uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Wei Xu

AUTEUR DE LA THÈSE / AUTHOR OF THESIS

Ph.D. (Mathematics)

GRADE / DEGREE

Department of Mathematics and Statistics

FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Mathematical Problems in Comparative Genomics

TITRE DE LA THÈSE / TITLE OF THESIS

David Sankoff

DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

Sylvia Boyd

Bin Ma

Zhicheng Cao

Daniel Panario

Gary W. Slater

Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

MATHEMATICAL PROBLEMS IN COMPARATIVE GENOMICS

Wei Xu

Thesis Submitted to the Faculty of Graduate and Postdoctoral Studies

In partial fulfillment of the requirements

for the degree of

Doctor of Philosophy in Mathematics[1]

Department of Mathematics and Statistics

Faculty of Science

University of Ottawa

[1]The Ph.D. Program is a joint program with Carleton University, administered by the Ottawa-Carleton Institute of Mathematics and Statistics

Your file Votre référence
ISBN: 978-0-494-48666-5
Our file Notre référence
ISBN: 978-0-494-48666-5

Abstract

In this thesis I look at several fundamental mathematical problems in the area of *comparative genomics*. To understand the probabilistic behaviors of genomic distances and to devise statistical tests to see whether there is significant evolutionary signal remaining in the gene orders, I derive the probability distributions for DCJ distance, reversal distance and breakpoint distance. To utilize these validated evolutionary signals in recovering the phylogeny of species, I develop a graph decomposition theory to effectively reduce the size of the *median problem*, which lies at the heart of the rearrangement based phylogeny problem and has been proven NP-hard. My decomposition theory enables recursive reductions of the size of the problem by discovering *adequate subgraphs* in the *multiple breakpoint graphs* which are the graphic representation of the median problems. The results on simulated data with varying parameters show the power of the theory and the effectiveness of the corresponding algorithm—*ASMedian*. With various possible improvements, this theory should lead to practical methods applicable to most biology instances.

Acknowledgements

I would like to thank my mentor David Sankoff. It is his knowledge and wisdom that have guided me to the frontier field of scientific research, avoiding all sorts of trivialities and distractions. And it is his encouragement and tolerance, that have helped me gain confidence in myself and faith in scientific research.

I would like to thank my granduncle in Beijing. Fourteen years ago, it was he who, for the first time in my life, showed me the power of knowledge and rationality, and the pleasure they can deliver.

Also I would like to thank my wife and her entire family for the love they give to me and to thank my caring friends for their invaluable help and suggestions.

Dedication

Dedicated to my wife Ziyi Zhang, for her wholehearted trust and endless support from the past to the long future.

Contents

Abstract ii

Acknowledgements iii

Dedication iv

List of Tables vii

List of Figures viii

1 Introduction 1

1.1 Genomes and gene orders 1

1.2 Breakpoint graph 3

1.3 Distance measures 5

1.3.1 The breakpoint distance 5

1.3.2 The edit distance measures 6

1.3.3 DCJ distance 8

1.4 The distance distributions among random breakpoint graphs . . . 9

1.4.1 Random breakpoint graphs 10

1.4.2 Cap optimization 11

1.4.3 Combinatorics and generating functions 12

1.4.4 Plasmids in random genomes 15

1.5 The Poisson distribution of common adjacencies 15
1.6 The median problem 16
1.6.1 Graphic representation and decomposition 18
1.7 The median of three problem 23

2 Paths and Cycles in Breakpoint Graphs of Random Multichromosomal Genomes 27

3 The distance between randomly constructed genomes 53

4 The distribution of distances between randomly constructed genomes: generating function, expectation, variance and limits 69

5 Poisson Adjacency Distributions in Genome Comparison: Multichromosomal, Circular, Signed and Unsigned Cases 91

6 Decompositions of multiple breakpoint graphs and rapid exact solutions to the median problem 114

7 A fast and exact algorithm for the median of three problem—a graph decomposition approach 134

8 Discussion and conclusion 158
8.1 The tests of remaining evolutionary signal 158
8.2 Adequate subgraphs and the median problem 161

Bibliography 163

List of Tables

8.1 p-values for given number of common adjacencies 159

8.2 Statistical tests based on the cycle numbers 160

List of Figures

1.1 Chromosomes, genes and permutations. 2

1.2 Breakpoint graph. 4

1.3 Breakpoints and breakpoint distance. 6

1.4 Various genomic rearrangement mechanisms. 7

1.5 Illustration of DCJ operations. 8

1.6 Cap assignments, breakpoint graphs and flower graphs. 13

1.7 Phylogeny problems and the median problems. 17

1.8 MBG and median graph 19

1.9 Edge shrinking. 22

Chapter 1

Introduction

1.1 Genomes and gene orders

In the past nine decades, with the advance of technology, increasing amount of information about genomes has been revealed, both in the number of genomes studies and the lengths of genomes that can be sequenced—from the usage of recombination-based linkage maps [28] in 1921 to the banding structure under the microscope for the fly genome [22] in 1933; from the modern banding structure [7] in 1970 with application extended to primate genomes to the radiation hybrid technique [11] in 1975 which enabled the possibility of DNA sequencing bases; from the first completely sequenced virus genome [24] in 1975, to the first completely sequenced organelle genome [3] in 1981, and the first prokaryotic genome [9] in 1995, then the first eukaryotic genome [10] in 1996 and the first draft of whole genome sequencing of humans in 2001 [32, 15].

A genome consists of one or more chromosomes, which are either linear or circular shaped sequences of DNA base pairs. Each chromosome consists of two strands of DNA, twisted in a double helix structure. The genes encoding the proteins and all sorts of functional RNAs are scattered along the chromosomes, with or without gaps

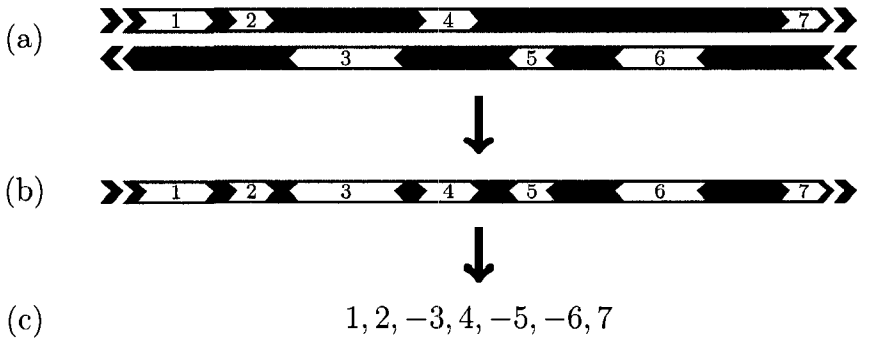


Figure 1.1: Chromosomes, genes and permutations. (a) denotes a double stranded DNA with 7 genes. The directions of the arrows represent the directions of each strand and the direction of the DNA takes the one for the upper strand ; (b) denotes the one line representation of the same DNA; (c) denotes the mathematical representation with permutations with signs indicating the direction of genes.

called non-coding areas in between. The information of each gene is stored on one of the two strands, and can only be read along the direction of that DNA strand (from its 5′-end to its 3′-end). By choosing one of the two DNA strands as the positive strand, the direction of the gene can be defined as positive if its information is stored on the positive strand; negative otherwise.

Genomes can be modeled mathematically by collections of linear or circular permutations, with each gene (in general, it can be any mark, such as a gene, a motif, a fragment of DNA, etc.) represented by a numerical number and its positive/negative direction can be represented by a plus/minus sign in front, as illustrated by Fig. 1.1. The representing permutations carrying plus/minus signs are called *signed permutations* and the ones without signs are called *unsigned permutations.* Very often, for convenience, the representing permutations are also called genomes, and hence we have *signed genomes* and *unsigned genomes.* Another related concept is *gene order*,

which emphasize the order in which genes (or markers in general) are arranged and how this order could possibly be changed.

When two genomes are compared, segments of chromosomes where the genes within it have the same order and signs in both genomes can be observed, which indicate a conserved evolutionary signal. The disruptions between conserved segments are called *breakpoints*, which are the traces of the rearrangements accumulated through the evolution all the way from the speciation of the two genomes.

Often we assume that the two genomes under comparison contain the same set of genes and every gene appears in each genome exactly once. This assumption on one hand simplifies the mathematical modeling, on the other its rationality can be well justified: 1) there are sets of genomes (e.g. most animal mitochondrial genomes) which fit this model exactly; 2) from the genes who appear in both genomes much more information of the rearrangements can be extracted, as they record the changes in both genomes; 3) the comparison of the two genomes can be decomposed into two phases— first to study the deletions/insertions of genes between genomes, then to study how the gene orders are shuffled from one to another. And with certain rearrangement mechanisms, this two-phase study proves valid [8, 19]; 4) the duplicates of genes can be represented by their exemplars [25], therefore every gene/exemplar only presents once in each genome.

1.2 Breakpoint graph

The comparisons of genomes can be also understood through graphs. For a signed genome, each gene is represented by a pair of vertices, called the head and tail to mark the two ends of the gene. For any two genes adjacent (no other genes/markers in between) on the genome, two vertices—one from each gene—representing the abutting

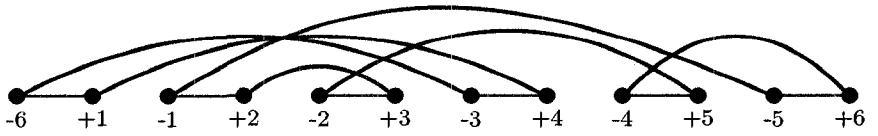


Figure 1.2: Breakpoint graph for blue genome (1 -5 -2 3 -6 -4) and red genome (1 2 3 4 5 6). Pairs of vertices denote the genes or mark the ends of linear chromosomes. The edges represent the adjacencies on the genomes.

parts of the two genes, are connected by an edge called an *adjacency edge*. For each linear chromosome, a pair of dummy vertices called *caps* are added to denote the beginning and ending of the chromosome. A cap is considered to be adjacent to the gene located at the same end of that linear chromosome, so an adjacency edge connecting them is added. Therefore in the graph, all vertices including caps are incident to exactly one edge, and the adjacency edges form a perfect matching of the graph as shown in Fig. 1.2.

To represent two genomes in comparison by the graph, adjacencies of genes in each genome are represented by adjacency edges labeled with colours (say red and blue) unique to each genome. Then the resultant graph is called a *breakpoint graph* (despite the fact that the edges in the graph denote the information about the adjacencies). For convenience we also extend the colours of the edges to the corresponding genomes and hence we call them red and blue genomes. Edges of each colour form a perfect matching, so the breakpoint graph is a 2-regular graph and it naturally decomposes into a set of cycles. The number of the cycles is denoted by c, called the cycle number of that breakpoint graph.

1.3 Distance measures

Genomic distance measures are used to indicate how far two genomes differs from each other. A valid distance measure should have a strong (either positive or negative) correlation with the total number of evolutionary events accumulated between genomes. Up to now there have existed two types genomic distance measures: the *breakpoint distance* and *edit distances*—a class of distance measures minimizing the number of operations needed to convert one genome into the other for a given set of allowed operations.

A distance measure is called *metric* if it satisfies the following triangular inequality,

$$d_{A,C} \leq d_{A,B} + d_{B,C} \qquad \text{for any genomes } A, B, C, \tag{1.3.1}$$

where $d_{A,B}$ denotes the genomic distance between genomes A and B. It can be easily verified that all of the following genomic distance measures are metric.

1.3.1 The breakpoint distance

The formal definition of a *breakpoint* is the place where an adjacency exists in one genome but not in the other. For two identical genomes, obviously there are no breakpoints. As the number of genomic rearrangements increases, more adjacencies on the genomes are broken and the number of breakpoints increases. The *breakpoint distance* is defined as the maximum number of breakpoints existing between two genomes, as illustrated by Fig. 1.3. For genomes with different number of genes and/or different number of chromosomes, different ways of counting breakpoints will lead to different total number of breakpoints. To make sure that the breakpoint distance is well defined, the maximum is taken. For example, for a pair of genomes

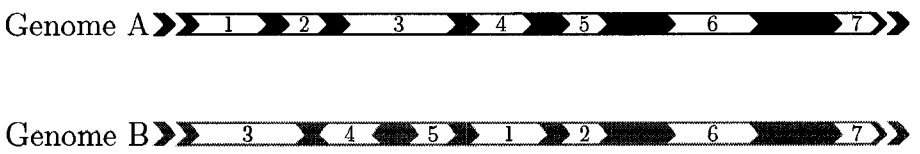


Figure 1.3: Breakpoints and breakpoint distance. This example shows 4 breakpoints existing between genomes A and B and hence their corresponding breakpoint distance is 4.

$A = 1, 2, 3, 4, 5, 6$ and $B = 1, 2, 3, 4, 2, 5, 3, 6, 1$. Two adjacencies (4-5 and 5-6) in A are missing in B; five adjacencies (4-2, 2-5, 5-3, 3-6 and 6-1) in B are absent in A. So their breakpoint distance is 5.

The calculation of the breakpoint distance is straightforward and no prior knowledge of rearrangement mechanisms is required. On the other hand, breakpoint distance does not tells us the possible sequence of rearrangements needed to convert one genome into the other.

1.3.2 The edit distance measures

Among many possible genomic rearrangement mechanisms proposed so far, the following ones are popular and well accepted: reversal, translocation, fission, fusion and transposition. A *reversal* of a fragment is to reverse the order of the genes contained in the fragment as well as the signs of the genes, illustrated by Figure 1.4(a). A *transposition* of a fragment is to excise out that fragment and reinsert it into the same chromosome but in a different location, as illustrated by Figure 1.4(b). A *translocation* between two chromosomes is to cut each chromosome in two and join fragments from different chromosomes together, as shown in Figure 1.4(c). A *fission* of a linear

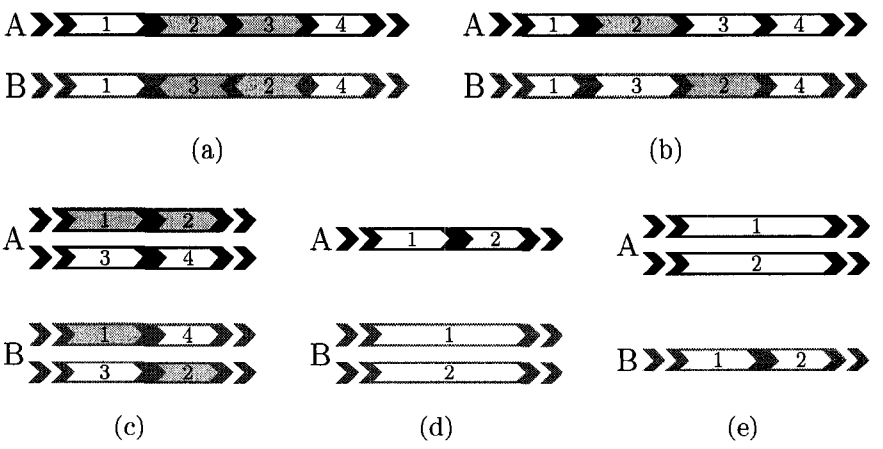


Figure 1.4: Various genomic rearrangement mechanisms: reversal (a), transposition (b), translocation (c), fission (d) and fusion (e) from genome A to genome B.

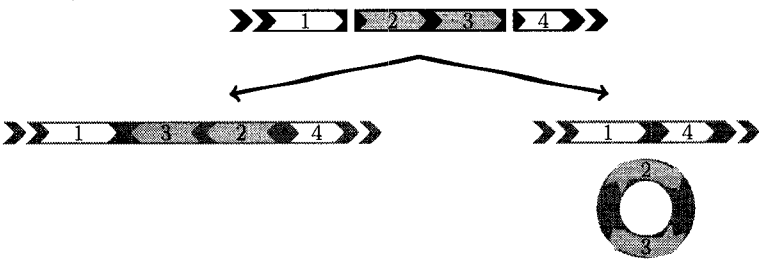
chromosome is to split this chromosome into two smaller linear chromosomes, as in Figure 1.4(d). A *fusion* of two linear chromosomes is to join the two into one longer chromosome as illustrated by Figure 1.4.(e).

The edit distance measures between two genomes are defined as the minimum number of operations needed to convert one genome into the other, given the allowed operation or the set of allowed operations.

The *reversal distance* is the minimum number of reversals needed to convert between genomes. It was first solved by Hannenhalli and Pevzner[14] for signed directed chromosomes, resulting in the following equation:

$$d = b - c + h + f, \tag{1.3.2}$$

where d is the reversal distance, b is the breakpoint distance, c is the number of cycles in the breakpoint graph, h is the number of structure called hurdles and f is a special structure called a fortress, which appears at most once. Finding the reversal distance for unsigned genomes has been proven NP-hard[5]. The minimum number of reversals, translocations, fissions and fusions for multi-chromosomes to convert between each



(a)

Figure 1.5: Illustration of DCJ operations: a reversal (at the left) and formation of a small circular chromosome (at the right).

other was first explored by Hannenhalli and Pevzner[13] and followed by [30], [21] and [16], and it is usually called HP-distance. The polynomial-time algorithm to calculate the translocation distance was solved by Hannenhalli [12].

1.3.3 DCJ distance

Recently Yancopoulos, Attie and Friedberg [33] proposed a generalized operation, called double-cut-and-join (DCJ for short). To perform a DCJ operation is to cut the genome twice in any two locations, and join the fragments back but in a different form, as illustrated by Figure 1.5. DCJ operations may perform operations like reversal, translocation, fission, fusion and two consecutive DCJ operations may perform transposition and block interchanges.

The DCJ distance is defined as the minimum number of DCJ operations between two genomes. For signed genomes containing any number of linear chromosomes and/or circular chromosomes, the DCJ distance is expressed as:

$$d = n + \chi - c, \tag{1.3.3}$$

where d is the DCJ distance, n is the number of genes contained in each genome, χ

is the number of linear chromosomes and c is the maximum number of cycles in the breakpoint graph.

The DCJ distance on one hand is an edit distance defined as the minimum number of DCJ operations; on the other hand it has a graph-theoretic formulation, whose value can be calculated by counting the number of cycles in the breakpoint graphs directly, similar to the calculation of the breakpoint distance by counting the multiple edges in the breakpoint graph.

1.4 The distance distributions among random breakpoint graphs

For the edit distance measures discussed above, a minimum sequence of events can be inferred for any two genomes necessary to transform one into the other. This inferred events sequence rarely corresponds to the true evolutionary scenarios and the inferred distance very often underestimates the total number of true events. Nevertheless, these inferred genomic distances can be viewed as indicators of the evolutionary signal remaining on the genomes and the closeness of two genomes in term of their phylogenetic relationship.

The problem can be formulated as a hypothesis test, where the null hypothesis says the two genomes in comparison do not show any significant evolutionary signal in their gene orders, while the alternative hypothesis says there is a strong signal remaining. To perform a test of the hypothesis, we need to know how the genomic distance is distributed under the null hypothesis—where genomes are randomly selected. Finding the distance distributions for random genome pairs consists of the first part of my Ph.D. research. The first three papers (Chapters 2,3,4) focus on the

DCJ distance distribution. Due to a probabilistic negligible difference with the DCJ distance, reversal distance/HP-distance have the same asymptotic distribution. An earlier paper [27] studies the DCJ distribution for circular genomes, where the expectation of number of cycles is expressed as $\log 2 + \frac{\gamma}{2} + \frac{1}{2}\log n$. My first paper (Chapter 2), as a continuation, studies the distribution for genomes containing the same number of linear chromosomes. The second paper (Chapter 3) extends the result to the genomes containing any number of linear and/or circular chromosomes and proposes the *flower graph*—a new graphic representation for genome with linear chromosomes, where all caps are merged into a single cap vertex. The third paper (chapter 4) then adopts a pure combinatorial enumeration method to provide more accurate analysis for the expectation and the variance of the distribution and uses the same technique to derive the distribution of the number of plasmids (smaller circular chromosomes) in a randomly constructed genomes.

1.4.1 Random breakpoint graphs

To find the distribution of distances we need to enumerate all pairs of random genomes. However since we can always relabel one of the two genomes as the identity genome, it suffices to keep one genome as the identity genome (and call it *the reference genome*), and construct the second genome randomly (called *the random genome*) from an equiprobabilistic distribution.

The strict ordering model constrains that the random genome should contain exactly the same number of linear chromosomes as the reference genome and no circular chromosomes are allowed. Under this model the adjacency edges from both the reference genome and random genome and another set of edges representing the genes and pairs of caps need to be considered to make sure that no circular chromosomes are

formed, as well as to form a desired number of cycles exist in the breakpoint graph. Then we are dealing with 3-regular graphs, which makes the problem relatively hard.

An alternative model relaxes the constraint against the circular chromosomes. Then edges representing the genes and pairs of caps are no longer needed. All information is contained in the breakpoint graph. Under this model, the random genome does not need to be explicitly constructed, instead we just construct all random breakpoint graphs.

There exists some difference between the strict and relaxed models, in terms of the distance distributions. However, this difference becomes negligible when the size of the genome (number of genes) become large, i.e. the two distributions are asymptotically the same as the number of genes goes to infinity, which is implied by the Kim-Wormald theory [17].

1.4.2 Cap optimization

Caps are introduced for linear chromosomes to denote their ends. Before constructing the breakpoint graph, we need to know how caps in one genome are mapped to the caps in the other genome, like the correspondence for the genes. This is called *cap assignment*. Different cap assignments do not make a difference from the view of biology, but they may lead to different breakpoint graphs and hence possible different cycle numbers, as illustrated by Figure 1.6. The two breakpoint graphs in subfigures (a) and (b) correspond to the same pair of genomes. With different cap assignments, the breakpoint graph in (a) contains 2 cycles while the one in (b) only has 1 cycle.

Since the edit distance is always defined as the minimum number of operations between two genomes and the DCJ distance is determined by Equation 1.3.3, the optimal cap assignment should be taken as the one whose corresponding breakpoint

graph contains the maximum number of cycles. The procedure for finding the optimal cap assignment is called *cap optimization.*

There is an easy rule for the cap optimization under the DCJ distance measure. Given a breakpoint graph with any cap assignment, decompose the 2-regular graph into a set of cycles not containing caps (*inner cycles*) and a set of paths which terminate at the caps. The paths ending with the same coloured edges are called *homogeneous* paths and the ones ending with different coloured edges are called *heterogeneous* paths. From the fact that one heterogeneous path can form a cycle by itself and two homogeneous paths with different ending colours can form a cycle, the rule about the cap optimization says that the maximum number of cycles is determined by

$$c = n + \chi + \frac{\psi_{HE}}{2}, \tag{1.4.1}$$

where ψ_{HE} is the number of heterogeneous paths.

