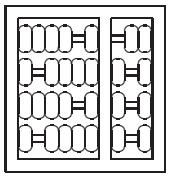

INSTITUTO DE COMPUTAÇÃO
UNIVERSIDADE ESTADUAL DE CAMPINAS

Algebraic Formalism for Genome Rearrangements (Part 1)

Cleber Mira *João Meidanis*

Technical Report - IC-05-10 - Relatório Técnico

June - 2005 - Junho

Algebraic Formalism for Genome Rearrangements (Part 1)

Cleber Mira * João Meidanis †

Abstract

One of the most important techniques of analysis of distinct genome sequences is the comparison of the order of their blocks of genes. Genome Rearrangements are mutational events that move large blocks of genes in a genome. We are interested in finding a minimum sequence of *rearrangement events* such as reversals and transpositions that transform one genome into another.

This work is composed by two parts. In the first part, we present the classical contributions for the genome rearrangement problem and discuss some of their characteristics. In the second part, we restate some problems and important results involving rearrangement events and present a new algebraic formalism based on Meidanis and Dias [16]. We prove the equivalence between the problems and some concepts of the classical theory and the new algebraic theory. The last objective of this work is to convince the reader that the algebraic model for genome rearrangement problems is more appropriate and complete in the sense that it extends the classical results and allow us to model not only rearrangement events but also *recombination events*.

1 Introduction

The advancements in the macromolecule sequencing in the last few decades allowed us to find complete genome sequences of several organisms [1, 9, 22]. The successful research on these sequences aroused the search for new computational techniques which were able to analyze that great amount of data. A better comprehension of that data may shed light on new evolutionary hypothesis, the speciation process and particular similarities and distinctions among genomes [20].

The genome of a species is composed by its set of macromolecules which are responsible for the encoding of the information necessary to build each protein used in the organism metabolism. These macromolecules are nucleic acids. The majority of the live beings encode the protein information in a few deoxyribonucleic acid (DNA) molecules called *chromosomes*. The DNA is a chain composed by two *strands* of smaller molecules. Each strand is a sequence of basic units that are composed by a sugar (deoxyribose), phosphate,

*Institute of Computing, University of Campinas, 13084-971 Campinas, SP. Research supported by FAPESP, grant #03/00731-3

†Scylla Bioinformática

and a *base*: adenine (A), guanine (G), cytosine (C), and thymine (T). The structure of these basic units induces an *orientation* on the DNA chain. The two strands of the DNA, which obey a double helix structure, are linked by the pairing between the bases: A – T and C – G. A *gene* is a contiguous stretch in one of the strands of the DNA that encode information for building a protein. It is important to notice that strands are *reverse complementary*, that is genes that belong to same strand have the same orientation, while genes in contrary strands have opposite orientations — they are reverse to each other — and are composed by complementary bases.

One of the most important techniques for the discovery of new hypothesis on the evolution of species is the comparison of their genome sequences. A well established technique for the comparison of two or more genomes is *sequence alignment* that consists in trying to determinate the *edit distance* between two genomes [21]. In this model a genome is viewed as a string over the alphabet $\{A, C, G, T\}$ and it is supposed that there are only insertions, deletions, and substitutions of bases as mutational events. On the other hand, some works in Molecular Biology [18, 19] demonstrated that sequence alignment is not the most adequate technique for genome comparison problem since there are mutational events in nature that move large blocks of genes instead of *local* operations over the basis. These *global* mutational events are called *genome rearrangement* events. As a matter of fact, mutational events involving large blocks of genes are rare in several species [11], therefore the analysis of this kind of event would be more appropriate to the comparison of genomes which are distant in the evolution time line like man and mice for instance.

Genome rearrangements analysis focus on the relative positions of the same block of genes at two or more distinct genome sequences. Assuming a parsimonious scenario for evolution, we are interested in identifying a minimum sequence of rearrangement events that transforms one genome into another. That is the *Genome Rearrangement Problem*. The minimum number of rearrangement events that transforms one genome into another is called *genome rearrangement distance* and this parameter can be used to infer the evolutionary relationship between distinct genomes.

Some examples of rearrangement events are *signed reversals* and *transpositions*. A signed reversal inverts the order and orientation of a contiguous sequence of blocks of genes, while a transposition moves a sequence of blocks of genes into another place in the same chromosome. In Figure 1, we can observe the application of signed reversals and transpositions over a genome.

In the example above, we can observe that each number in the sequence represents a block of genes of the genome. Two blocks of genes in distinct genomes identified by the same number are identical; i.e. they have the same genes in the same ordering.

Traditionally, genomes are represented through *permutations* in genome rearrangement theory, since although duplication of genes are biologically relevant, in the literature most results have been obtained considering that each gene (or block) appears uniquely in each genome. A permutation is a bijection from a set E to itself. A permutation can be represented through a matrix in which the first row indicates the position of an element while the second row indicates the element itself. To each block of genes it is assigned a value which identifies it uniquely and a *unichromosomical genome* is represented as a permutation built over these values and the ordering of the blocks of genes in the genome.

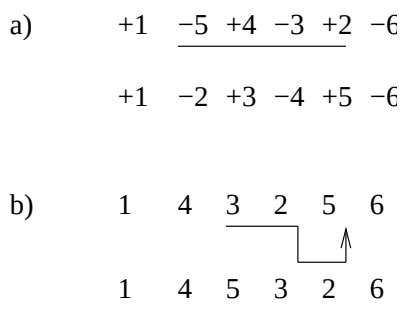

Figure 1: An example of a signed reversal and a transposition, where each number represents a block of genes. (a) A reversal inverts the order and changes the orientation of a sequence of blocks of genes. The block is underlined in the figure. (b) In a transposition, a sequence of blocks of genes is moved from its original position to the position immediately before a certain block, as indicated by the arrow.

The set $E(n) = \{1, 2, \ldots, n\}$ is often used to represent the positions of the block of genes. For the block labels, we define also

$$SE(n) = E(n) \cup -E(n) = \{-n,\ -(n-1),\ \ldots,\ -1,\ 1,\ \ldots,\ n\}$$

as the set of signed blocks of genes, where the sign conveys the orientation of a block. We omit the number of elements n, unless it is particularly necessary.

A *permutation* is simply a bijective mapping $\pi : E(n) \to E(n)$. A *signed permutation* is a mapping $\pi : E(n) \to SE(n)$ such that the function that maps $i \in E(n)$ to $|\pi(i)| \in E(n)$ is a permutation over E. We represent both kinds of permutations as:

$$\pi \leftrightarrow \begin{pmatrix} 1 & 2 & ... & n \\ \pi(1) & \pi(2) & \ldots & \pi(n) \end{pmatrix}.$$

This notation indicates an association between the position i and the element $\pi(i)$, for $1 \leq i \leq n$. Since the reordering of the elements is defined through a *positional index*, we sometimes simplify the notation and write a genome (which is a permutation, signed or not) as:

$$\pi = [\pi(1),\ \pi(2),\ \ldots,\ \pi(n)].$$

Every permutation has an *extended version*, which is a mapping denoted usually by the same letter π but over the extended domain

$$X(n) = \{0, 1, 2, \ldots, n, n+1\}$$

such that $\pi(0) = 0$ and $\pi(n+1) = n+1$ for both signed and unsigned permutations.

We define the *composition* of permutations $\pi\phi$ as a composition of functions in the usual sense, that is for each index i in E:

$$(\pi\phi)(i) = \pi(\phi(i)).$$

Observe that the composition $\pi\sigma$ is defined only when σ is an unsigned permutation. However, in the next section we will see a way of giving it a sound meaning for signed permutations.

