução para computar $\sigma \pi^{-1}$ e encontrar a sua norma, embora encontrar uma sequência de eventos bons que transforme o genoma π em σ tenha complexidade de tempo quadrática.

5.4 Considerações Finais

Nesse capítulo apresentamos alguns resultados obtidos por meio da exploração do formalismo algébrico para problemas de rearranjo em genomas. Demonstramos que é possível deduzir por meio do formalismo algébrico um limitante inferior para a distância de transposição de mesmo valor numérico que a fórmula apresentada por Bafna e Pevzner [7] e com uma formulação equivalentemente simples.

Definimos o grafo de transposições boas e apresentamos algumas suas propriedades como, por exemplo, o fato de ser um grafo bipartido e sem ciclos orientados. Discutimos como o grafo pode ser utilizado para identificar genomas facilmente transformáveis e concluímos essa

discussão com a caracterização de genomas facilmente transformáveis e um resultado que nos informa que o conjunto de genomas atingíveis a partir de um dado genoma não pode ser obtido em tempo polinomial.

Projetamos um algoritmo de tempo de execução linear no número de genes do genoma na entrada que soluciona o problema de rearranjo em genomas ponderado por transposições generalizadas. Utilizando a mesma estrutura de dados do Capítulo 4, resolvemos o problema de rearranjo em genomas ponderado por fusões, fissões e reversões com sinais por meio de um algoritmo de tempo de execução linear. Finalmente, apresentamos um algoritmo de tempo de execução quadrática para o problema de rearranjo em genomas ponderado por reversões com sinais e intercâmbio de blocos.

Capítulo 6

Conclusões

Diversos problemas de rearranjo em genomas envolvendo, separadamente, reversões com sinais, intercâmbios de blocos, fusões e fissões podem ser resolvidos por meio de algoritmos polinomiais eficientes. Por outro lado, o problema de rearranjos em genomas envolvendo combinações de diferentes eventos de rearranjo foi especialmente investigado nesse trabalho devido ao fato de ter sido pouco explorado até o momento. Além disso, o problema de rearranjos em genomas envolvendo transposições demonstrou ter propriedades combinatórias mais complexas do que o problema envolvendo intercâmbio de blocos ou reversões com sinais. Logo, decidimos estudar uma versão do problema de rearranjo em genomas para transposições generalizadas, pois acreditamos que a investigação desse problema permitiria um fundamento para o uso do formalismo algébrico em futuras pesquisas do problema envolvendo transposições.

Apresentamos nesse trabalho novos resultados algébricos relacionados a 3-ciclos (3-norma, 3-divisibilidade, relação entre paridade de uma permutação e sua norma).

Na Seção 3.6, realizamos algumas modificações conceituais no formalismo algébrico apresentado por Meidanis e Dias [39] (introdução do conceito de sistema de genes, definição de genomas por meio da propriedade de divisibilidade, análise de compatibilidade entre genomas).

No Capítulo 4, projetamos uma nova estrutura de dados eficiente para problemas de rearranjos em genomas baseada em estruturas mais simples como vetores, listas e uma tabela de espalhamento.

Na Seção 5.1, apresentamos um limitante inferior para a distância de transposição baseado no formalismo algébrico e demonstramos que esse limitante inferior é equivalente ao limitante inferior da distância de transposição oferecido pelo formalismo clássico. A análise de parâmetros e fórmulas provenientes de dois formalismos distintos para rearranjo em genomas pode ajudar a compreender melhor a estrutura combinatória de problemas envolvendo transposições. Apresentamos também uma caracterização das permutações cuja distância de transposição satisfaz o limitante inferior baseado em pontos de quebra. Discutimos algumas propriedades do grafo de transposições boas e caracterizamos por meio do grafo os genomas que são facilmente

transformáveis em um certo genoma. Além disso, demonstramos que é possível encontrar uma sequência de distância ponderada mínima de transposições generalizadas que transforma um genoma origem em um outro genoma destino e oferecemos um algoritmo de tempo de execução linear para encontrar essa sequência de eventos. Como uma transposição generalizada é aplicável a um genoma origem apenas quando os elementos do suporte de um dos ciclos do evento pertencem à mesma fita no genoma destino, então demonstramos que elementos que pertencem a uma mesma órbita do quociente dos dois genomas na entrada do problema pertencem a uma mesma órbita no genoma destino. Utilizamos uma versão modificada da estrutura de dados apresentada no Capítulo 4 no projeto do algoritmo que determina a sequência mínima de transposições generalizadas. As modificações na estrutura de dados podem ser úteis no projeto de algoritmos mais eficientes para problemas relacionados a transposições, como o problema de rearranjo em genomas por intercâmbio de blocos.

Na Seção 5.2, projetamos um algoritmo que encontra em tempo de execução linear uma sequência de fissões, fusões e reversões com sinais que transforma um dado genoma em outro com a somatória dos pesos dos eventos mínima. Demonstramos que para quaisquer pares de genomas π e σ em um sistema de genes (E, Γ) é possível encontrar um evento 2-bom (fissão, fusão ou reversão com sinal) para (π, σ) que seja aplicável a π . Reversões com sinais, fusões e fissões possuem a mesma estrutura de ciclos e por isso apresentam um efeito relacionado quando aplicadas em genomas, decorrente das propriedades de produtos envolvendo 2-ciclos. Tiramos proveito dessas propriedades para projetar um algoritmo simples e eficiente para esse problema.

Na Seção 5.3, apresentamos um algoritmo de tempo de execução quadrático para o problema de rearranjos em genomas ponderado por intercâmbio de blocos e reversões com sinais. A estrutura de dados usada nos problemas anteriores não permite, até onde sabemos, aplicar um intercâmbio de bloco em tempo de execução constante. Por essa razão, representamos o algoritmo por um pseudo-código que não detalha a estrutura de dados utilizada. A estrutura de dados escolhida para implementar o algoritmo poderá ter vantagens ou desvantagens dependendo das instâncias de entrada do problema.

6.1 Publicações e Resumo de resultados

Os principais resultados presentes nessa tese foram apresentados nos seguintes trabalhos:

- Pôsteres:

1. *Analysis of Sorting by Transpositions based on Algebraic Formalism* [43]

Pôster apresentado no “The Eighth Annual International Conference on Research in Computational Molecular Biology (RECOMB2004)”, em Março de 2004, San Diego, EUA. Esse trabalho introduz o problema de rearranjo em genomas envolvendo

eventos de rearranjo da forma de 3-ciclos, como transposições e splits, para o caso especial de genomas unicromossomais com uma única fita. Mostramos uma solução para o problema proposto e fornecemos uma fórmula para o cálculo da distância envolvendo transposições e splits. Com base nesse trabalho, definimos o que é uma transposição generalizada para o caso de um problema envolvendo genomas em um sistema de genes.

2. *A new Data Structure for Genome Rearrangement Problems*

Pôster aceito na conferência “The 15th Annual International Conference on Intelligent Systems for Molecular Biology (ISMB) & 6th European Conference on Computational Biology (ECCB)”, em Julho de 2007, Viena, Áustria. Esse trabalho apresenta a estrutura de dados do Capítulo 4 baseada na representação de genomas do formalismo algébrico.

- Relatório Técnico:

1. IC-05-10 *Algebraic formalism for genome rearrangements (part 1)*.

Relatório técnico do Instituto de Computação (Unicamp), de Junho de 2005, em inglês, 23 páginas. Esse artigo sumariza alguns dos conceitos e abordagens do formalismo clássico, principalmente no que concerne o problema de rearranjo por reversões com sinais. Discutimos as vantagens e desvantagens do formalismo clássico e apresentamos algumas das vantagens que consideramos relevantes do formalismo algébrico. Não existiu a parte 2 desse relatório. Os resultados que resultariam na parte 2 do relatório técnico estão contidos nessa tese.

2. IC-06-01 *Sorting by block-interchanges and signed reversals*.

Relatório técnico do Instituto de Computação (Unicamp), de Janeiro de 2006, em inglês. Nesse relatório, apresentamos uma medida de rearranjo em genomas baseada nos pesos de reversões com sinais e intercâmbios de blocos em uma sequência mínima de tais rearranjos que transforme um genoma em outro. Mostramos um algoritmo de tempo de execução quadrático para encontrar uma solução para o problema de rearranjo em genomas envolvendo reversões com sinais e intercâmbios de blocos e uma fórmula para calcular o peso da solução.

- Em Anais de Congresso:

Sorting by block-interchanges and signed reversals. Artigo apresentado no congresso “Fourth International Conference on Information Technology: New Generations ITNG 2007”, em Las Vegas (EUA), Abril de 2006.

- Artigos submetidos a Periódicos:

Sorting by Fissions, Fusions, and Signed Reversal in $O(n)$. Artigo submetido a revista “Transactions on Computational Biology and Bioinformatics”. O conteúdo do artigo é abordado na Seção 5.2.

6.2 Considerações Finais e Trabalhos Futuros

Diversos problemas em aberto são derivados diretamente dos problemas abordados nesse trabalho.

Em relação a representação e implementação de estrutura de dados para genomas do Capítulo 4, uma direção de pesquisa relevante seria a de identificar quais seriam os eventos de rearranjo em genomas que podem ser implementados em tempo constante, além da fissão, fusão, transposições generalizadas e reversão com sinais. O ponto central dessa pesquisa deve ser a investigação sobre se é possível encontrar para qualquer par de genomas um evento de rearranjo aplicável a um dos genomas e que seja um evento 2-bom ou 3-bom (divida o quociente) em tempo constante. Problemas de rearranjo em genomas que envolvam apenas eventos de rearranjo para os quais sempre é possível encontrar um evento 2-bom (ou um evento 3-bom) e aplicável ao genoma de origem em tempo constante podem ser resolvidos em tempo de execução linear no tamanho da entrada. Além disso, é importante determinar se é possível resolver um problema de rearranjo em genomas envolvendo intercâmbio de blocos em tempo linear, pois implementações eficientes para esses eventos de rearranjo poderiam ser utilizadas em outros eventos (e.g. translocações).

Uma linha de pesquisa mais geral consiste em encontrar subestruturas no quociente de pares de genomas que permitam identificar eventos de rearranjo que sejam os mais “adequados” segundo um critério biológico, por exemplo. As decomposições em 2-ciclos e 3-ciclos do quociente de dois genomas podem revelar propriedades interessantes a respeito do relacionamento entre eventos de rearranjo em uma sequência de eventos que transforma um genoma em outro.

Sugerimos a utilização da norma de um evento de rearranjo para ponderar a sua frequência de ocorrência em uma solução de um problema de rearranjo em genomas. A norma é uma fórmula que parece ser adequada do ponto de vista biológico em problemas envolvendo fissões, fusões, intercâmbio de blocos, transposições e reversões com sinais, pois a fórmula do peso baseada na norma coincide com a proporção entre os pesos geralmente atribuídos a esses eventos, no entanto em certos problemas envolvendo transreversões e outros eventos de rearranjo não sabemos se a norma continua sendo um parâmetro adequado. Uma análise mais precisa da frequência de eventos de rearranjo provavelmente envolverá uma investigação mais pormenorizada de como ocorrem rearranjos em genomas (de maneira aleatória ou atuando em “hot spots” no genoma) e de qual o seu “poder” (número de segmentos no qual o genoma é “quebrado” durante o evento).

Há algumas questões interessantes que permanecem sem resposta a respeito do problema

de rearranjo em genomas por transposições abordado por meio da análise do grafo de transposições boas. Por exemplo, quantos componentes o grafo de transposições boas possui? Como encontrar um caminho mínimo entre genomas pertencentes a componentes distintas? É possível definir algo como uma distância das componentes em relação a σ -componente? Além disso, ainda não sabemos qual é o tamanho da σ -componente. Suspeitamos que o número de permutações facilmente transformáveis tende a zero a medida que a ordem do grafo de transposições boas aumenta, pois esse comportamento já foi verificado em outros problemas [57, 21]. Por outro lado, há experimentos [23] que sugerem que a distância de transposição não é muito maior que o limitante inferior baseado na decomposição em 3-ciclos. Tentaremos responder a algumas dessas questões em trabalhos futuros.

O problema de rearranjos em genomas por transposições generalizadas permite apenas a aplicação de eventos de rearranjo cujo suporte seja contido no suporte de um cromossomo do genoma destino. Se eliminarmos essa restrição e exigirmos, por outro lado, que no máximo dois elementos possam ser iguais em cada ciclo do evento de rearranjo, então definimos um novo evento de rearranjo que incluirá fissões, fusões, reversões com sinais, transposições e outros variantes desses eventos de rearranjo. Ao se permitir tal flexibilidade nos tipos de eventos de rearranjo, esse problema pode ser mais adequado para modelar a evolução de algumas espécies e portanto mais relevante do ponto de vista biológico.

Finalmente, o formalismo algébrico e a estrutura de dados proposta para representar genomas podem ser utilizados para obter melhorias na teoria que lida com problemas de rearranjo envolvendo múltiplos genomas [52, 46]. Esse problema consiste em encontrar uma árvore filogenética que seja a mais “similar” possível ao cenário evolutivo baseado em eventos de rearranjo envolvendo diversas espécies. Se considerarmos o problema de rearranjo em genomas envolvendo múltiplos genomas no qual são permitidos apenas reversões com sinais, então o problema é *NP*-difícil [14], mesmo na versão envolvendo apenas três genomas na entrada (Problema da Média). Bourque e Pevzner [13] projetaram um algoritmo para o problema de rearranjos em genomas envolvendo reversões com sinais, translocações, fusões e fissões cuja heurística depende de um procedimento demorado de busca por eventos bons (esses tipos de eventos são definidos de maneira diferente da utilizada no formalismo algébrico). Durante o período de estágio de pesquisa no exterior (programa PDEE – CAPES), propusemos [46] uma versão simplificada da heurística de Bourque e Pevzner que se baseia em evidências experimentais de que reversões com sinais próprias atuando em ciclos curtos são frequentemente eventos bons. Por meio da análise de experimentos envolvendo dados de entrada gerados aleatoriamente e de origem biológica, argumentamos que a nova heurística é mais rápida que a heurística anterior sem comprometer a qualidade das soluções para entradas de dados envolvendo poucos genomas. Uma linha de pesquisa interessante consiste em avaliar o impacto da representação do problema por meio do formalismo algébrico e da estrutura de dados sobre a eficiência da heurística. Acreditamos que o tempo de execução de heurísticas para problemas envolvendo outros

eventos eventos de rearranjo (fusões e fisões por exemplo) poderia ser melhorado por meio dos algoritmos lineares apresentados nesse trabalho.

Apêndice A

Revisão Bibliográfica

A.1 Bafna e Pevzner (1998)

Bafna e Pevzner [7] foram os primeiros autores a apresentar um algoritmo de aproximação de tempo de execução polinomial para o problema de ordenação de permutações por transposições. Para entender esse algoritmo, precisamos primeiramente apresentar os conceitos fundamentais envolvidos nesse problema de rearranjo em genomas e as fórmulas dos limitantes inferiores e superiores para a distância de transposição. Nessa apresentação resumida da teoria envolvendo transposições, utilizaremos a terminologia adotada por Bafna e Pevzner [7].

Um genoma é modelado como uma permutação (uma função bijetora) $\pi = [\pi_1, \dots, \pi_n]$ sobre o conjunto de inteiros $\{1, 2, \dots, n\}$. Quando nos referirmos a uma permutação sem especificar um conjunto, assumimos que ela é definida sobre o conjunto $\{1, 2, \dots, n\}$. A versão estendida de uma permutação π inclui os elementos $\pi_0 = 0$ e $\pi_{n+1} = n + 1$, embora estes elementos adicionais sejam geralmente omitidos na representação final da permutação (o conjunto sobre o qual a permutação é definida também é estendido com os inteiros 0 e $n + 1$). Dada uma permutação

$$\pi = [\pi_1, \dots, \pi_i, \dots, \pi_j, \dots, \pi_k, \dots, \pi_n],$$

uma *transposição* $\tau(i, j, k)$, onde $1 \leq i < j \leq n + 1$ e $1 \leq k \leq n + 1$ tal que $k \notin [i, j]$, é o evento de rearranjo tal que

$$\pi\tau(i, j, k) = [\pi_1 \dots \pi_{i-1} \pi_j \dots \pi_{k-1} \pi_i \dots \pi_{j-1} \pi_k \dots \pi_n].$$

Observe que o evento $\tau(i, j, k)$ corresponde à permutação

$$\tau(i, j, k) = [1, 2, \dots, j, \dots, k - 1, i, \dots, j - 1, k, \dots, n].$$

Caso $i < j < k$ então a transposição $\tau(i, j, k)$ realiza uma troca dos blocos consecutivos π_i, \dots, π_{j-1} e π_j, \dots, π_{k-1} e, além disso, temos $\tau(i, j, k) = \tau(j, k, i)$.

Descrevemos agora o problema de ordenação por transposições conforme apresentado no artigo de Bafna e Pevzner.

Problema 112 (Problema de Rearranjo em Genomas por Transposições). Dadas duas permutações π e σ , o *problema de rearranjo em genomas por transposições* consiste em encontrar uma seqüência mínima de transposições

$$\tau_1, \dots, \tau_t$$

tal que:

$$\pi \cdot \tau_1 \cdot \tau_2 \cdot \dots \cdot \tau_t = \sigma$$

onde t é chamada de *distância de transposição* de π a σ , denotada por $d_t(\pi, \sigma)$.

Observamos que a mesma seqüência de transposições que transforma π em σ também transforma $\sigma^{-1}\pi$ na permutação identidade ι , usando-se composições de permutações.

Problema 113 (Ordenação por Transposições). Dada a permutação π , o *problema de ordenação por transposições* consiste em encontrar uma seqüência mínima τ_1, \dots, τ_t de transposições tal que:

$$\pi \cdot \tau_1 \cdot \tau_2 \cdot \dots \cdot \tau_t = \iota$$

onde t é chamada de *distância de transposição* de π , denotada por $d_t(\pi)$.

O conjunto de transposições é um conjunto gerador para o grupo simétrico $Gen(E)$, ou seja, qualquer permutação $\pi \in Gen(E)$ pode ser representada como o produto (composição) de transposições $\tau_1 \cdot \dots \cdot \tau_t = \pi^{-1}$. Qualquer permutação pode ser representada, por exemplo, como um produto das transposições $\tau(i, i+1, i+2)$ que trocam elementos adjacentes da identidade e tais transposições podem ser obtidas por meio de um procedimento análogo ao *bubble sort*. Considere o problema de determinar o número mínimo de geradores cujo produto é igual a π . Bafna e Pevzner observaram que o problema de ordenação por transposições pode ser reduzido ao problema anterior uma vez que encontrar o produto mínimo de geradores que resulta em π^{-1} é equivalente a ordenar π por transposições. Infelizmente, o problema de determinar o produto mínimo de geradores é *NP*-difícil. Além disso, o conjunto de geradores do problema de ordenação por transposições é fixo, ao contrário do problema anterior e a complexidade de ordenação por transposições permanece desconhecida. Tais fatos têm incentivado os pesquisadores a realizar uma análise mais pormenorizada das estruturas combinatoriais “escondidas” nas permutações e como tais estruturas são afetadas pelas transposições.