We can also represent the genomes with linear chromosomes by *flower graphs*, a variant of breakpoint graph containing only one cap, that a vertex is incident to a cap means it is connected to this cap node. For every pair of genomes, there only exists one corresponding flower graph, unlike breakpoint graphs.

1.4.3 Combinatorics and generating functions

The ordinary generating function (OGF) and the exponential generating function (EGF) are defined as the following formal power series correspondingly:

$$F(z) = \sum_n A_n z^n, \tag{1.4.2}$$

$$F(z) = \sum_n \frac{A_n}{n!} z^n, \tag{1.4.3}$$

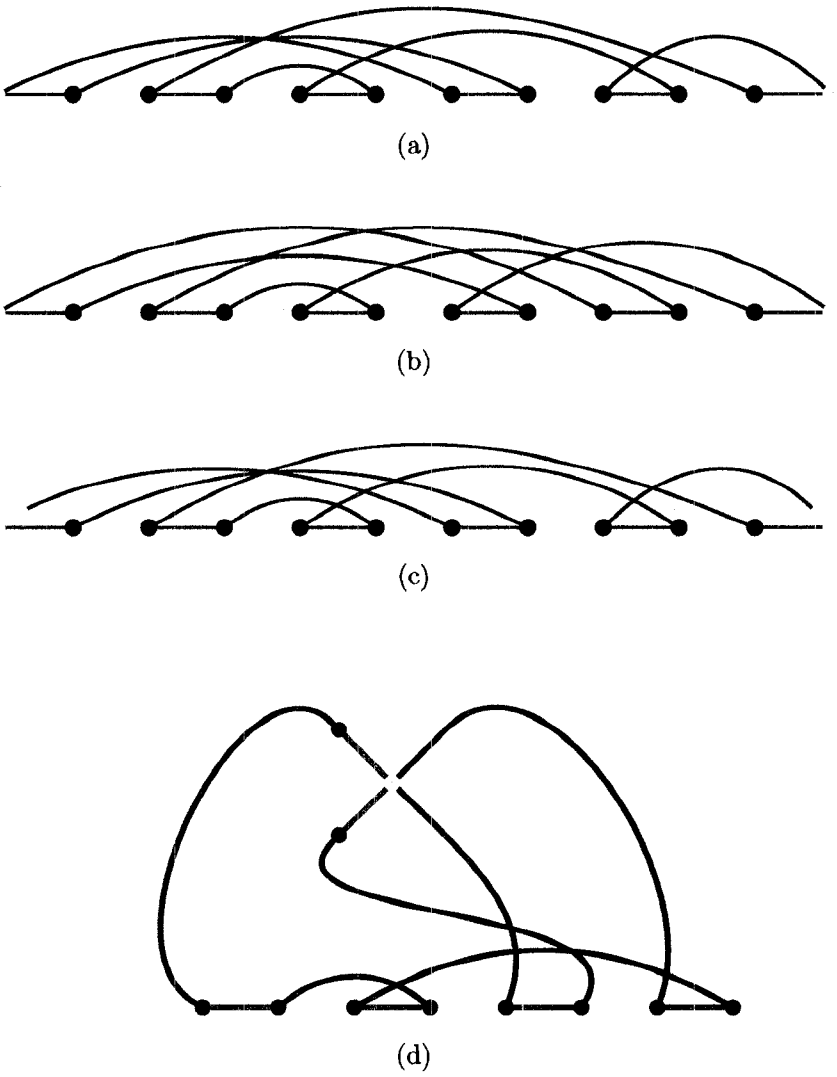


Figure 1.6: Cap assignments, breakpoint graphs and flower graphs. Subfigure (a) and (b) are two breakpoint graphs corresponding to the same pairs of genomes, due to different cap assignments, they contain different number of cycles, two and one correspondingly. Subfigure (c) illustrates the decomposition of the graph into a set of cycles not containing caps, and a set of paths ending with caps. Subfigure (d) shows the flower graph corresponding to the same pair of genomes.

where A_n counts the number of objects of size n and the notation $[z^n]$ takes the coefficient of z^n in F.

The bivariable generating functions are defined similarly, where u marks some particular structure of the object (such as a cycle in the breakpoint graph or a common adjacency between two genomes (see Section 1.5))

$$F(z,u) = \sum_n \sum_{m=0}^{n} A_{n,m}\, u^m z^n, \tag{1.4.4}$$

$$F(z,u) = \sum_n \sum_{m=0}^{n} \frac{A_{n,m}}{n!}\, u^m z^n, \tag{1.4.5}$$

$$F(z,u) = \sum_n \sum_{m=0}^{n} \frac{A_{n,m}}{m!\, n!}\, u^m z^n. \tag{1.4.6}$$

If random variable X denotes the number of that particular structure existing in the object, then its kth factorial moment $X_{(n)} = X(X-1)\ldots(X-k-1)$ can be expressed as

$$\mathbf{E}[X_{(n)}] = \frac{[z^k]\left(\partial_u^k F(z,u)\right)|_{u=1}}{[z^k]F(z,1)}. \tag{1.4.7}$$

From this formula, the expectation and variance of X can be derived.

In the third paper (see Chapter 4), by deriving the generating function that counts the number of random breakpoint graphs with certain numbers of inner cycles, heterogeneous paths and homogeneous paths, we can derive the expectation and the variance for the total number of cycles. Hence the DCJ distance distribution is asymptotically a normal distribution centered at $n - \frac{2\chi_1\chi_2}{2\chi_1+2\chi_2-1} - \frac{1}{2}\log\left(\frac{n+\max(\chi_1,\chi_2)}{\chi_1+\chi_2}\right)$ with variance $\frac{8\chi_1^2\chi_2^2}{(2\chi_1+2\chi_2-1)^2(2\chi_1+2\chi_2-3)} + \frac{1}{2}\log\left(\frac{n+\max(\chi_1,\chi_2)}{\chi_1+\chi_2}\right) - \frac{1}{4(\chi_1+\chi_2)} + \frac{1}{4(n+\max(\chi_1,\chi_2))}$, where χ_1 and χ_2 are the numbers of linear chromosomes in the two genomes.

1.4.4 Plasmids in random genomes

By using the same techniques, the expected number of plasmids formed in the random constructed genome is derived as $\frac{1}{2}\log\frac{n}{\chi}$, where n is the number of genes and χ is the number of linear chromosomes. Taking n as 10000 and χ as 23, which is a profile similar to the one of the human genome, the expected number of plasmids is 3.

1.5 The Poisson distribution of common adjacencies

In the fourth paper (see Chapter 5), we study the breakpoint distance distributions between pairs of random genomes with a similar goal. The breakpoint distance, denoted by b, and the number of common adjacencies, denoted by a, are related by the following equation:

$$a + b = n - \chi, \tag{1.5.1}$$

where n is the number of genes and χ is the number of linear chromosomes in each genome. For given numbers of genes and linear chromosomes, it suffices to study the distribution for the number of common adjacencies a. Again we can fix one of the genomes as the identity genome and let the other one be any random one.

We have studied all kinds of genomes—signed or unsigned, circular or linear, single-chromosome or multichromosomal. The unsigned single-chromosome case is related to the dinner table problem [23] and non-attacking kings problem [1]. G. Tesler has previously derived some results for linear, single-chromosome genomes [31].

For single-chromosome genomes, we find the generating functions counting the number of genomes containing certain numbers of genes and common adjacencies for

each signed or unsigned, linear or circular cases. Through these generating functions, either the probability mass functions (for signed genomes) or the factorial moments (for unsigned genomes) of the adjacency numbers can be derived. Then through the above results, we can get the asymptotic probability distributions.

For multi-chromosomal genomes containing any number of chromosomes, the exact expectations and variances of the adjacency numbers are derived and so are their factorial moments. Through these factorial moments, the asymptotic probability distributions are found.

These results can be summarized as: for all kinds of signed genomes, the asymptotic probability distribution of the adjacency number is Poisson with parameter 2; for all kinds of unsigned genomes, the asymptotic distribution is Poisson with parameter $\frac{1}{2}$.

1.6 The median problem

The evolutionary signal remaining in the gene orders, in the form of distance measures, can be used to construct the phylogeny relationship for a group of species. Phylogeny relationships can be represented by a rooted or unrooted tree called the phylogenetic tree with each node representing a modern or ancestral species. The edges between the nodes represent the distance between the corresponding two species. A fully resolved phylogenetic tree is a binary tree.

For phylogenetic trees with specified edge lengths, the summation of these lengths is called the total distance of this tree. Because the large scale genomic rearrangements occurred with small probabilities, among all possible trees the one with smallest total distance most likely reflects the real phylogeny. Following this parsimony philosophy, the problem of uncovering the real phylogeny becomes that of finding the tree

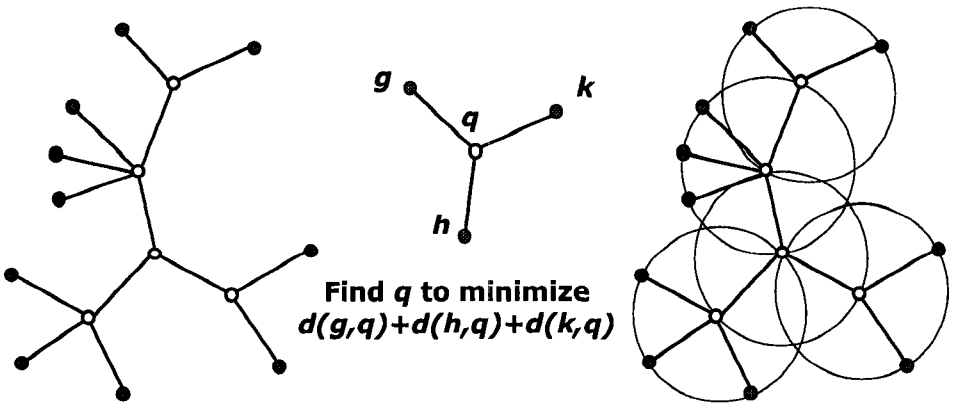


Figure 1.7: Left: unrooted phylogeny with open dots representing ancestral genomes to be inferred. Middle: median problem with three given genomes g, h and k and median q to be inferred. Right: decomposition of phylogeny into overlapping median problems.

with smallest total distance, which consists of two smaller problems: i) the "small phylogeny problem" of finding the minimum total distance for a given phylogeny tree; ii) the "big phylogeny problem" of seeking the minimum lengthed tree among all possible trees, on top of the "small phylogeny problem".

To solve the small phylogeny problem, we can optimize each ancestral genome by its three or more immediate neighbors, and iterate the same process through the tree until this whole process converges to either a global or local optima, as illustrated by Fig. 1.7. Enumerating all possible phylogeny trees and solving each corresponding small phylogeny problems are just enough to solve the big phylogeny problem.

The key step in this approach is to optimize the ancestral genome by its immediate neighbors. This problem is called the *median problem*, which is formally defined as to find a genome q, which minimizes the total distance $\sum_{g \in \mathcal{G}} d_{q,g}$, for a given set of three or more genomes $\mathcal{G}$, and a genomic distance measure d. When $\mathcal{G}$ only consists of three genomes, the problem is called the *median of three problem*, which lies

at the heart of optimizing binary phylogeny trees. This approach was first explored by Sankoff and Blanchette[26] and then followed by Moret et al. [20], Bourque and Pevzner [4], Adam and Sankoff [2].

The median problem under reversal distance measure has been proven NP-hard and APX-hard, as well as the one under DCJ distance measure[6, 29]. Exact algorithms have been developed for small instances[20, 6] and heuristic algorithms with varying degree of efficiency and accuracy have also been developed[4, 2, 18].

In the second part of my Ph.D. research, I look at the possibility of decomposing the median problem into a set of smaller problems, to reduce the computational complexity. I consider the DCJ distance measure, because of its simple mathematical nature. Up to now, I have considered the genomes with circular chromosomes, but the results can be extended to the linear case just with some modifications.

1.6.1 Graphic representation and decomposition

To model the median problem, the breakpoint graph is extended to the *multiple breakpoint graph* (MBG for short), which contains three or more perfect matchings and each of them is assigned a unique colour (denoted by $1, 2, 3, \ldots$) representing the genomes, as illustrated by Fig. 1.8. The size of the MBG is defined as half the number of vertices in the graph, which also equals to the number of genes and the size of each perfect matching. The rank of the MBG is the number of genomes in the set $\mathcal{G}$, denoted by $N_{\mathcal{G}}$. We use *i-matching* and *j-edges* to refer to the perfect matching bearing colour i and any set of edges with colour j correspondingly.

To model the candidate of the median genome, we add another perfect matching to the MBG and assign the colour 0 to it. The resultant graph is called the *median graph*. The size and the rank of the median graph are the same as the ones for its

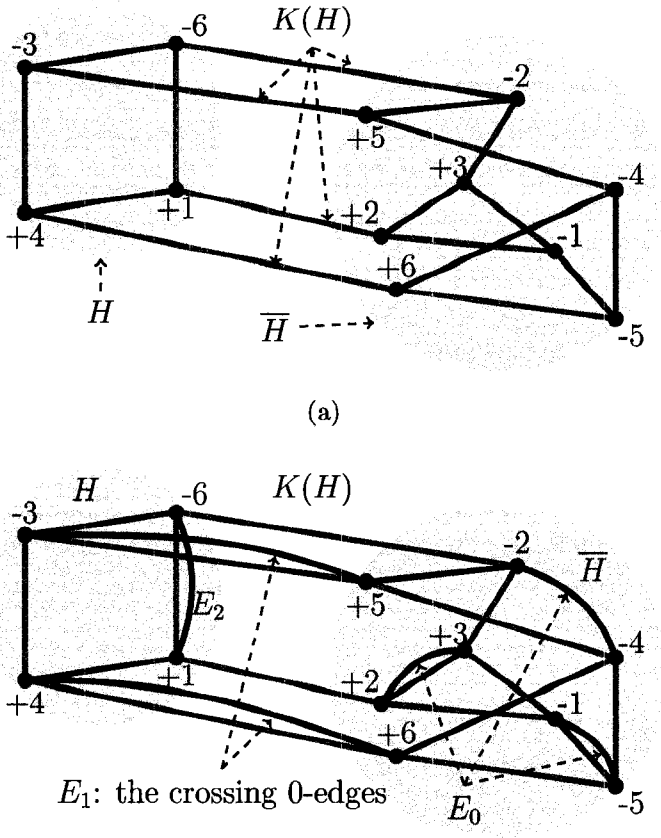


Figure 1.8: MBG and median graph. Red, blue, green and black denote colours 1, 2, 3 and 0. (a) An MBG based on three genomes, red (1 2 3 4 5 6), blue (1 -5 -2 3 -6 -4) and green (1 3 5 -4 6 -2). A subgraph H, the connecting edge set $K(H)$ and the complementary subgraph $\overline{H}$ are illustrated. (b) A median graph. The candidate matching is divided into three 0-edge sets: E_0, E_1 and E_2.

corresponding MBG.

An i-matching and a j-matching in either an MBG or a median graph form a 2-regular graph, consisting of a set of colour-alternating cycles. These cycles are referred to as *i-j cycles* and the total number of i-j cycles is denoted by $c_{i,j}$.

Because of Equation 1.3.3 for DCJ distance,

$$\min \sum_{g \in \mathcal{G}} d_{q,g} = N_{\mathcal{G}} \cdot n - \max \sum_{1 \leq i \leq N_{\mathcal{G}}} c_{0.i}. \tag{1.6.1}$$

Hence minimizing the total DCJ distance is equivalent to maximizing the total number of colour-alternating cycles $c = \sum_{1 \leq i \leq N_{\mathcal{G}}} c_{0.i}$. The 0-matching maximizing the total cycle number c is called the *optimal matching*. Usually there are more than one optimal matching for a given MBG.

Now consider an induced subgraph H of an MBG B, and denote $\bar{H}$ as its complementary induced subgraph induced by the vertex set $\mathbf{V}(B) - \mathbf{V}(H)$. A 0-matching is called *non-H-crossing*, if none of its 0-edges connects H to the remaining subgraph $\bar{H}$; otherwise it is H-crossing. A non-H-crossing 0-matching consists of a perfect 0-matching of H and a perfect 0-matching of $\bar{H}$. For a subgraph H, if there always exists an optimal matching which is non-H-crossing for any MBG containing H, then it is called a *decomposer*; if all optimal matchings are non-H-crossing for any MBG containing H, it is called a *strong decomposer*.

For an MBG of size n, the number of all possible 0-matching we need to search for an optimal 0-matching is $\frac{(2n)!}{2^n n!}$. By finding a decomposer H of size m, the corresponding search space is reduced to $\frac{(2m)!(2n-2m)!}{2^n m!(n-m)!}$, which is a reduction by a factor of about $\frac{n^n}{\sqrt{2}m^m\,(n-m)^{n-m}}$.

The induced subgraph H of size m is an *adequate subgraph* if $\frac{N_{\mathcal{G}} \cdot m}{2}$ or more cycles can be formed from H with one of its perfect 0-matchings. H is *strongly adequate* if the maximum number of cycles it can form is larger than $\frac{N_{\mathcal{G}} \cdot m}{2}$.

In the fifth paper (Chapter 6), we show that for an adequate subgraph H, if a 0-matching is H-crossing, then we can always construct another 0-matching which is non-H-crossing and the number of cycles formed by the new 0-matching is no less than the number from the original one. Similarly for a strong adequate subgraph H, given an H-crossing 0-matching, there always exists a non-H-crossing 0-matching, with more cycles. Theorem 2 in Chapter 6 using the above results, proves that any adequate subgraph is a decomposer and any strongly adequate subgraph is a strong decomposer.

By finding adequate subgraphs iteratively, an optimal 0-matching can be repeatedly partitioned into a set of subgraphs, hence the number of the 0-matchings we need to search to solve the median problem can be reduced repeatedly to a much smaller one.

Instead of partitioning the optimal 0-matching, we discuss the methods of decomposing the MBG (the median problem) into a set of smaller MBGs (smaller median problems). But before that we first introduce an edge operation called *edge shrinking* as in Fig. 1.9. To shrink a 0-edge e in an MBG, is to delete e and join any two edges neighboring e of the same colour into a new edge. To shrink a set of 0-edges is to shrink them one by one.

If an MBG contains an adequate subgraph H by shrinking all possible perfect 0-matchings of H (called partial 0-matchings), the MBG can be reduced to a number of smaller MBGs, where the total number is determined by the size of connecting edges of H. Because several partial 0-matchings can lead to the same smaller MBG, for each of these smaller MBGs, among the partial 0-matchings leading to it, we take the one forming the largest number of cycles with H, and put them into a set of partial 0-matchings called *the major set*. Obviously the major set guarantees that at least one partial 0-matching in it is contained by an optimal 0-matching of the

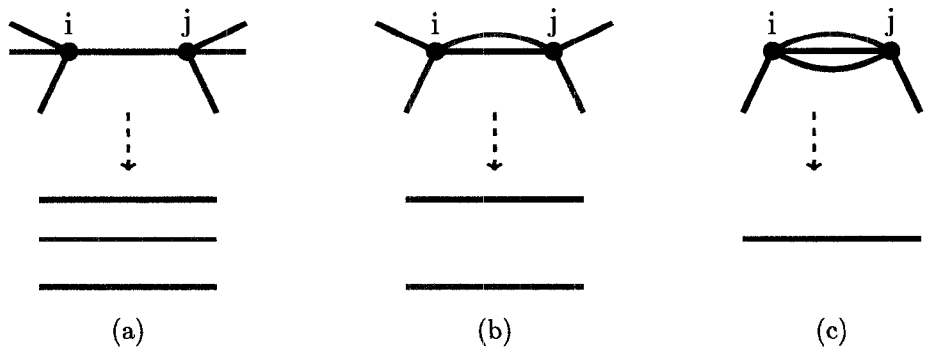


Figure 1.9: Edge shrinking.To shrink a 0-edge e in an MBG, is to delete e and join any two edges neighboring e of the same colour into a new edge. To shrink a set of 0-edges is to shrink them one by one.

original MBG. We can further reduce the size of the major set by excluding some of the partial 0-matchings according to some specific rule—the ones that construct too few cycles compared to other partial 0-matchings.

Let μ denote the number of partial 0-matchings in the major set. When H is small μ is usually very small. Hence by detecting an adequate subgraph, the original MBG (hence the original median problem) is decomposed into a small set of smaller MBGs (smaller median problems). At each step, the complexity of the problem reduces by a factor of $\frac{(2n)^m}{\mu}$, where n is the size of the MBG and m is the size of the adequate subgraph.

By recursively detecting the adequate subgraphs and decomposing of the median problems and MBGs, the original median problem might be solved rapidly and exactly.

1.7 The median of three problem

In the sixth paper (Chapter 7), the theory of adequate subgraphs is applied to the median of three problems. The *simple adequate* subgraphs are the ones not containing any smaller adequate subgraphs. The adequate subgraphs of rank 3 are the ones contained in the MBGs modeling the median of three problems. In this paper, we prove some important properties about the simple adequate subgraphs of rank 3, and discuss the algorithm inventorying these simple adequate subgraphs and give some improving techniques. We argue that it is more practical to use the simple adequate subgraphs of small size to decompose the MBGs instead of using general adequate subgraphs, or simple ones of large size. We then introduce *ASMedian*, the algorithm which solves the median of three problem fast and exactly for most instances. Finally we generate simulated data sets with varying parameters, and use these data to test the performance of this graph decomposition based algorithm.

We prove the following properties in various aspects, about the simple adequate subgraphs of rank 3, which enhances our understanding about the adequate subgraphs and can be used to speed up the algorithm inventorying them.