1.1 Sorting by Signed Reversals

A *signed reversal* $\rho(i, j)$, where $1 \leq i \leq j \leq n$, is a rearrangement event that transforms the genome

$$\pi = [\pi(1),\ \pi(2),\ \ldots, \pi(i-1),\ \pi(i),\ \ldots,\ \pi(j-1),\ \pi(j),\ \pi(j+1)\ \ldots,\ \pi(n)]$$

into the genome

$$\pi\rho(i, j) = [\pi(1),\ \pi(2),\ \ldots, \pi(i-1),\ -\pi(j),\ \ldots,\ -\pi(i),\ \pi(j+1),\ \ldots,\ \pi(n)].$$

Example 1.1 *Let* $\pi = [+1,\ -5,\ +4,\ -3,\ +2]$. *The reversal* $\rho(3, 4)$ *inverts and changes the orientation of elements* $\pi(3)$ *and* $\pi(4)$*:*

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

Given two genomes π and σ, the *genome rearrangement by signed reversals problem* consists in finding a minimum sequence of signed reversals $\rho_1,\ \rho_2,\ \ldots,\ \rho_k$ such that $\pi\rho_1 \ldots \rho_{k-1}\rho_k = \sigma$. We call k the *signed reversal distance* from π to σ, denoted by $d_\rho(\pi, \sigma)$.

Given a genome π, the *sorting by signed reversals problem* consists in finding a minimum sequence of signed reversals $\rho_1,\ \rho_2,\ \ldots,\ \rho_k$ such that $\pi\rho_1 \ldots \rho_{k-1}\rho_k = \iota$, where ι is the *identity permutation* $\iota = [1,\ 2,\ \ldots,\ n]$. We call k the *signed reversal distance* of π, denoted by $d_\rho(\pi)$.

The genome rearrangement problem can always be viewed as a sorting problem. Given two genomes π and σ, finding a sequence of rearrangement events that transform π into σ is equivalent to sort $\sigma^{-1}\pi$, where σ^{-1} is the permutation whose domain is extended to the set $SE(n)$ by the rule $\sigma^{-1}(-x) = -\sigma^{-1}(x)$. Consider a sequence of reversals $\rho_1, \rho_2, \ldots, \rho_k$ such that $\pi\rho_1\rho_2 \ldots \rho_k = \sigma$. Composing by σ^{-1} to the left, we have:

$$\sigma^{-1}\pi\rho_1\rho_2 \ldots \rho_k = \sigma^{-1}\sigma = \iota$$

Example 1.2 *Let*

$$\pi = [3,\ -2,\ -1,\ -7,\ 5,\ -4,\ 6,\ -10,\ 9,\ 8]$$

and

$$\sigma = [1,\ 2,\ -3,\ -7,\ 4,\ -5,\ 6,\ -8,\ -9,\ 10]$$

be signed permutations.

One possible sequence of signed reversals that transforms π *into* σ *is:*

$$\begin{aligned}
\pi\rho(1,3) &= [1,\ 2,\ -3,\ -7,\ 5,\ -4,\ 6,\ -10,\ 9,\ 8] \\
\pi\rho(1,3)\rho(5,6) &= [1,\ 2,\ -3,\ -7,\ 4,\ -5,\ 6,\ -10,\ 9,\ 8] \\
\pi\rho(1,3)\rho(5,6)\rho(8,10) &= [1,\ 2,\ -3,\ -7,\ 4,\ -5,\ 6,\ -8,\ -9,\ 10]
\end{aligned}$$

We will show that the reversals that transform π into σ can also be used to sort $\sigma^{-1}\pi$. At first, we extend the domain of σ^{-1} to SE.

$$\sigma^{-1}(1) = 1,\ \sigma^{-1}(2) = 2,\ \sigma^{-1}(3) = -3,\ \sigma^{-1}(4) = 5,\ \sigma^{-1}(5) = -6,$$
$$\sigma^{-1}(6) = 7,\ \sigma^{-1}(7) = -4,\ \sigma^{-1}(8) = -8,\ \sigma^{-1}(9) = -9,\ \sigma^{-1}(10) = 10,$$
$$\sigma^{-1}(-1) = -1,\ \sigma^{-1}(-2) = -2,\ \sigma^{-1}(-3) = 3,\ \sigma^{-1}(-4) = -5,\ \sigma^{-1}(-5) = 6,$$
$$\sigma^{-1}(-6) = -7,\ \sigma^{-1}(-7) = 4,\ \sigma^{-1}(-8) = 8,\ \sigma^{-1}(-9) = 9,\ \sigma^{-1}(-10) = -10,$$

So, we have

$$\sigma^{-1}\pi = [-3,\ -2,\ -1,\ 4,\ -6,\ -5,\ 7,\ -10,\ -9,\ -8]$$

and applying the previous sequence of reversals we sort $\sigma^{-1}\pi$:

$$\begin{aligned}
\sigma^{-1}\pi\rho(1,3) &= [1,\ 2,\ 3,\ 4,\ -6,\ -5,\ 7,\ -10,\ -9,\ -8] \\
\sigma^{-1}\pi\rho(1,3)\rho(5,6) &= [1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7,\ -10,\ -9,\ -8] \\
\sigma^{-1}\pi\rho(1,3)\rho(5,6)\rho(8,10) &= [1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7,\ 8,\ 9,\ 10]
\end{aligned}$$

For a given genome π, a *sorting signed reversal*, or sorting reversal for short, is a signed reversal ρ such that $d_\rho(\pi\rho) = d_\rho(\pi) - 1$.

Let Σ_n be the set of genomes. We call the maximum value of $d_\rho(\pi)$ for any $\pi \in \Sigma_n$, the *signed reversal diameter*, and denote it by $d_\rho(n)$.

1.2 Sorting by Transpositions

The previous rearrangement event changes the order and orientation of contiguous blocks of genes. On the other hand, the rearrangement event treated in this section, the transposition, does not change the orientation of blocks of genes. Instead, it exchanges the positions of two contiguous sequences of blocks of genes. A *transposition* $\tau(i, j, k)$, where $1 \leq i < j < k \leq n + 1$, is a rearrangement event that transforms the genome

$$\pi = [\pi(1),\ \ldots,\ \pi(n)],$$

into the genome

$$[\pi(1),\ \ldots,\ \pi(i-1),\ \pi(j),\ \ldots,\ \pi(k-1),\ \pi(i),\ \ldots,\ \pi(j-1),\ \pi(k),\ \ldots,\ \pi(n)].$$

Observe that transpositions can be viewed as permutations as follows.

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

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

Given two unsigned genomes π and σ, the *genome rearrangement by transpositions problem* consists in finding a minimum sequence of transpositions $\tau_1,\ \tau_2,\ \ldots,\ \tau_k$ such that $\pi\tau_1 \ldots \tau_{k-1}\tau_k = \sigma$. We call k the *transposition distance* from π to σ, denoted by $d_\tau(\pi, \sigma)$.

The *sorting by transpositions problem* consists in finding a minimum sequence of transpositions $\tau_1,\ \tau_2,\ \ldots,\ \tau_k$ that transforms a genome π into ι. The number k is simply the transposition distance $d_\tau(\pi)$ from π.

Since transpositions do not change orientations, we represent genomes using unsigned permutations only. Just like with signed reversals, the genome rearrangement by transpositions problem is equivalent to the sorting by transpositions problem. Given genomes π and σ, every series of transpositions that transforms π into σ also transforms $\sigma^{-1}\pi$ into ι.