Dada uma permutação π , o par (π_i, π_{i+1}) é um *ponto de quebra* quando $\pi_{i+1} \neq \pi_i + 1$, para $0 \leq i \leq n$. Denote por $b(\pi)$, o número de pontos de quebra da permutação π . A identidade é a única permutação que possui zero pontos de quebra e portanto, poderíamos estar tentados a acreditar que transposições que reduzam o número de pontos de quebra sejam transposições que pertençam a uma seqüência ordenante mínima. Na realidade, permutações com menos

pontos de quebra podem estar mais distantes da identidade do que permutações que possuam mais pontos de quebra.

Exemplo 114. Considere as permutações

$$\pi_1 = [3, 1, 2, 5, 4, 7, 6] \quad \text{e} \quad \pi_2 = [6, 7, 4, 5, 3, 2, 1].$$

Observamos que $b(\pi_1) = 7$ e $b(\pi_2) = 6$, no entanto $d_t(\pi_1) = 2$ e $d_t(\pi_2) = 3$.

Uma vez que uma transposição pode eliminar no máximo 3 pontos de quebra de uma permutação, Bafna e Pevzner [7] obtiveram o seguinte limitante inferior para a distância de transposição de π :

Proposição 115.

$$d_t(\pi) \geq \frac{b(\pi)}{3}$$

O *diagrama de ciclos* $B(\pi)$ de uma permutação π é o grafo cujo conjunto de vértices é formado pelos inteiros x tal que $0 \leq x \leq n+1$ e para $1 \leq i \leq n+1$, há uma aresta cinza orientada ligando $i-1$ a i e uma aresta preta orientada de π_i a π_{i-1} . A Figura A.1 apresenta o grafo de ciclos de uma permutação (as orientações das arestas foram omitidas). Um ciclo de $B(\pi)$ é chamado *alternante* se o mesmo é um ciclo no qual as arestas cinzas e pretas se alternam. Iremos chamar os ciclos alternantes simplesmente de ciclos. Uma vez que em cada vértice do grafo existe uma aresta cinza e uma aresta preta de cor contrária a uma aresta entrante, então existe uma única decomposição em ciclos alternantes. Seja $c(\pi)$ o número de ciclos de π , isto é o número de ciclos em $B(\pi)$. O comprimento de um ciclo é o número de arestas pretas no ciclo e dizemos que um ciclo é um *k-ciclo* quando seu comprimento é k . Um *k-ciclo* é *longo* caso $k > 2$, caso contrário é um *ciclo curto*. Dada uma permutação π e uma transposição τ , seja $\Delta c(\tau) = c(\pi\tau) - c(\pi)$.

Por meio de um argumento “gráfico”, Bafna e Pevzner determinaram a variação no número de ciclos de $B(\pi\tau)$ com relação a $B(\pi)$ como sendo $\Delta c(\tau) \in \{-2, 0, 2\}$ para qualquer transposição τ . Esse resultado permitiu a Bafna e Pevzner deduzir o seguinte limitante inferior para a distância de transposição baseado no número de ciclos de π :

Proposição 116.

$$d_t(\pi) \geq \frac{n+1-c(\pi)}{2}$$

É possível melhorar um pouco o limitante inferior para a distância de transposições dado na Proposição 116. Um ciclo no grafo de ciclos é *ímpar* se o número de arestas pretas no ciclo é ímpar, caso contrário o ciclo é *par*. Dada uma permutação π , considere os seguintes dois novos parâmetros: $c_{\text{odd}}(\pi)$ e $c_{\text{even}}(\pi)$ que respectivamente denotam os número de ciclos

ímpares e pares de $B(\pi)$. Define-se $\Delta_{c_{odd}}(\tau) = c_{odd}(\pi\tau) - c_{odd}(\pi)$ como a variação no número de ciclos ímpares de π devido a aplicação da transposição τ a π . De maneira semelhante ao caso envolvendo ciclos quaisquer, uma transposição pode aumentar de no máximo dois o número de ciclos ímpares em $\pi\tau$ em relação a π . De fato, o seguinte é válido: $\Delta_{c_{odd}} \in \{-2, 0, 2\}$. Uma vez que a identidade é a única permutação que possui $n + 1$ ciclos ímpares e cada transposição pode aumentar no máximo de dois o número de ciclos ímpares, então um novo limitante inferior mais justo decorre imediatamente:

Proposição 117.

$$d_t(\pi) \geq \frac{n + 1 - c_{odd}(\pi)}{2}.$$

Seja $d_t(n) = \max_{\pi \in Gen(E)} d_t(\pi)$ o diâmetro de transposição do grupo simétrico $Gen(E)$, i.e. a maior distância de transposição dentre quaisquer elementos em $Gen(E)$. Um limitante superior trivial para a distância de transposição é o número de elementos n da permutação. Por exemplo, poderíamos ordenar a permutação reversa $R(n) = [n, n - 1, \dots, 2, 1]$ por meio de uma sequência de transposições na qual cada evento elimina um ponto de quebra por vez.

Bafna e Pevzner [7] oferecem uma análise mais refinada do grafo de ciclos de uma permutação que permite identificar quais transposições (se existirem) incrementam de dois o número de ciclos. Dada uma permutação π , uma transposição τ é um x -movimento quando $\Delta c(\tau) = x$ onde $x \in \{-2, 0, 2\}$. Considere a rotulação das arestas pretas de $B(\pi)$ de 1 a $n + 1$ tal que a aresta (π_i, π_{i-1}) é rotulada como i . Diz-se que a transposição $\tau(i, j, k)$ atua sobre as arestas i, j e k . Uma transposição $\tau(i, j, k)$ atua sobre um ciclo C quando as arestas i, j e k pertencem ao ciclo C . Se uma transposição τ atuar em um ciclo de uma permutação π e aumentar o número de ciclos em $\pi\tau$ então a transposição será um 2-movimento. Por outro lado, se a transposição atua sobre arestas pertencentes a dois ciclos, então a transposição será um 0-movimento. Um ciclo pode ser descrito pelos rótulos de suas arestas pretas, porém como há diversas maneiras de escolher qual será o primeiro elemento no ciclo, decidiu-se por começar com o rótulo da aresta sainte incidente ao vértice mais a direita em $B(\pi)$, i.e. se $C = (i_1, \dots, i_k)$, onde i_t são rótulos das arestas pretas, então $i_1 = \max_{1 \leq t \leq k} i_t$. Para $k > 1$ um k -ciclo $C = (i_1, \dots, i_k)$ é *não orientado* quando i_1, \dots, i_k é uma sequência decrescente, caso contrário C é um ciclo *orientado*.

Exemplo 118. Dada a permutação π sobre o conjunto estendido $E = \{0, 1, 2, 3, 4, 5, 6, 7\}$:

$$\pi = [0, 2, 5, 4, 3, 6, 1, 7]$$

A permutação π possui 3 ciclos dos quais dois são não-orientados((5 3) e (4 2)) e um é orientado ((7 1 0)).

A partir da definição de ciclos orientados e não orientados, Bafna e Pevzner [7] demonstram o seguinte resultado.

Proposição 119. Dada uma permutação não ordenada π , existe um 2-movimento ou um 0-movimento seguido de um 2-movimento em π .

Como na pior das hipóteses, apenas um ciclo seria acrescentado por transposição em média, então um corolário imediato da Proposição 119 segue-se:

Proposição 120. Qualquer permutação π pode ser ordenada com $n + 1 - c(\pi)$ transposições.

A Proposição 120 apresenta um bom limitante superior para a distância de transposição encontrado pelos autores, além disso essa proposição fornece um algoritmo com um fator de aproximação 2 para o problema de ordenação por transposições. Além disso, a Proposição 120 mostra que é possível encontrar seqüências de transposições em que o número de 2-movimentos é maior que os demais. De fato, a Proposição 120 é coerente com a adoção explícita por parte dos autores de uma estratégia gulosa na qual tenta-se maximizar o número de 2-movimentos por meio de escolhas locais, embora seja importante destacar que esses 2-movimentos possam não participar de nenhuma seqüência mínima de transposições que ordene a permutação. Por outro lado, essa estratégia gulosa beneficia o algoritmo de aproximação pois caso seja encontrada uma maneira de garantir que em qualquer seqüência de transposições que ordena π o número de 2-movimentos é superior ao de 0-movimentos, então a qualidade do fator de aproximação será melhorada.

Dada uma tripla de arestas pretas x, y, z pertencentes ao mesmo ciclo C de $B(\pi)$, verificamos que o ciclo induz uma ordem circular sobre x, y, z . Considere como a representação canônica da tripla x, y, z como aquela na qual a primeira aresta é a que aparece mais à direita em $B(\pi)$. Uma tripla em sua representação canônica é chamada *não-orientada* se $x > y > z$ ou *orientada*, caso contrário. Observe que todas as triplas de um ciclo não orientado são não orientadas, enquanto em um ciclo orientado existe ao menos uma tripla orientada. Duas seqüências de inteiros $\{v_1 < \dots < v_k\}$ e $\{w_1 < \dots < w_k\}$ estão *entrelaçadas* se $v_1 < w_1 < v_2 < w_2 < \dots < v_k < w_k$ ou $w_1 < v_1 < w_2 < v_2 < \dots < w_k < v_k$. Duas triplas x, y, z e i, j, k se entrelaçam se $\{x, y, z\}$ e $\{i, j, k\}$ se entrelaçam. Dada uma transposição $\tau(i, j, k)$ e uma tripla x, y, z , dizemos que a transposição anterior é uma *shuffling transposition* (ou transposição de arrasto) com respeito a tripla x, y, z se $\{x, y, z\}$ e $\{i, j, k\}$ se entrelaçam.

A Proposição 121 identifica o comportamento de uma subestrutura do diagrama de ciclos de uma permutação π para a qual é possível determinar se existe um ciclo orientado ou não orientado em $\pi\tau$ para uma dada transposição τ .

Proposição 121. Dada a permutação π , seja x, y, z uma tripla do ciclo C e sejam $i, j, k \notin C$ arestas pretas de $B(\pi)$. Então $\tau(i, j, k)$ modifica a orientação da tripla x, y, z se e somente se τ é uma shuffling transposition para a tripla x, y, z .

Proposição 122. Dada a permutação π , se C é um ciclo não-orientado de $B(\pi)$, então para todas as triplas $x, y, z \in C$, a transposição $\tau(z, y, x) = \tau(y, x, z)$ transforma C em um ciclo não-orientado em $B(\pi\tau)$.

A Proposição 123 oferece uma maneira de determinar um 2-movimento a partir de uma tripla de um ciclo orientado:

Proposição 123. Se x, y, z é uma tripla orientada, então a transposição $\tau(y, z, x) = \tau(z, x, y)$ é um 2-movimento.

A Figura A.2 ilustra o efeito de transposições sobre uma tripla orientada (Bafna e Pevzner [7] utilizam essa figura para demonstrar a Proposição 123).

Dados dois ciclos C e C' de $B(\pi)$, dizemos que C e C' *se cruzam* caso existir uma tripla orientada em C e uma tripla não-orientada em C' que são entrelaçadas. Veja o item (a) da Figura A.3. Dois ciclos orientados C e C' são *não-interferentes* quando houver triplas orientadas em C e em C' que não se entrelaçam. Veja o item (b) da Figura A.3.

Dada a permutação π , pares de ciclos que se cruzam ou que não são interferentes no diagrama $B(\pi)$ são particularmente relevantes na teoria de Bafna e Pevzner, pois em ambos os casos é possível aplicar dois 2-movimentos em π . No caso de ciclos que se cruzam, a transposição τ que age sobre a tripla orientada é um 2-movimento e como essa tripla é entrelaçada a uma tripla não orientada, digamos (x, y, z) de outro ciclo, então (x, y, z) torna-se orientada em $\pi\tau$. Portanto, há um 2-movimento atuante sobre (x, y, z) . Por outro lado, se houver dois ciclos não-interferentes em $B(\pi)$, então a aplicação de um 2-movimento τ sobre uma das triplas não modificará a orientação da outra tripla que permanecerá orientada em $\pi\tau$. Logo, existe um 2-movimento em $\pi\tau$ atuante sobre a tripla orientada remanescente. Esse resultado é expresso na Proposição 124:

Proposição 124. Se para uma dada permutação π , o diagrama $B(\pi)$ possui ciclos que se cruzam ou ciclos não-interferentes, então existem dois 2-movimentos consecutivos para π .

Se houver um ciclo orientado C em $B(\pi)$ então C possui ao menos uma tripla orientada e pela Proposição 123 existe um 2-movimento aplicável em $B(\pi)$. Caso não haja ciclos orientados, mas se houver dois ciclos não orientados C e D com triplas entrelaçadas em $B(\pi)$, então existe uma transposição de arrasto (shuffling transposition) que atua sobre uma tripla em C (sem perda de generalidade) que é entrelaçada a outra tripla em D . Logo em $B(\pi\tau)$ o ciclo D se torna orientado. Esses dois ciclos C e D são cruzados em $B(\pi\tau)$ e portanto há dois 2-movimentos consecutivos. Os demais casos exigem uma análise detalhada de cada uma das configurações de entrelaçamento entre ciclos tais que não ocorram triplas entrelaçadas. A partir dessa análise e assumindo-se a estratégia gulosa de buscar seqüências de transposições nas quais é maximizado o número de 2-movimentos, Bafna e Pevzner obtiveram o seguinte resultado.

Proposição 125. Dada a permutação π , se existe um ciclo longo em $B(\pi)$, então é possível encontrar um 2-movimento ou um 0-movimento seguido de dois 2-movimentos consecutivos.

A Proposição 125 juntamente com a Proposição 120 permitem projetar um algoritmo para a ordenação de π (ver Figura A.4).

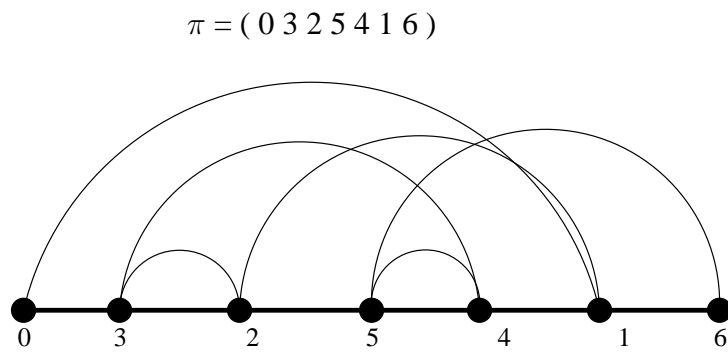


Figura A.1: Diagrama de Ciclos de um genoma. No topo encontra-se a permutação correspondente a um genoma em que são desconsideradas as orientações de seus blocos de genes. Em baixo é mostrado o Grafo de Ciclos da permutação π .

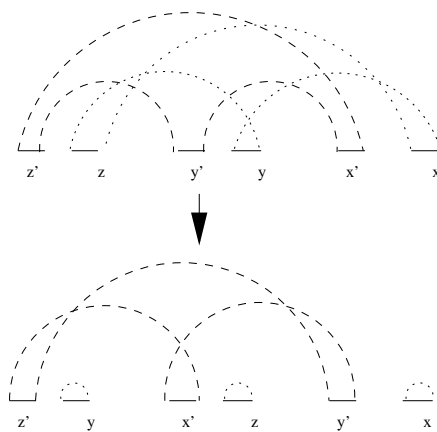


Figura A.2: Mudanças de orientação de arestas causadas por shuffling transposition $\tau(z, y, x)$.

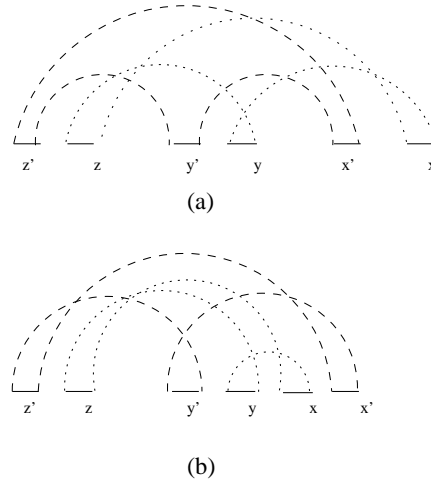


Figura A.3: Cruzamento e Interferência entre ciclos. (a) Dois ciclos que se cruzam. (b) Dois ciclos que não se interferem. Observe que as triplas $\{x, y, z\}$ e $\{x', y', z'\}$ se entrelaçam em ciclos que se cruzam, mas não em ciclos não-interferentes.

Algoritmo TSort(π)

```

1: while  $B(\pi)$  possuir um ciclo longo do
2:   if Houver um 2-movimento em  $\pi$  then
3:     Aplique um 2-movimento
4:   else
5:     Aplique um 0, 2, 2-movimento
6:   end if
7: end while
8: while  $B(\pi)$  possuir ciclos curtos do
9:   Aplique um 0-movimento seguido de um 2-movimento.
10: end while

```

Figura A.4: Algoritmo aproximado de ordenação por transposições com fator de aproximação 1,75.

A Proposição 125 permite criar 4 ciclos em 3 transposições na média ao invés da média de 1 ciclo por transposição se aplicarmos apenas operações baseadas na Proposição 120. Por outro lado, a Proposição 125 exige que os ciclos sejam longos e por isso é necessário combinar as operações de ambos os teoremas para obter o algoritmo. Discutiremos agora o fator de aproximação desse novo algoritmo. O algoritmo baseado na Proposição 120 garante que $\Delta c_{odd}(\tau) = 2$ para cada 2-movimento aplicado, entretanto a Proposição 125 não possui essa garantia para quaisquer 2-movimentos. Por isso, se analisarmos isoladamente cada um desses parâmetros que se deseja maximizar ($c(\pi)$ ou $c_{odd}(\pi)$) o algoritmo *TSort* não reduz o fator de aproximação de 2. Para resolver o aparente dilema de não haver uma redução no fator de aproximação do algoritmo, apesar do refinamento na identificação de 2-movimentos, Bafna e Pevzner consideraram uma função objetivo $f = xc_{odd}(\pi) + c_{even}(\pi)$, para $x \geq 1$, que combina os ciclos ímpares com os ciclos pares. O ganho máximo na variação da função objetivo para uma transposição τ é $\Delta f(\tau) = f(\tau\pi) - f(\pi) = 2x$. O algoritmo *TSort* provê um máximo ganho na variação de ciclos de $\min\{\frac{4}{3}, 2x - 1\}$. Logo, para um certo valor de ponderação dos ciclos ímpares ($x = 7/6$), é obtido um fator 1,75 de aproximação para a distância de transposição.

Proposição 126. O algoritmo *TSort* garante um fator de aproximação de 1,75 para o problema de ordenação por transposições.