- The vertex degrees of simple adequate subgraphs of rank 3 are either 2 or 3;
- except for the multiple edges, there are no more other odd sized simple adequate subgraphs of rank 3;
- except for the multiple edges, the maximum number of cycles of any simple adequate subgraph of rank 3 is exactly $\frac{3m}{2}$, where the size of the graph m is an even number.

Then we ask what is the total number of all possible adequate subgraphs? Because the adequate subgraphs are the key in the decomposition method, we hope

there are as many as possible adequate subgraphs so that it is probable for any MBG to contain at least one of them. By showing there exists a family of simple subgraphs called *mirrored tree* graphs, whose sizes can be arbitrary large, we prove that there are infinite number of simple adequate subgraphs and hence an infinite number of adequate subgraphs.

To inventory all simple adequate subgraphs for a given size, we need to search all possible subgraphs and to find their optimal 0-matching. Then its computational complexity is at least $\frac{((2m)!)^4}{2^{4m}(n!)^4}$. To speed up the program, we use several techniques either to eliminate the isomorphisms of the graphs or to restrict the vertex degrees to 2 or 3.

The facts that finding an adequate subgraph can reduce the search space significantly and that there are infinite many adequate subgraphs, tempt us to discover all adequate subgraphs in each MBG. However it may be very costly to discover some of them, so we need to balance the benefit and the cost.

First we argue that it is advisable to just use simple adequate subgraphs in decomposing the MBGs. That is because the number of simple adequate subgraphs is much smaller than the one of general adequate subgraphs; and usually the non-simple adequate subgraphs consist of several simple ones embedded into each other, so many non-simple ones can be discovered through the constituent simple ones.

For the simple subgraphs of large size, several problems arise. First, the time needed to inventory them can be inhibitingly long. Second, the total number of simple subgraphs for a given large size can be huge. Plus the increasing time spent on discovering each of these subgraphs, the total time spent on discovering these large size simple adequate subgraphs can be unaffordable.

From a rough estimation, we can show that the existing probability of simple adequate subgraphs of small sizes in a random MBG is much higher than the ones

for large sizes. Considering these facts, it is more efficient to just use a set of small simple adequate subgraphs in decomposing the MBGs.

The ASMedian algorithm uses a branch-and-bound method. The algorithm starts from the original MBG modeling the median of three problem and maintains a list of unexamined intermediate MBGs. At each step the algorithm chooses an intermediate MBG according to a certain rule. After examining that intermediate MBG, a set of smaller intermediate MBGs are generated. If a simple adequate subgraph is discovered, the smaller MBGs are constructed according to its major set; otherwise, pick up a vertex in the graph, and generate the smaller intermediate MBGs by shrinking any possible 0-edge incident to it respectively.

The upper bound of the cycle number comes from the metric property of the DCJ distance measure, $d_{i,k} \leq d_{i,j}+d_{j,k}$, from which a lower bound for the total distance can be derived, $d \geq \frac{d_{1,2}+d_{2,3}+d_{1,3}}{2}$. Because of Equation 1.3.3, we have $c \leq \frac{3n}{2}+\frac{c_{1,2}+c_{1,3}+c_{2,3}}{2}$. The upper bound of cycle numbers for 0-matchings containing the current existing partial 0-matching is the summation of the upper bound of the corresponding intermediate MBG and the number of cycles formed by the original MBG with that partial 0-matching. If this upper bound is no larger than the largest cycle number of the perfect 0-matchings found by the algorithm, this partial 0-matching together with the corresponding intermediate MBG are discarded, for they will never lead to any larger cycle number.

Sets of data are simulated to see the performance of the ASMedian algorithm. These simulated data are controlled by two parameters, n the number of genes each genome contains and π, the ratio of the number of reversals over n. Under each combination of n and π, we generate 10 instances of genome triplets by applying πn reversals to the identity genome. The number of genes n varies among 10, 20, 30, 40, 50, 60, 80, 100, 200, 300, 500, 1000, 2000, 5000 and π starts from 0.1 and increases by

intervals of 0.1. From the results on simulated data, we can see that huge speedups—from thousands to millions—are achieved by using adequate subgraphs, and genomes containing hundreds or even thousands of genes can be solved quickly, as long as the ratio of the distance to number of genes is not too high.

Chapter 2

Paths and Cycles in Breakpoint Graphs of Random Multichromosomal Genomes

Wei Xu, Chunfang Zheng and David Sankoff. Journal of Computational Biology. 14(4): 423-435. 2007. Mathematical work all done by Wei Xu. Simulations and part of the writing by Chunfang Zheng and David Sankoff.

Abstract

We study the probability distribution of the distance $d = n + \chi - \kappa - \psi$ between two genomes with n markers distributed on χ chromosomes and with breakpoint graphs containing κ cycles and ψ "good" paths, under the hypothesis of random gene order. We interpret the random order assumption in terms of a stochastic method for constructing the bicoloured breakpoint graph. We show that the limiting expectation of $E[d] = n - \frac{1}{2}\chi - \frac{1}{2}\log\frac{n+\chi}{2\chi}$. We also calculate the variance, the effect of different numbers of chromosomes in the two genomes, and the number of plasmids, or circular chromosomes, generated by the random breakpoint graph construction. A more realistic model allows intra- and interchromosomal operations to have different probabilities, and simulations show that for a fixed number of rearrangements, κ and d depend on the relative proportions of the two kinds of operation.

1 Introduction

Though there is a large literature on chromosomal rearrangements in genome evolution and algorithms for inferring them from comparative maps, there is a need for ways to statistically validate the results. Are the characteristics of the evolutionary history of two related genomes as inferred from an algorithmic analysis different from the chance patterns obtained from two unrelated genomes? Implicit in this question is the notion that the null hypothesis for genome comparison is provided by two genomes, where the order of markers (genes, segments or other) in one is an appropriately randomized permutation of the order in the other. In a previous paper [11], we formalized this notion for the case of the comparison of two random circular genomes, such as are found in prokaryotes and in eukaryotic organelles. We found that the expected number of inversions necessary to convert one genome into the

other is $n - O(\frac{1}{2}\log n)$, where n is the number of segments (or other markers). Related work has been done by R. Friedberg (personal communication) and by Eriksen and Hultman [2].

In another paper [10], we used simulations to throw doubt on whether the order of synteny blocks on human and mouse retains enough evolutionary signal to distinguish it from the case where the blocks on each chromosome are randomly permuted.

In this paper, we begin to bridge the gap between mathematical analysis of simple genomes and simulation studies of advanced genomes. We extend the mathematical approach in [11] to the more difficult case of genomes with multiple linear chromosomes, such as those of eukaryotic nuclear genomes, which not only undergo inversion of chromosomal segments, but also interchromosomal translocation. The presence of chromosomal endpoints changes the problem in a non-trivial way, requiring new mathematical developments. Key to our approach in this and previous papers is the introduction of randomness into the construction of the breakpoint graph rather than into the genomes themselves, which facilitates the analysis without materially affecting the results. One aspect of this is that the random genomes with multiple linear chromosomes may also include one or more small circular fragments, or *plasmids.*

Our main result is that the number of operations necessary to convert one genome into the other is $n - O(\frac{1}{2}\log\frac{n+\chi}{2\chi} + \frac{3}{2}\chi)$, where χ is the number of chromosomes in each genome. This result is validated by exact calculations of a recurrence up to large values of n and χ, by simulations, by analytic solution of a somewhat relaxed model, and by solving the limiting differential equation derived from the recurrence.

We also propose models where the randomness is constrained to assure a realistic predominance of inversion over translocation. We use simulations of this model to demonstrate how key properties of the breakpoint graph depend on the proportion of intra- versus interchromosomal exchanges.

2 The Breakpoint Graph: Definitions and Constructions

In the comparison of two genomes, they each can be considered to be made up of a set of markers, be they genes, probes or chromosomal segments, disposed sequentially along a number of linear chromosomes or, in some primitive genomes, a single circular chromosome. Each marker has two ends, called 5' and 3', and its orientation (or strandedness) is defined by whether the 5' end is to the left or right of the 3' end. Each marker in one genome corresponds to exactly one identical (or orthologous) marker in the other, but the markers are generally partitioned differently among chromosomes in the two genomes and may be oriented differently. For mathematical purposes, all the markers on one genome may be assigned positive sign while all those on the other genome are assigned positive or negative signs depending on whether they have the same or opposite orientations, respectively, in the two genomes.

In the computational theory of genome rearrangements, whose literature may be chronologically sampled in references [13, 8, 9, 5, 3, 4, 12, 15], the differences between the genomes are assumed to be due to a series of operations of a limited number of types. For our purposes, we consider inversion: the reversal of the order of a number of contiguous markers on a chromosome, accompanied by a change of sign of each of these markers, reciprocal translocation: the exchange of prefixes or suffixes of two chromosomes, and generalized transposition: the excision and circularization of a chromosomal fragment containing a number of contiguous markers from a chromosome and the re-linearization and re-insertion of the same fragment, reversed or not, between two other markers on the same chromosome.

The genomic distance between the two genomes is defined to be the number of operations necessary to convert one genome into another. For generalized trans-

positions, we count excision/circularization and re-linearization/re-insertion as two operations, and for inversions, we allow the reversal of an entire chromosome without counting it in the distance, since physically this simply corresponds to different ways of looking at the same chromosome, without any structural disruption. For our purposes, the prefixes or suffixes exchanged during translocation must contain a proper subset of the markers on each chromosome, otherwise the number of chromosomes changes, a mathematically tractable process, but not within the scope of the present paper.

This distance can be efficiently computed using the bicoloured *breakpoint graph*. In this graph, $2n$ vertices represent the 5' and the 3' ends of each marker. The edges represent the *adjacencies* between the ends of successive markers on a chromosome. We colour the edges from one genome (R) red, the other (B) black. We denote by χ the number of chromosomes in each genome. With the addition of dummy vertices (*caps*) at the endpoints of the χ red linear chromosomes, and dummy red edges connecting each cap to one marker end, the breakpoint graph decomposes automatically into alternating colour cycles and alternating colour paths. Caps occur at the start or termination of some paths, but a path can also start or terminate with a non-cap vertex. The number χ of linear chromosomes, the number κ of cycles and the number ψ of paths having at least one cap, are the components in the formula for genomic distance

$$d = n + \chi - \kappa - \psi, \tag{2.1}$$

as adapted from [12] and [15]. Despite the apparent asymmetry in our treatment of the red and black genomes, d is a symmetric function. Reversing which genome is coloured red and which is black would not affect d. Note that our use of "red" and "black" edges here corresponds to "black" and "gray" edges, respectively, in some

other papers, e.g., [3, 4, 15].

The breakpoint graph has $2n + 2\chi$ vertices corresponding to the $2n$ marker ends and the 2χ caps. The adjacencies in R determine $n - \chi$ red edges and the adjacencies in B determine $n - \chi$ black edges. The caps adjacent to chromosome ends determine a further 2χ red edges, for a total of $n + \chi$ red edges and $n - \chi$ black edges. Because each marker vertex, except black vertices at the end of paths, is incident to exactly one red and one black edge, the graph decomposes naturally into 2χ alternating colour *paths*, with or without one or more disjoint alternating colour *cycles*.

3 The Randomness Hypothesis and the Relaxation of Linearity

The key to a mathematically tractable model of random genomes is to relax the constraint that genome B is composed only of linear chromosomes. (We may retain this constraint for genome R.) The only structure we impose on B that there are 2χ vertices that represent the starting points or terminations of chromosomes, and each of the other $2n - 2\chi$ vertices is adjacent to one other vertex: furthermore the 2χ start and end points and the $n - \chi$ pairings, which define the black edges from genome B, are chosen at random from the $2n$ vertices.

Studying the statistical structure of the set of paths and cycles in the breakpoint graph is facilitated by relaxing the condition that genome B is composed only of χ linear chromosomes, but the consequence is that the random choice of vertices defines a genome that contains not only this number of linear chromosomes, but also in general several circular plasmids. There are partial mathematical results [6] which strongly suggest that this relaxation does no violence to the probabilistic structure

of the breakpoint graph.

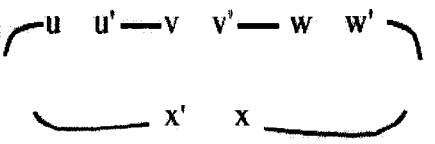


Figure 1: Random vertex pairing can give rise to plasmids .

For example, consider any vertex v, as in Figure 1. The chromosome containing v in genome B also contains v', the vertex at the other end of the same marker. It also contains u' and w, where u' and v are chosen by the random process to be adjacent in that genome and the vertex w adjacent to v'. It will also contain w', the other end of the marker containing w, and u, the other end of the marker containing u', and so on. Eventually, the two ends of construction will arrive at the two ends of a single marker, such as x and x' in the figure, closing the circle, or two end vertices, defining a linear chromosome.

Note that these considerations are independent of the properties of the alternating cycle containing v in the breakpoint graph, which involves edges determined by *both* genomes R and B.

4 How Many Paths and Cycles?

In Section 3, we discussed the structure of the individual genomes. We now examine the structure of the breakpoint graph determined by the two genomes.

4.1 The Case of No Caps – Circular Chromosomes

The combinatorial calculations that produce the well-known result (e.g., [1], p. 72, example 5.6 and p. 86, example 6.3) that the expected number of cycles in a random

permutation is $\sum_{i=1}^{n} 1/i$ extend directly to prove that in the breakpoint graph of the relaxed model of a random genome without caps, the expected number of cycles κ is

$$E(\kappa) = \sum_{i=1}^{n} \frac{1}{2i-1}. \tag{4.1}$$

In [11], we discussed the asymptotic formula

$$E(\kappa) \sim \log 2 + \frac{\gamma}{2} + \frac{1}{2}\log n, \tag{4.2}$$

where $\gamma = \lim_{n\to\infty}\left[\sum_{i=1}^{n} \frac{1}{i} - \log n\right] = 0.577\ldots$ is Euler's constant. We also cited the partial mathematical results in [6][1] and carried out simulations, both of which indicate that (4.1) and (4.2) also hold true without the relaxation, i.e. where not only the red, but also the black, genome consists of a *single* DNA circle, and there are no additional plasmids.

4.2 Linear Chromosomes

Where there are $\chi > 0$ linear chromosomes we can take a simplified approach to the construction of the breakpoint graph as a random ordering of $2n$ non-cap vertices and 2χ cap vertices, with alternating red and black lines connecting successive vertices. Wherever a cap appears, we delete the incident black edge and consider the cap and its erstwhile neighbour to be the end points of two paths, except for the last occurring cap, which simply terminates the last (2χ-th) path. The vertices ordered after this last cap are assumed to be on cycles rather than paths, and will be reordered in a

[1]Theorem 4 of Kim and Wormald states that given any two matchings B and R, and a third, random, matching S, the events that $B \cup S$ and $R \cup S$ are Hamiltonian (i.e., no plasmids) are asymptotically independent, under weak constraints on the number of cycles in $B \cup R$. This may be rephrased in terms of a fixed S, which we interpret as the matching linking the two ends of all the individual markers, and random B and R. Then it follows that the probabilistic structure of the set of cycles in $B \cup R$ (asymptotically) does not depend on whether B and R are Hamiltonian or not.

later step. There are some special cases that are not interpretable, such as two caps attached by a red or black edge. This corresponds to a "null" chromosome in the R or B genome, respectively, i.e., a chromosome without any markers. In other words, there will be less than χ linear chromosomes in such a genome. This represents a deviation of our simplified construction from a random breakpoint graph involving exactly χ chromosomes in each genome. As discussed in Section 5 below, for a fixed χ, this occurs with $O(n^{-1})$ probability.

What proportion of the vertices are on each path? As $n \to \infty$, the model becomes simply that of a random uniform distribution of 2χ points (the caps) on the unit interval. The probability density f_k of the difference between two order statistics x_k and x_{k+1}, representing the length of an alternating colour path, is the same for all $0 \leq k \leq 2\chi - 1$, where $x_0 = 0$:

$$f_k(y) = 2\chi(1-y)^{2\chi-1}, \tag{4.3}$$

with mean $1/(2\chi+1)$ and variance $\chi/[(2\chi+1)^2(\chi+1)]$. The probability density of the last order statistic, f_Σ, representing the sum of the lengths of all 2χ paths, is

$$f_\Sigma(y) = 2\chi y^{2\chi-1}, \tag{4.4}$$

with mean $2\chi/(2\chi+1)$ and variance $\chi/[(2\chi+1)^2(\chi+1)]$.

Recall that ψ is the number of paths having at least one cap. In our model, the proportion of such paths is $\frac{3}{4}$, i.e., the proportion with two caps ($\frac{1}{4}$) plus the proportion with one cap ($\frac{1}{2}$), so that the expected value of ψ is

$$E[\psi] = \frac{3}{2}\chi. \tag{4.5}$$

A derivation of the variance,

$$\mathrm{Var}(\psi) = \frac{\chi}{8}, \tag{4.6}$$

is given in reference [14].

4.3 Cycles

Let $\kappa_{n,\chi}$ be the number of cycles in the breakpoint graph. The proportion of the genomes that is in cycles is just what is left over after the paths are calculated, namely $1 - x_{2\chi}$. We ignore the initial linear ordering of these remaining vertices and instead simply calculate the number of cycles expected to be constructed from two random circular genomes with $(n+\chi)(1-x_{2\chi})$ markers, namely

$$E[\kappa_{n,\chi}(x_{2\chi})] = \sum_{i=1}^{(n+\chi)(1-x_{2\chi})} \frac{1}{2i-1} \tag{4.7}$$

from (4.1), ignoring the negligible effect of a non-integer limit of summation as n gets large. Thus, from (4.4), the expectation of of the random variable $\kappa_{n,\chi}$ is

$$\begin{aligned} E(\kappa_{n,\chi}) &= \int_0^1 f_\Sigma(y) E[\kappa_{n,\chi}(y)] dy \\ &= \int_0^1 2\chi y^{2\chi-1} \sum_{i=1}^{(n+\chi)(1-y)} \frac{1}{2i-1} dy \\ &= \int_0^1 2\chi y^{2\chi-1} \left(\sum_{j=1}^{2(n+\chi)(1-y)} \frac{1}{j} - \sum_{i=1}^{(n+\chi)(1-y)} \frac{1}{2i} \right) dy. \end{aligned} \tag{4.8}$$

On any fixed interval $[0, Y], Y < 1$, as n increases, the integrand is uniformly approximated by

$$g(n, y) = 2\chi y^{2\chi-1}[\log 2 + \frac{1}{2}\gamma + \frac{1}{2}\log(n + \chi)(1 - y)], \tag{4.9}$$

based on Young's bounds [16]:

$$\frac{1}{2r - 1} < \sum_{i=1}^{r} \frac{1}{i} - \log r - \gamma < \frac{1}{2r}. \tag{4.10}$$

Thus

$$\int_0^Y g(n, y)dy - \int_0^Y f_\Sigma(y)E[\kappa_{n,\chi}(y)] \to 0 \tag{4.11}$$

as $n \to \infty$. But since

$$\begin{aligned} \lim_{Y \nearrow 1} \int_0^Y g(n, y)dy &= \int_0^1 g(n, y)dy \\ &= \log 2 + \frac{1}{2}\left[-\sum_{i=1}^{2\chi} \frac{1}{i} + \gamma + \log(n + \chi)\right], \end{aligned} \tag{4.12}$$

based on the identity $\int_0^1 r \log(1 - x)x^{r-1}dx = -\sum_{i=1}^{r} \frac{1}{i}$, we may then conclude

$$E(\kappa_{n,\chi}) \to \log 2 + \frac{1}{2}\log\frac{n + \chi}{2\chi} \tag{4.13}$$

as $n \to \infty$.

While we will confirm the second term in (4.13) in subsequent sections, we will conclude that the $\log 2$ term is due to the difference between the order statistic-based model in Section 4.2 we have just analyzed and the original random breakpoint graph model.

5 A Recurrence for the Expected Number of Cycles

In Section 4, we derived a limiting expression for the expected number of cycles in a continuous analog of the random breakpoint graph problem, making use of order statistics on $[0, 1]$ to predict the distribution of the proportion of vertices in cycles. In effect we combined two separately derived limit results, one for the paths and one for the cycles. In this section, we derive an exact recurrence for the number of cycles for finite n, and the expectation of this number, by slightly relaxing the constraint on the number of linear chromosomes in one of the genomes.

We build the random breakpoint graph as follows. We construct a random matching R, using red edges, on the $2n + 2\chi$ labeled vertices and caps representing the markers and chromosome ends in the red genome, under the condition that no caps are matched to each other, as on the left of Fig. 2. Then we construct B as any random matching of the same $2n + 2\chi$ vertices, by adding black edges one at a time.

Consider the connected components of the graph after the addition of a number of black edges, as on the right of Fig. 2. They are either cycles (containing no caps), inner edges (paths containing no cap), cap edges (paths containing at least one cap) or composite cycles (containing caps). Let $N(\kappa, l, m)$ be the number of (equiprobable) ways graphs with κ cycles are produced by the process of adding black edges, starting with l inner edges and m cap edges[2]. Initially $m = 2\chi$ and $l = n - \chi$, composed entirely of red edges.