Example 1.3 *Consider the signed permutations* $\pi = [5,\ 3,\ 1,\ 6,\ 7,\ 2,.\ 8,\ 4]$ *and* $\sigma = [3,\ 6,\ 1,\ 5,\ 7,\ 8,\ 2,\ 4]$. *We can transform* π *into* σ *through the sequence of transpositions:*

$$\begin{aligned} \pi\tau(3,4,5) &= [5,\ 3,\ 6,\ 1,\ 7,\ 2,\ 8,\ 4] \\ \pi\tau(3,4,5)\tau(1,2,5) &= [3,\ 6,\ 1,\ 5,\ 7,\ 2,\ 8,\ 4] \\ \pi\tau(3,4,5)\tau(1,2,5)\tau(6,7,8) &= [3,\ 6,\ 1,\ 5,\ 7,\ 8,\ 2,\ 4] \end{aligned}$$

The same sequence of reversals can be used to sort $\sigma^{-1}\pi$. *Firstly, we present the inverse permutation of* σ.

$$\sigma^{-1}(1) = 3,\ \sigma^{-1}(2) = 7,\ \sigma^{-1}(3) = 1,\ \sigma^{-1}(4) = 8$$

$$\sigma^{-1}(5) = 4,\ \sigma^{-1}(6) = 2,\ \sigma^{-1}(7) = 5,\ \sigma^{-1}(8) = 6$$

$$\sigma^{-1}(-1) = -3,\ \sigma^{-1}(-2) = -7,\ \sigma^{-1}(-3) = -1,\ \sigma^{-1}(-4) = -8$$

$$\sigma^{-1}(-5) = -4,\ \sigma^{-1}(-6) = -2,\ \sigma^{-1}(-7) = -5,\ \sigma^{-1}(-8) = -6$$

Therefore $\sigma^{-1}\pi = [4,\ 1,\ 2,\ 3,\ 5,\ 7,\ 6,\ 8]$.
Now we apply the sequence of transpositions above to $\sigma^{-1}\pi$.

$$\begin{aligned} \sigma^{-1}\pi\tau(3,4,5) &= [4,\ 1,\ 2,\ 3,\ 5,\ 7,\ 6,\ 8] \\ \sigma^{-1}\pi\tau(3,4,5)\tau(1,2,5) &= [1,\ 2,\ 3,\ 4,\ 5,\ 7,\ 6,\ 8] \\ \sigma^{-1}\pi\tau(3,4,5)\tau(1,2,5)\tau(6,7,8) &= [1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7,\ 8] \end{aligned}$$

For a given genome π, a *sorting transposition* is a transposition τ such that $d_\tau(\pi\tau) = d_\tau(\pi) - 1$.

Let S_n be the set of unsigned permutations now. We denote by $d_\tau(n)$ the *transposition diameter* of S_n, which is the maximum transposition distance $d_\tau(\pi)$ for all $\pi \in S_n$.

In evolution, different events may occur. It is therefore natural to look for sorting sequences involving more than one kind of rearrangement event. For instance, we could consider signed reversals and transpositions as possible events that separate two genomes. Given a genome π, the *sorting by signed reversals and transpositions problem* consists in finding a minimum sequence of events $\rho_1,\ \rho_2,\ \ldots,\ \rho_k$, each one being either a signed reversal or a transposition, such that $\pi\rho_1 \ldots \rho_{k-1}\rho_k = \iota$. We call k the *signed reversal and transposition distance* of π, denoted as $d_{\rho\tau}(\pi)$. The two-input rearrangement problem and the diameter can be defined as well for this set of events.

1.3 Breakpoints, Cycle Diagrams, and Overlap Graphs

Since genome rearrangement (two-genome) problems are always equivalent to the corresponding simpler sorting (one-genome) problems, the classical theory has been developed mainly with regard to sorting problems. The following concepts and results are present in several works [11, 12, 14, 6, 21] and are fundamental to understand the formula for the signed reversal distance or the bounds for the transposition distance.

Given a genome π, a pair of elements $(\pi(i-1), \pi(i))$, where $1 \leq i \leq n+1$, is an *adjacency* when $\pi(i) - \pi(i-1) = 1$, otherwise it is a *breakpoint*. We denote the number of breakpoints of a permutation π by $b(\pi)$.

Given a genome π, the *image* of π is the permutation π' built through the replacement of each element $\pi(i)$, where $1 \leq i \leq n$, by elements $2\pi(i) - 1$ and $2\pi(i)$, in this order, if $\pi(i) > 0$, or by $2\pi(i)$ and $2\pi(i) - 1$, in this order, otherwise. The element $\pi(0) = 0$ remains unchanged, and the element $\pi(n+1) = n+1$ is replaced by $2n+1$.

Example 1.4 *Given* $\pi = [-1,\ 4,\ 2,\ -5,\ -3]$*, we get:*

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

Some authors use $-\pi(i)$ and $+\pi(i)$ instead of $2\pi(i) - 1$ and $2\pi(i)$ [21]. Both representations are equivalent.

Given a signed reversal $\rho(i, j)$, defining $\rho^* = \rho(2i - 1, 2j)$ we get the property $(\pi\rho)' = \pi'\rho^*$. Therefore, every sorting of π can be viewed as a sorting of π'.

Example 1.5 *For* $\pi = [-1,\ 4,\ 2,\ -5,\ -3]$ *the reversal* $\rho(2, 4)$ *has the corresponding reversal* $\rho^*(3, 8)$*, which inverts the positions* 3 *until* 8*, including both* 3 *and* 8*:*

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

We define a certain diagram for a genome π called the *cycle diagram*. A cycle diagram, also called *breakpoint graph*, $B(\pi)$ is built by placing elements of π' as vertices, including 0 and $2n+1$, in their natural ordering (see Figure 2, top). These vertices are linked by two kinds of edges: *gray edges* and *black edges*. There is a gray edge (x, y) when $x = 2i$, $y = 2i+1$ for $0 \leq i \leq n$. A black edge (x, y) connects $x = \pi'(2i)$ to $y = \pi'(2i + 1)$ for $0 \leq i \leq n$. As an example, Figure 2 (top) shows the cycle diagram for genome $\pi = [-3,\ 2,\ -5,\ -4,\ 1]$.

The cycle diagram defined as before is slightly different from the original definition in Hannenhalli and Pevzner [11] because we allow multiple edges between two vertices. In addition, some authors consider an alternative, more convenient way to graphically represent the diagram. The cycle diagram is traditionally represented linearly, as in the top of Figure 2. However, it is sometimes more convenient to present the cycle diagram in a "circular" way, as depicted on the bottom of Figure 2. As Hannenhalli and Pevzner [11], and Kaplan *et al.* [14] have pointed out, circularising the image of a permutation allows more uniformity in dealing with a certain special case related to hurdles, which are defined in Section 1.4.

The cycle diagram has a fundamental property: since the degree of each vertex in the diagram is exactly 2, there is a unique decomposition into cycles in the diagram. The *length*

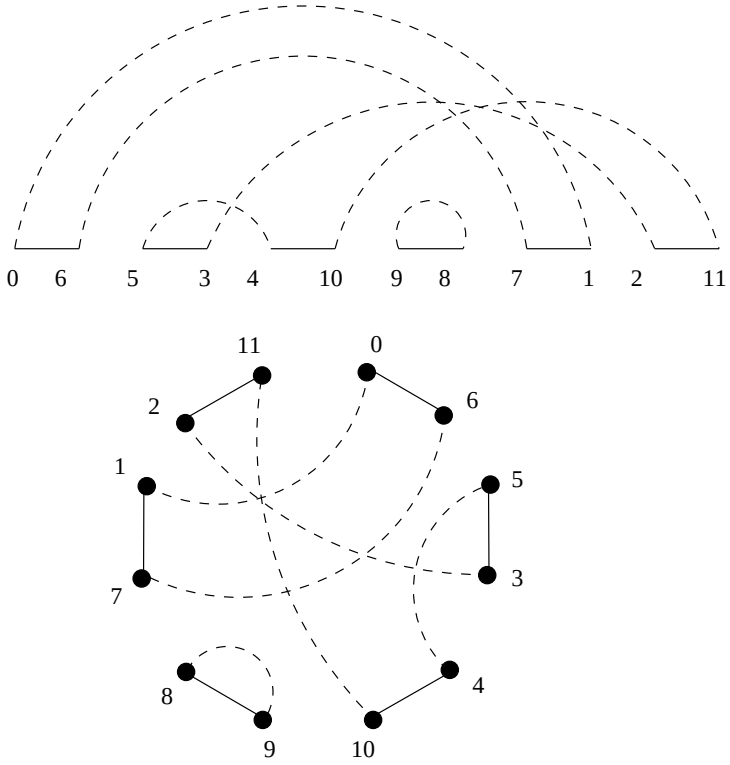

Figure 2: Two different drawings of the cycle diagram of genome $\pi = [-3,\ 2,\ -5,\ -4,\ 1]$. In the top, the diagram is drawn with black edges aligned over a straight line. In the bottom, the same diagram is depicted with black edges around a circle. In both cases, the ordering of vertices is given by π'.

of a cycle is the number of black edges in it. Let $c(\pi)$ denote the number of cycles in $B(\pi)$. As a consequence of allowing cycles of length 1, our $c(\pi)$ is larger by $n+1-b(\pi)$ than the value defined as $c(\pi)$ in most papers on reversal distance. Denoting by $c_{\geq 2}(\pi)$ the number of long cycles (cycles with length at least 2), we have

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

In order to analyze the complex structure of a cycle diagram and the consequences of a reversal applied to a permutation, Hannenhalli and Pevzner construct an equivalent, simpler permutation, through *padding operations* whose cycle diagram is composed solely by cycles of length two. Such an equivalent permutation conveys the same information as the original permutation and simplifies the analysis on the influence of a reversal on the number of cycles and hurdles. Later, Kaplan *et al.* [14] discovered that focusing on *arcs*, i.e. gray edges , instead of *cycles*, one can abandon the padding operations, working instead directly with the original permutation. Here we do not discuss the effects of padding operations on permutations.