Uma melhoria significativa no fator de aproximação de um algoritmo para o problema de ordenação por transposições está atrelada a métodos que escolham transposições que aumentem o número de ciclos ímpares. Essa questão incentivou os autores a buscarem uma maneira de garantir que a Proposição 125 aumente o número de ciclos **ímpares** ao invés de aumentar o número de ciclos de maneira indiscriminada.

Definição 127 (Transposição Válida). Dada uma permutação π , chamamos a transposição τ de uma *transposição válida* quando $\Delta c(\tau) = \Delta c_{odd}(\tau)$.

A partir de uma análise de padrões de entrelaçamento entre os ciclos de $B(\pi)$ para uma dada permutação π , Bafna e Pevzner determinaram em quais condições é possível maximizar o número de 2-movimentos válidos.

Proposição 128. Dada a permutação π , se houver um ciclo orientado em $B(\pi)$, então existe um 2-movimento válido ou um $0 - 2 - 2$ -movimento válido em π .

Em um ciclo com uma estrutura de auto-entrelaçamento semelhante ao da Figura A.5 não é possível aplicar um 2-movimento válido diretamente. Sempre é necessário aplicar um 0-movimento válido seguido de dois 2-movimentos válidos. Qual estrutura um ciclo orientado deve obedecer para permitir um 2-movimento válido? Para responder a essa pergunta os autores buscaram identificar quais seriam os ciclos com uma estrutura de auto-entrelaçamento o mais simples possível.

Dada a permutação π , seja C um ciclo (i_1, i_2, \dots, i_k) de $B(\pi)$ e $C^* = (i_1 = j_1 > \dots > j_k)$ uma sequência em ordem decrescente de elementos de C . Se C é um ciclo orientado tal que existe uma transposição que transforme C^* em C , então C é um *ciclo fortemente orientado*. Observe que C e C^* são iguais caso C é um ciclo não orientado.

Proposição 129. Um ciclo fortemente orientado possui um 2-movimento válido.

Como anteriormente, é possível realizar uma análise dos cruzamentos dos ciclos fortemente orientados com ciclos não-orientados e de dois ciclos fortemente orientados visando identificar qual deve ser a estrutura desses entrelaçamentos tal que dois 2-movimentos válidos consecutivos possam ser encontrados em π .

Proposição 130. Dada a permutação π , se $B(\pi)$ possui dois ciclos fortemente cruzados, então existem dois 2-movimentos válidos consecutivos em π .

Proposição 131. Dada a permutação π , se $B(\pi)$ possui dois ciclos fortemente não-interferentes, então existem dois 2-movimentos válidos consecutivos em π .

Finalmente, Bafna e Pevzner encontraram uma versão da Proposição 125 para o caso de movimentos válidos:

Proposição 132. Dada a permutação, se existe um ciclo longo em $B(\pi)$, então existe um 2-movimento válido em π ou um 0-movimento válido seguido de dois 2-movimentos válidos consecutivos em π .

Observe entretanto que a Proposição 132 se aplica apenas a permutações cujo diagrama de ciclos é composto por algum ciclo longo. Uma questão imediata que se coloca é sobre como lidar com diagramas $B(\pi)$ que possuam apenas ciclos curtos. Existem transposições atuantes sobre arestas de dois ciclos curtos que aumentam o número de ciclos ímpares, embora não aumentem o número de ciclos com relação a permutação original [7]. Chame um 0-movimento τ de *bom* quando ele cria 2 ciclos ímpares em $B(\pi\tau)$. Logo:

Proposição 133. Dada a permutação π , se $B(\pi)$ possui apenas ciclos curtos, então existe um bom 0-movimento seguido de um 2-movimento válido π .

A combinação da Proposição 132 e da Proposição 133 permite projetar o seguinte algoritmo aproximado *TransSort* mostrado na Figura A.6:

Proposição 134. O algoritmo *TransSort* ordena π em $O(n^2)$ utilizando não mais do que $\frac{3}{4}(n + 1 - c_{\text{odd}}(\pi))$ transposições, garantindo um fator de aproximação 1,5.

A partir do algoritmo *TransSort* é possível ordenar qualquer permutação com no máximo $\frac{3}{4}(n + 1 - c_{\text{odd}}(\pi))$ transposições. Um limitante superior mais justo para o diâmetro de transposição para $\text{Gen}(E)$ decorre naturalmente desse resultado.

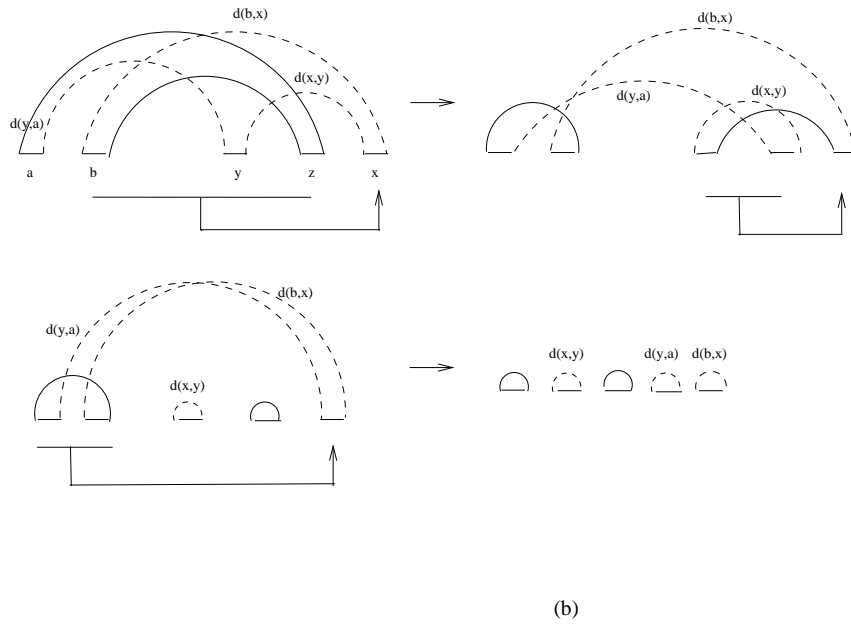


Figura A.5: Um 0 – 2 – 2-movimento válido. Os parâmetros $d(v, w)$ indicam a distância em número de arestas pretas entre os vértices π_v e π_w .

Algoritmo TransSort(π)

- 1: **while** $B(\pi)$ possuir um ciclo longo **do**
- 2: **if** Houver um 2-movimento válido em π **then**
- 3: Aplique um 2-movimento válido em π
- 4: **else**
- 5: Aplique um 0 – 2 – 2-movimento válido em π
- 6: **end if**
- 7: **end while**
- 8: **while** $B(\pi)$ possuir apenas ciclos curtos **do**
- 9: Aplique um 0-movimento bom seguido de um 2-movimento válido em π
- 10: **end while**

Figura A.6: Algoritmo aproximado para o problema de ordenação por transposições. O fator de aproximação do algoritmo é 1,5.

Corolário 135. O diâmetro de transposição do grupo simétrico $Gen(E)$ é de no máximo $\frac{3n}{4}$.

O algoritmo aproximado de Bafna e Pevzner de fator de aproximação 1,5 para o problema de ordenação por transposições foi por muito tempo o algoritmo mais eficiente projetado para este problema. Dentre os aspectos mais relevantes da teoria oferecida pelos autores podemos destacar a apresentação dos parâmetros e conceitos fundamentais que norteiam a pesquisa do problema mesmo nos artigos mais recentes da área. Além disso, é necessário destacar que as fórmulas para os limitantes inferiores e superiores para a distância de transposição e o diâmetro de transposição foram as primeiras fórmulas não triviais oferecidas para o problema. A implementação do algoritmo utiliza estruturas de dados simples que permitem realizar transposições em tempo linear (no tamanho da permutação). Alguns resultados novos sobre o problema foram conseguidos nos últimos anos; principalmente melhorias no tempo de execução do algoritmo de Bafna e Pevzner e simplificações na teoria. Christie [17] encontrou um algoritmo mais simplificado com o mesmo fator de aproximação, porém com complexidade $O(n^4)$. Hartman e Shamir [30] realizaram uma simplificação significativa da teoria de Bafna e Pevzner e propuseram um algoritmo aproximado de mesmo fator de aproximação e tempo de execução subquadrático. Recentemente, Elias e Hartman [22] propuseram um novo algoritmo aproximado de fator 1,375. A principal desvantagem encontrada no formalismo utilizado por esses autores é a necessidade de uma análise que envolve padrões diversos e complexos do entrelaçamento de ciclos e outras subestruturas de $B(\pi)$ na demonstração dos principais resultados do artigo. O tempo de execução e o fator de aproximação do valor de distância de transposição do algoritmo de Bafna e Pevzner é fortemente influenciado pela estrutura da permutação. Vale destacar também a ausência de uma análise das modificações da estrutura do diagrama de ciclos causada por -2 -movimentos e -2 -movimentos válidos.

A.2 Kaplan, Shamir e Tarjan (2000)

O artigo de Kaplan, Shamir e Tarjan [32] apresenta um algoritmo para o problema de rearranjo em genomas por reversões com sinais mais eficiente que o algoritmo proposto por Hannenhalli e Pevzner [29]. Nesse trabalho os autores definem uma nova estrutura combinatorial derivada do grafo de pontos de quebra, o “grafo de sobreposição”, que é mais eficiente e fácil de ser analisada do que o grafo de pontos de quebra. Por exemplo, na teoria de Hannenhalli e Pevzner eram necessárias algumas transformações equivalentes sobre a permutação original para que se obtivesse estruturas mais simples (ciclos curtos). Tais transformações eram utilizadas para obter estruturas para as quais a detecção das reversões mais adequadas para ordenar uma permutação fosse uma tarefa fácil. A teoria desenvolvida por Kaplan, Shamir e Tarjan dispensa o processo de transformação (*padding*) de permutações para a obtenção de estruturas adequadas à detecção de reversões que ordenam a permutação. A eliminação das transformações equivalentes

simplificou significativamente a compreensão dessa teoria. Além disso, esses autores conseguiram simplificar as estruturas de dados utilizadas no algoritmo, o que permitiu uma considerável redução em sua complexidade.

A teoria de Kaplan, Shamir e Tarjan [32] especifica uma representação de permutações semelhante a utilizada por Hannenhalli e Pevzner [29]. Dada uma permutação com sinais π , a função $u(\pi)$ que associa o conjunto de permutações com sinais a permutações sem sinais da seguinte maneira: a cada elemento x da permutação π , substitua x por $2x - 1$ e $2x$ se x é positivo, ou por $2|x|$ e $2|x| - 1$, caso contrário. Além disso, para $\pi' = u(\pi)$ acrescente os elementos $\pi'_0 = 0$ e $\pi'_{2n+1} = 2n + 1$. Por exemplo:

$$[3, -2, 5, -4, -1] \rightarrow [0, 5, 6, 4, 3, 9, 10, 8, 7, 2, 1, 11].$$

Uma reversão ρ realizada em uma permutação com sinais π possui uma correspondente reversão ρ' sobre uma permutação sem sinais π' .

Definição 136. Uma reversão $\rho(i, j)$ tal que i é ímpar e j é par é chamada de *reversão par*. Uma reversão par $\rho' = \rho(2i + 1, 2j)$ sobre $u(\pi)$ é equivalente a uma reversão $\rho(i + 1, j)$ em π .

Logo encontrar uma seqüência mínima de reversões com sinais que ordene a permutação π é equivalente a encontrar uma seqüência mínima de reversões pares que ordenem $u(\pi)$. A partir desse ponto, representaremos as permutações sem sinais obtidas pela transformação u apenas por π . Além disso, quando houver referência a reversão de uma permutação, de fato, estaremos nos referindo a reversão par sobre a permutação sem sinais π . Dada a permutação sem sinal $\pi = [0, \pi_1, \pi_2, \dots, \pi_n, n + 1]$, o par (π_i, π_{i+1}) , onde $0 \leq i \leq n$, é chamado de *gap*. Um gap é chamado de ponto de quebra se $|\pi_{i+1} - \pi_i| > 1$, caso contrário o gap é chamado de adjacência. O número de pontos de quebra da permutação π é denotado por $b(\pi)$. Dada uma reversão com sinais $\rho(i, j)$ dizemos que a reversão *age sobre* os gaps (π_{i-1}, π_i) e (π_j, π_{j+1}) .

De maneira semelhante a usada por Hannenhalli e Pevzner [29], Kaplan, Shami e Tarjan [32] utilizam uma definição de grafo de pontos de quebra, $B(\pi)$, tal que cada elemento de π torna-se um vértice do grafo e dois vértices são ligados por uma aresta preta se o par de vértices é um ponto de quebra em π , ou ligados por uma aresta cinza se o par de vértices é um ponto de quebra em π^{-1} . Observe que o grafo de pontos de quebra possui uma única decomposição em ciclos disjuntos, uma vez que cada vértice do grafo possui grau dois. Logo, o número de ciclos de $B(\pi)$ é bem definido e iremos denotá-lo por $c(\pi)$. A partir da análise do efeito de reversões com sinais sobre o número de ciclos e de pontos de quebra do grafo $B(\pi)$ foram definidos os seguintes conceitos que indicam quais seriam as reversões com sinais desejadas para ordenar a permutação.

Definição 137. Dada uma permutação π , uma reversão ρ é chamada de *reversão própria* quando $\Delta b - \Delta c = -1$, onde Δb é a variação no número de pontos de quebra resultante da aplicação

de ρ a π , ou seja, temos $\Delta b = b(\pi\rho) - b(\pi)$, enquanto Δc é a variação no número de ciclos no grafo de pontos de quebra devido a aplicação de ρ a π , isto é, temos $\Delta c = c(\pi\rho) - c(\pi)$.

Uma reversão ρ atua sobre uma aresta cinza e se a aresta e é incidente aos vértices pertencentes aos pontos de quebra nos quais a reversão ρ age. Uma aresta cinza é *orientada* se a reversão que atua sobre a aresta é uma reversão própria, caso contrário a aresta é *não orientada*.

Proposição 138. Dado $B(\pi)$, uma aresta cinza $(\pi_k \pi_l)$ é orientada se e somente se $k + l$ é par.

Kaplan, Shamir e Tarjan descrevem uma relação entre as arestas cinzas segundo o seu cruzamento no grafo de pontos de quebra. Dois intervalos de números reais *se sobrepõem* quando a interseção entre os intervalos não é vazia e um intervalo não estiver contido no outro. Considere a associação do intervalo $[i, j]$ a aresta (π_i, π_j) , duas arestas *se sobrepõem* quando os seus respectivos intervalos associados se sobrepõem. A partir da relação de sobreposição, definimos o *grafo de sobreposição* $OV(\pi)$ tal que seu conjunto de vértices é o conjunto de arestas cinzas do grafo de pontos de quebra e dois vértices em $OV(\pi)$ são adjacentes se suas arestas correspondentes no grafo de pontos de quebra se sobrepõem. Para evitar confusões entre os conceitos dos dois grafos, iremos nos referir a vértices e arestas com relação ao grafo de pontos de quebra e respectivamente a nós e arcos do grafo de sobreposição, desse modo denotaremos por N o conjunto de nós e por A o conjunto de arcos do grafo de sobreposição. Um componente conexo de $OV(\pi)$ é chamado de *componente orientado* se houver algum nó no componente que represente uma aresta cinza orientada, caso contrário será um *componente não orientado*.

A demonstração de Kaplan, Shamir e Tarjan de que é possível ordenar π em tempo de execução $O(n^2)$ baseia-se no algoritmo de eliminação de obstáculos de Hannenhalli e Pevzner, que exige um tempo de execução linear no tamanho da permutação, e na demonstração de que dada uma permutação π sem componentes não orientados sempre é possível encontrar uma reversão própria ρ tal que $\pi\rho$ não possua componentes não orientados. Em outros termos, os autores demonstram que:

Proposição 139. Dada uma permutação π tal que $OV(\pi)$ não possua componentes não orientados, então $d(\pi) = b(\pi) - c(\pi)$.

Para demonstrar a proposição anterior os autores avaliam o efeito de reversões orientadas sobre o grafo de sobreposição e desenvolvem o conceito de “clique feliz” para determinar dentre os nós orientados do grafo de sobreposição qual é o nó que possui uma *reversão segura* atuando sobre ele, isto é, uma reversão que não crie nenhum componente não orientado quando aplicada a permutação inicial. Denote por $N(e)$ os nós vizinhos a e no grafo de sobreposição incluindo e . Chamaremos de $ON(e)$ o subconjunto de nós orientados de $N(e)$ e de $UN(e)$ ao subconjunto de nós não orientados de $N(e)$. A Proposição 140 mostra o efeito de uma reversão atuando em um nó do grafo de sobreposição.

Proposição 140. Seja e um nó do grafo de sobreposição $OV(\pi)$ e seja $\pi' = \pi\rho_e$ onde ρ_e é a reversão que atua sobre e . O grafo $OV(\pi')$ pode ser obtido por meio das seguintes operações:

- Complemente o subgrafo de $OV(\pi)$ induzido pelos nós $N(e) - e$ e troque a orientação de cada vértice no conjunto $N(e) - e$.
- Se e é um nó orientado em $OV(\pi)$ então remova-o de $OV(\pi)$.
- Se houver algum nó orientado e' em $OV(\pi)$ tal que $\rho_{e'} = \rho_e$ então remova o nó e' de $OV(\pi)$.

Seja $OV(\pi)$ um grafo de sobreposição da permutação π , uma *clique feliz* $C \subseteq N$ é uma clique de nós orientados de $OV(\pi)$ tal que para todo nó orientado $y \notin C$, se $(x, y) \in A$ e $x \in C$, então existe um nó orientado $z \notin C$ onde $(z, y) \in A$ e $(z, x) \notin A$.

Proposição 141. Dado o grafo de sobreposição $OV(\pi)$, seja C uma clique feliz e e um nó em C tal que $|UN(e')| \leq |UN(e)|$ para cada $e' \in C$. Então a reversão que atua sobre o nó e , ρ_e é segura.

Segundo a Proposição 141, basta realizar uma busca pelo nó com número máximo de vizinhos não orientados em uma clique feliz para encontrarmos uma reversão segura. É necessário demonstrar, entretanto, que em um grafo de sobreposição contendo apenas bons componentes sempre existe uma clique feliz.

Proposição 142. Dado o nó orientado e de $OV(\pi)$, existe um nó orientado $f \in ON(e)$ tal que para $\pi' = \pi\rho_f$, onde ρ_f é a reversão atuando em f , todos os componentes em $OV(\pi')$ são orientados.