[2]This counts each graph or partial graph, not just once, but according to the number of times it is produced by different sequences of black edge placements

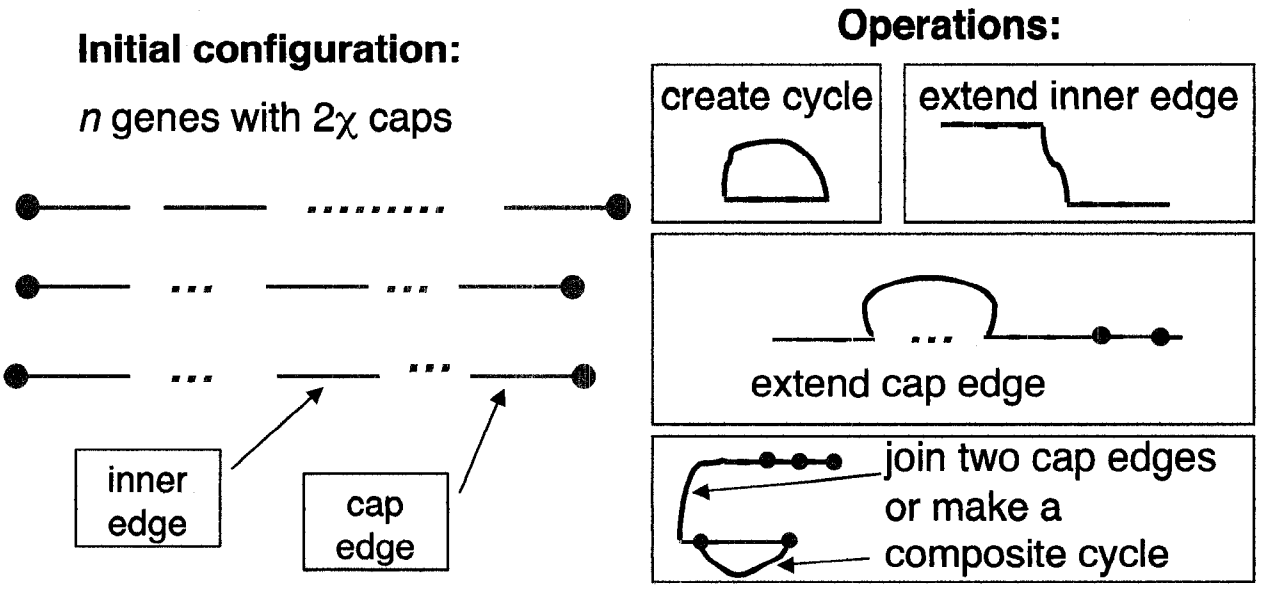


Figure 2: Left: Initial configuration of edges and caps. Right: Operations of extending inner edges or cap edges and completing cycles or paths.

Lemma 5.1.

$$\begin{aligned} N(\kappa,l,m) &= lN(\kappa-1,l-1,m) \\ &+ \left(\binom{2l}{2} - l + 4lm\right)N(\kappa,l-1,m) \\ &+ \binom{2m}{2}N(\kappa,l,m-1), \end{aligned} \tag{5.1}$$

where $N(-1,l,m) = N(\kappa,l,-1) = N(\kappa,\kappa-1,m) = 0$ *are boundary conditions for the recurrence.*

Proof. There are $\binom{2m+2l}{2}$ ways of adding a black edge:

- The number of cycles can be increased by 1 if the two ends of an inner edge are connected. This decreases l by 1 and may happen in l ways.
- Two inner edges can be connected to form one extended inner edge. Again l

decreases by 1. This can be done in $\binom{2l}{2} - l$ ways.

- One end of an inner edge is connected to a cap edge to form an extended cap edge. Again l decreases by 1. This can be done in $4lm$ ways. (N.B. An extended edge, whether inner or cap, behaves exactly as an unextended edge of the same type in this construction, so we need not specify if an edge is extended or not.)

- Two cap edges are connected or one is closed to make a composite cycle. Here $2m$ decreases by 2, and m decreases by 1. This can be done in $\binom{2m}{2}$ ways

Collecting terms gives recurrence (5.1). □

Theorem 5.1.1. *The expected number of cycles constructed, starting with l inner edges and $2m$ cap edges, is*

$$\begin{aligned} E[\kappa(l,m)] &= \frac{2m(2m-1)}{(2l+2m)(2l+2m-1)}E[\kappa(l,m-1)] \\ &\quad +[1-\frac{2m(2m-1)}{(2l+2m)(2l+2m-1)}]E[\kappa(l-1,m)] \\ &\quad +\frac{2l}{(2l+2m)(2l+2m-1)}, \end{aligned} \tag{5.2}$$

where $E[\kappa(0,m)] = E[\kappa(l,-1)] = 0$ are boundary conditions for the recurrence.

Proof. The expected number of cycles produced during the construction of matching B will be

$$\begin{aligned} E[\kappa(l,m)] &= \frac{\sum_\kappa \kappa N(\kappa,l,m)}{\sum_\kappa N(\kappa,l,m)} \\ &= \frac{\sum_\kappa \kappa N(\kappa,l,m)}{\prod_{i=1}^{l+m}\binom{2i}{2}} \end{aligned} \tag{5.3}$$

since there are $\prod_{i=1}^{l+m}\binom{2i}{2}$ ways of adding black edges until the number of inner edges

and the number of cap edges are both zero.

The theorem follows directly from (5.3) and Lemma 5.1. □

Note that the matching B includes a black edge incident to each cap, whereas a breakpoint graph contains no such edges. In the construction, however, these black edges are all in cap edges or composite cycles, and may simply be deleted without affecting the number of inner cycles or its expectation. The affected cap edges or composite cycles just decompose into one or more paths. However, if a black edge connects two caps, this corresponds to a "null" chromosome in the B genome, i.e., one without any markers. In other words, there will be less than χ linear chromosomes in the B genome. For a fixed χ, this occurs with $O(n^{-1})$ probability. Just as with the inevitability of "plasmids" in genome B as discussed in Section 3 and to be further detailed in Section 10 below, this does not detract from the exactness of our result, only from the correspondence between our model and the strict comparison of two random genomes with exactly χ chromosomes, all of which are linear.

When this construction is completed, we can delete the black edges incident to caps to reveal the linear paths in the breakpoint graph. In the rare $(O(\frac{1}{n}))$ case that a black edge directly connects two caps, there is one less chromosome in B, so that we cannot claim that equation (5.2) is an exact solution for the case of χ black chromosomes, except in the limit.

6 Limiting Behavior of $E[\kappa(n, \chi)]$

Motivated by equation (4.13), if we calculate $E[\kappa(l, m)]$ for a large range of values of l and m, we find that to a very high degree of precision, the values fit

$$E[\kappa(n, \chi)] = \frac{1}{2} \log \frac{n + \chi}{2\chi}, \tag{6.1}$$

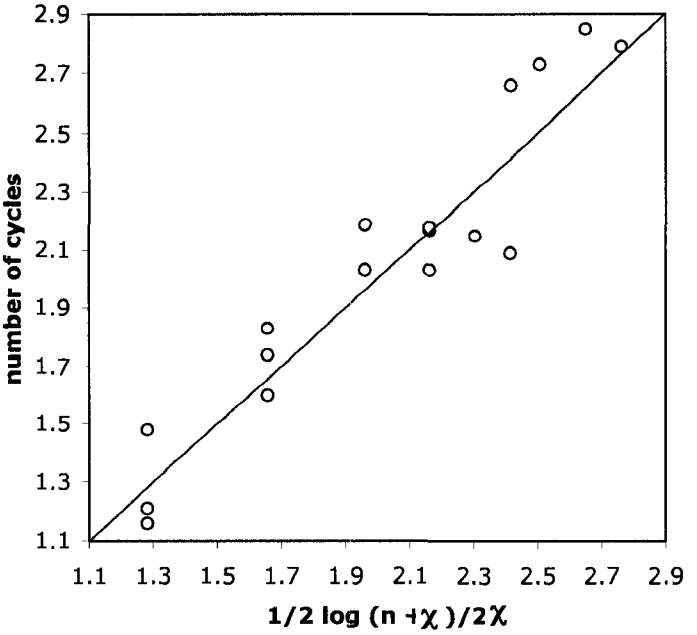


Figure 3: Simulations for $\chi = 20$ and n ranging from 500 to 10,000. Each point represents the average of 100 pairs of random genomes.

without the $\log 2$ term in equation (4.13).

Furthermore, when we simulate 100 pairs of random genomes with 20 chromosomes, for a large range of values of n, using a strictly ordered model rather than the relaxed models in Sections 4-5 above, and count the number of cycles in their breakpoint graphs, the average trend corresponds well to equation (6.1). This is seen in Figure 3.

We rewrite recurrence (5.2) for $t(l,m) = E[\kappa(l,m)]$ as

$$\begin{aligned} t(l,m) - t(l-1,m) &= \frac{2m(2m-1)}{(2l+2m)(2l+2m-1)}[t(l,m-1) - t(l-1,m)] \\ &\quad + \frac{2l}{(2l+2m)(2l+2m-1)}. \end{aligned} \tag{6.2}$$

As l and m both increase, the formula $t = \frac{1}{2}\log\frac{l+m}{m} + C$ approximates a solution of (6.2), since the first-order approximations $\frac{1}{2}\frac{1}{l+m}$ and $\frac{1}{2}\frac{1}{m}$ for the difference terms

$t(l,m) - t(l-1,m)$ and $t(l,m-1) - t(l-1,m)$ on the left and right hand sides, respectively, satisfy (6.2) exactly. Recall that initially, $l = n - \chi$ and $m = 2\chi$, so that $t = \frac{1}{2}\log\frac{n+\chi}{2\chi} + C$.

Comparison with the boundary condition $\chi = n$, where each chromosome in our construction starts with two cap edges and no inner edges, so that $\kappa \equiv 0$, further reinforces the computationally suggested value of $C = 0$.

7 Differential Rates of Inversion and Translocation

The models we have been investigating assume that adjacencies between vertices are randomly established in one genome independently of the process in the other genome. For multichromosomal genomes, this means that the probability that any particular pair of adjacent vertices in the black genome are on the same chromosome in the red genome is of the order of χ^{-1}. This suggests that there are far fewer intrachromosomal exchanges during evolution than interchromosomal, in the approximate ratio of χ^{-1} : 1, which, in the mammalian case, comes to about 0.05 : 1, a tiny minority. In point of fact, intrachromosomal processes such as inversion represent not a minority, but a clear majority of evolutionary events.

Table 1 gives the estimated ratio of intrachromosomal events to interchromosomal events among six vertebrate species. The estimator is based on the number of synteny blocks on each B genome chromosome compared to the number of different chromosomes in the R genome represented among these blocks – more different chromosomes in R for a given number of synteny blocks on the same B chromosome result in a higher estimate of translocations, while more synteny blocks on the B chromosome involving the same chromosomes in R result in a higher estimate of inversions [7]. This ratio in Table 1 depends on the resolution of the synteny block evidence

used to estimate the events; at finer resolutions than the 1 Mb used for the table, the ratio increases considerably. Even for the mouse-dog comparison the ratio is more than 1 at a 300 Kb resolution, while most of the other comparisons have a $2:1$ ratio or more.

What is the importance of this tendency for our theoretical analysis? In the break-

$B \setminus R$	human	mouse	chimp	rat	dog	chicken
human	\	1.2	-	1.6	1.7	2.9
mouse	1.3	\	1.1	2.3	0.7	1.3
chimp	15	1.4	\	-	-	-
rat	1.5	1.7	-	\	-	-
dog	1.9	0.7	-	-	\	-
chicken	4.5	1.8	-	-	-	\

Table 1: Ratio of intrachromosomal events to interchromosomal ones, at a resolution of 1Mb. Calculated from estimates in [7]. Asymmetries between $B\backslash R$ and $R\backslash B$ due to construction of primary data sets in the UCSC Genome Browser and to asymmetry in the estimator used.

point graph, the number of adjacent vertex pairs in one genome that are on different chromosomes in the other is a good indication of the number of translocations among pairs of chromosomes, though there is no simple mathematical connection. Furthermore, the number of edges connecting, for example, vertices on the same R genome chromosome to vertices on different B genome chromosomes, is a property of the breakpoint graph that we can easily influence in our model. For example, in our derivation of the recurrence (5.2) in Section 5, we could divide the end vertices of inner edges into χ classes corresponding to the χ chromosomes, as in Figure 4. Then by adjusting the relative probabilities of choosing intra-class edges versus inter-class edges, we can indirectly model differing proportions of inversions versus translocations. The removal of the simplifying assumption of equiprobable edge choice, however, would greatly complicate the analysis leading up to (5.2) and hence to (6.1).

Leaving the theoretical aspects open, then, we propose a simulation approach to the

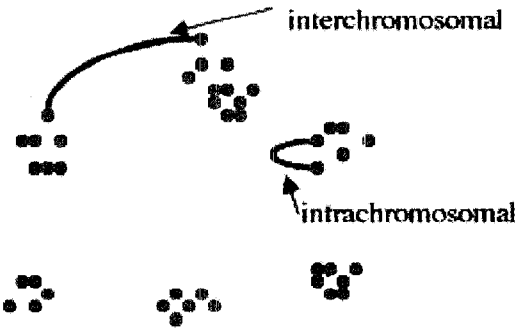


Figure 4: Partitioning vertices into classes according to chromosomes in genome R. Two kinds of edges with differing probabilities, corresponding roughly to inversion versus translocation rates.

question of the how the inversion-translocation ratio affects the breakpoint graph. For this simulation, our choice of parameters is inspired by the human-mouse comparison with 270 autosomal synteny blocks at a resolution of 1 Mb [7]. For simplicity we set $\chi = 20$ in both genomes as a compromise between the 19 of mouse and the 22 of human. We wish to rearrange the genomes so that the genomic distance between them is about 240. It requires 405 random rearrangements for the algorithm to infer 240, since with such large distances, the algorithm finds a reconstruction that is shorter than the true rearrangement trajectory. It goes without saying that there will be little relation between the operations inferred by the algorithm and the operations actually producing the genomes.

We initialized the simulations with a genome having a distribution of chromosome sizes, in terms of numbers of blocks, patterned roughly after the human genome when it is compared to the mouse genome. We then used random inversions and random translocations to simulate the mouse genome. The translocations were conditioned not to result in chromosomes smaller than a certain threshold or larger than a certain

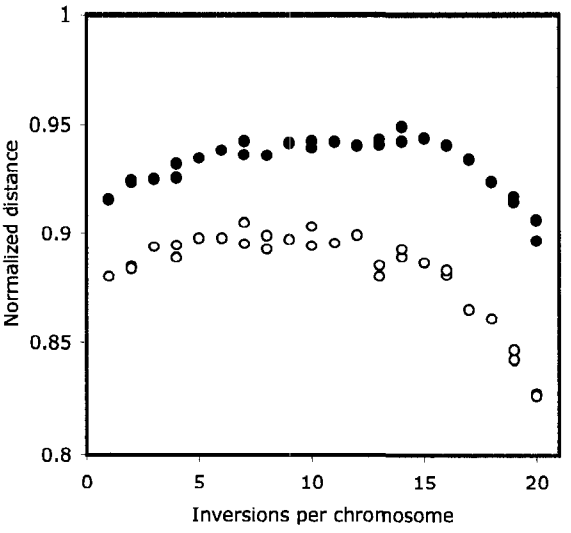


Figure 5: Effect of changing inversion-translocation proportions. Open dots: before discarding 2-cycles. Filled dots: after discarding 2-cycles.

maximum.

We sampled 10 runs with r inversions per chromosome and $405 - 20r$ translocations, for each $r = 1, \ldots, 20$. In Figure 5, we show that the average inferred distance (normalized by dividing by 270, the number of blocks) rises slowly with the increasing proportion of inversions, then falls precipitously as translocations became very rare. One artifact in this result is due to "2-cycles", representing genes that are adjacent in both genomes. In the breakpoint graphs of random genomes, 2-cycles occur rarely; the expected number of them has a limiting probability of $\frac{1}{2}$. And there are no 2-cycles in breakpoint graphs created from real genome sequence data. (If two synteny blocks were adjacent and in the same orientation in both genomes, they would simply be amalgamated and treated as a single, larger, block.) Breakpoint graphs created from random inversions and translocations, however, will tend to retain some 2-cycles even after a large number of operations. It takes a very large number of operations before we can be sure that all adjacencies will be disrupted. The effect of

these remaining 2-cycles is to decrease the distance in a way irrelevant to our interest in comparing synteny in real genomes (with no 2-cycles) to random genomes (with virtually no 2-cycles). For the sake of comparability, therefore, we should discard all two cycles and reduce n by a corresponding amount. This done in Figure 5 and it does reduce somewhat the variability of the normalized distance with respect to the inversion-translocation proportion, because the number of 2-cycles rises from about 10 per run when there are few inversions per chromosome, to more than 20 per run when there are 19 or 20 inversions per chromosome, and very few translocations.

Nevertheless, even when 2-cycles are purged, there remain two clear effects, an initial rise in the genomic distance, which will not discuss here, and a larger drop in the distance when nearly all the operations are inversions. This drop is largely accounted for by an increase in the number of cycles from an average of 2 per run when there are less than 15 inversions per chromosome to 10 cycles per run, when there are 20 inversions per chromosome. To explain this, we observe that insofar as translocations do not interfere, the evolution of the genomes takes place as if each chromosome was evolving independently on is own. But from (6.1), when there are only inversions and no translocations, we could then expect about $\frac{1}{2}\log(\frac{1}{2} + 270/40) \sim 1$ cycle per chromosome or 20 for the whole genome. In our simulations, when there are 20 inversions per chromosome, there remain a total of only 5 translocations. Were these translocations removed from our simulation, we could extrapolate a further increase of almost 10 in the number of cycles, as predicted by (6.1).

8 Variance

To test whether the comparison of two genomes reveals anything non-random about the order of synteny blocks on the chromosomes of the genomes, we need not only the

expected distance between two genomes of a given size and number of markers, but also the variance. The expected distance, based on (2.1) can be found by using (4.2) in Section 4.1 or (4.5) and (6.1) in Section 6. The variance of ψ is given by (4.6) and the variance of κ, found using other means in [14], is:

$$\text{Var}(\kappa) = \log 2 + \frac{1}{2}(\gamma + \log n) - \frac{\pi^2}{8}, \text{ for } \chi = 0; \tag{8.1}$$

$$\text{Var}(\kappa) = \frac{1}{2}\log\frac{(n+\chi)}{2\chi} - \frac{1}{8\chi}, \text{ for } \chi > 0. \tag{8.2}$$

9 Unequal χ

Allowing different numbers of chromosomes χ_R and χ_B in the red and black genomes, respectively, the procedures of Section 5 and 6 have been shown [14] to result in the limiting result:

$$E[\kappa(n, \chi_R, \chi_B)] = \frac{1}{2}\log\frac{n + \max[\chi_R, \chi_B]}{\chi_R + \chi_B}. \tag{9.1}$$

10 Number of plasmids

In our relaxed model, we allow the random black genome to contain circular plasmids in addition to the χ linear chromosomes desired. Calculations of the distribution of the number Π of these plasmids is similar to the calculation of the number of cycles κ in the breakpoint graph. This can be seen by considering *white* edges connecting the two vertices of a marker or gene in B as playing an analogous role to the red edges from genome R in the breakpoint graph. Only small modifications to the previous derivation result in:

$$E[\Pi(n, \chi)] = \frac{1}{2}\log\frac{n}{\chi}, \tag{10.1}$$

and

$$\mathrm{Var}(\Pi) = \frac{1}{2}\log\frac{n}{\chi} - \frac{1}{4\chi}. \tag{10.2}$$

11 Discussion

We have continued the development of probabilistic models of random genomes, with a view to testing the statistical significance of genome rearrangement inferences. Here, we have focused on the breakpoint graphs of multichromosomal genomes and found that the limiting expectation of the distance between two random genomes, based on equation (2.1), is $n - \frac{1}{2}\chi - \frac{1}{2}\log\frac{n+\chi}{2\chi}$, with variance $\frac{\chi^2}{8} + \frac{1}{2}\log\frac{(n+\chi)}{2\chi} - \frac{1}{8\chi}$.

A test based on these quantities, however, should be considered preliminary, for two reasons. First, our random breakpoint graphs imply exaggerated rates of translocations, compared to inversions. We have explored a more realistic problem, how to generate random breakpoint graphs reflecting differential rates of inversion and translocation. Our simulations show that the cycle structure of these graphs is sensitive to this differential and so analytical work on this problem is important to the eventual utility of our approach in testing the significance of rearrangement inferences. Second, this kind of test is too powerful, sensitive to small deviations from randomness. Thus where part of the rearrangement trajectory between two genomes inferred by an algorithm is unequivocal and part is uncertain, the test may reject the null hypothesis of randomness, leading perhaps to the incorrect conclusion that the entire inferred trajectory is historically correct. It is advisable instead to consider details of the cycle structure in the breakpoint graphs of the real and random genome pairs to see where any departure from randomness occurs, as illustrated in [10].

Acknowledgements

Research supported in part by grants from the Natural Sciences and Engineering Research Council of Canada (NSERC). DS holds the Canada Research Chair in Mathematical Genomics and is a Fellow of the Evolutionary Biology Program of the Canadian Institute for Advanced Research. We grateful to the referees for their careful reading and constructive criticism.

References

[1] Billingsley, P. 1995. Probability and measure, 3rd edition. New York: Wiley-Interscience. Patrick Billingsley

[2] Eriksen, N. and Hultman, A. 2004. Estimating the expected reversal distance after a fixed number of reversals. Advances in Applied Mathematics 32, 439–453.

[3] Hannenhalli, S. and Pevzner, P. A. 1995. Transforming cabbage into turnip (polynomial algorithm for sorting signed permutations by reversals). Proceedings of the 27th Annual ACM Symposium on the Theory of Computing, 178–189 (full version in Journal of ACM 46, 1–27, 1999).

[4] Hannenhalli, S. and Pevzner, P. A. 1995. Transforming men into mice (polynomial algorithm for genomic distance problem. Proceedings of the 36th Annual IEEE Symposium on Foundations of Computer Science, 581–592. Milwaukee, WI.

[5] Kececioglu, J. and Sankoff, D. 1994. Efficient bounds for oriented chromosome inversion distance. Combinatorial Pattern Matching. Fifth Annual Symposium (Crochemore, M. and Gusfield, D., eds.) Lecture Notes in Computer Science 807, Springer, 307–325.

[6] Kim, J.H. and Wormald, N.C. 2001. Random matchings which induce Hamilton cycles, and Hamiltonian decompositions of random regular graphs, Journal of Combinatorial Theory, Series B 81, 20–44.

[7] Mazowita, M., Haque, L. and Sankoff, D. 2006. Stability of rearrangement measures in the comparison of genome sequences. Journal of Computational Biology 13, 554–566.

[8] Sankoff, D. 1989. Mechanisms of genome evolution: models and inference. Bulletin of the International Statistical Institute 47.3, 461–475.

[9] Sankoff, D. 1992. Edit distance for genome comparison based on non-local operations. In Combinatorial Pattern Matching. Third Annual Symposium (Apostolico, A., Crochemore, M., Galil, Z., Manber, U., eds.), Lecture Notes in Computer Science 644, Springer, 121–135.

[10] Sankoff, D. 2006. The signal in the genomes. PLoS Computational Biology 2, e35.