An *odd cycle* is a cycle with odd length. Let $g = (\pi'(i), \pi'(j))$ with $\pi'(i)$ even, be a gray edge in the cycle diagram $B(\pi)$. We define the *interval induced by* g as the set of nonnegative integers $[i, j] = \{k \mid i \leq k \leq j\}$. Gray edges $(\pi'(i), \pi'(k))$ and $(\pi'(a), \pi'(b))$ are *interleaving* if the intervals $[i, j]$ and $[a, b]$ are such that $[i, j] \cap [a, b] \neq \emptyset$, $[i, j] \not\subseteq [a, b]$, and $[a, b] \not\subseteq [i, j]$. Two cycles C_1 and C_2 are *interleaving* when there are interleaving gray edges $g_1 \in C_1$ and $g_2 \in C_2$ in the cycle diagram.

Let $\Delta b(\rho, \pi)$ denote $b(\pi\rho) - b(\pi)$. Since a reversal inverts a contiguous segment of the permutation, it can create or eliminate at most two breakpoints in π. In fact, we have $-2 \leq \Delta b(\rho, \pi) \leq 2$. An immediate lower bound for the signed reversal distance is:

$$\frac{b(\pi)}{2} \leq d_\rho(\pi)$$

Let $\Delta c(\rho, \pi)$ denote $c(\pi\rho) - c(\pi)$. A reversal is *proper* when $\Delta c(\rho, \pi) = 1$. Each pair of black edges $(\pi'(2i), \pi'(2i+1))$ and $(\pi'(2j), \pi'(2j+1))$ in the cycle diagram, where $0 \leq i < j \leq n$, defines a reversal $\rho(i, j)$ to be applied to π. A reversal ρ *acts on* a gray edge g when it is defined by the black edges incident to the gray edge g. A gray edge is *oriented* when the reversal acting on it is proper, otherwise it is *unoriented*. A cycle is *oriented* when it has an oriented gray edge among its edges, otherwise it is *unoriented*. If there is a sequence of proper reversals that sorts π, then $d_\rho(\pi) = n+1-c(\pi)$. A result from Bafna and Pevzner [4] implies that we can increase $\Delta c(\rho, \pi)$ by at most 1 when applying a reversal. Therefore we have the following alternative lower bound:

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

since the identity permutation has $n+1$ cycles. Furthermore, the identity permutation is the only one that has $n+1$ cycles, and for this reason we are tempted to find reversals that increase the number of cycles. Unfortunately, reversals that increase the number of cycles (proper reversals) are not always sorting reversals.

Example 1.6 *Consider the signed permutation* $\pi = [4,\ 3,\ 1,\ -5,\ -2]$*, which has* $c(\pi) = 3$*. There are just two proper reversals that we can apply to* π*:* $\rho_1 = \rho(4, 5)$ *and* $\rho_2 = \rho(2, 4)$*. In the first case, the resulting permutation* $\pi\rho_1$ *has no proper reversal and therefore its distance is at least 3. On the other hand, permutation* $\pi\rho_2$ *not only has a proper reversal but also admits a sequence of proper reversals that sorts it, witnessing that* $d_\rho(\pi) = 3$*. We conclude that* ρ_2 *is a sorting reversal while* ρ_1 *is not.*

Kaplan, Shamir, and Tarjan [14] define a graph based on the interleaving relation among gray edges of the cycle diagram. Given a signed permutation π, the *overlap graph* of π, denoted by $OV(\pi)$, is the graph whose vertex set is the set of gray edges of $B(\pi)$ and two vertices are linked by an edge in $OV(\pi)$ if their corresponding gray edges in $B(\pi)$ are interleaving. A vertex in $OV(\pi)$ is *oriented* if its corresponding gray edge in $B(\pi)$ is oriented, otherwise the vertex is *unoriented*. A component in $OV(\pi)$ is *oriented* if it contains an oriented vertex, otherwise it is an *unoriented component*. Figure 3 illustrates an overlap graph.

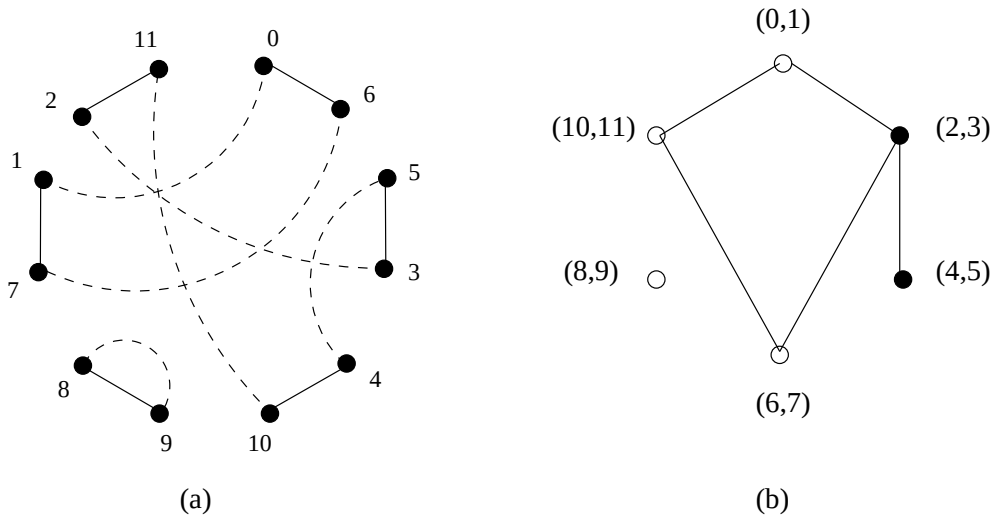

Figure 3: a) The cycle diagram of genome $\pi = [-3,\ 2,\ -5,\ -4,\ 1]$. b) The corresponding overlap graph $OV(\pi)$ of π. For each vertex in the overlap graph there is a corresponding gray edge in the cycle diagram. Oriented vertices are black while the unoriented ones are white.

A new, considerable simplification to the the Sorting by Signed Reversals theory was achieved by Anne Bergeron [6]. In her work, she focused on the analysis of the signed permutation itself rather than in the other, derived structures such as the cycle diagram and the overlap graph. Bergeron identified some of the previous concepts and combinatorial structures in the permutation itself, such as some of its proper reversals and its hurdles (see section 1.4).