A Proposição 142 afirma, em outros termos, que na vizinhança orientada de qualquer nó orientado de $OV(\pi)$ existe uma clique feliz. Essa proposição sugere um meio de encontrar uma clique feliz em um grafo de sobreposição sem componentes ruins. Primeiramente, os autores utilizaram uma representação implícita do grafo de sobreposições $OV(\pi)$ baseada em dois vetores que representam respectivamente π e π^{-1} , ao invés da representação usual de grafos que utiliza um espaço de $O(n^2)$. Dada essa representação de π e π^{-1} é possível realizar as seguintes operações:

- Identificação dos extremos de uma aresta cinza. Dado um elemento qualquer y da permutação π , determinar o outro extremo y' da aresta cinza a qual y pertence é obtido da seguinte maneira: se y é par, então o outro extremo da aresta cinza é $y' = y + 1$, caso contrário, $y' = y - 1$.

- Determinar se um elemento é o extremo esquerdo ou direito de uma aresta cinza. Dado o elemento y , encontre o outro extremo y' e compare os valores de $\pi^{-1}[y]$ e $\pi^{-1}[y']$. O elemento cujo valor em π^{-1} é menor será o elemento esquerdo enquanto o outro o direito na aresta cinza.
- Determinar se uma aresta cinza é orientada ou não orientada. Dada a aresta cinza $e = (y \ y')$, basta verificar a paridade do valor $\text{parity}(e) = \pi^{-1}[y] + \pi^{-1}[y']$. Se $\text{parity}(e)$ é par então e é orientada, caso contrário a aresta cinza é não orientada.
- Determinar se duas arestas cinzas se sobrepõe. Dadas duas arestas cinzas $(y \ y')$ e $(z \ z')$, as mesmas irão se sobrepor caso $\pi^{-1}[y] < \pi^{-1}[z] < \pi^{-1}[y'] < \pi^{-1}[z']$ ou $\pi^{-1}[z] < \pi^{-1}[y] < \pi^{-1}[z'] < \pi^{-1}[y']$, caso contrário as arestas cinzas não se sobrepõe.
- Efetuar uma reversão $\rho(i, j)$. Basta inverter o intervalo $\pi[i..j]$ e efetuar as modificações necessárias no vetor π^{-1} .

Claramente essas operações sobre os vetores podem ser realizadas em tempo de execução constante, exceto pela aplicação da reversão sobre a permutação que exige um tempo de execução proporcional ao tamanho do intervalo que sofre a reversão, ou seja, $O(n)$. Essas operações são fundamentais para encontrar uma clique feliz em $OV(\pi)$.

Proposição 143. Dado o grafo $OV(\pi)$ em sua representação implícita, é possível encontrar uma clique feliz em $OV(\pi)$ em tempo de execução $O(n)$.

Resumidamente, o algoritmo para encontrar uma clique feliz em um grafo de sobreposição consiste em percorrer uma lista de nós orientados em ordem crescente segundo o extremo esquerdo das arestas cinzas correspondentes. Os nós na seqüência tais que seu extremo esquerdo pertença a todos os intervalos anteriores são adicionados a uma lista que indica a clique corrente. Caso o próximo nó a ser adicionado obedeça a uma invariante da clique durante a execução do algoritmo, ele não é adicionado a clique. O algoritmo devolve a clique corrente caso não haja mais nenhum vértice a ser adicionado. O tempo de execução desse algoritmo é proporcional ao número de nós do grafo de sobreposição, ou seja, $O(n)$.

Após a localização de uma clique feliz no grafo de sobreposição é necessário realizar uma busca que determine o nó orientado na clique feliz que possua o número máximo de vizinhos não orientados. Uma implementação ingênua dessa busca pelo nó de máximo grau de vizinhos não orientados verificaria o grau de todos os nós da clique e armazenaria o nó de maior grau encontrado durante o cálculo dos graus. Esse algoritmo exigiria um tempo de execução da ordem de $O(n^2)$. Dada a clique feliz $C = \{e_1, e_2, \dots, e_j\}$, Kaplan, Shamir e Tarjan conseguem obter o nó orientado com o tempo de $O(n)$ por meio da utilização de um vetor $o[i]$ como estrutura de dados, de tal forma que a soma $\sum_{i=1}^l o[i]$, onde $1 \leq l \leq j$, é o número de vizinhos não orientados do vértice $e_l \in C$. Para cada nó não orientado no grafo de sobreposição, no

Algoritmo KSTSort(π)

- 1: Ache os componentes do grafo de sobreposição $OV(\pi)$
 {procedimento de Hannenhalli e Pevzner}
- 2: Elimine os Obstáculos
- 3: **while** π não estiver ordenada **do**
- 4: Encontre uma clique feliz C no grafo $OV(\pi)$
- 5: Ache um vértice $e_f \in C$ com o máximo número de vizinhos não orientados
- 6: Aplique uma reversão segura sobre o vértice e_f
- 7: Inclua a reversão na seqüência de reversões ordenantes
- 8: Atualize a permutação π e o grafo $OV(\pi)$
- 9: **end while**
- 10: **return** A seqüência de reversões ordenantes

Figura A.7: Ordenação por Reversões de Permutações com sinais.

máximo quatro das posições dos vetores são alteradas segundo uma análise de casos que avalia a sobreposição do nó não orientado aos nós da clique feliz. Ao fim da inicialização do vetor, o nó com o número máximo de vizinhos não orientados será e_f , onde:

$$f = \arg \max_l \left\{ \sum_{i=1}^l o[i] \mid 1 \leq l \leq j \right\}.$$

O algoritmo de Kaplan, Shamir e Tarjan é mostrado na Figura A.7

A relevância do algoritmo de Kaplan, Shamir e Tarjan se deve ao fato de ser um algoritmo eficiente para o problema de rearranjo por reversões com sinais. Um dos principais benefícios oferecido pelos autores foi a simplificação de diversas estruturas e processos de transformações que eram utilizadas em algoritmos anteriores, principalmente pela teoria de Hannenhalli e Pevzner. Esse algoritmo apresenta também uma organização engenhosa de suas estruturas de dados o que explica sua grande eficiência. Além disso, é o primeiro algoritmo que busca por reversões seguras apenas entre as reversões próprias. Por outro lado; Kaplan, Shamir e Tarjan recorrem a argumentos visuais como na descrição da invariante 4.1 de seu artigo [32].

A.3 Bergeron (2005)

Uma nova simplificação considerável da teoria de ordenação de genomas por reversões com sinais foi realizada por Anne Bergeron [8]. Em sua teoria, ela dispensou a utilização de estruturas combinatórias utilizadas nas teorias anteriores como o grafo de pontos de quebra e o grafo de sobreposição. Ao invés de realizar a análise dessas estruturas, ela identificou estruturas

combinatórias equivalentes na própria permutação com sinais e sua correspondente transformação em permutação sem sinais e conseguiu formular conceitos como o de reversão orientada e obstáculos de maneira extremamente intuitiva.

Seja $\pi = [\pi_1, \pi_2, \dots, \pi_n]$ a permutação com sinais que representa um genoma e acrescente a essa permutação os elementos $\pi_0 = 0$ e $\pi_{n+1} = n + 1$. Dada essa permutação aumentada π , define-se o *par orientado* (π_i, π_j) como sendo um par de inteiros consecutivos de sinais opostos tais que $||\pi_i| - |\pi_j|| = 1$. Por exemplo na permutação

$$[0; \quad -2; \quad 3; \quad -5; \quad 4; \quad 1; \quad 6],$$

os pares orientados são $(-2, 1)$, $(-2, 3)$, $(-5, 4)$ e $(-5, 6)$. Um par orientado na permutação π corresponde a uma aresta cinza orientada no grafo de sobreposição de π . Logo um par orientado induz uma reversão própria sobre π , ou *reversão orientada* na terminologia de Bergeron, que é a reversão que atua sobre a aresta cinza correspondente. A reversão induzida pelo par orientado (π_i, π_j) é:

$$\rho(i, j - 1), \text{ se } \pi_i + \pi_j = +1$$

ou

$$\rho(i + 1, j), \text{ se } \pi_i + \pi_j = -1.$$

Se observarmos apenas o efeito da reversão induzida ρ sobre a permutação π , verificamos que um par de elementos consecutivos e adjacentes é criado em $\pi\rho$, em outros termos, o número de pontos de quebra da permutação $\rho\pi$ foi reduzido de pelo menos 1. Esse comportamento é observado no exemplo abaixo:

$$\pi = [0; \quad -2; \quad \underline{3, \quad -5, \quad 4, \quad 1}, \quad 6].$$

Os elementos sublinhados indicam a região afetada pela reversão $\rho(2, 5)$ induzida pelo par orientado $(-2, 1)$. O resultado da aplicação da reversão é:

$$\pi\rho(2, 5) = [0, \quad -2, \quad -1, \quad -4, \quad 5, \quad -3, \quad 6].$$

A permutação π possui 6 pontos de quebra e 4 pares orientados, enquanto a permutação $\pi\rho(2, 5)$ possui 5 pontos de quebra e 2 pares orientados. Como durante uma ordenação de π desejamos encontrar a permutação que possua o número mínimo de pontos de quebra, então de maneira intuitiva seria desejável encontrar a reversão ρ induzida por algum par orientado de π tal que em $\pi\rho$ apareça o número máximo possível de novos pares orientados, pois desse modo sempre é possível eliminar ao menos um ponto de quebra da permutação original. Essa discussão nos remete ao conceito de “score” de uma reversão com sinais ρ . O *score* de uma reversão com sinais ρ é o número de pares orientados em $\pi\rho$.

Bergeron afirma que uma estratégia baseada na escolha em cada passo da ordenação das reversões de *score* máximo é uma estratégia ótima tal que a seguinte proposição vale:

Proposição 144. Se a aplicação de uma seqüência de k reversões orientadas em uma permutação π resultar em uma permutação em que todos os elementos são positivos π' , então $d(\pi) = d(\pi') + k$.

O algoritmo de ordenação de Bergeron possui dois conjuntos de operações ordenantes: reversões orientadas de *score* máximo e determinadas reversões “especiais” sobre permutações cujos elementos são todos positivos. Iremos discutir agora quais são as estruturas encontradas em permutações com todos os elementos positivos.

Seja π uma permutação cujos todos os elementos são positivos e considere também que π seja uma permutação *reduzida*, ou seja, π não possui elementos adjacentes consecutivos. Como anteriormente, a permutação π possui os elementos auxiliares $\pi_0 = 0$ e $\pi_{n+1} = n + 1$. Além disso, considere que a permutação induz uma ordem circular de sucessão dos elementos, isto é 0 é sucessor de $n + 1$.

Definição 145 (Intervalo de Moldura). Um *intervalo de moldura* em π é uma seqüência de elementos $\pi_j, \pi_{j+1}, \dots, \pi_{j+k}$ em π que obedeça a seguinte forma: $\pi_j = i$, $\pi_{j+k} = i + k$ e $\pi_t \in [i, i + k]$ para $j + 1 \leq t \leq j + k$.

Em qualquer permutação sem sinais o intervalo envolvendo todos os elementos é um intervalo de moldura. Considere a seguinte permutação reduzida como exemplo:

$$\pi = [0, \quad 2, \quad 1, \quad 3, \quad 5, \quad 4, \quad 6],$$

além do intervalo de moldura trivial $0 \quad 2 \quad 1 \quad 3 \quad 5 \quad 4 \quad 6$, há os intervalos $0 \quad 2 \quad 1 \quad 3$ e $3 \quad 5 \quad 4 \quad 6$. Dizemos que os intervalos de moldura $0 \quad 2 \quad 1 \quad 3$ e $3 \quad 5 \quad 4 \quad 6$ estão *contidos* no intervalo $0 \quad 2 \quad 1 \quad 3 \quad 5 \quad 4 \quad 6$, ou também que este intervalo *contém* os anteriores.

Definição 146 (Obstáculo). Dada uma permutação reduzida de elementos positivos, um intervalo de moldura é um *obstáculo* quando esse intervalo não contém nenhum outro intervalo de moldura.

Segundo Bergeron [8] um obstáculo definido como acima é equivalente aos conceitos de obstáculos utilizados nas duas teorias anteriores de Hannenhalli e Pevzner [29] e Kaplan *et al.* [32]. Bergeron redefine duas operações para a eliminação de obstáculos de uma permutação que já haviam sido utilizadas no trabalho de Hannenhalli e Pevzner: *corte de obstáculos* e *junção de obstáculos*. Dada uma permutação com apenas um obstáculo $i \quad \pi_j \dots \pi_{j+k-1} \quad i + k$ é possível eliminar esse obstáculo por meio da reversão $\rho(\pi^{-1}(i) + 1, \pi^{-1}(i + 1) - 1)$, que corta o obstáculo. Observe o seguinte exemplo:

$$\pi = [0, \quad 2, \quad 1, \quad 3, \quad 5, \quad 4, \quad 6] \quad \rightarrow \quad \pi\rho(1, 1) = [0, \quad -2, \quad 1, \quad 3, \quad 5, \quad 4, \quad 6]$$

Uma vez que $i < i + 1 < j$, então a aplicação da reversão $\rho(\pi^{-1}(i) + 1, \pi^{-1}(i + 1) - 1)$ sempre cria pelo menos um par orientado em $\pi\rho$: na pior das hipóteses será apenas o par $(-(j - 1) \quad j)$.

Dada uma permutação π que possua mais de um obstáculo, a operação de junção de obstáculos consiste em realizar uma reversão dos elementos de π que aparecem no intervalo entre o elemento final de um obstáculo até o elemento inicial de outro intervalo. Em outros termos, dados os obstáculos $i \dots i+k$ e $i' \dots i'+k'$, a reversão $\rho(\pi^{-1}(i+k), \pi^{-1}(i'))$ é uma operação de junção de obstáculos. Considere o seguinte exemplo:

$$\pi = [0, 3, 5, 7, 4, 6, 8, 2, 1, 9, 12, 10, 11, 13]$$

e

$$\pi\rho(6,9) = [0, 3, 5, 7, 4, 6, -9, -1, -2, -8, 12, 10, 11, 13].$$

Observe que a operação de junção sempre cria pelo menos dois pares orientados em $\pi\rho$.

A escolha de qual obstáculo deve sofrer uma operação de corte ou junção deve ser bem avaliada em uma permutação com mais de dois obstáculos pois é possível que após a aplicação da reversão um novo obstáculo surja na permutação. Tome a seguinte permutação como exemplo:

$$\pi = [0, 3, 5, 7, 4, 6, 8, 2, 1, 9, 12, 10, 11, 13].$$

Considere que foi realizada a escolha de aplicar um corte no obstáculo 3 5 7 4 6 8. Logo a permutação resultante será:

$$\pi\rho(2,4) = [0, 3, -4, -7, -5, 6, 8, 2, 1, 9, 12, 10, 11, 13].$$

Aplicando o algoritmo de ordenação em $\pi\rho(2,4)$ baseado na estratégia da escolha de uma reversão de máximo *score* obtemos a seguinte permutação:

$$\pi' = [0, 3, 4, 5, 6, 7, 8, 2, 1, 9, 12, 10, 11, 13].$$

Se aglutinarmos os elementos adjacentes consecutivos em um novo elemento da permutação e renomearmos todos os elementos segundo a sua posição relativa na nova permutação obtemos:

$$\pi'' = [0, 3, 2, 1, 4, 6, 5, 7, 8].$$

A permutação π'' possui dois obstáculos (são 0 3 2 1 4 e 4 6 5 7 8), ou seja, o número de obstáculos não foi reduzido após a aplicação do corte de obstáculos. Essa discussão remete ao seguinte conceito:

Definição 147 (Superobstáculo). Um obstáculo é *simples* se a aplicação de uma operação de corte ou junção ρ sobre esse obstáculo reduz o número de obstáculos em $\pi\rho$. Um obstáculo que não é simples é chamado de *super obstáculo*.

A partir desse conceito de super obstáculos, Bergeron [8] adaptou o algoritmo de eliminação de obstáculos de Hannenhalli e Pevzner de acordo com a sua teoria:

Dada uma permutação reduzida, então:

- Se a permutação possui $2k$ obstáculos, para $k \geq 2$, aplique a junção de quaisquer dois obstáculos não consecutivos.
- Se a permutação possui $2k + 1$ obstáculos, para $k \geq 1$, então se houver apenas um obstáculo simples corte-o, caso contrário aplique a junção de dois obstáculos não consecutivos (ou dois obstáculos consecutivos caso $k = 1$).

A partir da análise de alguns resultados da teoria de Hannenhalli e Pevzner [29], Bergeron encontra os subsídios necessários para provar que os dois algoritmos anteriores para eliminar obstáculos e realizar reversões orientadas de *score* máximo, de fato, ordenam uma permutação qualquer π . Primeiramente, chamaremos de uma *reversão segura* à reversão que não cria componentes ruins no grafo de sobreposição de uma permutação. Hannenhalli e Pevzner [29] haviam provado o seguinte resultado:

Proposição 148 (Hannenhalli e Pevzner). Qualquer seqüência de reversões seguras é ótima na ordenação da permutação π .

Bergeron demonstrou que reversões orientadas de máximo *score* são reversões que ordenam π .

Proposição 149 (Bergeron). Uma reversão orientada de *score* máximo é uma reversão segura.

Além disso, Bergeron mostrou também que um obstáculo em sua teoria é equivalente a um obstáculo na teoria de Hannenhalli e Pevzner [29] e Kaplan *et al.* [32]. Essa equivalência entre os conceitos de obstáculos é expressa em outros termos na Proposição 150:

Proposição 150. Se π é uma permutação reduzida, um componente não orientado do grafo de sobreposição é minimal se e somente se o seu *span* é um intervalo de moldura que não contém nenhum outro intervalo de moldura.

Para realizar a manipulação das permutações nos algoritmos, Bergeron optou por utilizar uma representação matricial e de vetores de bits. Uma matriz representa o grafo de sobreposição da permutação π , além disso um vetor de bits é utilizado para armazenar a orientação dos nós do grafo de sobreposição enquanto uma matriz armazena em cada uma de suas colunas a representação binária dos valores dos *scores* de cada reversão associada a um vértice do grafo de sobreposição. Por meio de um conjunto de procedimentos de manipulação bits, é possível realizar a inicialização das estruturas de dados e a aplicação de reversões sobre a permutação. Uma análise dos procedimentos utilizados por Bergeron demonstram que uma das operações

que demandam maior tempo de execução é o novo cálculo do *score* de cada reversão (nó) aplicável a permutação. O cálculo do *score* é limitado pelo número de nós vizinhos do nó corrente e portanto o tempo de execução do cálculo de seu *score* é $O(n^2)$. Uma vez que a distância de reversão é limitada por n , então o tempo de execução total do algoritmo é $O(n^3)$.