[11] Sankoff, D. and Haque, L. 2006. The distribution of genomic distance between random genomes. Journal of Computational Biology 13, 1005–1012.

[12] Tesler, G. 2002. Efficient algorithms for multichromosomal genome rearrangements. Journal of Computer and System Sciences 65, 587–609.

[13] Waterston, G., Ewens, W., Hall, T. and Morgan, A. 1982. The chromosome inversion problem. Journal of Theoretical Biology 99, 1–7.

[14] Xu, W. 2007. The distance between randomly constructed genomes. Submitted.

[15] Yancopoulos, S., Attie, O. and Friedberg, R. 2005. Efficient sorting of genomic permutations by translocation, inversion and block interchange. Bioinformatics 21, 3340 – 3346

[16] Young, R. M. 1991. Euler's constant. Mathematical Gazette 75, 187–190.

Chapter 3

The distance between randomly constructed genomes

Wei Xu. Proceedings of the 5th Asia-Pacific Bioinformatics Conference. David Sankoff, Lusheng Wang and Francis Chin (Eds.) 227-236. 2007. Work of Wei Xu.

Abstract

In this paper, we study the exact probability distribution of the number of cycles c in the breakpoint graph of two random genomes with n genes or markers and χ_1 and χ_2 linear chromosomes, respectively. The genomic distance d between the two genomes is $d = n - c$. In the limit we find that the expectation of d is $n + \max(\chi_1, \chi_2) - \frac{2\chi_1\chi_2}{2\chi_1+2\chi_2-1} - \frac{1}{2}\ln\frac{n+\min(\chi_1,\chi_2)}{\chi_1+\chi_2}$.

1 Introduction

The study of genome rearrangements has developed a sophisticated technology for inferring a minimizing sequence of operations necessary to transform one genome into another, where the genomes are represented by signed permutations on $1, \cdots, n$ and the operations are modeled on the biological processes of inversion, reciprocal translocation, chromosome fusion and fission, transposition of chromosomal segments, excision and reintegration of circular chromosomal segments, among others. Once these inferences are made, however, there is a need for some way to statistically validate both the inferences and the assumptions of the evolutionary model.

Our approach has been to see to what extent there is an signal remaining in the comparative structure of the two genomes, or whether evolution has largely scrambled the order of each one with respect to the other, in terms of the evolutionary model assumed. This has led to the study of completely scrambled, i.e., randomized, genomes as a null baseline for the detection of a evolutionary signal. Insofar as a pair of genomes retain some evidence of evolutionary relationship, this should be detectible by contrast to randomized genomes. In previous papers, we have worked out the statistical properties of random genomes consisting of one or more circular chromosomes, [1] and those of two random genomes containing the same number c of linear chromosomes.[2]. The latter paper concentrated on showing

that the number of circular chromosomes inevitably associated with random linear chromosomes is very small with realistic numbers of chromosomes. It only included a rough estimation of the statistical properties of the linear chromosomes.

The present paper introduces a new way of representing the comparison of linear genomes, requiring only a single source/sink vertex in the breakpoint graph of the two genomes, instead of the numerous "chromosomal caps" used in other treatments. This facilitates a more rigorous treatment of the case of linear chromosomes, including the more realistic situation where the number of linear chromosomes may be different (χ_1 and χ_2) in the two genomes being compared.

2 Genome rearrangement with linear chromosomes

In our framework, each genome consists of n markers (genes, chromosomal segments, etc.), divided among a number of disjoint chromosomes. We fix the number of linearly ordered chromosomes, but in our construction of random genomes we will permit some additional, circularly ordered, chromosomes as well. In graph-theoretical terms, we usually represent each marker by two distinct vertices, marking the beginning and end of the marker, respectively. We call all of these *inner vertices*. For each linear chromosome, two extra vertices, named *caps* are added to represent the ends of the chromosome. In comparing two genomes containing different numbers χ_1 and χ_2 of linear chromosomes, we equalize their numbers at $\chi = \max(\chi_1, \chi_2)$ by adding an appropriate number of null chromosomes, each of which consists only of two caps, to one of the genomes.

2.1 The Breakpoint Graph

When two genomes, say a red one and a black one, containing the same n markers, are compared, we use red edges to connect the the nearest vertices of two adjacent markers according to their order in the red genome – this may be the end of one mark and the beginning of the other, or two ends, or two beginnings, depending on the orientation or "strandedness" of the markers on the chromosome. The first and last inner vertices are connected to caps. Each cap may only be connected to one inner vertex. We also connect the two caps of any null chromosome in the red genome by a red edge. Similarly, we use black edges to connect the vertices and caps in the black genomes. There are thus $2n$ inner vertices, 2χ caps, $n + \chi$ red edges and $n + \chi$ black edges in the graph.

Since each vertex is connected to one red and one black edge (one adjacency in each genome), a 2-regular graph is formed. A 2-regular graph always can be decomposed into number of cycles c, and in our bicoloured graph, the edge colours alternate around each cycle. Yancopoulos, Attie and Friedberg[3] showed that the edit distance d is related to the number of cycles c by

$$d = n + \chi - \max c, \tag{2.1}$$

when block interchanges (each counting as two operations) are allowed besides inversions and reciprocal translocations. The number of cycles depends on which red chromosome and which black chromosome are incident to the same cap, a choice which is left free in the graph definition. The maximal number of cycles in equation (2.1) refers to the optimal choice of this cap assignment. We refer to this particular graph as the breakpoint graph of the two genomes.

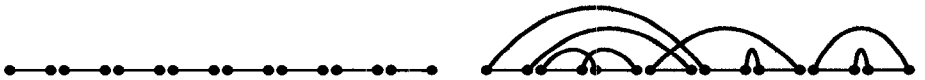


Figure 1: The construction of a random breakpoint graph. We start with the red genome, represented by a set of cap edges (in blue) and a set of inner edges (in red), and add the black edges randomly, one by one, until every vertex is connected by one black edge. In (b) there are 3 cycles. Caps denoted by blue dots and inner vertices by black ones.

2.2 Random Genomes

Were we to construct genomes by successively adding markers or caps in random order, it would be very difficult to say anything precise about the breakpoint graph, because the linearity condition on chromosomes induces great complexity to the events whose probabilities we wish to calculate. Instead, we introduce the randomness directly in the construction of the breakpoint graph, leading to simple expressions for probabilities of the sizes and numbers of cycles. This simplicity comes at a cost, however, since the construction of a random genome at the level of the breakpoint graph does not exclude some circular chromosomes. As we shall mention later, there is good reason to believe that this feature does not affect our results on the limits of expectations.

To obtain two genomes randomized with respect to each other, it suffices to fix the gene order in one of them, say the red genome, and to introduce randomness into the black genome only. Because we are interested in calculations pertaining to the breakpoint graph, we simply postulate that at each step a black edge may be added to connect any two inner vertices that are not already incident to black edges. We do not at this stage really connect caps to inner vertices using black edges, because these edges are implicitly determined by the cycle optimization procedure applied to the rest of the graph. Thus we start with $2n + 2\chi$ vertices (inner vertices and caps), with red edges connected. We distinguish between two kinds of edges: 2χ *cap edges* incident to a cap and $n - \chi$ *inner edges* not

incident to a caps. To construct the random breakpoint graph, we connect two inner vertices at random by a black edge until every vertex is incident to a black edge. Note that in randomly adding black edges we are not guaranteed to end up with linear chromosomes, since there is the possibility that the black genome so constructed will contain one or more circular chromosomes, with no caps. As χ becomes large, the number of such circles and the number of markers in them, will be small. Nevertheless this possibility is not part of the original problem involving two random genomes with linearly ordered chromosomes. Fortunately, partial mathematical results indicate that in the limit, the possible presence of circular chromosomes does not affect the probability structure of the breakpoint graph[4].

2.3 Cap Optimization

In the procedure of cap optimization, the breakpoint graph is decomposed into cycles and 2χ paths (whose two ends are caps or inner vertices incident to only one cap edge). The ψ_{HO} *homogenous paths* terminate with caps via two red edges (type 1) or with two inner vertices (type 2), with an equal number of the two types, and the ψ_{HE} *heterogenous paths* end with one cap and one inner vertex. The optimization principle developed by Hannenhalli and Pevzner[5] and Tesler[6], comes down to, in the reformulation by Yancopoulos *et al.*[3], to the addition of two black edges joining one homogenous path of type one to another homogeneous path of type 2 to form a cycle and the addition of a single black edge to each heterogeneous path to form a cycle. It can be seen that the maximized cap cycle number ψ is

$$\max \psi = \chi + \frac{1}{2}\psi_{HE} \tag{2.2}$$

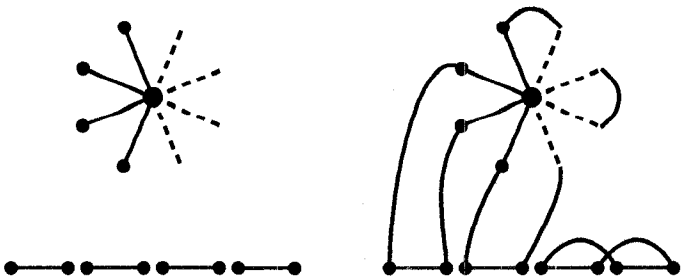


Figure 2: The illustration of the modified model. At the initial state (a), all the caps have been merged into one source/sink vertex C. The dashed black edges are reserved for the 2χ black cap edges to be added later. At the end (b), all the cap edges should be connected via inner edges, except for some that are composed of two cap edges or a single edge with C at both ends. The rest of the inner edges form the inner cycles. In the figure, two homogenous paths, two heterogenous paths and one inner cycle are depicted.

2.4 The *Flower* Representation

To facilitate the construction of the random breakpoint graph, including the cycle optimization, we abandon the regular graph representation and introduce a modified model as follows.

We replace all the caps by a single source/sink vertex C. Then we may portray the cap edges as distributed around C as in Figure 2, while the inner edges are unaffected. In Figure 2(a), there are 2χ red cap edges and the same number of dashed edges incident to C indicating where the black cap edges will eventually connect. Some same-coloured pairs of these cap edges may represent null chromosomes. The construction proceeds by adding black edges one by one at random as detailed in the next section, and terminates when a complete structure as in Figure 2(b) is achieved.

The cycles are of two sorts, those in the flower structure, named *cap cycles* and the rest, *inner cycles*. In the next section, recurrence equations will derived for both kinds. Note that each "petal" of the flower, connected to the source/sink vertex, represents a path, either

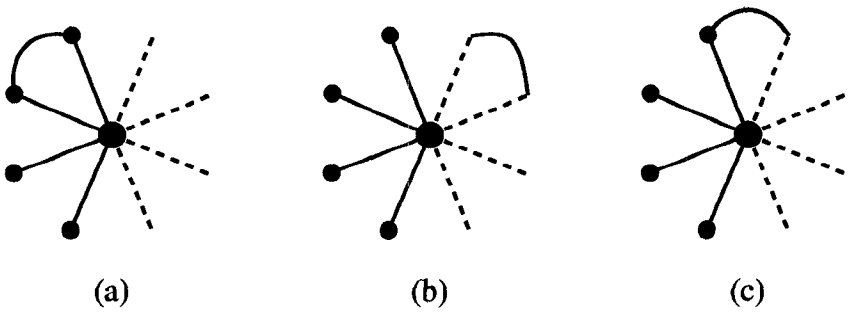


Figure 3: The three possible ways of completing a cap path. Homogenous paths are shown in (a) and (b) and a heterogenous path in (c).

a homogenous or heterogeneous. The cap cycles are not explicitly depicted in the graph. Their total number is determined by the capping optimization formula.

3 The Recurrence Equations

3.1 The Number of Heterogenous Paths ψ_{HE}

From the cap optimization principle, the number of cap cycles should be equal to $\chi + \frac{1}{2}\psi_{HE}$.

During the construction, at each step it suffices to keep track only of the number of extended red cap edges, where this includes paths with C connected to a red edge at one end and a red edge at the other, the number of extended black cap edges, where this includes either dashed edges of paths with C connected to a black edge at one end and a red edge at the other.

We start from a general situation where there are r extended red cap edges and s extended black cap edges. The problem is denoted as (r, s).

At each step, one black edge is added, connecting two extended red cap edges, two dashed

or extended black cap edges or one extended red cap edge and one dashed or extended black cap edge. Once a path forms, the total number of extended paths (i.e., the edges that remain to be connected) decreases by 2. The three possible ways of adding the black edge lead to the smaller problems $(r-2, s)$, $(r, s-2)$, $(r-1, s-1)$, respectively. The numbers of ways of doing each are $\binom{r}{2}$, $\binom{s}{2}$ and rs, respectively. Only in the last situation is a heterogenous path completed. Denote $\mathbf{n}(\psi_{HE}, r, s)$ as the total number of ways to get a breakpoint graph with ψ_{HE} heterogenous paths for the (r, s) problem. Since each problem with size (r, s) can be constructed from three smaller problems of sizes $(r-2, s)$, $(r, s-2)$ and $(r-1, s-1)$, respectively, we have the recurrence :

$$\mathbf{n}(\psi_{HE}, r, s) = \binom{r}{2} \mathbf{n}(\psi_{HE}, r-2, s) + \binom{s}{2} \mathbf{n}(\psi_{HE}, r, s-2) + rs\, \mathbf{n}(\psi_{HE}-1, r-1, s-1) \tag{3.1}$$

Denote by $\bar{\psi}_{HE}(r, s)$ the average number of heterogenous paths in the breakpoint graph for (r, s), defined as:

$$\begin{aligned} \bar{\psi}_{HE}(r, s) &= \frac{\sum_{\psi_{HE}=0}^{min(r,s)} \mathbf{n}(\psi_{HE}, r, s)\, \psi_{HE}}{\sum_{\psi_{HE}=0}^{min(r,s)} \mathbf{n}(\psi_{HE}, r, s)} \\ &= \frac{\sum_{\psi_{HE}=0}^{min(r,s)} \mathbf{n}(\psi_{HE}, r, s)\, \psi_{HE}}{\prod_{i=0}^{\frac{r+s}{2}} \binom{r+s-2i}{2}} \end{aligned}$$

where $\sum_{\psi_{HE}=0}^{min(r,s)} \mathbf{n}(\psi_{HE}, r, s) = \prod_{i=0}^{\frac{r+s}{2}} \binom{r+s-2i}{2}$ is the total number of ways to construct the breakpoint graph.

By summing over equation (3.1), we get the recurrence equation for the average num-

ber of heterogenous paths.

$$\bar{\psi}_{HE}(r,s) = \frac{\binom{r}{2}\bar{\psi}_{HE}(r-2,s) + \binom{s}{2}\bar{\psi}_{HE}(r,s-2) + rs\bar{\psi}_{HE}(r-1,s-1) + rs}{\binom{r+s}{2}} \quad (3.2)$$

Equation (3.2) has a probabilistic interpretation, since (r,s) can be decomposed into $(r-2,s)$, $(r,s-2)$ and $(r-1,s-1)$ with probabilities $\frac{r(r-1)}{(r+s)(r+s-1)}$, $\frac{s(s-1)}{(r+s)(r+s-1)}$ and $\frac{2rs}{(r+s)(r+s-1)}$, respectively.

3.2 The Number of Inner Cycles

The number of the inner cycles depends on the number of inner edges not used by the paths. Suppose we start with $2m$ extended cap edges (the extended red cap edges and the dashed or extended black edges) and l inner edges, which it will be convenient to denote (m,l).[1] In the random construction of the breakpoint graph, each addition of one black edge can lead to four different situations:

1. two cap edges are connected – there are $\binom{2m}{2}$ ways of doing this – and the size of the problem becomes $(m-1,l)$

2. one cap edge and one inner edge are connected – there are $4ml$ ways of doing this – and the size of the problem becomes $(m,l-1)$

3. two different inner edges are connected – there are $2l(l-1)$ ways of doing this – and the size of the problem becomes $(m,l-1)$

4. the two ends of the same inner edges are connected – there are l ways of doing this. The size of the problem becomes $(m,l-1)$ and the number of inner cycles increases by one.

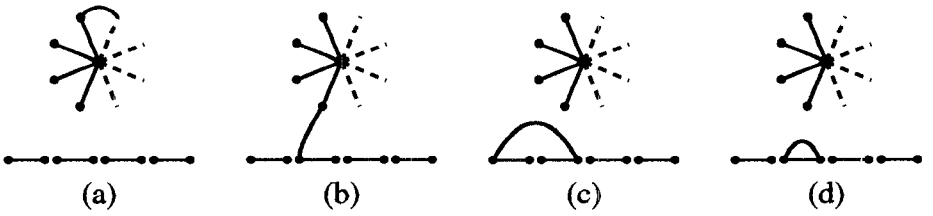


Figure 4: The four possible ways to build a black edge in counting the inner cycles. Two cap edges are connected (a); one cap edge and one inner edge are connected (b); two different inner edges are connected (c); the two ends of the same inner edge are connected (d). And only in the last case, one inner cycle is formed.

Denote by $\mathbf{n}(\kappa, m, l)$ the number of ways to get a breakpoint graph with κ inner cycles for a (m, l) problem. Similarly define $\bar{\kappa}(m, l)$ as the average number of inner cycles for the problem (m, l)

$$\begin{aligned}\bar{\kappa}(m,l) &= \frac{\sum_{\kappa=0}^{l} \mathbf{n}(\kappa,m,l)\,\kappa}{\sum_{\kappa=0}^{l} \mathbf{n}(\kappa,m,l)} \\ &= \frac{\sum_{\kappa=0}^{l} \mathbf{n}(\kappa,m,l)\,\kappa}{\prod_{i=0}^{m+l}\binom{2m+2l-2i}{2}}\end{aligned}$$

We then get the corresponding recurrences

$$\begin{aligned}\mathbf{n}(\kappa,m,l) &= \binom{2m}{2}\mathbf{n}(\kappa,m-1,l) + [4ml+2l(l-1)]\,\mathbf{n}(\kappa,m,l-1) \\ &+ l\,\mathbf{n}(\kappa-1,m,l-1) \qquad (3.3)\end{aligned}$$

$$\bar{\kappa}(m,l) = \frac{\binom{2m}{2}\bar{\kappa}(m-1,l) + [4ml+2l(l-1)]\,\bar{\kappa}(m,l-1) + l\,\bar{\kappa}(m,l-1) + l}{\binom{2m+2l}{2}} \qquad (3.4)$$

Equation (3.4) also has a probabilistic interpretation, associating the probabilities $\frac{2m(2m-1)}{(2m+2l)(2m+2l-1)}$, $\frac{8ml}{(2m+2l)(2m+2l-1)}$, $\frac{4l(l-1)}{(2m+2l)(2m+2l-1)}$ and $\frac{2l}{(2m+2l)(2m+2l-1)}$ with the four pos-

[1]Rather than $(2m, l)$.

sible smaller problems.

4 The solution to the problems

4.1 The cap cycles

The recurrence equations (3.2) and (3.4) enable rapid calculation of $\bar{\psi}_{HE}$ and $\bar{\kappa}$, but there is no easy way to convert them into a closed form solution.

We can, however, deduce these quantities through another combinatoric approach. The total number of ways to form any kind of flower structure is $\prod_{i=0}^{\frac{r+s}{2}} \binom{r+s-2i}{2}$. The ways to form a result with $2\psi_{HE}$ heterogenous paths (which should be always even) is

$$\begin{aligned}
\mathbf{n}(2\psi_{HE}, r, s) &= \binom{\frac{r+s}{2}}{2\psi_{HE}} \frac{(r)!}{(r-2\psi_{HE})!} \frac{(s)!}{(s-2\psi_{HE})!} \binom{\frac{r+s}{2} - 2\psi_{HE}}{\frac{r}{2} - \psi_{HE}} \\
&\quad \prod_{i=0}^{\frac{r+s}{2}} \binom{r - \psi_{HE} - 2i}{2} \prod_{i=0}^{\frac{r+s}{2}} \binom{s - \psi_{HE} - 2i}{2} \\
&= \frac{(r)!(s)!(r+s)!}{(\frac{r+s}{2})!} \frac{4^{\psi_{HE}}}{(2\psi_{HE})!(\frac{r}{2} - \psi_{HE})!(\frac{s}{2} - \psi_{HE})!}
\end{aligned} \tag{4.1}$$

Averaging over $\mathbf{n}(2\psi_{HE}, r, s)$ and rewriting $r = 2\chi_1$ and $r = 2\chi_2$, we get

$$\bar{\psi}_{HE}(\chi_1, \chi_2) = \frac{4\chi_1\chi_2}{2\chi_1 + 2\chi_2 - 1} \tag{4.2}$$

So the average number of cap cycles is

$$\psi = \max(\chi_1, \chi_2) + \frac{2\chi_1\chi_2}{2\chi_1 + 2\chi_2 - 1} \tag{4.3}$$

When $\chi_1 = \chi_2 = \chi$, it becomes $\chi + \frac{2\chi^2}{4\chi - 1}$, approaching 1.5χ as χ becomes large, confirm-

ing a result which we have previously derived in another way.[2]

4.2 The Inner Cycles

In the flower structure where the numbers of chromosomes are equal, suppose we traverse all the edges, starting with a black cap edge, and each time we visit C, we choose an outgoing edge of colour different from the incoming edge. This will order the edges as in Figure 5(a). The last edge will be a red cap edge and C will be the last vertex. We then add the edges in the inner cycles to the right of the flower structure edges.

We define the position x of an edge, as the number of edges to the left of, and including, that edge. We assume there are χ linear chromosomes in each genome. So the smallest value possible for x is 1 and the largest one is $n - \chi$. The 2χ cap edges occupy random positions in the sequence. The constraints on the model are that the last cap edge should have C on its right, the ith cap edge can only distributed from $x_{i-1} + 1$ to $n + \chi - (2\chi - i) = n - \chi + i$. Only the inner edges to the right of the variable $x_{2\chi}$, the position of the last cap edge, are in inner cycles. Once we know the distribution of $x_{2\chi}$, we then use the formula[1] for the expected number of cycles in circular genomes to calculate the number of inner cycles.