We will briefly describe the main points of her new view of the theory. Let π be a signed permutation in its extended version. An *oriented pair* $(\pi(i),\ \pi(j))$ of π is a pair of integers with opposite signs that would be consecutive if taken in absolute value, i.e.,

$||\pi(i)| - |\pi(j)|| = 1$. For instance, in the permutation

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

the oriented pairs are $(-2,\ 1)$, $(-2,\ 3)$, $(-5,\ 4)$, and $(-5,\ 6)$. For each oriented pair of a permutation there is an unique corresponding oriented gray edge. Therefore, an oriented pair induces a proper reversal that is the reversal acting on the corresponding gray edge. The reversal induced by the oriented pair $(\pi(i),\ \pi(j))$, which Bergeron calls *oriented reversal*, is

$$\rho(i, j-1),\ \text{if } \pi(i) + \pi(j) = +1$$

or

$$\rho(i+1, j),\ \text{if } \pi(i) + \pi(j) = -1.$$

Observing the effect of the induced reversal ρ when applied to π, one can verify that at least one adjacency is created in $\pi\rho$; in other terms, the number of cycles with respect to $B(\pi)$ increases by at least one, since a new cycle of length one appears in $B(\pi\rho)$. For example, consider the following permutation

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

The reversal induced by oriented pair $(-2,\ 1)$ is $\rho(2, 5)$. We have:

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

The permutation π above has 6 breakpoints and 4 oriented pairs, while $\pi\rho(2, 5)$ has 5 breakpoints and 2 oriented pairs. Every oriented reversal is proper, although there are proper reversals that are not oriented in the sense of Bergeron. We have discussed earlier that it is tempting to use proper reversals, even though they might not be sorting reversals. Indeed, the crucial question is: which oriented reversals are also sorting reversals? Bergeron proposes a new parameter associated with a proper reversal that partially settles this question. Given a permutation π, the *score* of an oriented reversal ρ is the number of oriented pairs in $\pi\rho$. Bergeron asserts that choosing an oriented reversal with maximum score in each step of the sorting process is an optimal strategy. This fact is explicitly stated as:

Proposition 1.1 (Bergeron [6]) *If the successive application of a sequence of k oriented reversals with maximum score to the permutation π yields a permutation θ with no negative elements, then $d(\pi) = d(\theta) + k$.*

Proposition 1.1 assures that oriented reversals with maximum score are sorting reversals. Applying reversals of this kind to a permutation π will eventually lead to a permutation that does not contain any oriented pair, so we cannot apply this strategy further. If the resulting permutation is not the identity permutation, we need some other way to proceed. In the next section we discuss the internal combinatorial structures of *positive permutations* and find sorting reversals that can be applied to them.

1.4 Interleaving Graph and Hurdles

Hannenhalli and Pevzner [12] define a graph over the cycle diagram which shows the underlying relationship among the cycles in that diagram. Apart from cycles of length one, we follow their terminology in this section. Let $\mathcal{C}(\pi)$ be the set of cycles in the cycle diagram $B(\pi)$. The *interleaving graph* $H(\pi)$ is composed by vertex set $\mathcal{C}(\pi)$ and there is an edge between two cycles C_1 and C_2 in $\mathcal{C}(\pi)$ when C_1 and C_2 are interleaving. Moreover the vertices of $H(\pi)$ are labeled as *oriented* or *unoriented* based on their corresponding cycles. If a component of $H(\pi)$ is composed solely by unoriented cycles, then it is called an *unoriented component*, otherwise it is an *oriented component*. In our terminology, based on Bergeron [6], each cycle of length one in the cycle diagram forms a singleton, unoriented component in the interleaving graph.

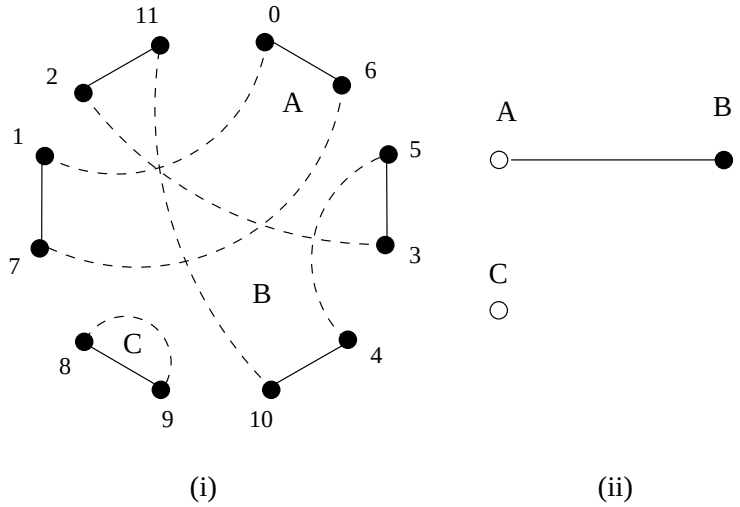

Figure 4: The cycle diagram (i) of genome $\pi = [-3,\ 2,\ -5,\ -4,\ +1]$ and its corresponding interleaving graph (ii). Vertices in the interleaving graph corresponding to oriented cycles in the cycle diagram are depicted in black. The vertex B in the interleaving graph is black since its corresponding cycle in the cycle diagram is oriented.

Kaplan, Shamir, and Tarjan [14] have found an important relationship between the interleaving graph and the overlap graph: both graphs have the same number of components when derived from the same permutation.

Given a connected component U in $H(\pi)$, define the *leftmost position* of U as

$$U_{min} = \min_{C \in U} \min_{\pi'(i) \in C} i,$$

and the *rightmost position* as

$$U_{max} = \max_{C \in U} \max_{\pi'(i) \in C} i.$$

Denote by $Extent(U)$ the interval $[U_{min}, U_{max}]$.

Example 1.7 *In Figure 4, the interleaving graph of* $\pi = [-3,\ 2,\ -5,\ -4,\ +1]$ *has two components: one composed by cycles A and B (call it U), and one composed solely by the cycle C (call it U′). We have* $U_{min} = 0$, $U_{max} = 11$, $U'_{min} = 6$, *and* $U'_{max} = 7$. *Moreover, the extensions of the two components are:* $Extension(U) = [0, 11]$ *and* $Extension(U') = [6, 7]$.

To define hurdles, simple hurdles, super hurdles, and fortresses we need the concept of *component separation*. Two slightly different definitions for separation appear in the literature, one by Hannenhalli and Pevzner [12], and the other by Setubal and Meidanis [21]. We will use the latter here because it corresponds better with the intuitive notion of separation, as the following example shows.

Example 1.8 *Consider the genome* $\pi = [11,\ 10,\ 1,\ 8,\ 7,\ 2,\ 5,\ 4,\ 3,\ 6,\ 9]$. *In Figure 5, component B does not separate A and C according to Hannenhalli and Pevzner, however it is clearly located "between" them.*

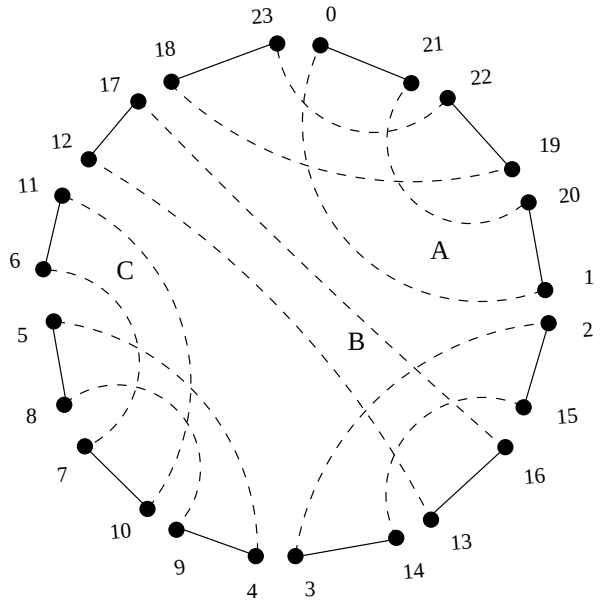

Figure 5: The cycle diagram of genome $\pi = [11,\ 10,\ 1,\ 8,\ 7,\ 2,\ 5,\ 4,\ 3,\ 6,\ 9]$. The diagram is composed by three unoriented components A, B, and C. According to Setubal and Meidanis [21], component B separates components A and C. However, B does not separate A and C according to Hannenhalli and Pevzner [12].

Hannenhalli and Pevzner [12] define component separation as follows. A component U in $H(\pi)$ separates components U' and U'' in π when there exists a gray edge $(\pi'(i), \pi'(j))$ in U such that $Extent(U') \subset [i, j]$ and $Extent(U'') \cap [i, j] = \emptyset$.