A teoria de Bergeron sobre o problema de ordenação por reversões de permutações com sinais tem sido a teoria mais fácil e intuitiva para a compreensão do problema e a que fornece um algoritmo de implementação mais simples entre todos os algoritmos projetados para esse problema. Por outro lado, essa apresentação do formalismo clássico apresenta alguns pontos negativos comparativamente às teorias anteriores. O algoritmo de Bergeron possui um tempo de execução superior ao algoritmo de Kaplan *et al.* [32]. Além disso, enquanto os algoritmos discutidos anteriormente poderiam ser utilizados após algumas alterações para resolver o problema de encontrar todas as seqüências de reversões que ordenam uma permutação com sinais, o algoritmo de Bergeron seria incapaz de ser adaptado para a solução desse problema, uma vez que a escolha de uma reversão para o próximo passo na ordenação é realizada em um conjunto menor do que o conjunto de todas as reversões ordenantes. De maneira semelhante aos demais trabalhos sobre o problema de ordenação por reversões com sinais, a teoria de Bergeron não eliminou a necessidade de diagramas para justificar alguns de seus teoremas, como na demonstração do efeito de uma reversão com sinais no grafo de sobreposição, o que acreditamos ser um recurso inadequado.

A.4 Hartman e Shamir (2006)

Uma simplificação da teoria de ordenação de permutações por transposições foi realizada por Hartman e Shamir [30]. Primeiramente, os autores demonstram que qualquer algoritmo projetado para resolver o problema de ordenação por transposição de permutações “lineares” pode ser aplicado para a resolução do mesmo problema no caso de permutações “circulares”. No entanto a grande contribuição do artigo é a redução dos conceitos e estruturas que devem ser analisadas no diagrama de ciclos e o projeto de um novo algoritmo aproximado para o problema de ordenação por transposições. Em sua teoria, os autores utilizaram estruturas de dados projetadas por Kaplan e Verbin [33] que permitiram uma expressiva redução no tempo de execução do algoritmo. A estrutura de dados desenvolvida por Kaplan e Verbin é uma combinação de uma árvore *splay* com listas de blocos da permutação a ser ordenada.

Apresentamos a definição de transposição segundo o trabalho de Hartman e Shamir [30]. Dada uma permutação $\pi = [\pi_1, \dots, \pi_n]$ sobre $\{1, 2, \dots, n\}$ que representa um genoma, define-se o *segmento* A em π como uma seqüência de elementos consecutivos π_i, \dots, π_j , onde $j \geq i$. Dois segmentos $A = \pi_i, \dots, \pi_k$ e $B = \pi_j, \dots, \pi_l$ são *contíguos* caso $j = k + 1$ ou $i = l + 1$. Uma transposição τ em π é o evento que troca dois segmentos contíguos. Em outros

termos, dada a permutação

$$\pi = [\pi_1, \dots, \pi_i, \dots, \pi_j, \dots, \pi_k, \dots, \pi_n]$$

a permutação $\tau(i, j, k)$ troca os segmentos $\pi_i \dots \pi_{j-1}$ e $\pi_j \dots \pi_{k-1}$ em $\pi\tau(i, j, k)$:

$$\pi\tau(i, j, k) = [\pi_1 \pi_2 \dots \pi_j \dots \pi_{k-1} \pi_i \dots \pi_{j-1} \dots \pi_k \dots \pi_n].$$

Uma permutação circular é uma permutação $\pi = (\pi_1 \pi_2 \dots \pi_n)$ sobre $[n]$ tal que se considera os seus índices como uma seqüência circular, i.e. $\pi(\pi_i) = \pi_{i+1}$ para $1 \leq i \leq n-1$ e $\pi(\pi_n) = \pi_1$. Existem n seqüências de elementos que representam uma permutação circular de acordo com a escolha de qual será o primeiro elemento da seqüência. Infelizmente, os autores não deixam claro se foi adotada uma convenção a respeito de qual seria a representante canônica de uma permutação circular. Uma convenção adequada para a representante canônica seria a de que o primeiro elemento dessa seqüência representante seja o elemento 1.

Uma transposição $\tau(i, j, k)$ sobre uma permutação circular π é o evento que troca dois segmentos disjuntos e contíguos π_i, \dots, π_{j-1} e π_j, \dots, π_k em π . De fato, uma transposição define uma partição dos elementos de uma permutação circular em três segmentos contíguos dois a dois. Uma vez que a permutação é circular, os segmentos são contíguos disjuntos e é possível ordenar três segmentos em uma ordem cíclica apenas de duas maneiras, então uma transposição em uma permutação circular pode ser realizada por meio da troca de quaisquer dois segmentos da permutação.

Dada uma permutação π (linear ou circular) o *problema de ordenação por transposições* consiste em determinar uma seqüência mínima de transposições τ_1, \dots, τ_k tal que

$$\pi\tau_1\tau_2\dots\tau_k = \iota$$

onde ι é a permutação identidade $[1 \ 2 \ \dots \ n]$ (observe que no caso de permutações circulares a permutação a qual π deve ser transformada é $(1 \ 2 \ \dots \ n)$ que **não** é a identidade). O número de transposições k em uma seqüência mínima é chamado de *distância de transposição* da permutação π e é denotado por $d_t(\pi)$.

Hartman e Shamir mostram que os problemas de ordenação de permutações circulares e lineares são equivalentes pois existe uma seqüência de transposições que ordena uma permutação linear π e sua equivalente permutação circular π^c .

Proposição 151. O problema de ordenar permutações lineares por transposições é equivalente ao problema de ordenar permutações circulares por transposições.

Nas definições que se seguem Hartman e Shamir adotaram a representação de genomas por meio de permutações circulares. Dada a permutação π sobre $[n]$ defina $f(\pi)$ como a permutação sobre $[2n]$ tal que cada elemento i de π é substituído pelos elementos $2i-1$ e $2i$, nessa ordem,

em $f(\pi)$. Uma transposição $\tau(i, j, k)$ é chamada *legal* quando i, j e k são ímpares. Uma vez que a permutação é circular, os índices e elementos $2n + 1$ e 1 serão considerados idênticos.

Dadas a permutação circular $\pi = (\pi_1 \dots \pi_n)$ e $f(\pi) = \pi' = (\pi'_1 \dots \pi'_{2n})$, o *grafo de pontos de quebra* $B(\pi)$ é um diagrama que possui como conjunto de vértices os elementos $\{1, 2, \dots, 2n\}$ e π'_{2i} é ligado a π'_{2i+1} por uma aresta preta, chamada b_i e o elemento $2i$ é ligado a $2i + 1$ por meio de uma aresta cinza, para $1 \leq i \leq n$.

O grafo de pontos de quebra distingue-se do diagrama (ou grafo) de ciclos devido a ser construído sobre $f(\pi)$ o invés de de ser construído diretamente da permutação π como na teoria de Bafna e Pevzner [7].

Uma vez que cada vértice possui duas arestas incidentes, o diagrama possui uma única decomposição em ciclos. O número de ciclos de $B(\pi)$ é denotado por $c(\pi)$. O comprimento de um ciclo é obtido de seu número de arestas pretas. Um ciclo é chamado de k -ciclo se possui comprimento k . Um ciclo é *ímpar* quando o seu comprimento é ímpar, caso contrário será um ciclo *par*. O número de ciclos ímpares em $B(\pi)$ é denotado por $c_{odd}(\pi)$. A variação do número de ciclos devido a aplicação de uma transposição τ à permutação π é denotada como $\Delta c(\pi, \tau) = c(\pi\tau) - c(\pi)$, enquanto a variação do número de ciclos ímpares é denotada por $\Delta c_{odd}(\pi, \tau) = c_{odd}(\pi\tau) - c_{odd}(\pi)$. Bafna e Pevzner [7] haviam provado anteriormente que $\Delta c(\pi, \tau) \in \{-2, 0, 2\}$ e $\Delta c_{odd}(\pi, \tau) \in \{-2, 0, 2\}$. Além disso, Bafna e Pevzner ofereceram o melhor limitante inferior para a distância de transposição que conhecemos até o momento: $d(\pi) \geq \frac{n+1-c_{odd}(\pi)}{2}$. Embora esses resultados tenham sido demonstrados para permutações lineares, os mesmos são válidos para permutações circulares.

Há apenas dois tipos possíveis de configurações de 3-ciclos que podem ser obtidos por meio da aplicação de transposições legais. A Figura A.8 mostra os dois tipos de configurações de um 3-ciclo.

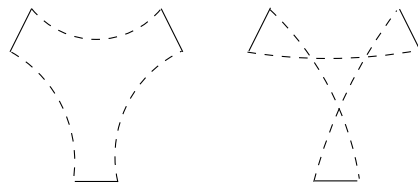


Figura A.8: Configurações de 3-ciclos que podem ser obtidos por meio da aplicação de transposições legais. O 3-ciclo da esquerda é não orientado enquanto o da direita é orientado.

Dada uma permutação π , uma transposição legal τ atua sobre as arestas pretas b_i, b_j e b_k em $B(\pi)$ caso $\tau = \tau(i, j, k)$. Uma transposição τ é uma k -transposição se $\Delta c_{odd}(\pi, \tau) = k$. Um ciclo é chamado de *ciclo orientado* se existe uma 2-transposição que atua em três de suas arestas pretas, caso contrário o ciclo será *não orientado*. Nas duas possíveis configurações de

um 3-ciclo em $B(\pi)$, observados na Figura A.8, o ciclo da esquerda é não orientado enquanto o da direita é orientado.

Seja π uma permutação circular e i_1, \dots, i_k uma seqüência cíclica de elementos, i.e. i_1 é o sucessor de i_k , então chama-se de *arco* um subconjunto de elementos contíguos na seqüência anterior. O par (i_j, i_l) , tal que $i \neq j$, define dois arcos disjuntos i_j, \dots, i_{l-1} e i_l, \dots, i_{j-1} . Por outro lado uma tripla (b_i, b_j, b_k) , onde $i \neq j \neq k$, define uma partição da seqüência cíclica em três arcos disjuntos. Os pares de arestas pretas (a, b) e (c, d) *se cruzam* caso as arestas a e b pertençam a arcos distintos definidos por (c, d) . Um par de arestas pretas (a, b) *cruza* um ciclo C se existe um par de arestas pretas (c, d) em C , i.e. c e d pertencem a C , tal que (a, b) e (c, d) se cruzam. Os ciclos C e D *se cruzam* se houver pares de arestas pretas em C e D que se cruzam. Duas triplas de arestas pretas *se entrelaçam* quando cada uma das arestas de uma das triplas pertence a um arco distinto definido pela segunda tripla, nesse caso diz-se que as triplas são *entrelaçadas*. Do mesmo modo, dois 3-ciclos *se entrelaçam* ou são *entrelaçados* caso suas arestas se entrelaçam.

Um k -ciclo em $B(\pi)$ é *curto* se $k \leq 3$, caso contrário será um ciclo *longo*. O grafo de pontos de quebra, bem como a permutação que o define, é chamado *simples* quando possui apenas ciclos curtos. Hartman e Shamir [30], com a expectativa de simplificar a estrutura do grafo de pontos de quebra, isto é, as relações de entrelaçamento e cruzamento entre os ciclos e arestas, utilizaram uma transformação de permutações arbitrárias em permutações simples semelhante às transformações equivalentes utilizadas por Hannenhalli e Pevzner [29] no diagrama de ciclos.

Dado o grafo de pontos de quebra $B(\pi)$, seja $b = (v_b, w_b)$ uma aresta preta e $g = (v_g, w_g)$ uma aresta cinza ambas pertencentes ao ciclo

$$C = (\dots, v_b, w_b, \dots, v_g, w_g, \dots),$$

de $B(\pi)$. Um (g, b) -split de $B(\pi)$ consiste na seguinte série de operações sobre $B(\pi)$:

- Remoção das arestas b e g de $B(\pi)$.
- Adição de dois novos vértices v e w a $B(\pi)$.
- Adição de duas novas arestas pretas (v_b, v) e (w, w_b) .
- Adição de duas novas arestas cinzas (w_g, w) e (v, v_g) .

O grafo resultante da aplicação de um (g, b) -split em $B(\pi)$ é denotado por $\hat{B}(\pi)$. Observe que um (g, b) -split divide o ciclo C em dois novos ciclos em $\hat{B}(\pi)$. Hannenhalli e Pevzner [29] demonstraram que existe uma permutação $\hat{\pi}$ sobre o conjunto $\{1, \dots, n+1\}$ que define o grafo $\hat{B}(\pi)$ e portanto $\hat{B}(\pi) = B(\hat{\pi})$, nesse caso as permutações $\hat{\pi}$ e π são chamadas *equivalentes*. Seja $n(\pi)$ o número de arestas pretas de π . Um (g, b) -split é chamado *seguro* quando essa operação preserva o parâmetro $n(\pi) - c_{odd}(\pi)$, ou em outros termos, temos $n(\pi) - c_{odd}(\pi) = n(\hat{\pi}) - c_{odd}(\hat{\pi})$.

Proposição 152 (Lin e Xue [36]). Qualquer permutação pode ser transformada em uma permutação simples por meio de splits seguros.

De fato, os splits seguros preservam o limitante inferior para a distância de transposição, devido a garantia de que $n(\pi) - c_{\text{odd}}(\pi) = n(\hat{\pi}) - c_{\text{odd}}(\hat{\pi})$. Além disso, uma sequência de transposições que ordena π é relacionada a uma sequência de transposições que ordena $\hat{\pi}$ por meio da seguinte Proposição 153 de Hannenhalli e Pevzner [29]:

Proposição 153 (Hannenhalli e Pevzner [29]). Dada uma permutação π , seja $\hat{\pi}$ uma permutação simples equivalente a π . Uma sequência de transposições que ordena $\hat{\pi}$ é equivalente (imita) uma sequência de transposições que ordena π com o mesmo número de operações.

A importância da Proposição 153 se deve a possibilidade de transformar uma permutação cujo grafo de pontos de quebra é composto de ciclos longos e com estruturas complexas de entrelaçamento e cruzamento (às vezes auto-cruzamentos) que exigiam a análise de diversos casos na teoria de Bafna e Pevzner [7] em uma permutação simples envolvendo apenas ciclos curtos com um número reduzido de relacionamentos entre os ciclos. Resta determinar as sequências de transposições que lidem com os ciclos curtos e obedeçam ao fator de aproximação de 1,5 para a distância de transposição.

Proposição 154 (Christie [17]). Se π é uma permutação que contém um 2-ciclo, então existe uma 2-transposição sobre π .

É possível eliminar todos os 3-ciclos orientados de π ao aplicar-se as 2-transposições atuantes em cada um desses ciclos. Tais transposições existem pela própria definição de 3-ciclos orientados. Por outro lado, por meio da Proposição 154 é possível eliminar todos os 2-ciclos de π . Logo resta encontrar uma maneira de eliminar os 3-ciclos não-orientados que utilize o máximo possível de 2-transposições. Uma $(0, 2, 2)$ -sequência é uma sequência de três transposições tal que a primeira transposição é uma 0-transposição enquanto as demais são 2-transposições.

Proposição 155. Dada uma permutação π que possui dois 3-ciclos não orientados que se entrelaçam, então existe uma $(0, 2, 2)$ -sequência de transposições sobre π .

Proposição 156. Dada uma permutação π , sejam C e D dois 3-ciclos não orientados que se cruzam e não são entrelaçados. Então, existe uma transposição que transforma C e D em um 1-ciclo e um 5-ciclo orientado.

Dada uma permutação π , a Proposição 155 garante que existe uma $(0, 2, 2)$ -sequência de transposições sobre π caso existam ciclos entrelaçados. Isso significa que de cada seis novos ciclos ímpares que poderiam ser criados em três transposições, de fato são obtidos apenas quatro ciclos ímpares novos o que garante a razão de 1,5 entre as transposições e o fator de aproximação. Por outro lado, a Proposição 156 afirma que existe uma 0-transposição τ que transforma

os dois 3-ciclos não orientados que se cruzam em um 1-ciclo e um 5-ciclo orientado. Uma vez que o 5-ciclo é orientado então existe uma 2-transposição atuante sobre arestas desse ciclo. Por meio da análise de casos, Hartman e Shamir determinaram que a aplicação da 2-transposição sobre o 5-ciclo resulta em dois novos ciclos de comprimento um e um 3-ciclo não orientado.

Um ciclo E é *fragmentado* pelos ciclos C e D se todos os pares de arestas de E cruzam com um par de arestas em C ou com uma par de arestas em D .

Proposição 157. Considere a permutação π que contém três 3-ciclos não orientados C , D e E , tais que E é fragmentado por C e D . Logo existe uma $(0, 2, 2)$ -seqüência de transposições sobre π .

Uma par de arestas pretas é chamado *adjacente* se ambas as arestas são incidentes sobre uma mesma aresta cinza.

Proposição 158 (Bafna e Bevezner [7]). Dada uma permutação π , seja (b_i, b_j) uma par de arestas adjacentes em um ciclo não orientado de π , então o par (b_i, b_j) cruza algum outro ciclo em π .

A Proposição 158 destaca o fato de que em qualquer ciclo de $B(\pi)$ existe um par de arestas que se cruza com um ciclo distinto em $B(\pi)$. Esse resultado garante que ao longo do processo de ordenação não surgirão 3-ciclos não orientados que não se cruzem a nenhuma aresta de outro ciclo e portanto os 3-ciclos de $B(\pi)$ sempre podem ser classificados em algumas das categorias anteriores: ciclos que se cruzam, ciclos que se entrelaçam ou ciclos fragmentados. Além disso, a Proposição 158 oferece uma maneira de realizar escolhas sobre pares de arestas que garantidamente existem em $B(\pi)$.

Segue abaixo o algoritmo de ordenação por transposições com fator de aproximação 1,5.

O algoritmo *Sort* mostrado na Figura A.9 consiste em primeiramente transformar uma permutação qualquer na entrada em uma permutação simples que mantenha o limitante inferior da distância de transposição. O segundo passo do algoritmo é eliminar os 2-ciclos ao se aplicar 2-transposições como garantido pela Proposição 154. Para os demais 3-ciclos restantes no grafo de pontos de quebra é escolhido um par adjacente de arestas pretas e verificado se o ciclo ao qual o par pertence é orientado ou não. Se o ciclo é orientado então existe uma 2-transposição atuante sobre o mesmo, caso contrário o ciclo poderá se entrelaçar a outro ciclo ou ser fragmentado por outros dois ciclos e portanto existe uma $(0, 2, 2)$ -seqüência aplicável a permutação. Se forem utilizadas estruturas de dados triviais para a representação da permutação, como por exemplo um vetor, então as operações fundamentais do algoritmo de aplicar uma transposição e de encontrar um par de arestas que cruzem a um par dado podem ser realizadas em tempo de execução linear no tamanho vetor. Logo, como o número de iterações da linha 2 e da linha 5 do algoritmo é limitado pelo número de ciclos do grafo de pontos de quebra, a complexidade desse algoritmo é quadrática no número de genes (elementos do conjunto).