When n becomes large, we may define a continuous approximation to this construction. The x_i become the order statistics of χ uniformly distributed points on $(0, n + \chi)$. Using the distribution for the position of the x_χ, we find the expected number of inner cycles is

$$c(m = 2\chi, l = n - \chi) = \frac{1}{2} \ln \frac{\chi + n}{2\chi} + B \tag{4.4}$$

where B is some constant.[2]

Note that equation (4.4) is the asymptotic solution of equation (3.4), with $m = 2\chi, l =$

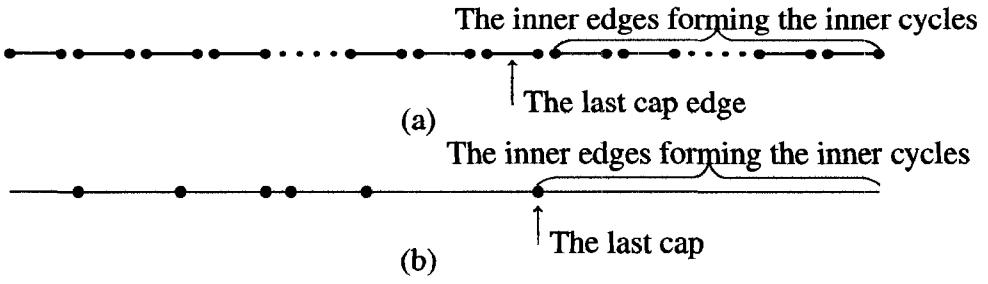


Figure 5: The exact model (a) for counting the inner cycle number and the approximate model (b). Model (a) is discrete. The cap in the last cap edge should be in the right side in order to correspond to the flower structure. Model (b) is a continuous approximation of model (a) when the number of inner edges are large enough and has no constraint on the last cap edge.

$n-\chi$. B can be found from an initial condition: when $n = \chi$, then there are no inner edges and hence no inner cycles. So $\frac{1}{2}\ln\frac{\chi+\chi}{2\chi} + B = 0$, i.e., $B = 0$. In numerical comparison as well, the equation $c = \frac{1}{2}\ln\frac{\chi+n}{2\chi}$ confirms the recurrence equation (3.4).

4.3 Two Genomes Having Different Numbers of Linear Chromosomes

Suppose the two genomes being compared have χ_1 and χ_2 linear chromosomes, respectively. We have already found the formula for the cap cycles which is $\max(\chi_1, \chi_2) + \frac{2\chi_1\chi_2}{2\chi_1+2\chi_2-1}$. For the number of inner cycles, the approximate model only deals with the case where $\chi_1 = \chi_2 = \chi$. But that solution is also the asymptotic solution for the recurrence equation (3.4), which depends only on m and l. We can thus substitute the values for m and l in the case of unequal number of linear chromosomes. Note that in the case of equality,

$m = 2\chi$ and $l = n - \chi$ and in the unequal case $m = \chi_1 + \chi_2$ and $l = n - \max(\chi_1, \chi_2)$.

$$\begin{aligned} c &= \frac{1}{2}\ln\frac{\chi+n}{2\chi} = \frac{1}{2}\ln\frac{2\chi+n-\chi}{2\chi} = \frac{1}{2}\ln\frac{m+l}{m} \\ &= \frac{1}{2}\ln\frac{\chi_1+\chi_2+n-\max(\chi_1,\chi_2)}{\chi_1+\chi_2} \\ &= \frac{1}{2}\ln\frac{n+\min(\chi_1,\chi_2)}{\chi_1+\chi_2} \end{aligned} \tag{4.5}$$

Hence in the limit the total number of cycles is

$$c = \max(\chi_1, \chi_2) + \frac{2\chi_1\chi_2}{2\chi_1 + 2\chi_2 - 1} + \frac{1}{2}\ln\frac{n+\min(\chi_1,\chi_2)}{\chi_1+\chi_2} \tag{4.6}$$

And the genomic distance d is

$$d = n - \frac{2\chi_1\chi_2}{2\chi_1 + 2\chi_2 - 1} - \frac{1}{2}\ln\frac{n+\min(\chi_1,\chi_2)}{\chi_1+\chi_2}. \tag{4.7}$$

5 Conclusion

The mathematical essence of the question with two genomes with linear chromosomes, is the number of the cycles in the 2-regular breakpoint graph whose vertices consist of a set of labeled vertices and another set of interchangeable caps vertices. We have shown that collapsing all the caps to a single source/sink facilitates the optimal capping problem as well as the calculation of cycle expectations.

The final result equation (4.7) can be applied to the comparison of two genomes with the same or different number of linear chromosomes plus any number of circular chromosomes. This is true under the condition that inversions, translocations and block interchanges are the mechanism of genomic rearrangement, where the latter count as if they were each two operations.[3]

References

[1] Sankoff, D. and Haque, L. 2006. The distribution of genomic distance between random genomes. Journal of Computational Biology 13, 1005–1012.

[2] Xu, W., Zheng, C. and Sankoff, D. 2006. Paths and cycles in breakpoint graphs of random multichromosomal genomes. Proceedings of RECOMB Satellite Conference on Comparative Genomics 2006, G. Bourque and N. El-Mabrouk, eds., Lecture Notes in Computer Science 4205. Heidelberg: Springer, 51–62.

[3] Yancopoulos, S., Attie, O. and Friedberg, R. 2005. Efficient sorting of genomic permutations by translocation, inversion and block interchange. Bioinformatics 21, 3340 – 3346

[4] Kim, J.H. and Wormald, N.C. 2001. Random matchings which induce Hamilton cycles, and Hamiltonian decompositions of random regular graphs, Journal of Combinatorial Theory, Series B 81, 20–44.

[5] Hannenhalli, S. and Pevzner, P.A. 1995. Transforming men into mice (polynomial algorithm for genomic distance problem. Proceedings of the IEEE 36th Annual Symposium on Foundations of Computer Science. 581–92.

[6] Tesler, G. 2002. Efficient algorithms for multichromosomal genome rearrangements. Journal of Computer and System Sciences 65, 587–609.

Chapter 4

The distribution of distances between randomly constructed genomes: generating function, expectation, variance and limits

Wei Xu. Journal of Bioinformatics and Computational Biology. 6(1): 23-36. 2008.
Work of Wei Xu.

Abstract

Based on a large repertoire of chromosomal rearrangement operations, the genomic distance d between two genomes with χ_r and χ_b linear chromosomes, respectively, both containing the same (or orthologous) n genes or markers, is $d = n + \max(\chi_r, \chi_b) - c$, where c is the number of cycles in the breakpoint graph of the two genomes. In this paper, we study the exact probability distribution of c. We derive the expectation and variance, and show that, in the limit, the expectation of d is $n - \frac{2\chi_r\chi_b}{2\chi_r+2\chi_b-1} - \frac{1}{2}\ln\frac{n+\max(\chi_r,\chi_b)}{\chi_r+\chi_b}$.

1 Introduction

The study of genome rearrangements has developed a sophisticated technology for inferring a minimizing sequence of operations necessary to transform one genome into another, where the genomes are represented by signed permutations on $1, \cdots, n$ and the operations are modeled on the biological processes of inversion, reciprocal translocation, chromosome fusion and fission, transposition of chromosomal segments, excision and reintegration of circular chromosomal segments, among others. [1] Once these inferences are made, however, there is a need for some way to statistically validate both the inferences and the assumptions of the evolutionary model.

Our approach has been to see to what extent there is an signal remaining in the comparative structure of the two genomes, or whether evolution has largely scrambled the order of each one with respect to the other. This has led to the study of completely scrambled, i.e., randomized, genomes as a null baseline for the detection of a evolutionary signal. Insofar as a pair of genomes retain some evidence of evolutionary relationship, this should be detectible by contrast to randomized genomes. In

[1] We do not deal directly with these operations in our analysis, but their definitions and biological pertinence are extensively discussed in the literature, e.g., in the paper of Yancopoulos *et al.*[3]

previous papers, we have studied the statistical properties of random genomes consisting of one or more circular chromosomes, [1] and those of two random genomes containing the same number χ of linear chromosomes.[2]

The present paper contains a more rigorous treatment by generating functions of the general case, where the n markers in the first genome are partitioned into χ_r linear chromosomes, while there is a possibly different number χ_b of linear chromosomes in the second genome. We derive the expectation and standard deviation for the genomic distance d – defined to be the minimum number of operations necessary to transform the first genome into the second – in the same tractable model studied previously,[1, 2] where randomness is introduced through unconditionally equiprobable adjacencies between markers on the second genome. Our results, which pertain to randomized genomes containing at most χ_b linear chromosomes and possibly a few, $O(\log \sqrt{n})$, circular chromosomes, are asymptotically (with increasing n) applicable to the less tractable case where adjacencies are constrained so that the markers are partitioned into exactly χ_b linear chromosomes.

2 The breakpoint graph based on linear chromosomes

In our framework, each genome consists of n markers (genes, chromosomal segments, etc.), divided among a number of disjoint chromosomes. In graph-theoretical terms, we represent each marker by two distinct labeled vertices, marking the beginning and end of the marker, respectively. We call all of these *inner vertices.* For each linear chromosome, two extra unlabeled vertices, named *caps* are added to represent the ends of the chromosome. In comparing two genomes containing different numbers χ_r and χ_b of linear chromosomes, we equalize their numbers at $\chi = \max(\chi_r, \chi_b)$ by

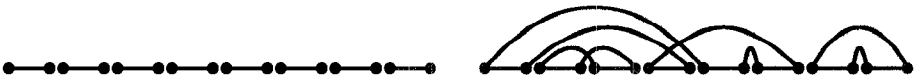
adding an appropriate number of *null chromosomes*, each of which contains only two cap vertices, to one of the genomes.

When two genomes, say a red one and a black one, containing the same n markers, are compared, we use red edges to connect the nearest vertices of two adjacent markers according to their order in the red genome – this may be the end of one marker and the beginning of the other, or two ends, or two beginnings, depending on the orientation or "strandedness" of the markers on the chromosome, as given in the biological data. The first and last inner vertices on a linear chromosomeare connected to caps. Each cap may only be connected to one inner vertex. We also connect the two caps of any null chromosome in the red genome by a red edge. Similarly, we use black edges to connect the vertices of adjacent markers in the black genome. We also connect the first and last vertices to caps with black edges, using each of the same caps as with the red genome exactly once; since the caps are unlabeled, there are many ways to carry out this last step. There are thus $2n$ inner vertices, 2χ caps, $n + \chi$ red edges and $n + \chi$ black edges in the graph.

Since each vertex is connected to one red and one black edge (one adjacency in each genome), forming a 2-regular graph, it can be decomposed into number c of cycles, with the edge colours alternating around each cycle. Yancopoulos, Attie and Friedberg[3] showed that the genomic distance d is related to the number of cycles c by

$$d = n + \max(\chi_r, \chi_b) - \max c. \tag{2.1}$$

The number of cycles depends on which red chromosome and which black chromosome are incident to the same cap, a choice that we left free in the graph definition above. The maximal number of cycles in equation (2.1) refers to the optimal choice of this cap assignment. We refer to this particular graph as the breakpoint graph of



(a) (b)

Figure 1: The construction of a random breakpoint graph. We start with the red genome, where the cap vertices are coloured red and the inner vertices are coloured black , and add the black edges randomly, one by one, until every vertex is connected by one black edge. In (b) there are 3 cycles.

the two genomes.

2.1 Random genomes

Were we to construct genomes by successively adding markers or caps in random order, it would be very difficult to say anything precise about the number of cycles in the breakpoint graph, because the linearity condition on chromosomes induces great complexity to the events whose probabilities we wish to calculate. Instead, we introduce the randomness through the choice of edges in the construction of the breakpoint graph, leading to simple expressions for probabilities of the sizes and numbers of cycles. This simplicity comes at a cost, however, since the construction of a random genome at the level of the breakpoint graph does not exclude some circular chromosomes. As we shall mention later, there is good reason to believe that this feature does not affect our results on the limits of expectations.

To obtain two genomes randomized with respect to each other, it suffices to fix the marker order in one of them, say the red genome[2], and to introduce randomness into the black genome only. At each step we simply add a black edge to connect any two vertices that are not already incident to black edges. Any inner vertex connected

[2]Without loss of generality we assume $\chi_r \leq \chi_b$.

to a cap by a black edge is called *terminal.* Thus we start with $2n+2\chi$ vertices (inner vertices and caps), with red edges connected. We distinguish between two kinds of red edges: 2χ red *cap edges* incident to a cap and $n-\chi$ red *inner edges* not incident to a cap. To construct the random breakpoint graph, we connect any two vertices at random by a black edge until each of the $2n+2\chi$ vertices are connected to one other vertex by a black edges.

Note that in randomly adding black edges we are not guaranteed to end up with linear chromosomes only, since there is the possibility that the black genome so constructed will contain one or more circular chromosomes (or *plasmids*), with no caps. In Section 6 we show that as χ becomes large, the number of such circles and the number of markers in them, tends to zero. As n becomes large, the number of circles remains relatively small. Nevertheless the possibility of circular chromosomes is not included in the statement of the original problem of finding the expected distance between two randomized genomes with linearly ordered chromosomes. Fortunately, mathematical results by Kim and Wormald[4] suggest that, in the limit, the production of circular chromosomes during the addition of black edges does not affect the probability structure of the breakpoint graph. Their results pertain most directly to single-chromosomal genomes under constraints on the number of cycles, but we may conjecture that they hold more generally.

Furthermore, there is the possibility that one or more of the random pairs of vertices connected by a black edge consists of two caps, thus defining a null chromosome (i.e., containing no inner vertices) in the black genome, so that our comparison of genomes with χ_r and χ_b chromosomes is affected by the presence of some black genomes with less than χ_b chromosomes. In Section 5 we will discuss the size of this effect, and how it disappears with large n.

2.2 Cap optimization

In cycles containing caps, we may define *paths* which start and end with edges incident to a cap or terminal. Thus by suppressing the black edges connecting terminals to caps, the breakpoint graph is decomposed into cycles containing no caps and 2χ paths, whose two ends are caps or terminals.[3] The ψ_{ho} *homogenous paths* terminate with caps via two red edges (type 1) or with two terminals (type 2), with an equal number of the two types, and the ψ_{he} *heterogenous paths* end with one cap and one terminal. The optimization principle developed by Hannenhalli and Pevzner[5] and Tesler[6], comes down to, in the reformulation by Yancopoulos *et al.*[3], to the addition of two black edges joining one homogenous path of type 1 to another homogeneous path of type 2 to form a cycle and the addition of a single black edge to each heterogeneous path to form a cycle. It can be seen that the maximum number ψ of cycles containing caps thus constructed is

$$\max \psi = \chi + \frac{1}{2}\psi_{he}. \tag{2.2}$$

3 The generating functions

Counting random breakpoint graphs with a given number of heterogeneous paths ψ_{he} requires enumerating how many ways ψ_{he} caps can be selected and matched with the same number of terminals. This is the problem of counting perfect matchings in a complete graph with two colours of vertex, where the total number of vertices is even, and where the number of edges between caps and terminals has a given value.

Theorem 1. *The exponential generating function G for the number of different perfect matchings in complete graphs on a set of labeled vertices, each coloured either red*

[3]Except where there are ν null chromosomes , in which case the number of paths will be $2\chi - \nu$.

or black, containing certain number of edges incident to vertices of different colors, is

$$G(x,y,z) = \exp(\frac{x^2+2xyz+y^2}{2}) \tag{3.1}$$

where the formal variables x, y and z mark red vertices, black vertices and edges incident to vertices of both colours, respectively.

Proof. Consider two positive even integers r and b. We first choose i red vertices and i black vertices, such that both $r-i$ and $b-i$ are even. We pair each of these red vertices with one of the black ones. Connecting pairs of the remaining red vertices produces $\frac{r-i}{2}$ edges, and pairs of the remaining black vertices produce $\frac{b-i}{2}$ edges. The number of ways of doing this is:

$$\begin{aligned} a_{r,b,i} &= \binom{r}{i}\binom{b}{i} i! \frac{(r-i)!}{2^{\frac{r-i}{2}}(\frac{r-i}{2})!} \frac{(b-i)!}{2^{\frac{b-i}{2}}(\frac{b-i}{2})!} \\ &= \frac{r!b!}{(\frac{r+b}{2})!2^{\frac{r+b}{2}}} \frac{(\frac{r+b}{2})!}{i!(\frac{r-i}{2})!(\frac{b-i}{2})!} 2^i \end{aligned} \tag{3.2}$$

Using the formal variables x, y, z to mark the red vertices, black vertices and the edges incident to two colours, respectively, we write

$$a_{r,b,i}x^r y^b z^i = \frac{r!b!}{(\frac{r+b}{2})!2^{\frac{r+b}{2}}} \frac{(\frac{r+b}{2})!}{i!(\frac{r-i}{2})!(\frac{b-i}{2})!} x^r y^b (2z)^i. \tag{3.3}$$

Fixing the total number vertices to be $2n$, and summing over all pairs r, b such that $r+b=2n$,

$$\sum_{\substack{r+b=2n \\ 0\le i\le \min(r,b)}} \frac{a_{r,b}}{r!b!} x^r y^b z^i = \sum_{\substack{r+b=2n \\ 0\le i\le \min(r,b)}} \frac{1}{n!2^n} \frac{n!}{i!(\frac{r-i}{2})!(\frac{b-i}{2})!} x^{r-i} y^{b-i} (2xyz)^i. \tag{3.4}$$

Setting $j = \frac{r-i}{2}$ and $k = \frac{b-i}{2}$

$$\begin{aligned}\sum_{\substack{r+b=2n \\ 0\leq i\leq \min(r,b)}} \frac{a_{r,b}}{r!b!}x^r y^b z^i &= \frac{1}{n!2^n}\sum_{i+j+k=n}\frac{n!}{i!j!k!}x^{2j}y^{2k}(2xyz)^i \\ &= \frac{(x^2+2xyz+y^2)^n}{n!2^n}. \end{aligned} \tag{3.5}$$

by summation of the trinomial terms. So that

$$\begin{aligned} G(x,y,z) &= \sum_{n=0}^{\infty}\frac{(x^2+2xyz+y^2)^n}{n!2^n} \\ &= \exp\left(\frac{x^2+2xyz+y^2}{2}\right) \end{aligned}$$

□

The inner edges in the breakpoints graphs form a set of inner cycles whose total number is denoted by κ, and the interior of the paths, excluding the cap edges. The number of paths is determined by the number of linear chromosomes via the number of caps. In the following the number of paths[4] is set to be m.

Theorem 2. *The exponential generating function for the number of cycles and the number of interior edges, given that there are m paths in the breakpoint graph is:*

$$H(u,v) = (1-2u)^{-m-\frac{v}{2}} \tag{3.6}$$

where the formal variables u and v pertain to inner edges and inner cycles respectively.

Proof. The exponential generating function for the number of ways of constructing

[4]For these purposes, we also count black null chromosomes as paths; so that m always takes the value $\chi_r + \chi_b$ under the convention $\chi_r \leq \chi_b$.

an inner cycle containing one or more inner edges is

$$\sum_{i=1}^{\infty} \frac{2^{i-1}(i-1)!}{i!} u^i = \frac{1}{2}\left(\sum_{i=1} \frac{1}{i}(2u)^i\right) = -\frac{1}{2}\log(1-2u). \tag{3.7}$$

By the exponential formula, the exponential generating function for all inner cycles in the breakpoint graph is $\exp(-\frac{1}{2}\log(1-2u)) = (1-2u)^{-\frac{1}{2}}$. If we use v to mark the inner cycles, then the generating function becomes

$$\exp(-\frac{1}{2}\log(1-2u)\cdot v) = (1-2u)^{-\frac{v}{2}} \tag{3.8}$$

For each path where the two ends are distinguished, the exponential generating function is

$$\sum_{i=0}^{\infty} \frac{2^i i!}{i!} u^i = \frac{1}{1-2u}, \tag{3.9}$$

where the $2^i i!$ counts the number of different paths that can be constructed with i inner edges.

Then, by the product rule for generating functions and the assumption that these m paths are ordered,[5] $H(u,v) = (1-2u)^{-m-\frac{v}{2}}$ is the generating function for inner edges and inner cycles, given m paths. □

4 The expectations and variances for ψ_{he} and κ

Let $\left[\frac{x^n}{n!}\right] G$ be $n!$ times of the coefficient of the term x^n in the exponential generating function G. In the perfect matching problem, the number of total perfect matchings of r and b red and black vertices is $\left[\frac{x^r y^b}{r!b!}\right] G(x,y,z=1)$. The number of perfect

[5]The order of the paths is determined by the caps serving as their endpoints or the caps adjacent to their terminals.

matchings with h edges incident to vertices with different colors is $\left[\frac{x^r y^b z^h}{r!b!}\right] G(x,y,z)$. Similarly, the number of different configurations for $2\chi_r$ red cap edges and $2\chi_b$ black cap edges are $\left[\frac{x^{2\chi_r} y^{2\chi_b}}{2\chi_r!2\chi_b!}\right] G(x,y,z=1)$, and the number of configurations with ψ_{he} heterogeneous paths is $\left[\frac{x^{2\chi_r} y^{2\chi_b} z^{\psi_{he}}}{2\chi_r!2\chi_b!}\right] G(x,y,z)$.

We can find the expectations and variances for ψ_{he} and κ from the generating functions (3.8) and (3.6).