Let $\mathcal{U}(\pi)$ be the set of non-singleton, unoriented components in $H(\pi)$. Define the containment partial order on $\mathcal{U}(\pi)$ with the relation $U \prec W$ when $Extent(U) \subset Extent(W)$ for $U, W, \in \mathcal{U}(\pi)$. A component U in $\mathcal{U}(\pi)$ is minimal with respect to $\prec$ when there is no component $W \in \mathcal{U}(\pi)$ such that $W \prec U$. The greatest component U in $\mathcal{U}(\pi)$ with respect to

$\prec$ is the component such that $W \prec U$ for any $W \in \mathcal{U}(\pi)$. The minimal elements according to the partial order on $\mathcal{U}(\pi)$ are the minimal hurdles of π. The greatest element $U \in \mathcal{U}(\pi)$ in the partial order is the greatest hurdle if U does not separate any two minimal hurdles.

Several authors [12, 14, 6] have observed that by circularizing the permutation the differences between the greatest hurdle and minimal hurdles disappear. Therefore it is more convenient to deal with a circular cycle diagram, as the one depicted in Figure 6, and this is the setting in our official definition of component separation, due to Setubal and Meidanis [21].

Consider then a circular cycle diagram. A *chord* is a straight line connecting two arbitrary vertices in $B(\pi)$. Chords do not have to be gray nor black edges, actually chords may connect vertices that are not connected by any edge in $B(\pi)$. A component B *separates* two components A and C in $B(\pi)$ if all chords connecting a vertex in A and a vertex in C cross at least one gray edge of B. We say that B is a *separating component* for A and C. Given a black edge in A and another black edge in C, we may think of the (unique) reversal that breaks exactly on those two edges. Such a reversal may change the orientation of a separating component for A and C. If the separating component is unoriented, then it becomes oriented after the application of the reversal. However, if the separating component is oriented, it can remain oriented or become unoriented. A *hurdle* is an unoriented, nonsingleton component that does not separate any two other unoriented, nonsingleton components. All other unoriented components are called *nonhurdles*. Bergeron gives an alternative way to define hurdles, although valid only on a restricted class of permutations, namely, those with no singleton or oriented components [6].

Denote by $h(\pi)$ the number of hurdles of π. Similarly to breakpoints and cycles, we have $\Delta h(\rho, \pi) = h(\pi\rho) - h(\pi)$.

Hurdles are actually classified into simple hurdles and super hurdles, and this distinction is important during the sorting process of a permutation. Hannenhalli and Pevzner observed that a reversal ρ acting on an edge of an unoriented cycle of a hurdle U of a permutation π, when applied to π, transforms the unoriented component U into an oriented one with the same number of cycles, where each cycle maintains the same length as before. They called such an operation of eliminating a hurdle *hurdle cutting*, and say that the reversal ρ *cuts off* the hurdle. These definitions help distinguish simple hurdles from super hurdles.

A hurdle A *protects* a nonhurdle B if the removal of the hurdle A would turn B into a hurdle. By "removal" here we mean a hurdle cutting operation as described previously. A hurdle is called a *super hurdle* if it protects some nonhurdle, otherwise it is a *simple hurdle*. Figure 6 illustrates those concepts.

Given a genome π, according to Hannenhalli and Pevzner [12], a reversal ρ is a *safe reversal* if $\Delta b(\rho, \pi) - \Delta c_{\geq 2}(\rho, \pi) + \Delta h(\rho, \pi) = -1$, which corresponds to $\Delta b(\rho, \pi) - \Delta c(\rho, \pi) = -1$ in our notation . Kaplan *et al.* [14], as well as Bergeron [6], define the safe reversal in a more restrictive way, since they require that the reversal should also be proper to be a safe reversal. Recall that proper reversals are defined as the one satisfying $\Delta c(\rho, \pi) = 1$. Safe reversals are important when sorting oriented components, in which case both definitions agree and are equivalent to the concept of sorting reversal defined earlier (page 5).

A genome π is called a *fortress* if a permutation has an odd number of hurdles and all of them are super hurdles (see Figure 8). Let $f(\pi)$ be 1 when π is a fortress, and 0 otherwise.

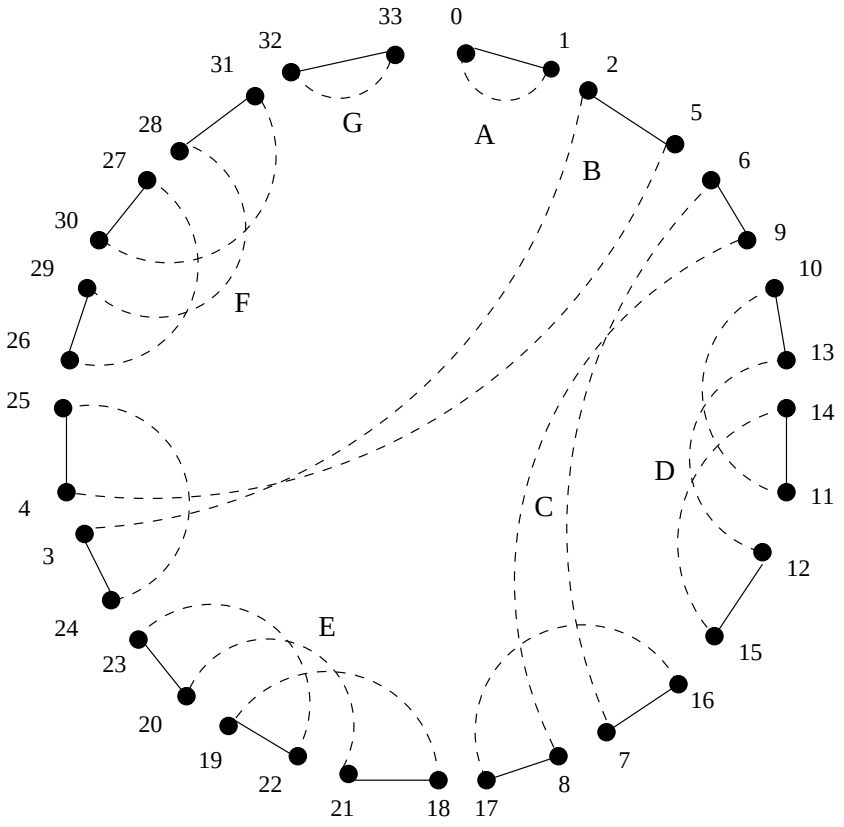

Figure 6: The cycle diagram of genome $\pi = [1, 3, 5, 7, 6, 8, 4, 9, 11, 10, 12, 2, 13, 15, 14, 16]$. Hurdles and nonhurdles, as well as simple and super hurdles are illustrated in the diagram. To begin with, note that the diagram contains no oriented components. Unoriented component B separates unoriented components E and F, and is therefore a nonhurdle. Likewise, component C separates, for instance, D and E, and is also a nonhurdle. On the other hand, components D, E, and F are hurdles, since they do not separate other unoriented, nonsigleton components. Component D protects component C, so D is a super hurdle. Since F protects B, it is also a super hurdle. On the other hand, E does not protect any component, therefore it is a simple hurdle. Components A and G are nonhurdles because they are singleton components.

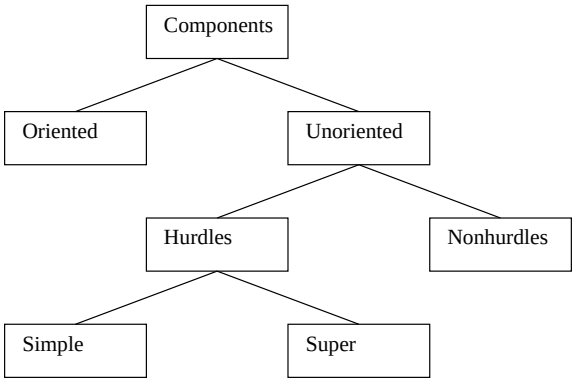

Figure 7: Classification of components in a cycle diagram.