Algoritmo Sort(π)

- 1: Transforme a permutação π em sua permutação simples equivalente $\hat{\pi}$ por meio de (g, b) -splits seguros (Proposição 152)
- 2: **while** $B(\hat{\pi})$ contiver um 2-ciclo **do**
- 3: aplique uma 2-transposição (Proposição 154)
- 4: **end while**
- 5: **while** $B(\hat{\pi})$ contiver um 3-ciclo **do**
- 6: Escolha um par adjacente de arestas pretas c de um ciclo C
- 7: **if** C é orientado **then**
- 8: Aplique a 2-transposição atuante sobre as arestas de C
- 9: **else**
- 10: Escolha um par adjacente d (de um ciclo D) que cruza com c (d existe devido a Proposição 158)
- 11: **if** os ciclos C e D são entrelaçados **then**
- 12: Aplique uma $(0, 2, 2)$ -seqüência (Proposição 155)
- 13: **else** {Existe um par adjacente c' em C que não cruza com D }
- 14: Escolha um par adjacente e (de um ciclo E) que cruza com o par c' (novamente a existência de e é garantida pela Proposição 158)
- 15: {O ciclo C é fragmentado pelos ciclos D e E }
- 16: Aplicar uma $(0, 2, 2)$ -seqüência (Proposição 157)
- 17: **end if**
- 18: **end if**
- 19: **end while**
- 20: **return** A seqüência de transposições que ordenam $\hat{\pi}$ em uma seqüência de transposições que ordenem π (Proposição 153)

Figura A.9: Algoritmo de Hartman e Shamir para o problema de Ordenação por Transposições. O algoritmo consiste primeiramente em encontrar uma permutação simples equivalente à permutação da entrada. Após isso, os 2-ciclos da permutação simples são eliminados por meio de 2-transposições, restando apenas 3-ciclos no grafo de pontos de quebra. Os 3-ciclos orientados são eliminados por 2-transposições, enquanto os 3-ciclos não orientados são eliminados por meio de uma $(0, 2, 2)$ -seqüência.

Proposição 159. O algoritmo *Sort* é um algoritmo de fator de aproximação 1,5 para o problema de ordenação de permutações por transposições e seu tempo de execução é $O(n^2)$, onde n é o número de genes.

Hartman e Shamir [30] utilizaram uma estrutura de dados alternativa a um vetor para representar a permutação. Os autores projetaram uma nova estrutura de dados que consiste em uma pequena modificação de uma estrutura de dados primeiramente desenvolvida por Kaplan e Verbin [33] para o problema de ordenação de uma permutação por reversões com sinais. Daremos uma breve exposição dessa estrutura de dados desenvolvida por Kaplan e Verbin.

Uma maneira alternativa de representar um par de arestas pretas (b_1, b_2) é através da aresta cinza $(2i, 2i + 1)$ que incide sobre as arestas b_1 e b_2 . Os elementos da aresta cinza são chamados *companheiros*. As operações básicas exigidas pelo algoritmo *Sort* são:

- $Query(\pi, e_1, e_2)$: Essa operação encontra um par que cruza com o par (e_1, e_2) em π .
- $Transp(\pi, e_1, e_2, e_3)$: Essa operação consiste na aplicação de uma transposição sobre π que atue sobre os elementos e_1, e_2 e e_3 .

A estrutura de dados consiste de uma lista de $\Theta(\sqrt{\frac{n}{\log n}})$ blocos com $\Theta(\sqrt{n \log n})$ elementos de $f(\pi)$ em cada bloco. Os elementos de cada bloco são ordenados de acordo com a ordem dos índices de seus respectivos elementos companheiros em $f(\pi)$. A cada um dos blocos é associada uma árvore *splay* contendo os elementos do bloco. Uma árvore *splay* é uma árvore binária balanceada auto-ajustável projetada por Sleator e Tarjan [55]. Além disso, cada elemento possui um apontador para o bloco a qual pertence.

Proposição 160. Considere a permutação π armazenada na estrutura de dados de lista de blocos/árvores *splay*. As operações de *Query* e *Transp* podem ser realizadas em tempo de execução $O(\sqrt{n \log n})$.

Se a estrutura de dados descrita anteriormente for utilizada para implementar o algoritmo *Sort* então o seu tempo de execução torna-se subquadrático.

Proposição 161. O algoritmo *Sort* é um algoritmo de fator de aproximação 1,5 para o problema de ordenação de permutações por transposições e seu tempo de execução é $O(n\sqrt{n \log n})$.

Hartman e Shamir [30] introduziram alguns conceitos provenientes da teoria de ordenação por reversões, como a idéia de permutações simples equivalentes, na teoria de ordenação por transposições, que mostraram-se fundamentais para uma significativa simplificação do tratamento do problema e ao mesmo tempo uma redução no tempo de execução do algoritmo de aproximação. A simplificação da teoria permitiu que a análise exaustiva de diversos casos de relacionamentos entre os ciclos fosse reduzida e a própria estrutura dos diagramas utilizados

fosse simplificada. Por outro lado, muitos dos argumentos utilizados pelos autores ainda são fortemente dependentes da análise de figuras e não foi apresentado nenhuma indicação sobre o papel das -2 -transposições no problema de ordenação (existiria alguma sequência mínima de transposições que ordenasse uma permutação e envolvesse -2 -transposições?). Além disso, uma questão importante que os autores não tratam em seu artigo é se o conjunto de sequências de transposições que o seu algoritmo fornece é o mesmo conjunto de sequências do algoritmo de Bafna e Pevzner.

A.5 Elias e Hartman (2006)

Elias e Hartman [22] apresentaram novas contribuições significativas para o problema de ordenação de permutações por transposições. A principal contribuição de Elias e Hartman é a proposição de um novo algoritmo aproximado com melhor fator de aproximação (fator 1,375) do que os anteriores para realizar a ordenação por transposições. Além disso, os autores apresentam uma melhoria no limitante inferior do diâmetro de transposição, contribuindo para o desenvolvimento dessa interessante linha de pesquisa. Relembrando, Christie [17] e Walter *et al.* [41] ambos conjecturaram que o valor do diâmetro seria $\lfloor (n-1)/2 \rfloor + 1$, que é a distância da permutação reversa. Eriksson e colegas [23] mostraram que para $n = 14$ e $n = 16$ (quando consideradas permutações circulares) há permutações mais difíceis de ordenar que a permutação reversa. Elias e Hartman estendem esta observação para todas as permutações de tamanhos pares a partir de 14. Finalmente, os autores apresentam o valor exato do diâmetro de permutações simples definidas na Seção A.4.

As definições de permutação e transposição dessa seção são as mesmas da Seção A.4. Elias e Hartman [22] tratam apenas de permutações circulares pois os resultados envolvendo estas permutações são mais claros do que os envolvendo permutações lineares, embora os resultados sejam equivalentes para ambos os tipos de permutações. Recordando as definições e notações da Seção A.4, uma permutação circular é uma permutação $\pi = (\pi_1 \pi_2 \dots \pi_n)$ sobre $\{1, \dots, n\}$ para a qual os índices de seus elementos formam uma sequência circular, ou seja $\pi(\pi_i) = \pi_{i+1}$ para $1 \leq i \leq n-1$ e $\pi(\pi_n) = \pi_1$. Um segmento de π é uma sequência de elementos consecutivos π_i, \dots, π_j , onde $j \geq i$. Dois segmentos $A = \pi_i, \dots, \pi_k$ e $B = \pi_j, \dots, \pi_l$ são contíguos caso $j = k+1$ ou $i = l+1$. Uma transposição τ em π é o evento de rearranjo que troca dois segmentos contíguos (quaisquer dois segmentos de uma partição em três segmentos da permutação circular π).

Dada uma permutação π (circular) o problema de ordenação por transposições consiste em determinar uma sequência mínima de transposições τ_1, \dots, τ_k tal que

$$\pi \tau_1 \tau_2 \dots \tau_k = (1 \ 2 \ \dots \ n).$$

O número de transposições k em uma sequência mínima é chamado de distância de transposição

da permutação π e é denotado por $d_t(\pi)$.

O diâmetro de transposição, denotado por $d_t(n)$ (Elias e Hartman denotam o diâmetro de transposição por $TD(n)$) é a máxima distância de transposição $d_t(\pi)$ para todo π no grupo de permutações definido sobre $\{1, \dots, n\}$.

Elias e Hartman [22] adotam uma definição do grafo de pontos de quebra equivalente à dada por Hartman e Shamir [30]. Além disso, são equivalentes às definições e notações referentes ao número de ciclos do grafo de pontos de quebra, aos ciclos ímpares, à variação do número de ciclos ímpares causada por uma transposição. Para facilitar a leitura, repetimos aqui algumas dessas definições e notações.

Dada a permutação circular $\pi = (\pi_0 \dots \pi_{n-1})$, o grafo de pontos de quebra $B(\pi)$ é um diagrama que possui como conjunto de vértices os elementos $\{l_0, r_0, \dots, l_{n-1}, r_{n-1}\}$. Para $0 \leq i \leq n-1$ os vértices r_i é ligado a l_{i+1} por uma aresta cinza e o elemento l_{π_i} é ligado a r_{π_i-1} por meio de uma aresta preta chamada b_i , para cada π_i . Essa definição do grafo de pontos de quebra distingue-se da definição de Hartman e Shamir [30] apenas pela utilização dos rótulos l e r ao invés de uma função f que dobra o valor de um elemento da permutação. Como se trata de um diagrama, a maneira de desenhá-lo é relevante. É conveniente, segundo Elias e Hartman, que o grafo seja desenhado em um círculo tal que as arestas pretas sejam dispostas na circunferência e as arestas cinzas sejam cordas. O grafo de pontos de quebra possui uma decomposição única em ciclos contendo arestas cinzas e pretas alternadas. Um k -ciclo possui k arestas pretas. Uma permutação π é uma 2-permutação (respectivamente, uma 3-permutação) quando $B(\pi)$ contém apenas 2-ciclos (respectivamente, apenas 3-ciclos).

Qualquer transposição em π corta três arestas pretas de $B(\pi)$ pela definição do grafo de pontos de quebra. Dada uma permutação π , uma transposição τ aplica-se sobre as arestas pretas b_i , b_j e b_k em $B(\pi)$ caso τ corte essas arestas pretas. A definição de uma “transposição que aplica-se sobre arestas pretas” distingue-se do conceito de uma “transposição que atua sobre arestas pretas” de Hartman e Shamir [30] apenas por este exigir que a transposição seja legal e seja definida para a permutação $f(\pi)$. Uma transposição τ é um k -movimento se $\Delta_{C_{odd}}(\pi, \tau) = k$ (o conceito de k -movimento é idêntico ao de k -transposição de Hartman e Shamir [30]). Um ciclo é chamado de ciclo orientado se existe uma 2-movimento que se aplica em três de suas arestas pretas, caso contrário o ciclo será não orientado.

Recordamos alguns conceitos sobre o relacionamento entre ciclos do grafo de pontos de quebra. Os pares de arestas pretas (a, b) e (c, d) se cruzam caso as arestas a, b, c e d apareçam na ordem a, c, b e d no grafo de pontos de quebra (essa definição é equivalente a definição de cruzamento de arestas pretas da Seção A.4). Um par de arestas pretas (a, b) cruza um ciclo C se existe um par de arestas pretas (c, d) em C tal que (a, b) e (c, d) se cruzam. Os ciclos C e D se cruzam se houver pares de arestas pretas um em C e outro em D que se cruzam. Duas triplas de arestas pretas (a, b, c) e (r, s, t) se entrelaçam quando as arestas ocorrem em ordem alternada, ou seja, na ordem a, r, b, s, c, t ou r, a, s, b, t, c o que equivale a cada uma das arestas de uma

das triplas pertencer a um arco distinto definido pela segunda tripla, caso em que diz-se que as triplas são entrelaçadas. Do mesmo modo, dois 3-ciclos se entrelaçam ou são entrelaçados caso suas arestas se entrelaçem. Um 3-ciclo C é *fragmentado* quando cada par de arestas pretas de C entrelaça com um par de arestas pretas de algum outro 3-ciclo. O conceito de 3-ciclo fragmentado de Elias e Hartman distingue-se do conceito de ciclo fragmentado por dois ciclos, pois permite que os ciclos de um componente composto apenas por dois 3-ciclos entrelaçados sejam também fragmentados.

Uma *configuração* de ciclos é um subgrafo do grafo de pontos de quebra que é induzido por estes ciclos. Uma configuração A é uma *subconfiguração* de uma configuração B quando os ciclos de A são um subconjunto dos ciclos de B . A configuração A é *conexa* quando, para quaisquer ciclos c_1 e c_k de A , existem ciclos c_2, \dots, c_{k-1} de A tais que c_i se entrelaça com c_{i+1} para $1 \leq i \leq k-1$. Um *componente* é uma configuração conexa maximal em um grafo de pontos de quebra. O *tamanho* de configurações e componentes é o número de ciclos que eles contêm. Uma configuração ou componente são chamados *orientados* (respectivamente *não orientados*) quando todos os ciclos da configuração ou componente são orientados (respectivamente, não orientados). Uma configuração ou componente são chamados de *pequenos* quando possuem no máximo 8 ciclos, caso contrário, são chamados *grandes*. Um *portão aberto* em um 2-ciclo ou 3-ciclo de uma configuração é um par de arestas pretas de um mesmo ciclo que não cruza nenhum outro ciclo da configuração. Uma configuração é chamada de *completa* quando não possui portões abertos.

Uma (x, y) -*seqüência* de transposições em uma permutação simples (para $x \geq y$) é uma seqüência de x transposições tal que pelo menos y dessas transposições são 2-movimentos e a permutação resultante da aplicação das transposições é uma permutação simples. Por exemplo, a $(0, 2, 2)$ -seqüência da Seção A.4 corresponde a uma $(3, 2)$ -seqüência. Uma $\frac{a}{b}$ -seqüência é uma (x, y) -seqüência tal que $x \leq a$ e $\frac{x}{y} \leq \frac{a}{b}$. Uma configuração (ou componente, ou permutação) possui uma (x, y) -seqüência se é possível aplicar tal seqüência em seus ciclos. Uma atenção especial é dada às configurações que possuem $\frac{11}{8}$ -seqüências.

A Proposição 162 é fundamental para a obtenção dos resultados desse artigo de Elias e Hartman. A Proposição 162 basea-se em resultados de Christie [17] e Hartman e Shamir [30] (ver Proposição 157).

Proposição 162. Para qualquer permutação, exceto a identidade, existe um 2-movimento ou uma $(3, 2)$ -seqüência de transposições.

Permutações simples, ou seja, permutações cujos ciclos têm comprimento no máximo 3, são importantes para o novo algoritmo apresentado nessa seção. Elias e Hartman [22] utilizam a mesma estratégia de Hartman e Shamir [30] de simplificar as relações de entrelaçamento e cruzamento entre os ciclos e arestas do grafo de pontos de quebra por meio da transformação de permutações arbitrárias em permutações simples. A Proposição 152 da Seção A.4 afirma que

qualquer permutação pode ser transformada de maneira segura em uma permutação simples. Além disso, segundo a Proposição 153 de Hannenhalli e Pevzner [29], dada uma permutação simples $\hat{\pi}$ equivalente a uma permutação π , uma seqüência de transposições que ordena $\hat{\pi}$ é equivalente (imita) uma seqüência de transposições que ordena π com o mesmo número de operações. O algoritmo de Elias e Hartman transforma a permutação em sua entrada em uma permutação simples antes de iniciar a ordenação por transposições.

A.5.1 Resultados de Diâmetro de Transposição

Elias e Hartman [22] discutem o diâmetro de transposição no caso geral e para alguns subconjuntos do grupo de permutações. Eles demonstram que $\lfloor n/2 \rfloor + 1$ é um limitante inferior mais justo para o diâmetro de transposição. Em seguida, demonstram que $\lfloor n/2 \rfloor$ é o valor do diâmetro de transposição para permutações simples. Finalmente, os autores apresentam um limitante superior para o diâmetro de transposição para 3-permutações.

De maneira semelhante a definição do diâmetro de transposição, o *diâmetro de transposição de permutações simples* denotado por $TDS(n)$ é o máximo valor de $d_t(\pi)$ para qualquer permutação simples π de n elementos. Do mesmo modo são definidos o diâmetro de transposição de 2-permutações (denotado por $TD2$) e o diâmetro de transposição de 3-permutações (denotado por $TD3$). Elias e Hartman fazem uma observação relevante de que o termo “diâmetro de transposição” não é o mais adequado para os conjuntos de permutações simples, 2-permutações e 3-permutações, pois esses conjuntos não são grupos, embora os autores mantenham a nomenclatura por consistência.

A distância de transposição da permutação reversa $(0 \ n-1 \ \dots \ 1)$ é $\lfloor (n-1)/2 \rfloor + 1$. Diversos autores [41, 17, 23] conjecturaram que o diâmetro de transposição é igual à distância de transposição da permutação reversa (exceto para $n = 14$ e $n = 16$ na conjectura formulada por Erksson e al. [23]). Elias e Hartman demonstram que a conjectura baseada na distância de transposição da reversa é falsa.

Proposição 163 (Elias e Hartman [22]). O diâmetro de transposição $d_t(n)$ do grupo de permutações com n elementos obedece a desigualdade $d_t(n) \geq \lfloor \frac{n}{2} \rfloor + 1$.

O limitante inferior da Proposição 163 é satisfeito pela permutação reversa para n ímpar. Para n par, os autores demonstram que uma permutação da forma

$$\pi = (0 \ 4 \ 3 \ 2 \ 1 \ 5 \ 13 \ 12 \ 11 \ 10 \ 9 \ 8 \ 7 \ 6 \ 14 \ 17 \ 16 \ 15 \ \dots \ 4i + 14 \ 4i + 17 \ 4i + 16 \ 4i + 15)$$

para $n \equiv 2 \pmod{4}$ e $n = 14 + 2k$ (onde k é par e $0 \leq i \leq k-1$) possui distância de transposição igual a $\lfloor n/2 \rfloor + 1$. A demonstração de que permutações do formato anterior satisfazem o limitante inferior envolve uma análise de casos sobre as maneiras de ordenar a permutação. Por exemplo, cada um dos intervalos $[0, 5]$, $[5, 14]$ e $[14, 0]$ pode ser ordenado separadamente

ou pode-se utilizar transposições que envolvam elementos de intervalos distintos. Ao fim da análise de todas as maneiras de ordenar a permutação, verifica-se que a permutação satisfaz o limitante inferior.

O diâmetro de transposição para permutações simples é dado pela Proposição 164:

Proposição 164 (Elias e Hartman [22]). Dado o conjunto de permutações simples de n elementos temos $TDS(n) = \lfloor \frac{n}{2} \rfloor$.

A demonstração da Proposição 164 consiste em mostrar que a distância de transposição de qualquer permutação π obedece a $d_t(\pi) \geq \lfloor \frac{n}{2} \rfloor$ do que decorre $TDS(n) \geq \lfloor \frac{n}{2} \rfloor$. Elias e Hartman demonstram que $TDS(n) \leq \lfloor \frac{n}{2} \rfloor$ por indução em n . A demonstração do passo indutivo envolve uma análise de diversos casos de entrelaçamento entre os ciclos de $B(\pi)$ para determinar se é possível reduzir a permutação π a uma permutação simples π' por meio de no máximo m transposições tal que $m + d(\pi') \leq \lfloor \frac{n}{2} \rfloor$.