Theorem 3.

$$\mathbf{E}[\psi_{he}] = \frac{4\chi_r\chi_b}{2\chi_r + 2\chi_b - 1}, \qquad \mathbf{Var}[\psi_{he}] = \frac{32\chi_r^2\chi_b^2}{(2\chi_r + 2\chi_b - 1)^2(2\chi_r + 2\chi_b - 3)}.$$

Proof. Since $G(x,y,z) = \sum_{i,j,k} a_{i,j,k} x^i y^j z^k$, we have

$$z \cdot \frac{\partial G(x,y,z)}{\partial z} = \sum_{i,j,k} k \cdot a_{i,j,k} x^i y^j z^k. \tag{4.1}$$

Now $\left[\frac{x^{2\chi_r} y^{2\chi_b}}{2\chi_r!2\chi_b!}\right] G(x,y,z=1)$ is the number of configurations with $2\chi_r$ red cap edges and $2\chi_b$ black cap edges. And $\left[\frac{x^{2\chi_r} y^{2\chi_b}}{2\chi_r!2\chi_b!}\right] \left(z \cdot \frac{\partial G(x,y,z)}{\partial z}\right)|_{z=1}$ is the summation of numbers of heterogeneous paths among these configurations. The expectation of the number of heterogeneous paths is the summation of heterogeneous paths devided by the total number of configurations,

$$\mathbf{E}[\psi_{he}] = \frac{\left[\frac{x^{2\chi_r} y^{2\chi_b}}{2\chi_r!2\chi_b!}\right] \left(z \cdot \frac{\partial G(x,y,z)}{\partial z}\right)|_{z=1}}{\left[\frac{x^{2\chi_r} y^{2\chi_b}}{2\chi_r!2\chi_b!}\right] G(x,y,z=1)}, \tag{4.2}$$

where [6]

$$\begin{aligned}\left[\frac{x^{2\chi_r}y^{2\chi_b}}{2\chi_r!2\chi_b!}\right]G(x,y,z=1) &= \left[\frac{x^{2\chi_r}y^{2\chi_b}}{2\chi_r!2\chi_b!}\right]\exp\left(\frac{(x+y)^2}{2}\right)\\ &= \frac{2\chi_r!\,2\chi_b!\binom{2\chi_r+2\chi_b}{2\chi_r}}{(\chi_r+\chi_b)!2^{\chi_r+\chi_b}}\\ &= (2\chi_r+2\chi_b-1)!! \qquad (4.3)\end{aligned}$$

and

$$\begin{aligned}\left[\frac{x^{2\chi_r}y^{2\chi_b}}{2\chi_r!2\chi_b!}\right]\left(z\cdot\frac{\partial G(x,y,z)}{\partial z}\right)|_{z=1} &= \left[\frac{x^{2\chi_r}y^{2\chi_b}}{2\chi_r!2\chi_b!}\right]xy\cdot\exp\left(\frac{(x+y)^2}{2}\right)\\ &= \frac{2\chi_r!\,2\chi_b!\binom{2\chi_r+2\chi_b-2}{2\chi_r-1}}{(\chi_r+\chi_b-1)!2^{\chi_r+\chi_b-1}}\\ &= 4\chi_r\chi_b(2\chi_r+2\chi_b-3)!! \qquad (4.4)\end{aligned}$$

Thus

$$\mathbf{E}\left[\psi_{he}\right]=\frac{4\chi_r\chi_b}{2\chi_r+2\chi_b-1}. \qquad (4.5)$$

Similarly,

$$\mathbf{E}\left[\psi_{he}(\psi_{he}-1)\right] = \frac{\left[\frac{x^{2\chi_r}y^{2\chi_b}}{2\chi_r!2\chi_b!}\right]\left(z^2\cdot\frac{\partial^2 G(x,y,z)}{\partial^2 z}\right)|_{z=1}}{\left[\frac{x^{2\chi_r}y^{2\chi_b}}{2\chi_r!2\chi_b!}\right]G(x,y,z=1)}, \qquad (4.6)$$

[6] In the following equations, we use the double factorial $n!!$ defined recursively as $n!!=n(n-2)!!$ for all $n>1$, and $0!!=1!!=1$.

where

$$\begin{aligned}\left[\frac{x^{2\chi_r}y^{2\chi_b}}{2\chi_r!2\chi_b!}\right]\left(z^2\cdot\frac{\partial^2 G(x,y,z)}{\partial^2 z}\right)|_{z=1} &= \left[\frac{x^{2\chi_r}y^{2\chi_b}}{2\chi_r!2\chi_b!}\right]x^2y^2\cdot\exp\left(\frac{(x+y)^2}{2}\right)\\ &= \frac{2\chi_r!\,2\chi_b!\binom{2\chi_r+2\chi_b-4}{2\chi_r-2}}{(\chi_r+\chi_b-2)!\,2^{\chi_r+\chi_b-2}}\\ &= 4\chi_r\chi_b\,(2\chi_r-1)(2\chi_b-1)(2\chi_r+2\chi_b-5)!!\end{aligned} \tag{4.7}$$

so that

$$\mathbf{E}\left[\psi_{he}(\psi_{he}-1)\right] = \frac{4\chi_r\chi_b(2\chi_r-1)(2\chi_b-1)}{(2\chi_r+2\chi_b-1)(2\chi_r+2\chi_b-3)}. \tag{4.8}$$

Then

$$\begin{aligned}\mathbf{Var}\left[\psi_{he}\right] &= \mathbf{E}\left[\psi_{he}(\psi_{he}-1)\right]+\mathbf{E}\left[\psi_{he}\right]-\mathbf{E}^2\left[\psi_{he}\right]\\ &= \frac{32\chi_r^2\chi_b^2}{(2\chi_r+2\chi_b-1)^2(2\chi_r+2\chi_b-3)}\end{aligned} \tag{4.9}$$

□

Corollary 1.

$$\mathbf{E}\left[\psi\right]=\max(\chi_r,\chi_b)+\frac{2\chi_r\chi_b}{2\chi_r+2\chi_b-1},\mathbf{Var}\left[\psi\right]=\frac{8\chi_r^2\chi_b^2}{(2\chi_r+2\chi_b-1)^2(2\chi_r+2\chi_b-3)}.$$

Proof. This follows directly from Equation (2.2) and Theorem 3. □

Corollary 2. *When* $\chi_r=\chi_b=\chi$

$$\mathbf{E}\left[\psi\right]=\chi+\frac{2\chi^2}{4\chi-1}\sim\frac{3}{2}\chi,\qquad \mathbf{Var}\left[\psi\right]=\frac{8\chi^4}{(4\chi-1)^2(4\chi-3)}\sim\frac{\chi}{8}$$

To get the expectation and variance of κ, we could apply the same procedure to equation (3.6) as we did in Theorem 3, but it will be easier and more instructive to work with the quantity $L_l(v)$ describing the configurations with exactly l inner edges in $H(u,v)$.

Theorem 4. *The expectation and the variance of the number of cycles not containing caps are:*

$$\mathbf{E}[\kappa] = \sum_{i=0}^{l-1} \frac{1}{2m+1+2i}, \qquad \mathbf{Var}[\kappa] = \sum_{i=0}^{l-1} \frac{1}{2m+1+2i} - \sum_{i=0}^{l-1} \frac{1}{(2m+1+2i)^2}$$

Proof. We first derive the expression for $L_l(v)$ and extract the values of the expectation and variance from it.

$$L_l(v) = \left[\frac{u^l}{l!}\right] (1-2u)^{-m-\frac{v}{2}} = \prod_{i=0}^{l-1}(2m+v+2i). \tag{4.10}$$

So the expectation is:

$$\begin{aligned} \mathbf{E}[\kappa] &= \frac{\left(v \cdot \frac{\partial L_l(v)}{\partial v}\right)|_{v=1}}{L_l(v)|_{v=1}} \\ &= \frac{\sum_{j=1}^{l-1}\left[\prod_{i=0, i\neq j}^{l-1}(2m+1+2i)\right]}{\prod_{i=0}^{l-1}(2m+1+2i)} \\ &= \sum_{i=0}^{l-1} \frac{1}{2m+1+2i}. \end{aligned} \tag{4.11}$$

And the variance is:

$$\begin{aligned}
\mathbf{Var}[\kappa] &= \mathbf{E}[\kappa(\kappa-1)] + \mathbf{E}[\kappa] - \mathbf{E}^2[\kappa] \\
&= \frac{\left(v^2 \cdot \frac{\partial^2 L(v)}{\partial^2 v}\right)|_{v=1}}{L(v)|_{v=1}} + \mathbf{E}[\kappa] - \mathbf{E}^2[\kappa] \\
&= \frac{\sum_{0\le j\ne k\le l-1}\left[\prod_{i=0,i\ne j,k}^{l-1}(2m+1+2i)\right]}{\prod_{i=0}^{l-1}(2m+1+2i)} + \mathbf{E}[\kappa] - \mathbf{E}^2[\kappa] \\
&= \sum_{0\le i\ne j\le l-1}\frac{1}{(2m+1+2i)(2m+1+2j)} \\
&\quad + \sum_{i=0}^{l-1}\frac{1}{2m+1+2i} - \left(\sum_{i=0}^{l-1}\frac{1}{2m+1+2i}\right)^2 \\
&= \sum_{i=0}^{l-1}\frac{1}{2m+1+2i} - \sum_{i=0}^{l-1}\frac{1}{(2m+1+2i)^2}
\end{aligned} \tag{4.12}$$

□

Remark 1. *The expression in Equation (4.11) converges to* $\frac{1}{2}\log\left(\frac{l+m}{m}\right)$ *as* l *and* m *both increase, or to* $\log 2 + \frac{1}{2}(\log l + \gamma)$ *as* l *increases while* $m = 0$. *The expression in Equation (4.12) converges to* $\frac{1}{2}\log\left(\frac{l+m}{m}\right) - \frac{1}{4m} + \frac{1}{4l+4m}$ *as* l *and* m *both increase, or to* $\log 2 + \frac{1}{2}(\log l + \gamma) - \frac{\pi^2}{8} + \frac{1}{4l}$ *as* l *increases while* $m = 0$.

Since $l = n - \min(\chi_r, \chi_b)$ and $m = \chi_r + \chi_b$, we have

Corollary 3. *The expectation and the variance of the number of inner cycles are asymptotically*

$$\mathbf{E}[\kappa] \sim \begin{cases} \frac{1}{2}\log\left(\frac{n+\max(\chi_r,\chi_b)}{\chi_r+\chi_b}\right) & m \ge 1 \\ \log 2 + \frac{1}{2}(\log n + \gamma) & m = 0 \end{cases} \tag{4.13}$$

$$\mathbf{Var}[\kappa] \sim \begin{cases} \frac{1}{2}\log\left(\frac{n+\max(\chi_r,\chi_b)}{\chi_r+\chi_b}\right) - \frac{1}{4(\chi_r+\chi_b)} + \frac{1}{4(n+\max(\chi_r,\chi_b))} & m \ge 1 \\ \log 2 + \frac{1}{2}(\log n + \gamma) - \frac{\pi^2}{8} + \frac{1}{4n} & m = 0 \end{cases} \tag{4.14}$$

where $m = \chi_r + \chi_b$.

From Corollary 1 and equations (2.1), (4.13) and (4.14), we have the asymptotic results for the genomic distance

Corollary 4.

$$\mathbf{E}[d] \sim \begin{cases} n - \frac{2\chi_r\chi_b}{2\chi_r+2\chi_b-1} - \frac{1}{2}\log\left(\frac{n+\max(\chi_r,\chi_b)}{\chi_r+\chi_b}\right) & m \geq 1 \\ n - \log 2 - \frac{1}{2}(\log n + \gamma) & m = 0 \end{cases} \tag{4.15}$$

$$\mathbf{Var}[d] \sim \begin{cases} \frac{8\chi_r^2\chi_b^2}{(2\chi_r+2\chi_b-1)^2(2\chi_r+2\chi_b-3)} + \frac{1}{2}\log\left(\frac{n+\max(\chi_r,\chi_b)}{\chi_r+\chi_b}\right) & \\ \quad -\frac{1}{4(\chi_r+\chi_b)} + \frac{1}{4(n+\max(\chi_r,\chi_b))} & m \geq 1 \\ \log 2 + \frac{1}{2}(\log n + \gamma) - \frac{\pi^2}{8} + \frac{1}{4n} & m = 0 \end{cases} \tag{4.16}$$

5 Formation of null chromosomes

We previously mentioned that in the construction of the random breakpoint graph, null chromosomes can form if two caps are connected with a black edge. Creating black null chromosomes reduces the number of linear chromosomes in the black genomes, contrary to our goal of comparing random genomes with a fixed number of chromosomes. In this section, we first estimate the probability of such null chromosomes being formed and then we assess the impact on the expectation of κ when formation of null chromosomes is considered. From both viewpoints, the effect of null chromosomes will be seen to be $O(1/n)$.

For simplicity, we calculate the probability only for the case where both genomes contain χ linear chromosomes, together with n markers in common. When null chromosomes are allowed, there are $2\chi+2n$ vertices available for pairing in the breakpoint graph, and there are thus $(2\chi + 2n - 1)!!$ different perfect matchings. But if we wish

to exclude null chromosomes, we can count the number of configurations by first considering the caps. The first cap can connect to any one of the $2n$ marker vertices; the second cap can connect to the remaining $2n-1$ marker vertices and so on. So there are $\prod_{i=0}^{2\chi-1}(2n-i)$ different ways to connect the caps. Then there are $2n-2\chi$ marker vertices left and they have $(2n-2\chi-1)!!$ different configurations. So there are $\frac{(2n)!(2n-2\chi-1)!!}{(2n-2\chi)!}$ different configurations containing no null chromosomes.

Therefore, the probability p of getting a configuration with null chromosomes when a black edge is allowed to connect two caps is:

$$\begin{aligned} p &= 1-\frac{(2n)!(2n-2\chi-1)!!}{(2n-2\chi)!(2n+2\chi-1)!!} \qquad (5.1)\\ 1-p &= 2^{2\chi}\frac{(2n)!(n+\chi)!}{(n-\chi)!(2n+2\chi)!} \\ &\sim 2^{2\chi}\frac{(2n)^{2n+\frac{1}{2}}(n+\chi)^{n+\chi+\frac{1}{2}}}{(n-\chi)^{n-\chi+\frac{1}{2}}(2n+2\chi)^{2n+2\chi+\frac{1}{2}}} \qquad (5.2)\end{aligned}$$

by Stirling's approximation. Then

$$\begin{aligned} 1-p &\sim \frac{(1-\frac{\chi}{n})^{2\chi-\frac{1}{2}}}{(1-\frac{\chi^2}{n^2})^{n+\chi}} \\ &\sim \exp(-\frac{\chi^2}{n}+\frac{\chi}{2n}+\frac{\chi^3}{n^2}). \qquad (5.3)\end{aligned}$$

So as a first order approximation, we may write

$$p = \frac{\chi^2}{n}. \qquad (5.4)$$

The formation of null chromosomes does not affect the number of heterogeneous paths but increases the average number of inner cycles. The more null chromosomes, the

fewer inner edges in the paths and the more inner edges available to form inner cycles. In the rest of this section, we quantify the impact of null chromosomes formation on the expectation of κ.

Because of null chromosomes, the exponential generating function we used to obtain $\mathbf{E}[\kappa]$ reflects, not exactly χ_b linear chromosomes in the black genome, but rather χ_b or fewer. Here we use it to define the exponential generating function for paths, inner edges and inner cycles, with χ_b or less linear chromosomes in the black genomes, and denote it as $F_{\leq\chi_b}$.

The generating function $G(x, y, z)$ is not affected by the consideration of null chromosomes. . For given χ_r and χ_b the number of different configurations of cap edges is $\left[\frac{x^{2\chi_r}y^{2\chi_b}}{(2\chi_r)!(2\chi_b)!}\right] G(x, y, z = 1) = (2\chi_r + 2\chi_b - 1)!!$.

$$F_{\leq\chi_b}(u, v) = (2\chi_r + 2\chi_b - 1)!!(1 - 2u)^{-(\chi_r+\chi_b)-\frac{v}{2}}. \tag{5.5}$$

Similarly we can define a series of generating functions describing breakpoint graphs with i or fewer linear chromosomes in the black genomes.

$$F_{\leq i}(u, v) = (2\chi_r + 2i - 1)!!(1 - 2u)^{-(\chi_r+i)-\frac{v}{2}}. \tag{5.6}$$

By the Inclusion-Exclusion formula, the exponential generating function for exactly χ_b linear chromosomes in the black genomes is

$$F_{=\chi_b}(u, v) = \sum_{i=0}^{\chi_b}(-1)^{\chi_b-i}\frac{(2\chi_b)!}{(2i)!(\chi_b - i)!2^{\chi_b-i}}F_{\leq\chi_b-i}(u, v). \tag{5.7}$$

We can use $F_{\leq\chi_b}$ and $F_{\leq\chi_b-1}$ to calculate the first approximation of the expectation of κ. For simplicity, we only calculate for the case $\chi_r = \chi_b = \chi$, where $m = 2\chi$.

$$\left[\frac{u^l}{l!}\right] F_{\leq\chi}(u, v=1) = (2l+2m-1)!! \tag{5.8}$$

$$\left[\frac{u^l}{l!}\right] F_{\leq\chi-1}(u, v=1) = (2l+2m-3)!! \tag{5.9}$$

$$\left[\frac{u^l}{l!}\right] \left(v \cdot \frac{\partial F_{\leq\chi}(u,v)}{\partial v}\right)|_{v=1} = (2l+2m-1)!! \sum_{i=0}^{l-1} \frac{1}{2m+2i+1} \tag{5.10}$$

$$\left[\frac{u^l}{l!}\right] \left(v \cdot \frac{\partial F_{\leq\chi-1}(u,v)}{\partial v}\right)|_{v=1} = (2l+2m-3)!! \sum_{i=0}^{l-1} \frac{1}{2m+2i-1} \tag{5.11}$$

So that

$$\begin{aligned}
\mathbf{E}[\kappa] &\sim \frac{(2l+2m-1)!! \sum_{i=0}^{l-1} \frac{1}{2m+2i+1} - \binom{m}{2}(2l+2m-3)!! \sum_{i=0}^{l-1} \frac{1}{2m+2i-1}}{(2l+2m-1)!! - \binom{m}{2}(2l+2m-3)!!} \\
&= \frac{\sum_{i=0}^{l-1} \frac{1}{2m+2i+1} - \frac{m(m-1)}{2(2m+2l-1)} \sum_{i=0}^{l-1} \frac{1}{2m+2i-1}}{1 - \frac{m(m-1)}{2(2m+2l-1)}} \qquad (5.12) \\
&= \sum_{i=0}^{l-1} \frac{1}{2m+2i+1} + \frac{\frac{m(m-1)}{2(2m+2l-1)}\left(-\frac{1}{2m-1} + \frac{1}{2m+2l-1}\right)}{1 - \frac{m(m-1)}{2(2m+2l-1)}} \\
&\sim \sum_{i=0}^{l-1} \frac{1}{2m+2i+1} - \frac{ml}{8(m+l)^2}. \qquad (5.13)
\end{aligned}$$

Remark 2. *In equation (5.12), the term* $\frac{m(m-1)}{2(2m+2l-1)}$ *converges to* 0 *when* l *goes to infinity. The corresponding terms for higher approximations converge to* 0 *even faster.*

Remark 3. *In equation (5.13), the term* $-\frac{ml}{8(m+l)^2}$ *is in the order of* $O(\frac{m}{l})$.

6 The number of plasmids in black genomes

As discussed in Section 2.1, during the random construction of the black genomes, there is the potential of creating one or more circular chromosomes besides the desired linear chromosomes. In this section, we calculate the distribution of the number of

circular plasmids (Π) and show that the expectation is actually very small.

Assume there are n markers and χ linear chromosomes. The situation is very similar to the problem of counting inner cycles. Here we need to construct χ linear chromosomes with at least one marker in each, in contrast to the construction of m paths, some of which can contain zero inner vertices. If we use x to mark signed markers, the exponential generating function to describe a linear chromosome is:

$$\sum_{i=1}^{\infty} \frac{2^{i-1} i!}{i!} x^i = \sum_{i=1}^{\infty} 2^{i-1} x^i = \frac{x}{1-2x}. \tag{6.1}$$

Using the formal variable y to mark plasmids, the exponential generating function describing the black genomes is:

$$F(x,y) \;=\; x^{\chi}(1-2x)^{-\chi-\frac{y}{2}}. \tag{6.2}$$

Thus we have:

Theorem 5. *The expectation and variance of the number of plasmids in random construction of a genome with n signed marks and χ linear chromsomes are:*

$$\mathbf{E}\,[\Pi] \;=\; \sum_{i=0}^{n-\chi-1} \frac{1}{2\chi+2i+1} \sim \frac{1}{2}\log\frac{n}{\chi} \tag{6.3}$$

$$\mathbf{Var}\,[\Pi] \;=\; \sum_{i=0}^{n-\chi-1} \frac{1}{2\chi+2i+1} - \frac{1}{(2\chi+2i+1)^2} \sim \frac{1}{2}\log\frac{n}{\chi} - \frac{1}{4\chi} + \frac{1}{4n}. \tag{6.4}$$

7 Conclusion

The mathematical essence of the comparison of two genomes with linear chromosomes is the number of inner cycles and heterogeneous paths in the 2-regular breakpoint graph whose vertices consist of a set of labeled vertices and caps. The main result,

equations (4.15) and (4.16), can be used to test whether two genomes are significantly closer than random genomes in terms of a genomic distance counting the number of inversions, reciprocal translocations, chromosome fusions and fissions, and excisions and reintegrations of circular chromosomal segments, the latter a mechanism for transposing chromosomal fragments from one site to another in the genome.[3]

The contribution here, however, is not this particular test, based implicitly on a model that unrealistically weights all rearrrangement operations equally, but the mathematical approach that enables exact and asymptotic results about the distribution of genomic distance. We can hope that this may be extended to random models that mirror the predominance of inversion among rearrangement processes.[2]

We were able to achieve exact results thanks to the introduction of randomness, not in the order of markers on the chromosomes, but in the construction of the breakpoint graph. This, unfortunately, implies that one of the genomes being compared may have circular "plasmid" chromosomes as well as the linear ones postulated, and that occasional null chromosomes may alter the number of chromosomes in that genome. I.e., we have exact results, but on a somewhat different problem than intended. This deviation becomes negligible as the number of markers increases. Still, the accuracy of our results depends on the applicability of the Kim-Wormald theorem[4] and its conjectured generalization.

The direct introduction of randomness on the chromosomes themselves is an alternate possibility, but this would prevent the straightforward calculations of cycle probabilities in the breakpoint graph.

References

[1] Sankoff, D. and Haque, L. 2006. The distribution of genomic distance between random genomes. Journal of Computational Biology 13, 1005–1012.

[2] Xu, W., Zheng, C. and Sankoff, D. 2006. Paths and cycles in breakpoint graphs of random multichromosomal genomes. Journal of Computational Biology. 2007, 14(4): 423-435.