We have now all the ingredients needed to solve the sorting by reversals problem polynomially. The next proposition contains the celebrated formula to find the signed reversal distance involving the parameters discussed earlier: the number of cycles in $B(\pi)$, the number of hurdles in the cycle diagram $B(\pi)$, and the fortress indicator $f(\pi)$.

Proposition 1.2 (Hannenhalli and Pevzner[11]) *Given a genome π, then:*

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

This formula also allows one to calculate the signed reversal diameter. We have:

Proposition 1.3 *The value of the diameter $d_\rho(n)$ for genomes in Σ_n satisfies:*

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

The following lower bound for the transposition diameter was found by Christie [8], and Meidanis, Walter and Dias [17], while the upper bound was found by Eriksson *et al.* [10]

Proposition 1.4 (Transposition Diameter) *For S_n and $n \geq 9$, we have:*

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

1.5 Cycle Diagrams and Transpositions

Up to now, no one knows how to sort by transpositions in polynomial time. Bafna and Pevzner [5] presented the first polynomial time *approximation* algorithm for the sorting by transpositions problem. In this section, we present their approach to the problem, leading to an 1.5-approximation algorithm that provides the best performance guarantee known so far.

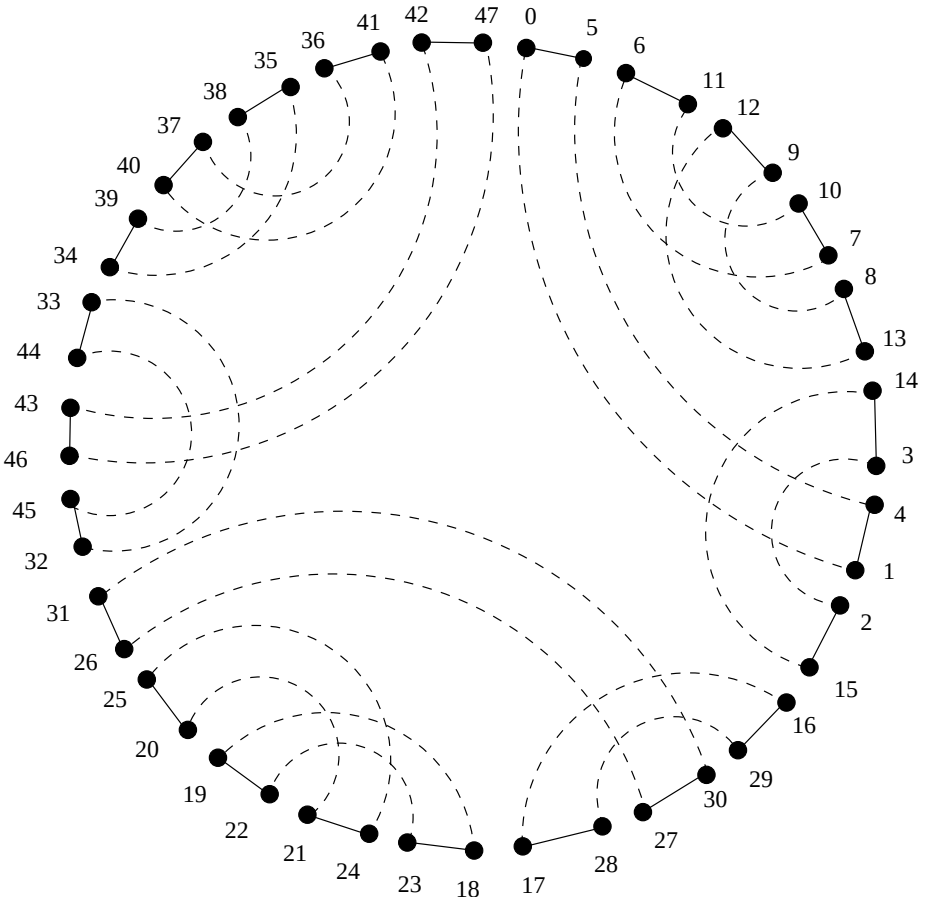

Figure 8: Genome π is a fortress: all three hurdles in the diagram are super hurdles.

In the sorting by transpositions problem, Bafna and Pevzner [5] and Hartman and Shamir [13] present different versions of the cycle diagram for a permutation π. Bafna and Pevzner define one vertex for each element in π, resulting in vertices of degree 4. Hartman and Shamir define their diagrams much like those we saw in previous sections for the reversal operation. The only difference is that Hartman and Shamir consider circular permutations, pointing out that there is a straightforward correspondence between the linear and circular versions of the problem. As we saw earlier, it is also more convenient to deal with circular permutations with respect to reversals.

We shall therefore follow the same line developed for reversals here. As before, a genome is a signed permutation. Since transpositions never change signs, we shall consider that all elements are nonnegative in the sorting by transpositions problem. Therefore, for a genome π its image π' is such that each element $\pi(i)$, for $1 \leq i \leq n$ is replaced by $2\pi(i) - 1$ and $2\pi(i)$ in this order. Moreover, we have $\pi'(0) = 0$ and $\pi'(2n + 1) = 2n + 1$. A transposition $\tau(k, l, m)$ on π' is called *legal* when k, l, m are non-negative, odd integers. Note that this is analogous to a notion implicit in the work of Hannehalli and Pevzner, who forbid cuts in π' at positions that do not exist in π. For any transposition τ on π, there is a unique corresponding legal transposition τ' such that $(\pi\tau)' = \pi'\tau'$. Since the diagram is defined in the same way as for reversals, we maintain the same notation of the number of cycles $c(\pi)$ (including singleton cycles). A cycle is a *k-cycle* when its length is k. A k-cycle is a *long cycle* when $k > 2$, otherwise it is a *short cycle*.

Given a permutation π and a transposition τ, let $\Delta c(\pi\tau, \pi) = c(\pi\tau) - c(\pi)$ be the increase in the number of cycles when τ is applied to π. Bafna and Pevzner [5] prove that $\Delta c(\pi\tau, \pi) \in \{-2, 0, 2\}$. From this observation, they deduce the following lower bound for the transposition distance:

Proposition 1.5 *For any permutation π on n blocks of genes we have*

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

Bafna and Pevzner [5] further improve the lower bound given in Proposition 1.5 taking into account the parity of cycles. A cycle is *odd* if its length is odd, otherwise it is *even*. Given a permutation π, let $c_{odd}(\pi)$ and $c_{even}(\pi)$ be respectively the number of odd and even cycles in $B(\pi)$. The following sharper bound holds:

Proposition 1.6 *For any permutation π on n blocks of genes we have*

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

Given a signed permutation π, a transposition τ is an *x-move* when $\Delta c(\pi\tau, \pi) = x$ where $x \in \{-2, 0, 2\}$. Bafna and Pevzner prove the following result.

Proposition 1.7 *Given an unsorted permutation π, there exists a 2-move or a 0-move followed by a 2-move in π.*

Given a permutation π, in the worst case, the number of cycles in $B(\pi\tau)$ would increase by one cycle only compared to $B(\pi)$. Therefore an upper bound for the transposition distance follows straightforwardly from Proposition 1.7:

Proposition 1.8 *Any permutation π can be sorted with $n + 1 - c(\pi)$ transpositions.*

Proposition 1.8 yields an approximation algorithm for the sorting by transpositions problem whose output is not higher than twice the optimum. The sorting algorithm is based on a greedy strategy of finding 2-moves or $0 - 2$-move sequences. Notice that a 2-move is not necessarily a sorting transposition. Bafna and Pevzner [5] showed that a significant improvement in an approximation algorithm can be achieved if one tries to find transpositions that increase the number of odd cycles in the cycle diagram.

Given a permutation π and a transposition τ, we call τ a *valid transposition* when $\Delta c(\pi\tau, \pi) = \Delta c_{odd}(\pi\tau, \pi)$. Investigating the role of some self-interleaving structures in the cycle diagram, Bafna and Pevzner [5] determined the conditions to find valid moves in a permutation.