A obtenção de um limitante superior para o diâmetro de transposição de 3-permutações é a base para a demonstração do algoritmo 1,375-aproximado. Os autores demonstram que qualquer 3-permutação com pelo menos oito ciclos tem uma $\frac{11}{8}$ -seqüência (Proposição 167). A demonstração desse resultado é realizada em duas etapas: primeiramente demonstra-se que todos os componentes grandes de uma 3-permutação possui uma $\frac{11}{8}$ -seqüência e em seguida demonstra-se que 3-permutações com pelo menos oito ciclos e que contenham apenas componentes pequenos possuem uma $(11, 8)$ -seqüência.

A demonstração de que todos os componentes grandes de uma 3-permutação possui uma $\frac{11}{8}$ -seqüência exige uma análise de um número extremamente grande de casos baseados em configurações do grafo de pontos de quebra. A enumeração desses casos pode ser realizada de maneira sistemática se analisado o relacionamento entre configurações e subconfigurações. As configurações mais básicas são as não orientadas, conexas e formadas por apenas dois 3-ciclos não orientados. Há apenas duas configurações desse tipo: dois ciclos cruzados não orientados ou dois ciclos entrelaçados não orientados. Qualquer configuração não orientada e conexa pode ser contruída a partir das duas configurações básicas por meio da *adição* de um novo ciclo não orientado tal que alguma de suas arestas pretas seja cruzada com algum dos ciclos na configuração. Se é possível criar uma configuração B ao adicionar um ciclo a uma configuração A , então B é chamado de uma *extensão* de A . Qualquer componente pode ser construído por meio de dois tipos de extensões chamadas *extensões suficientes*: (1) extensões que eliminam (*fecham* na terminologia de Elias e Hartman) um portão aberto e (2) extensões de configurações completas que criam no máximo um portão aberto. Configurações que podem ser construídas por meio de uma série de extensões suficientes a partir das configurações básicas são chamadas *configurações suficientes*.

Os componentes avaliados pela demonstração são as configurações suficientes de um grafo de pontos de quebra que contenha nove ciclos. A análise de casos para esse tipo de grafo de

Algoritmo Análise de Casos

```

1: Inicie uma fila de configurações com o par de ciclos não orientados que se entrelaçam e
   com o par de ciclos não orientados que se cruzam.
2: while a fila não estiver vazia do
3:   Remova a primeira configuração  $A$  da fila.
4:   for all cada extensão suficiente  $B$  de  $A$  do
5:     if  $B$  não possui uma  $\frac{11}{8}$ -seqüência then
6:       Adicione  $B$  a fila.
7:     else
8:       Retorne a seqüência que ordena  $B$ .
9:     end if
10:  end for
11: end while

```

Figura A.10: Algoritmo para a Análise de Casos da Demonstração do Diâmetro de Transposição de 3-Permutações.

pontos de quebra envolve cerca de 80000 diferentes maneiras de ordenar as configurações suficientes. Para avaliar cada um desses casos, Elias e Hartman [22] implementaram um programa que se assemelha a uma busca em largura no espaço de casos a serem verificados. O programa retorna uma seqüência de transposições que ordena um segmento de uma permutação que corresponde a configuração suficiente no grafo de pontos de quebra. A Proposição 165 sintetiza a primeira etapa que lida com componentes grandes.

Proposição 165 (Elias e Hartman [22]). Qualquer configuração suficiente não orientada com nove ciclos possui uma $\frac{11}{8}$ -seqüência.

A Figura A.10 ilustra um algoritmo que realiza a verificação de cada caso da demonstração da Proposição 165.

Finalizada a demonstração para o caso de componentes grandes, é necessário avaliar os grafos de pontos de quebra contendo apenas componentes pequenos e não orientados. Componentes pequenos de um grafo de pontos de quebra que não possuem uma $\frac{11}{8}$ -seqüência são chamados de *componentes pequenos ruins*. Elias e Hartman determinaram com ajuda de um programa que existem apenas cinco componentes pequenos ruins. Com o auxílio de outro programa para determinar as seqüências que ordenam os componentes, os autores demonstram o resultado enunciado na Proposição 166.

Proposição 166 (Elias e Hartman [22]). Qualquer 3-permutação com pelo menos oito ciclos e que contenham apenas componentes pequenos ruins possuem uma $(11, 8)$ -seqüência.

A Proposição 165 e a Proposição 166 permitem deduzir a Proposição 167.

Proposição 167 (Elias e Hartman [22]). Qualquer 3-permutação com pelo menos oito ciclos possui uma $\frac{11}{8}$ -seqüência.

Para 3-permutações de n elementos, o número de ciclos é $c = n/3$. Seja $g(c) = 11\lfloor c/8 \rfloor + \lfloor 3(c \bmod 8)/2 \rfloor$ e define-se uma função f (distinta da utilizada por Hartman e Shamir [30]) como:

$$f(c) = \begin{cases} g(c) + 1 & \text{se } c \bmod 8 = 1, \\ g(c) & \text{caso contrário.} \end{cases}$$

Os autores demonstram por indução em $f(c)$ que qualquer seqüência de transposições que ordena uma 3-permutação por meio de $\frac{11}{8}$ -seqüências até a permutação possuir menos de 8 ciclos e por meio de $(3, 2)$ -seqüências a partir disso utiliza no máximo $f(c)$ transposições. O limitante superior para o diâmetro de transposição de 3-permutações é enunciado na Proposição 168.

Proposição 168 (Elias e Hartman [22]). $TD3(n) \leq f(\frac{n}{3})$.

A.5.2 Algoritmo 1,375-Aproximado

O novo algoritmo de ordenação por transposições com fator de aproximação 1,375 é mostrado na Figura A.11.

O algoritmo *EliasHartmanSort* mostrado na Figura A.11 consiste em primeiramente transformar uma permutação qualquer na entrada em uma permutação simples que mantenha o limitante inferior da distância de transposição. O segundo passo do algoritmo é eliminar os 2-ciclos por meio de 2-movimentos como garantido pela Proposição 154 da Seção A.4. Após a eliminação de todos os 2-ciclos, a permutação resultante é uma 3-permutação. Todos os 3-ciclos restantes no grafo de pontos de quebra são marcados. Para cada 3-ciclo de $B(\pi)$ verifica-se se o ciclo é orientado ou não. Se o ciclo é orientado então existe uma 2-movimento atuante sobre o mesmo, caso contrário o ciclo é estendido por oito vezes. Se uma configuração grande foi encontrada então aplica-se uma $\frac{11}{8}$ -seqüência. Caso contrário, temos um componente pequeno. Se for possível aplicar uma $\frac{11}{8}$ -seqüência, então a aplicamos, senão todos os ciclos do componente pequeno são desmarcados. Após o fim do laço sobre 3-ciclos marcados, o grafo contém apenas componentes pequenos ruins. Enquanto o grafo contiver pelo menos oito ciclos, aplica-se uma $\frac{11}{8}$ -seqüência. Quando o grafo contiver menos de 8 ciclos, aplica-se uma $(3, 2)$ -seqüência até que todos os 3-ciclos tenham sido eliminados. Finalmente, as transposições que ordenaram a transposição simples são transformadas em transposições equivalentes na permutação original na entrada do algoritmo.

Proposição 169 (Elias e Hartman [22]). O algoritmo *EliasHartmanSort* é um algoritmo de fator de aproximação 1,375 para o problema de ordenação de permutações por transposições e seu tempo de execução é $O(n^2)$, onde n é o número de elementos.

Algoritmo EliasHartmanSort(π)

```

1: Transforme a permutação  $\pi$  em sua permutação simples equivalente  $\hat{\pi}$  (Proposição 152)
2: while  $B(\hat{\pi})$  contiver um 2-ciclo do
3:   Aplique um 2-movimento (Proposição 154)
4: end while
5: Marque todos os 3-ciclos da 3-permutação resultante
6: while  $B(\hat{\pi})$  contiver um 3-ciclo marcado  $C$  do
7:   if  $C$  é orientado then
8:     Aplique um 2-movimento atuante sobre as arestas de  $C$ 
9:   else
10:    Tente estender suficientemente o ciclo  $C$  oito vezes
11:    if Uma configuração grande é atingida then
12:      Aplique uma  $\frac{11}{8}$ -seqüência (Proposição 165)
13:    else {Trata-se de um componente pequeno}
14:      if É possível aplicar uma  $\frac{11}{8}$ -seqüência then
15:        Aplique-a
16:      else
17:        Desmarque todos os ciclos no componente
18:      end if
19:    end if
20:  end if
21: end while
    { A permutação resultante contém apenas componentes pequenos ruins }
22: while A permutação contiver pelo menos 8 ciclos do
23:   Aplique uma  $\frac{11}{8}$ -seqüência (Proposição 166 )
24: end while
25: while A permutação resultante contiver um 3-ciclo do
26:   Aplique uma  $(3, 2)$ -seqüência (Proposição 162)
27: end while
28: Transforme a seqüência de transposições que ordenaram  $\hat{\pi}$  em uma seqüência de transposições que ordenem  $\pi$  (Proposição 153)
29: return A seqüência de transposições que ordena  $\pi$ 

```

Figura A.11: Algoritmo de Elias e Hartman para o problema de Ordenação por Transposições. O algoritmo consiste primeiramente em encontrar uma permutação simples equivalente a permutação da entrada. Após isso, os 2-ciclos da permutação simples são eliminados por meio de 2-movimentos, restando apenas 3-ciclos no grafo de pontos de quebra, que são marcados. Os 3-ciclos orientados são eliminados por 2-movimentos, enquanto os 3-ciclos não orientados são estendidos até ser atingido uma configuração grande ou um componente pequeno. Se o componente ou configuração contiver uma $\frac{11}{8}$ -seqüência então essa seqüência é aplicada. Os 3-ciclos restantes são eliminados por meio de uma $(3, 2)$ -seqüência.

A.5.3 Relação entre Configurações e Grafos de Pontos de Quebra

Dois resultados que se referem ao relacionamento entre configurações e grafos de pontos de quebra são demonstrados por Elias e Hartman. O primeiro resultado permite testar se uma configuração descreve um grafo de pontos de quebra. O segundo resultado implica que uma configuração descreve um componente se e somente se ela descreve um grafo de pontos de quebra.

Dado um grafo de pontos de quebra $B(\pi)$ o *grafo de pontos de quebra complementar* $\bar{B}(\pi)$ é formado pela substituição da aresta preta $b_i = (l_{\pi_i}, r_{\pi_{i-1}})$ pela aresta preta complementar $\bar{b}_i = (l_{\pi_i}, r_{\pi_i})$ para cada i . O grafo de pontos de quebra complementar intuitivamente pode ser obtido por meio de uma transformação que move todas as arestas pretas no diagrama de uma posição em sentido horário na circunferência do diagrama. De maneira semelhante, define-se a *configuração complementar* \bar{C} de uma configuração C por meio do movimento de uma posição no sentido horário de todas as arestas pretas dos ciclos da configuração.

Proposição 170 (Elias e Hartman [22]). Uma configuração é um grafo de pontos de quebra se e somente se a configuração complementar é um grafo hamiltoniano.

A Proposição 170 permite identificar se uma configuração é um grafo de pontos de quebra.

Proposição 171 (Elias e Hartman [22]). Se C é um subgrafo induzido por um componente em um grafo de pontos de quebra $B(\pi)$, então a configuração complementar \bar{C} é um grafo hamiltoniano.

As Proposições 170 e 171 implicam que uma configuração é um componente se e somente se ela descreve um grafo de pontos de quebra.

A.5.4 Discussão

Elias e Hartman [22] tiraram proveito do conceito de permutações simples para analisar os diversos padrões de configurações desse tipo de permutação. Por meio de uma análise assistida por computador de um número grande de casos, os autores conseguiram encontrar uma $\frac{11}{8}$ -seqüência de transposições que ordena determinadas configurações de uma permutação simples (mais especificamente de uma 3-permutação). Esse resultado permitiu determinar um limitante superior para o diâmetro de transposição de 3-permutações e também foi utilizado para obter um fator de aproximação de 1,375 para um algoritmo de ordenação por transposições. Um estudo mais elaborado dos diversos tipos de configuração do grafo de pontos de quebra pode contribuir para a obtenção de melhorias no fator de aproximação do problema ou sugerir semelhanças com estruturas combinatórias de problemas NP -difíceis e permitir uma classificação segundo a complexidade de tempo de execução do problema de ordenação por transposições.

Referências Bibliográficas

- [1] M. D. Adams, S. E. Celniker, R. A. Holt, C. A. Evans, J. D. Gocayne, P. G. Amanatides, S. E. Scherer, P. W. Li, R. A. Hoskins, R. F. Galle, R. A. George, S. E. Lewis, S. Richards, M. Ashburner, S. N. Henderson, G. G. Sutton, J. R. Wortman, M. D. Yandell, Q. Zhang, L. X. Chen, R. C. Brandon, Y.-H. C. Rogers, R. G. Blazej, M. Champe, B. D. Pfeiffer, K. H. Wan, C. Doyle, E. G. Baxter, G. Helt, C. R. Nelson, G. L. G. Miklos, J. F. Abril, A. Agbayani, H.-J. An, C. Andrews-Pfannkoch, D. Baldwin, R. M. Ballew, A. Basu, J. Baxendale, L. Bayraktaroglu, E. M. Beasley, K. Y. Beeson, P. V. Benos, B. P. Berman, D. Bhandari, S. Bolshakov, D. Borkova, M. R. Botchan, J. Bouck, P. Brokstein, P. Brottier, K. C. Burtis, D. A. Busam, H. Butler, E. Cadieu, A. Center, I. Chandra, J. M. Cherry, S. Cawley, C. Dahlke, L. B. Davenport, P. Davies, B. de Pablos, A. Delcher, Z. Deng, A. D. Mays, I. Dew, S. M. Dietz, K. Dodson, L. E. Doup, M. Downes, S. Dugan-Rocha, B. C. Dunkov, P. Dunn, K. J. Durbin, C. C. Evangelista, C. Ferraz, S. Ferriera, W. Fleischmann, C. Fosler, A. E. Gabrielian, N. S. Garg, W. M. Gelbart, K. Glasser, A. Glodek, F. Gong, J. H. Gorrell, Z. Gu, P. Guan, M. Harris, N. L. Harris, D. Harvey, T. J. Heiman, J. R. Hernandez, J. Houck, D. Hostin, K. A. Houston, T. J. Howland, M.-H. Wei, C. Ibegwam, M. Jalali, F. Kalush, G. H. Karpen, Z. Ke, J. A. Kennison, K. A. Ketchum, B. E. Kimmel, C. D. Kodira, C. Kraft, S. Kravitz, D. Kulp, Z. Lai, P. Lasko, Y. Lei, A. A. Levitsky, J. Li, Z. Li, Y. Liang, X. Lin, X. Liu, B. Mattei, T. C. McIntosh, M. P. McLeod, D. McPherson, G. Merkulov, N. V. Milshina, C. Mobarri, J. Morris, A. Moshrefi, S. M. Mount, M. Moy, B. Murphy, L. Murphy, D. M. Muzny, D. L. Nelson, D. R. Nelson, K. A. Nelson, K. Nixon, D. R. Nusskern, J. M. Pacleb, M. Palazzolo, G. S. Pittman, S. Pan, J. Pollard, V. Puri, M. G. Reese, K. Reinert, K. Remington, R. D. C. Saunders, F. Scheeler, H. Shen, B. C. Shue, I. Sidén-Kiamos, M. Simpson, M. P. Skupski, T. Smith, E. Spier, A. C. Spradling, M. Stapleton, R. Strong, E. Sun, R. Svirskas, C. Tector, R. Turner, E. Venter, A. H. Wang, X. Wang, Z.-Y. Wang, D. A. Wassarman, G. M. Weinstock, J. Weissenbach, S. M. Williams, T. Woodage, K. C. Worley, D. Wu, S. Yang, Q. A. Yao, J. Ye, R.-F. Yeh, J. S. Zaveri, M. Zhan, G. Zhang, Q. Zhao, L. Zheng, X. H. Zheng, F. N. Zhong, W. Zhong, X. Zhou, S. Zhu, X. Zhu, H. O. Smith, R. A. Gibbs, E. W. Myers, G. M. Rubin, and J. C. Venter. The genome sequence of *Drosophila melanogaster*. *Science*, 287(5461):2185–2195,

March 2000.

- [2] M. Aigner and D. B. West. Sorting by insertion of leading element. *Journal of Combinatorial Theory*, 45:306–309, 1987.
- [3] S. Anderson, A.T. Bankier, B.G. Barrell, M.H.L. De Bruijn, A.R. Coulson, J. Drouin, I.C. Eperon, F. Sanger, P.H. Schreier, A.J.H. Smith, R. Staden, and I.G. Young. Sequence and organization of the human mitochondrial genome. *Nature*, 290:457 – 465, 1981.
- [4] D. A. Bader, B. M. E. Moret, and M. Yan. A linear-time algorithm for computing inversion distance between signed permutations with an experimental study. *Journal of Computational Biology*, 8(5):483–491, 2001.
- [5] V. Bafna and P. A. Pevzner. Sorting by transpositions. In *Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 614–623, San Francisco, USA, January 1995.
- [6] V. Bafna and P. A. Pevzner. Genome rearrangements and sorting by reversals. *SIAM Journal on Computing*, 25(2):272–289, 1996.
- [7] V. Bafna and P. A. Pevzner. Sorting by transpositions. *SIAM Journal on Discrete Mathematics*, 11(2):224–240, May 1998.
- [8] A. Bergeron. A very elementary presentation of the Hannenhalli-Pevzner theory. *Discrete Applied Mathematics*, 146(2):134–135, 2005. Versão preliminar apareceu em Proceedings of the 12th Annual Symposium of the Combinatorial Pattern Matching (CPM’2001).
- [9] A. Bergeron, J. Mixtacki, and J. Stoye. On sorting by translocations. *Journal of Computational Biology*, 13(2):567–578, 2006.
- [10] P. Berman and S. Hannenhalli. Fast sorting by reversal. In D. S. Hirschberg and E. W. Myers, editors, *Proceedings of Combinatorial Pattern Matching (CPM’96), 7th Annual Symposium*, volume 1075 of *Lecture Notes in Computer Science*, pages 168–185, Laguna Beach, USA, January 1996. Springer.
- [11] M. Blanchette, T. Kunisawa, and D. Sankoff. Parametric genome rearrangement. *Journal of Computational Biology*, 172:11–17, 1996.
- [12] G. Bourque, P. Pevzner, and G. Tesler. Reconstructing the genomic architecture of ancestral mammals: Lessons from human, mouse, and rat genomes. *Genome Research*, 4(14):507–516, 2004.

- [13] G. Bourque and P. A. Pevzner. Genome-scale evolution: Reconstructing gene orders in the ancestral species. *Genome Research*, 12(1):26–36, 2002.
- [14] A. Caprara. Formulations and hardness of multiple sorting by reversals. In S. Istrail, P. A. Pevzner, and M. Waterman, editors, *Proceedings of the 3rd Annual International Conference on Computational Molecular Biology (RECOMB'99)*, pages 84–93, Lyon, France, 1999. ACM Press.
- [15] A. Caprara. Sorting permutations by reversals and eulerian cycle decompositions. *SIAM Journal on Discrete Mathematics*, 12(1):91–110, February 1999.
- [16] D. A. Christie. Sorting permutations by block-interchanges. *Information Processing Letters*, 60(4):165–169, November 1996.
- [17] D. A. Christie. *Genome Rearrangement Problems*. PhD thesis, Glasgow University, 1998.
- [18] The Genome International Sequencing Consortium. Initial sequencing and analysis of the human genome. *Nature*, 409:860–921, 2001.
- [19] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms - Second Edition*. McGraw-Hill, 2001.
- [20] Z. Dias and J. Meidanis. Genome rearrangements distance by fusion, fission, and transposition is easy. In *Proceedings of the String Processing and Information Retrieval (SPIRE'2001)*, pages 250–253, Laguna de San Rafael, Chile, November 2001. IEEE Computer Society.
- [21] Z. Dias and J. Meidanis. Sorting by prefix transpositions. In A. H. F. Laender and A. L. Oliveira, editors, *Proceedings of the String Processing and Information Retrieval (SPIRE'2002)*, number 2476 in Lecture Notes in Computer Science, pages 65–76, Lisboa, Portugal, September 2002. Springer-Verlag, Berlin.
- [22] I. Elias and T. Hartman. A 1.375-approximation algorithm for sorting by transpositions. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 3(4):369–379, 2006.
- [23] H. Eriksson, K. Eriksson, J. Karlander, L. Svensson, and J. Wastlund. Sorting a bridge hand. *Discrete Mathematics*, 241:289–300, 2001.
- [24] S. Finnilä. *Phylogenetic analysis of mitochondrial DNA Detection of mutations in patients with occipital stroke*. PhD thesis, University of Oulu, 2000.
- [25] V. Fortuna and J. Meidanis. Sorting the reverse permutation by prefix transpositions. Technical Report IC-04-04, Computing Institute, University of Campinas, April 2004.

- [26] S. Hannenhalli. Polynomial-time algorithm for computing translocation distance between genomes. *Discrete Applied Mathematics*, 71:137–151, 1996.
- [27] S. Hannenhalli and P. A. Pevzner. Towards a computational theory of genome rearrangements. *Lecture Notes in Computer Science*, 1000:184–200, 1995.
- [28] S. Hannenhalli and P. A. Pevzner. Transforming men into mice (polynomial algorithm for genomic distance problem). In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science (FOCS'95)*, pages 581–592, Los Alamitos, USA, October 1995. IEEE Computer Society Press.
- [29] S. Hannenhalli and P. A. Pevzner. Transforming cabbage into turnip: Polynomial algorithm for sorting signed permutations by reversals. *Journal of the ACM*, 46(1):1–27, January 1999. Versão preliminar apareceu em Proceedings of the Twenty-Seventh Annual ACM Symposium on the Theory of Computing.
- [30] T. Hartman and R. Shamir. A simpler and faster 1.5-approximation algorithm for sorting by transpositions. *Information and Computation*, 204(2):275–290, 2006. Versão preliminar apareceu em Proceedings of CPM'03.
- [31] M. Jerrum. The complexity of finding minimum-length generator sequences. *Theoretical Computer Science*, 36:265–289, 1985.
- [32] H. Kaplan, R. Shamir, and R. E. Tarjan. Faster and simpler algorithm for sorting signed permutations by reversals. *SIAM Journal on Computing*, 29(3):880–892, January 2000. Versão preliminar apareceu em Proceedings of the Eighth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'97).
- [33] H. Kaplan and E. Verbin. Efficient data structures and a new randomized approach for sorting signed permutations by reversals. In *Combinatorial Pattern Matching, 14th Annual Symposium, CPM 2003, Morelia, Michocán, Mexico, June 25-27, 2003, Proceedings*, volume 2676 of *Lecture Notes in Computer Science*. Springer, 2003.
- [34] B. Lewin. *Genes VI*. Oxford University Press, 1997.
- [35] G. Li, X. Qi, X. Wang, and Zhu B. A linear-time algorithm for computing translocation distance between signed genomes. In *Proceedings of the 15th Annual Symposium of the Combinatorial Pattern Matching (CPM'04)*, volume 3109 of *Lecture Notes on Computer Science*, pages 323–332, 2004.
- [36] G.-H. Lin and G. Xue. Signed genome rearrangement by reversals and transpositions: Models and approximations. *Theoretical Computer Science*, 259:513–531, 2001.

- [37] Y. C. Lin, C. L. Lu, , H. Chang, and C. Y. Tang. An efficient algorithm for sorting by block-interchanges and its application to the evolution of vibrio species. *Journal of Computational Biology*, 12:102–112, 2005.
- [38] S. MacLane and G. Birkhoff. *Algebra*. The Macmillan Company, London, sixth printing edition, 1971.
- [39] J. Meidanis and Z. Dias. An alternative algebraic formalism for genome rearrangements. In D. Sankoff and J. H. Nadeau, editors, *Comparative Genomics: Empirical and Analytical Approaches to Gene Order Dynamics, Map Alignment and Evolution of Gene Families*, pages 213–223. Kluwer Academic Publishers, November 2000.
- [40] J. Meidanis, M. E. M. T. Walter, and Z. Dias. Distância de reversão de cromossomos circulares. In *Proceedings of the XXIV Seminário Integrado de Software e Hardware (SEMISH'97)*, pages 119–131, August 1997. In Portuguese.
- [41] J. Meidanis, M. E. M. T. Walter, and Z. Dias. Transposition distance between a permutation and its reverse. In R. Baeza-Yates, editor, *Proceedings of the 4th South American Workshop on String Processing (WSP'97)*, pages 70–79, Valparaiso, Chile, 1997. Carleton University Press.
- [42] J. Meidanis, M. E. M. T. Walter, and Z. Dias. Reversal distance of signed circular chromosomes. Technical Report IC-00-23, Institute of Computing - University of Campinas, December 2000.
- [43] C. Mira and J. Meidanis. Analysis of sorting by transpositions based on algebraic formalism, March 2004. Poster in The Eighth Annual International Conference on Research in Computational Molecular Biology (RECOMB2004).
- [44] C. Mira and J. Meidanis. Sorting by block-interchanges and signed reversals. In Sharham Latifi, editor, *Proceedings of The Fourth International Conference on Information Technology: New Generations ITNG 2007*, pages 670–676. IEEE Computer Society Press, April 2007.
- [45] C. Mira and J. Meidanis. Sorting by fissions, fusions, signed reversals in $o(n)$. *Transactions of Computational Biology and Bioinformatics*, 2007. submetido para avaliação.
- [46] C. Mira, Q. Peng, J. Meidanis, and P. Pevzner. A shortest-cycle based heuristics for the multiple genome rearrangement problem. Technical Report IC-06-23, Institute of Computing - University of Campinas, November 2006.
- [47] L. H. Jacy Monteiro. *Elementos de Álgebra*. Livros Técnicos e Científicos, 1969.

- [48] J. H. Nadeau and B. A. Taylor. Lengths of chromosomal segments conserved since divergence of man and mouse. *Proceedings of the National Academy of Sciences, USA*, 81:814–818, 1984.
- [49] M. Ozery-Flato and R. Shamir. Two notes on genome rearrangement. *Journal of Bioinformatics and Computational Biology*, 1(1):71–93, April 2003.
- [50] J. D. Palmer and L. A. Herbon. Plant mitochondrial DNA evolves rapidly in structure, but slowly in sequence. *Journal of Molecular Evolution*, 27:87–97, 1988.
- [51] P. A. Pevzner and M. S. Waterman. Open combinatorial problems in computational molecular biology. In *Proceedings of the 3rd Israel Symposium on Theory of Computing and Systems*, pages 158–163. IEEE Computer Society Press, 1995.
- [52] D. Sankoff and M. Blanchette. Multiple genome rearrangements. In S. Istrail, P. A. Pevzner, and M. Waterman, editors, *Proceedings of the 2nd Annual International Conference on Computational Molecular Biology (RECOMB-98)*, pages 243–247, New York, USA, March 1998. ACM Press.
- [53] J. C. Setubal and J. Meidanis. *Introduction to Computational Molecular Biology*. PWS Publishing Company, 1997.
- [54] A. J. G. Simpson, F.C. Reinach, P. Arruda, F. A. Abreu, M. Acencio, R. Alvarenga, L. M. C. Alves, J. E. Araya, G. S. Baia, C. S. Baptista, M. H. Barros, E. D. Bonaccorsi, S. Bordin, J. M. Bove, M. R. S. Briones, M. R. P. Bueno, A. A. Camargo, L. E. A. Camargo, D. M. Carraro, H. Carrer, N. B. Colauto, C. Colombo, F. F. Costa, M. C. R. Costa, C. M. Costa-Neto, L. L. Coutinho, M. Cristofani, E. Dias-Neto, C. Docena, H. El-Dorry, A. P. Facincani, A. J. S. Ferreira, V. C. A. Ferreira, J. A. Ferro, J. S. Fraga, S. C. Fran?a, M. C. Franco, M. Frohme, L. R. Furlan, M. Garnier, G. H. Goldman, M. H. S. Goldman, S. L. Gomes, A. Gruber, P. L. Ho, J. D. Hoheisel, M. L. Junqueira, E. L. Kemper, J. P. Kitajima, J. E. Krieger, E. E. Kuramae, F. Laigret, M. R. Lambais, L. C. C. Leite, E. G. M. Lemos, M. V. F. Lemos, S. A. Lopes, C. R. Lopes, J. A. Machado, M. A. Machado, A. M. B. N. Madeira, H. M. F. Madeira, C. L. Marino, M. V. Marques, E. A. L. Martins, E. M. F. Martins, A. Y. Matsukuma, C. F. M. Menck, E. C. Miracca, C. Y. Miyaki, C. B. Monteiro-Vitorello, D. H. Moon, M. A. Nagai, A. L. T. O. Nascimento, L. E. S. Netto, A. Nhani, F. G. Nobrega, L. R. Nunes, M. A. Oliveira, M. C. De Oliveira, R. C. De Oliveira, D. A. Palmieri, A. Paris, B. R. Peixoto, G. A. G. Pereira, H. A. Pereira, J. B. Pesquero, R. B. Quaggio, P. G. Roberto, V. Rodrigues, A. J. De M. Rosa, V. E. De Rosa, R. G. De Sá, R. V. Santelli, H. E. Sawasaki, A. C. R. Da Silva, A. M. Da Silva, F. R. Da Silva, W. A. Silva, J. F. Da Silveira, M. L. Z. Silvestri, W. J. Siqueira, A. A. De Souza, A. P. De Souza, M. F.

- Terenzi, D. Truffi, S. M. Tsai, M. H. Tsuchako, H. Vallada, M. A. Van Sluys, S. Verjovski-Almeida, A. L. Vettore, M. A. Zago, M. Zatz, J. Meidanis, and J. C. Setubal. The genome sequence of the plant pathogen *Xylella fastidiosa*. *Nature*, 406(6792):151–159, July 2000.
- [55] D. D. Sleator and R. E. Tarjan. Self-adjusting binary search trees. *Journal of ACM*, 32(3):652–686, 1985.
- [56] E. Tannier and M. Sagot. Sorting by reversals in subquadratic time. Technical Report 5097, INRIA, Institut National de Recherche en Informatique et en Automatique, January 2004.
- [57] N. Q. Tran. An easy case of sorting by reversals. *Journal of Computational Biology*, 5(4):741–746, 1998.
- [58] M. E. M. T. Walter. *Algoritmos para Problemas em Rearranjo de Genomas*. PhD thesis, University of Campinas, Brazil, 1999. In Portuguese.
- [59] L. Wang, D. Zhu, X. Liu, and S. Ma. An $O(n^2)$ algorithm for signed translocation. *J. Comput. Syst. Sci.*, 70(3):284–299, 2005.
- [60] Wolfram Research, Inc. Mathematica edition: Version 5.0, 2003. Software.
- [61] S. Yancopoulos, O. Attie, and R. Friedberg. Efficient sorting of genomic permutations by translocation, inversion and block interchange. *Bioinformatics*, 21(16):3340–3346, June 2005.
- [62] D. Zhu and L. Wang. On the complexity of unsigned translocation distance. *Theor. Comput. Sci.*, 352(1):322–328, 2006.

Índice Remissivo

- (g,b)-split, 171
 - seguro, 171
- árvore splay, 168
- órbita ímpar, 42
- órbita par, 42
- algoritmo
 - EliasHartmanSort, 182
 - exchange(), 129
 - FFSRSort, 131
 - Sort, 173
 - SRBISort, 136
 - TGSort, 121
 - TransSort, 27, 156
 - TSort, 155
 - updateLongPairQ, 124
 - updateShortPairQ, 124
- algoritmo aproximativo, 92
- alinhamento de seqüências, 8
- aresta cinza
 - entrelaçamento, 15
 - intervalo induzido por, 15
 - não orientada, 15
 - orientada, 15
- Brassica campestris, 8
- Brassica oleracea, 8
- bubble sort, 148
- célula, 61
- ciclo, 35
 - 1-ciclo, 35
 - k -ciclo, 26, 35
 - ímpar, 26, 149
 - alternante, 149
 - aparecer na ordem circular, 38
 - ciclos disjuntos, 36
 - curto, 26, 149
 - entrelaçamento, 15
 - fortemente orientado, 156
 - fragmentado, 173, 178
 - k-ciclo, 149
 - lista circular, 36
 - longo, 26, 149
 - não orientado, 15, 150
 - ordem circular, 38, 49
 - orientado, 15, 150
 - par, 26, 149
- ciclos
 - não-interferentes, 152
 - se cruzam, 152
- citosina, 60, 61
- clique feliz, 161
- cromossomo
 - circular, 7
 - linear, 7
- diâmetro
 - reversão com sinais, 10
 - transposição, 11, 27
- diagrama de ciclos, 13, 149
 - arcos, 15
 - aresta cinza, 13
 - aresta preta, 13

- ciclo alternante, 13
- comprimento de ciclo, 15
- distância de edição, 8
- distância genômica, 8
 - reversão com sinais, 10
 - reversão com sinais e transposição, 11
 - transposição, 11, 28
- estrutura de ciclos, 127
- estrutura de dados
 - lista de pares, 114, 129, 130
 - pLong, 114
 - pShort, 114
 - tabela de espalhamento, 127
 - updateArrays, 119
 - updateQ, 119
- evento 2-bom, 78, 128
- evento 3-bom, 101, 108, 109, 112, 113, 121
- evento de rearranjo, 8
- fissão, 127
- função, 36
 - restrição de uma função a um conjunto, 36
- função de mapeamento de genes em inteiros, 97
- fusão, 127
- gene, 60, 61
- genoma, 9
 - atingível, 106
 - atinge, 106
 - base, 7
 - bases complementares, 7
 - cromossomo, 8
 - cromossomos, 7
 - cromossomos correspondentes, 66
 - DNA, 61
 - facilmente transformável, 101
 - fita complementar, 67
 - fitas, 7
 - fitas complementares reversas, 7
 - gene, 7
 - multicromossomal, 7
 - orientação, 7
 - RNA, 60, 61
 - segmento, 168
 - segmentos contíguos, 168, 176
 - unicromossomal, 7
- genomas
 - compatíveis, 64
 - conorbitais, 71, 72, 108, 111, 113, 121
 - equicromossomais, 66
- grafo de pontos de quebra, 170, 177
- grafo de sobreposição, 16
 - componente não orientado, 16
 - componente orientado, 16
 - vértice não orientado, 16
 - vértice orientado, 16
- grafo de transposições boas, 102
 - σ -componente, 105
- grupo simétrico, 34
- guanina, 60, 61
- matriz, 35
- mitocôndria, 61
- mutações, 7
- não obstáculo, 20
- obstáculo, 19
 - conjunto de pontos finais, 20
 - corte de, 20, 165
 - simples, 20, 166
 - super, 20, 166
- obstáculos
 - consecutivos, 20
 - junção de, 23, 165
 - não consecutivos, 23

- par disjunto, 96
- par não-disjunto, 96
- permutação, 34
 - órbita, 35
 - órbita não trivial, 35
 - órbita trivial, 35
 - acessibilidade, 35
 - adjacência, 12
 - associatividade do produto, 34
 - com sinais, 8
 - composta, 34
 - conjugação, 39
 - conjugado, 39
 - conjugador, 39
 - decomposição em k -ciclos, 41
 - decomposição em ciclos disjuntos, 36
 - disjuntas, 37
 - elementos fixos, 37
 - estrutura de ciclos, 39
 - fortaleza, 23
 - identidade, 34
 - imagem, 12, 34
 - intervalo de moldura, 165
 - inversa, 34
 - mapear, 34
 - número de órbitas, 42
 - norma, 41
 - obstáculo, 165
 - operações de padding, 15
 - padding, 158
 - par não orientado, 18
 - par orientado, 18, 164
 - pares de pontos de quebra, 98
 - ponto de quebra, 12, 98, 148
 - pontuação, 19
 - posição, 8
 - produto, 34
 - reduzida, 165
 - representação como um produto de ciclos, 36
 - reversa, 28
 - suporte, 37
 - versão estendida, 12
- permutação com sinais
 - gap, 159
- permutações
 - equivalentes, 171
- polipeptídios, 60, 61
- problema de ordenação
 - reversão com sinais, 10
 - transposição, 11
- problema de rearranjo em genomas, 8
 - reversão com sinais, 10
 - reversão com sinais e transposição, 11
 - transposição, 11
- quociente
 - pares curtos, 114
 - pares longos, 114
- recombinações, 7
- reversão com sinais, 8, 10, 127
 - agir sobre gaps, 159
 - atuar em aresta cinza, 15
 - cortar obstáculo, 20
 - ordenante, 10
 - orientada, 18, 164
 - par, 159
 - própria, 15
 - score, 164
 - segura, 16, 20, 167
- transformação de órbitas em permutações com
 - sinais, 97
- transposição, 8, 11, 92, 147
 - x -movimento, 26
 - (x,y) -seqüência, 178

- 0-movimento bom, 27, 156
- atua sobre arestas, 150
- atua sobre um ciclo, 150
- boa, 101
- distância de, 97, 148
- legal, 26, 170
- limitante inferior da distância, 92, 97
- limitante inferior de distância, 99
- ordenante, 11
- shuffling transposition, 151
- válida, 27, 155
- x-movimento, 150
- transposição generalizada
 - algoritmo, 113, 121
 - estrutura de dados, 113
 - evento 3-bom, 108
- tripla
 - não-orientada, 151
 - orientada, 151
- triplas
 - entrelaçadas, 151