Theorem 1.1 *If there is a long cycle in $B(\pi)$, then there is a valid 2-move in π or a valid 0-move followed by two consecutive valid 2-moves.*

Observe that Theorem 1.1 is valid only for permutations that have long cycles. A separate analysis is necessary for permutations that are composed solely by short cycles. Bafna and Pevzner [5] have found transpositions acting on two short cycles that increase the number of odd cycles although the number of cycles in the permutation remains the same. Given a permutation π, a 0-move τ is called *good* if it creates two odd cycles in $B(\pi\tau)$.

Theorem 1.2 *If $B(\pi)$ has only short cycles, then there exists a good 0-move followed by a valid 2-move in π.*

Combining the previous theorems (Theorem 1.1 and Theorem 1.2) Bafna and Pevzner [5] have designed the following approximation algorithm *TransSort*:

```
Algorithm TransSort
While π is not sorted:
        If B(π) has a long cycle,
            then apply a valid 2-move or a valid 0 − 2 − 2-move sequence to
π.
            else //B(π) has only short cycles
            apply a good 0-move followed by a valid 2-move to π.
```

Theorem 1.3 *Algorithm* TransSort *sorts π in $O(n^2)$ time by using no more than $\frac{3}{4}(n + 1 - c_{odd}(\pi))$ transpositions, thereby achieving an 1.5 approximation factor.*

Algorithm *TransSort* sorts any permutation with at most $\frac{3}{4}(n + 1 - c_{odd}(\pi))$ transpositions. So, a sharper transposition diameter upper bound follows.

Corolary 1.1 (Transposition Diameter) *The transposition diameter of S_n is at most $\frac{3n}{4}$.*

1.6 Additional Results

The sorting by signed reversals problem is the best understood genome rearrangement problem until now. The first exact polynomial time algorithm was proposed by Hannenhalli and Pevzner [11]. Several improvements and new concepts have been suggested and added to those authors' theory [14, 6], culminating with a subquadratic algorithm by Tannier and Sagot [23]. Bader, Moret and Yan [2] have offered an algorithm that calculates the signed reversal distance in linear time.

Research on the transposition distance problem has resulted in several approximation algorithms [3, 8]. The best known approximation algorithm for the transposition distance has $O(n^2)$ complexity and guarantees a 1.5 approximation factor [5]. Despite that, this algorithm is considered very complicated and for this reason Christie [8] proposed an alternative, simpler algorithm with the same approximation factor, but with $O(n^4)$ time complexity. Eriksson *et al.* [10] gave an algorithm that requires at most $\lfloor (2n-2)/3 \rfloor$ transpositions to sort any permutation. Recently, Hartman and Shamir [13] proposed a new $O(n^{3/2}\sqrt{\log n})$ 1.5-approximation algorithm for the sorting by transpositions problem using new data structures developed by Kaplan and Verbin [15]. For a specific input instance (permutations of the form $[n, n-1, \ldots, 2, 1]$) there are exact polynomial time algorithms that solve the sorting by transpositions problem [10, 17, 8]. We do not know of any polynomial time algorithm that solves sorting by transpositions nor a proof that the problem is NP-hard.

Christie [7] proposed and solved the block interchange distance problem in genomes — a generalization of the transposition distance problem.

2 Discussion

In this section we discuss some characteristics of the classical theory for the Rearrangement Problem that we particularly see as limiting and inadequate.

The main disadvantages of classical theory are listed below:

1. When modeling a genome as a signed permutation in the classical theory, blocks of genes must be labeled with integers because of the positional framework of this representation.

2. Positional indexing does not fit circular genomes properly because it is necessary to choose a block of genes as a starting point.

3. There are subtle differences between cycle diagrams depending on the kind of genome (linear or circular) and which rearrangement event will be treated (reversals or trans-

positions). These differences can be a source of misunderstandings and do not express the real, essential differences among rearrangement problems.

4. The classical theory makes frequent use of graphical arguments in proofs, which may lead to confusion. An analogy can be drawn with the Euclidean Geometry, where arguments are mainly graphical, but has been enriched by Analytic Geometry.

5. The classical theory requires that the target permutation into which the other permutation will be sorted must be the identity. Such requirement is cumbersome for rearrangement problems involving more than two genomes because it is necessary to perform a relabeling for each pair of genomes of the problem.

Such disadvantages motivated the authors to look for a more suitable model for genomes and rearrangement events. Meidanis and Dias [16] present an algebraic formalism in which the role of permutations as a model for genomes is more deeply explored than in the classical theory. Some advantages of algebraic theory are:

1. The original labels of blocks of genes can be used as elements of the permutation representing a genome.

2. In the algebraic theory no indexing is necessary: each block of genes is mapped onto the next block in the genome. This simplifies the specification of rearrangement operations.

3. Permutations as product of cycles represents circular genomes more suitably than permutations in their matrix representation.

4. Well known results from Permutation Group Theory can be used in arguments of the algebraic theory instead of graphical arguments.

5. The majority of the fundamental concepts in the theory such as genomes, rearrangement events, the cycle diagram, cycles and breakpoints can be modeled by permutations. Such modeling expresses all the main concepts in a uniform way instead of using several structures as sets, pairs, graphs, strings, and etc.

6. The norm of a rearrangement event can be used as a "weight" in rearrangement problems involving more than one kind of event. Such "weight" would measure, for instance, the probability of occurrence of a certain event.

7. Since each block of genes is represented by its own "name" in the algebraic theory, dealing with several genomes does not involve any relabeling.

References

[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. Mobarry, 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. Sidn-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, Mar. 2000.

[2] 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.

[3] 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.

[4] V. Bafna and P. A. Pevzner. Genome rearrangements and sorting by reversals. *SIAM Journal on Computing*, 25(2):272–289, 1996.

[5] V. Bafna and P. A. Pevzner. Sorting by transpositions. *SIAM Journal on Discrete Mathematics*, 11(2):224–240, May 1998.

[6] A. Bergeron. A very elementary presentation of the hannenhalli-pevzner theory. *Discrete Applied Mathematics*, 146(2):134–135, 2005.

[7] D. A. Christie. Sorting permutations by block-interchanges. *Information Processing Letters*, 60(4):165–169, November 1996.

[8] D. A. Christie. *Genome Rearrangement Problems*. PhD thesis, Glasgow University, 1998.

[9] T. G. I. S. Consortium. Initial sequencing and analysis of the human genome. *Nature*, 409:860–921, 2001.

[10] H. Eriksson, K. Eriksson, J. Karlander, L. Svensson, and J. Wastlund. Sorting a bridge hand. *Discrete Mathematics*, 241:289–300, 2001.

[11] 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, Oct. 1995. IEEE Computer Society Press.

[12] 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, Jan. 1999.

[13] T. Hartman and R. Shamir. A simpler and faster 1.5-approximation algorithm for sorting by transpositions. In *Proceedings of CPM'03*, pages 156 – 169, 2003. extended version.

[14] 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, Jan. 2000.

[15] 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.

[16] J. Meidanis and Z. Dias. An alternative algebraic formalism for genome rearrangements. In D. Sankoff and J. H. Nadeau, editors, *Comparative Genomics: Empirical and Analyitical Approaches to Gene Order Dynamics, Map Alignment and Evolution of Gene Families*, pages 213–223. Kluwer Academic Publishers, Nov. 2000.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] J. C. Setubal and J. Meidanis. *Introduction to Computional Molecular Biology*. PWS Publishing Company, 1997.

[22] A. J. G. Simpson, F. 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. D. Oliveira, R. C. D. 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. D. M. Rosa, V. E. D. Rosa, R. G. D. S, R. V. Santelli, H. E. Sawasaki, A. C. R. D. Silva, A. M. D. Silva, F. R. D. Silva, W. A. Silva, J. F. D. Silveira, M. L. Z. Silvestri, W. J. Siqueira, A. A. D. Souza, A. P. D. Souza, M. F. Terenzi, D. Truffi, S. M. Tsai, M. H. Tsuhako, H. Vallada, M. A. V. 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.

[23] E. Tannier and M.-F. Sagot. Sorting by reversals in subquadratic time. Technical Report 5097, INRIA, Institut National de Recherche en Informatique et en Automatique, January 2004.