[3] Yancopoulos, S., Attie, O. and Friedberg, R. 2005. Efficient sorting of genomic permutations by translocation, inversion and block interchange. Bioinformatics 21, 3340 – 3346

[4] Kim, J.H. and Wormald, N.C. 2001. Random matchings which induce Hamilton cycles, and Hamiltonian decompositions of random regular graphs, Journal of Combinatorial Theory, Series B 81, 20–44.

[5] Hannenhalli, S. and Pevzner, P.A. 1995. Transforming men into mice (polynomial algorithm for genomic distance problem. Proceedings of the IEEE 36th Annual Symposium on Fo undations of Computer Science. 581–92.

[6] Tesler, G. 2002. Efficient algorithms for multichromosomal genome rearrangements. Journal of Computer and System Sciences 65, 587–609.

Chapter 5

Poisson Adjacency Distributions in Genome Comparison: Multichromosomal, Circular, Signed and Unsigned Cases

Wei Xu, Benot Alain and David Sankoff. Bioinformatics 24(18) 2008. Work of Wei Xu. Preliminary work and results by Benot Alain and David Sankoff served as motivation, but none of the derivations in that previous work remain in this article.

Abstract

The number of common adjacencies of genetic markers, as a measure of the similarity of two genomes, has been widely used as indicator of evolutionary relatedness and as the basis for inferring phylogenetic relationships. Its probability distribution enables statistical tests in detecting whether significant evolutionary signal remains in the marker order. In this paper, we derive the probability distributions of the number of adjacencies for a number of types of genome—signed or unsigned, circular or linear, single-chromosome or multichromosomal. Generating functions are found for single-chromosome cases, from which exact counts can be calculated. Probability approaches are adopted for multichromosomal cases, where we find the exact values for expectations and variances. In both cases, the limiting distributions are derived in term of numbers of adjacencies. For all unsigned cases, the limiting distribution is Poisson with parameter 2; for all signed cases, the limiting distribution is Poisson with parameter $\frac{1}{2}$.

1 Introduction

The linear order of markers, such as genes or other sites, on chromosomes is a characteristic structural feature of the genome shared by all individuals in a species. During evolution, various *rearrangement events* disrupt this order by moving or inverting segments of chromosomes. At the time of the event, the order of the markers within a segment is either conserved or inverted and the order of markers outside the segment is conserved. A *breakpoint* may be created between two markers that have hitherto been adjacent in the order if one is inside the segment while the other remains outside the segment affected.

The number of breakpoints b in the comparison of two genomes is the oldest and

simplest metric representing the evolutionary divergence of species through chromosomal rearrangements, e.g. [4, 7].

Chromosomes are generally *circular* in prokaryotes, mitochondria and chloroplasts, and a single chromosome contains the entire genome, although sometimes smaller circles, *plasmids*, contain some of this information. Eukaryotic nuclear genomes are partitioned among *linear* chromosomes, from a few to a few dozen in number.

Available data on chromosomes may or may not identify the DNA strand on which the markers are found (called *signed* or *unsigned* data, respectively).

Let a be the number of pairs of markers adjacent in two genomes with the same markers $1, 2, \ldots, n$ and the same number of linear chromosomes χ_l. Then

$$a + b = n - \chi_l.$$

As evolutionary rearrangements continue to disrupt adjacencies, b increases and a decreases. Now, even pairs of randomly constructed genomes may have some adjacencies, and pairs of genomes clearly related at the DNA sequence level may have highly scrambled marker order. Then, for any given pair of genomes the question arises of whether a is significantly larger than the random case. To answer this, i.e., to test a for statistical significance, we need its distribution under the null hypothesis of randomness.

In this paper, we study the distribution of the number of common adjacencies under the null hypothesis that the n markers are ordered completely randomly on the genomes (N.B. it suffices to randomize just one of the genomes, since relabeling markers can convert one of the genome to a canonical order, e.g., $1, 2, \ldots, n$, without changing a). For multichromosomal genomes, the number of markers on each chromosome is also random. We study the cases of single or multiple chromosomes, which

can be linear or circular, and signed or unsigned. The unsigned single-chromosome case is related to the dinner table problem [6] and non-attacking kings problem [1]. G. Tesler has previously derived results for linear, single-chromosome genomes [8] .

In Section 2, we find the generating functions for the case of a single-chromosome genome and approximate the probability distributions for large n. In Section 3, we extend our discussion to multichromosomal genomes and directly derive the exact formulae for the expectation and variance of a as well as an approximation of the probability distribution for large n.

The remainder of this section introduces notation and recalls fundamental theorems used in this paper.

1.1 Notation and terminology

In the single-chromosome unsigned case we call the genome R where the markers are ordered from 1 to n the *reference genome*, i.e., $R = 1, 2, \ldots, n$. The *random genome* $G = g_1, g_2, \ldots, g_n$ is sampled from a uniform distribution on the set of permutations of $1, \ldots, n$. For multichromosomal genomes, the beginning and ending marker of each chromosome in the reference genome are given, while in the random genome only the number of chromosomes is given. In the signed case, all the markers in the reference genome R have positive sign, while in the random genome G, a positive or negative sign is assigned at random to each marker independently.

If for some i we have $|g_i - g_{i+1}| = 1$ in the unsigned case, or $g_{i+1} - g_i = 1$ in the signed case, we say there is an adjacency in common between the two genomes, otherwise there is a breakpoint. In the multichromosomal case, if g_i is the last term on a chromosome and/or g_{i+1} is the first, we identify neither an adjacency nor a breakpoint.

1.2 Generating functions

The ordinary generating function (OGF) is defined as the formal power series:

$$F(z) = \sum_{n=0}^{\infty} A_n z^n, \tag{1.1}$$

where $[z^n]$ denotes A_n the coefficient of z^n in F.

If $A_{n,m}$ denotes the number of (random) genomes with n markers and m adjacencies in common with the reference genome, consider the bivariate generating function

$$F(z,u) = \sum_{n,m} A_{n,m} z^n u^m. \tag{1.2}$$

For a given n, if X is the random variable counting the number of adjacencies in a random genome and $A_n = \sum_m A_{n,m}$, then the probability

$$\mathbf{P}_n(X = k) = \frac{A_{n,k}}{A_n} = \frac{[z^n u^k]F(z,u)}{[z^n]F(z,1)}. \tag{1.3}$$

Let $X_{(k)} = X(X-1)\dots(X-k+1)$. Then the kth factorial moment is

$$\mathbf{E}[X_{(k)}] = \mathbf{E}[X(X-1)\dots(X-k+1)] \tag{1.4}$$

$$= \frac{[z^k]\left(\partial_u^k F(z,u)\right)|_{u=1}}{[z^k]F(z,1)}, \tag{1.5}$$

where ∂_u^k denotes the k-th partial derivative with respect to u.

The probability generating function

$$P_n(u) = \frac{[z^n]F(z,u)}{[z^n]F(z,1)} \tag{1.6}$$

$$= \sum_m \frac{A_{n,m}}{A_m} u^m \tag{1.7}$$

$$= \sum_m \mathbf{P}_n(X = m)u^m. \tag{1.8}$$

Substituting u by e^t gives the familiar moment generating function.

1.3 Convergence of probability distributions

A probability measure is *determined by its moments* if it has finite moments $\alpha_k = \mathbf{E}[x^k]$ of all orders and the power series $\sum_k \alpha_k \frac{r^k}{k!}$ has a positive radius of convergence for r.

Theorem 1 (Theorem 30.2 in [2]). *Suppose that the distribution of X is determined by its moments and the X_n have moments of all orders and that $\lim_n \mathbf{E}[X_n^r] = \mathbf{E}[X^r]$ for $r = 1, 2, \ldots$. Then the distribution of X_n converges to the distribution of X.*

Theorem 2. *For probability distributions of X_n, if their kth factorial moment converges to μ^k, then their probability distributions converge to Poisson distribution with mean μ.*

Proof. A distribution is determined by its moments if $\sum_k \mathbf{E}[x^k]\frac{r^k}{k!}$ converges for any value of r. For a Poisson$[\mu]$ distribution, $\sum_k \mathbf{E}[x^k]\frac{r^k}{k!} = e^{\mu(e^r-1)}$, which converges for any μ and r. Hence Theorem 1 applies. Since regular moments $\mathbf{E}[X^r]$ are just the linear combinations of factorial moments $\mathbf{E}[X_{(r)}]$, the conclusion in Theorem 1 is also true for factorial moments. Finally for Poisson$[\mu]$, $\mathbf{E}[X_{(r)}] = \mu^r$. □

2 The generating function approach to single-chromosome genomes

In this section we consider four different cases of genomes containing only one chromosome: unsigned linear, unsigned circular, signed linear and signed circular chromosomes. We first derive the generating functions for each case. For unsigned cases, the limiting probability distributions are derived via factorial moments while for signed cases the direct derivation of the exact distribution from generating functions is possible.

We first introduce an operation (see [3]) that we call the *star operation.* For any genome, from the existing adjacencies, this operation distinguishes an arbitrary set of adjacencies and labels them with stars. For a genome with m adjacencies, there are 2^m different ways of picking starred adjacencies.

By using starred genomes, we can avoid complications due to overcounting certain nested configurations. We can then make use of a straightforward relation (Lemma 1) between starred genomes and genomes without stars to derive the main result in Theorem 3.

Lemma 1. *If $F(z,u)$ is the bivariate generating function counting the number of genomes with n elements and m adjacencies and $G(z,v)$ is the bivariate generating function counting the number of genomes with n elements and l starred adjacencies, then the star operation corresponds to the substitution $u \longrightarrow 1+v$ and*

$$G(z,v) = F(z,1+v) \tag{2.1}$$

$$F(z,u) = G(z,u-1). \tag{2.2}$$

Proof. If $f_{n,m}$, the coefficient of $z^n u^m$ in F is the number of genomes with n elements

and m adjacencies, the star operation on these genomes will produce $f_{n,m}\binom{m}{l}$ genomes with l starred adjacencies, where $l = 0, 1, \ldots, m$. We have $\sum_{l=0}^{m} f_{n,m}\binom{m}{l} z^n v^l = f_{n,m} z^n (1+v)^m$. Comparing $G(z,v) = \sum_{n,m} f_{n,m} z^n (1+v)^m$ with $F(z,u) = \sum_{n,m} f_{n,m} z^n u^m$, we have the desired result. □

2.1 The four generating functions

Theorem 3. *Denote by $F^{u,l}(z,u), F^{u,c}(z,u), F^{s,l}(z,u)$ and $F^{s,c}(z,u)$ the generating functions correspondingly to unsigned linear, unsigned circular, signed linear and signed circular single-chromosome genomes, where the coefficients of powers of z and u count markers and adjacencies, respectively. Then we have:*

$$F^{u,l}(z,u) = \sum_{n=0}^{\infty} n! z^n \left(\frac{1+uz-z}{1-uz+z}\right)^n \tag{2.3}$$

$$F^{u,c}(z,u) = \sum_{n=0}^{\infty} (n-1)! z^n \left(\frac{1+uz-z}{1-uz+z}\right)^n \tag{2.4}$$

$$F^{s,l}(z,u) = \sum_{n=0}^{\infty} n! (2z)^n \left(\frac{1}{1-uz+z}\right)^n \tag{2.5}$$

$$F^{s,c}(z,u) = \sum_{n=0}^{\infty} (n-1)! (2z)^n \left(\frac{1}{1-uz+z}\right)^n. \tag{2.6}$$

Proof. We count the numbers for the corresponding starred configurations first and derive the generating functions for the original questions via equation (2.2). Define a *synteny block* as a block of markers numbered successively either in a increasing or in a decreasing order (for the signed case, there is a minus sign before decreasing ordered markers). For a synteny block of size s, there are $s-1$ adjacencies. The *starred synteny block* is just the synteny block where each adjacency is starred. The generating function $S(z,v)$ counting the number of starred synteny blocks is $2z^2v + 2z^3v^2 + 2z^4v^3 + \cdots = \frac{2z^2v}{1-zv}$. The starred genomes are the compositions of starred synteny blocks and

free markers—markers that not involved in any starred adjacencies. Call these starred synteny blocks and free markers *free components*. Now we derive the expressions for $G(z,v)$ using the fact that starred genomes are permutations (signed/unsigned, linear/circular) of these free components.

1. Unsigned linear genomes. If there are n free components, there are $n!$ unsigned linear permutations of them. Each free component can be a free marker or a starred synteny block, so the generating function for free components is $z + S(z,v) = z + \frac{2z^2v}{1-zv}$. Then we have

$$G^{u,l}(z,v) = \sum_{n=0}^{\infty} n! \left(z + \frac{2z^2v}{1-zv}\right)^n.$$

So

$$F^{u,l}(z,u) = G^{u,l}(z,u-1) = \sum_{n=0}^{\infty} n! \left(z\frac{1+zu-z}{1-zu+z}\right)^n.$$

2. Unsigned circular genomes. Given n free components, there are $(n-1)!$ circular permutations. So that:

$$G^{u,c}(z,v) = \sum_{n=0}^{\infty} (n-1)! \left(z + \frac{2z^2v}{1-zv}\right)^n$$
$$F^{u,c}(z,u) = \sum_{n=0}^{\infty} (n-1)! \left(z\frac{1+zu-z}{1-zu+z}\right)^n.$$

3. Signed linear genomes. Given n free components there are $n!$ linear permutations. Each free component is either a free marker, which may take two different signs, or a starred synteny block. So the generating function for free components

is $2z + S(z,v) = 2z + \frac{2z^2v}{1-zv}$. We have:

$$G^{s,l}(z,v) = \sum_{n=0}^{\infty} n! \left(2z + \frac{2z^2v}{1-zv}\right)^n$$
$$F^{s,l}(z,u) = \sum_{n=0}^{\infty} n!2^n \left(z\frac{1}{1-zu+z}\right)^n.$$

4. Signed circular genomes. Similarly we have:

$$G^{s,c}(z,v) = \sum_{n=0}^{\infty} (n-1)! \left(2z + \frac{2z^2v}{1-zv}\right)^n$$
$$F^{s,c}(z,u) = \sum_{n=0}^{\infty} (n-1)!2^n \left(z\frac{1}{1-zu+z}\right)^n.$$

□

2.2 From factorial moments to limiting probability distributions for unsigned genomes

After expanding the generating functions $F(z,u)$, the coefficient of the term $z^n u^m$ is just the number of permutations with n elements and m adjacencies. While this approach is easily followed for signed genomes, it leads to complicated multiple summations for the unsigned cases. Next we derive the probability distribution for unsigned cases by means of factorial moments.

Theorem 4. *X is the random variable counting the number of adjacencies for unsigned linear or circular single-chromosome genomes. Its k-th factorial moment is*

$$\mathbf{E}[X_{(k)}] = 2^k(1 + o(\frac{1}{n})), \tag{2.7}$$

while its probability distributions

$$\mathbf{P}[X = k] = e^{-2}\frac{2^k}{k!}(1 + o(\frac{1}{n})). \tag{2.8}$$

The limiting distribution is Poisson[2].

Proof. Set $P(z,u) = 1 + uz - z, Q(z,u) = 1 - uz + z$.

Then

$$F^{u,l} = \sum_{n=0}^{\infty} n! z^n P^n Q^{-n}$$
$$F^{u,c} = \sum_{n=0}^{\infty} (n-1)! z^n P^n Q^{-n}.$$

The kth derivative of $P^n Q^{-n}$ can be expanded as

$$\partial_u^k \left[P^n Q^{-n}\right] = \sum_{i=0}^{k} \binom{k}{i} \partial_u^i P^n \cdot \partial_u^{k-i} Q^{-n}.$$

Then we have

$$\partial_u^i P^n = n_{(i)}\, z^i P^{n-i}$$
$$\partial_u^i Q^{-n} = (n+i-1)_{(i)}\, z^i Q^{-n-i},$$

where $n_{(i)}$ stands for $n(n-1)\ldots(n-i+1)$ and $n_{(0)} = 1$.

Since $P(z,u)|_{u=1} = 1$ and $Q(z,u)|_{u=1} = 1$, we have

$$\partial_u^i P^n|_{u=1} = n_{(i)}\, z^i \quad \text{and} \quad \partial_u^i Q^{-n}|_{u=1} = (n+i-1)_{(i)}\, z^i$$

and

$$\partial_u^k \left[P^n Q^{-n}\right] |_{u=1} = z^k \sum_{i=0}^{k} \binom{k}{i} n_{(i)} \cdot (n+i-1)_{(i)}. \tag{2.9}$$

1. Unsigned linear case:

$$\partial_u^k F^{u,l}|_{u=1} = \sum_{l=0}^{\infty} l! z^{l+k} \sum_{i=0}^{k} \binom{k}{i} (l+k-i+1)_{(k-i)} \cdot l_{(i)}. \tag{2.10}$$

Then for the ith factorial moment, we can calculate:

$$\begin{aligned}
\mathbf{E}[X_{(k)}^{u,l}] &= \frac{1}{n!}[z^n]\partial_u^k F^{u,l}|_{u=1} \\
&= \frac{(n-k)!}{n!} \sum_i \binom{k}{i} (n-i-1)_{(k-i)} \cdot (n-k)_{(i)} \\
&= 2^k \left(1 - \frac{k(k+1)}{2n} + O(\frac{k^4}{n^2})\right).
\end{aligned} \tag{2.11}$$

2. Unsigned circular case:

$$\partial_u^k F^{u,c}|_{u=1} = \sum_{l=0} (l-1)!\, z^{l+k} \sum_{i=0}^{k} (l+k-i-1)_{(k-i)} \cdot l_{(i)} \tag{2.12}$$

$$\begin{aligned}
\mathbf{E}[X_{(k)}^{u,c}] &= \frac{1}{(n-1)!}[z^n]\partial_u^k F^{u,c}|_{u=1} \\
&= \sum_i \binom{k}{i} \frac{n-k}{n-1} \frac{n-k-1}{n-2} \cdots \frac{n-k-i+1}{n-i} \\
&= 2^k \left(1 - \frac{k(k-1)}{2n} + O(\frac{k^4}{n^2})\right).
\end{aligned} \tag{2.13}$$

Let X^u be the number of common adjacencies for unsigned single-chromosome genomes, either linear or circular.

$$\mathbf{E}[X_{(k)}^u] = 2^k(1 + O(\frac{1}{n})) \overset{n\to\infty}{\longrightarrow} 2^k.$$

From Theorem 2, we conclude that the limiting distribution for the number of adjacencies is Poisson[2].

The probability generating function:

$$\begin{aligned} P^u(u) &= \sum_{k=0}^{n} \mathbf{P}[X^u = k]u^k \\ &= \sum_{k=0}^{n} \frac{\mathbf{E}[X^u_{(k)}]}{k!}(u-1)^k \\ &= \sum_{k=0}^{\infty} \frac{2^k(1+O(\frac{1}{n}))}{k!}(u-1)^k \\ &= e^{2u-2}(1+O(\frac{1}{n})). \end{aligned} \tag{2.14}$$

$$\mathbf{P}[X^u = k] = [u^k]P^u(u) = e^{-2}\frac{2^k}{k!}(1+O(\frac{1}{n})). \tag{2.15}$$

□

2.3 Derivation of distributions for signed cases

The relatively simpler generating functions for signed genomes enable the direct derivation of probability distributions. We have

Theorem 5. *For signed linear or circular single-chromosome genomes, the probability distributions of the number of adjacencies are:*

$$\mathbf{P}[X^{s,c} = k] \overset{n\to\infty}{\longrightarrow} e^{-\frac{1}{2}}\frac{1}{k!2^k}(1-\frac{2k-1}{2n}) \tag{2.16}$$

$$\mathbf{P}[X^{s,l} = k] \overset{n\to\infty}{\longrightarrow} e^{-\frac{1}{2}}\frac{1}{k!2^k} \tag{2.17}$$

and their limiting distributions are Poisson$[\frac{1}{2}]$.

Proof. From the generating functions $F(z,u)$ we get the corresponding probability generating functions $P_n(u)$, which give us the probability distribution immediately.

$$\begin{aligned} P_n^{s,l}(u) &= \frac{[z^n]F^{s,l}(z,u)}{[z^n]F^{s,l}(z,1)} \\ &= \frac{\sum_{l=0}^{n} l!2^l\binom{n-1}{n-l}(u-1)^{n-l}}{n!2^n} \\ &= \exp(\frac{u-1}{2})\left(1-\frac{u-1}{2n}+O\left(\frac{(\frac{u-1}{2})^n}{(n+1)!}\right)\right) \end{aligned} \tag{2.18}$$

$$\begin{aligned} P_n^{s,c}(u) &= \frac{[z^n]F^{s,u}(z,u)}{[z^n]F^{s,u}(z,1)} \\ &= \frac{\sum_{l=0}^{n}(l-1)!2^l\binom{n-1}{n-l}(u-1)^{n-l}}{(n-1)!2^n} \\ &= \sum_{i=0}^{n}\frac{2^{n-i}}{(n-1)!2^n}\frac{(n-1)!(n-i-1)!}{i!(n-i-1)!}(u-1)^i \\ &= \sum_{i=0}^{n}\frac{(\frac{u-1}{2})^i}{i!} \\ &= \exp(\frac{u-1}{2})\left(1+O\left(\frac{(\frac{u-1}{2})^n}{(n+1)!}\right)\right). \end{aligned} \tag{2.19}$$

From $\mathbf{P}[X^s = k] = [u^k]P_n^s(u)$ we have

$$\begin{aligned} \mathbf{P}[X^{s,l}=k] &= [u^k]P_n^{s,l}(u) \\ &= e^{-\frac{1}{2}}\frac{1}{k!2^k}\left(1-\frac{2k-1}{2n}+O\left(\frac{1}{(n-k)!2^{n-k}}\right)\right) \end{aligned} \tag{2.20}$$

$$\begin{aligned} \mathbf{P}[X^{s,c}=k] &= [u^k]P_n^{s,c}(u) \\ &= e^{-\frac{1}{2}}\frac{1}{k!2^k}\left(1+O\left(\frac{1}{(n-k)!2^{n-k}}\right)\right). \end{aligned} \tag{2.21}$$

□

3 The probability approach for multichromosomal genomes

For multichromosomal genomes, the variation in the number of chromosomes, shape (linear or circular) and length (the number of markers) of each chromosome complicate the exact calculation. However, some dominant tendencies emerge when the number of markers is much larger than the number of linear chromosomes. We use a probabilistic approach to characterize these tendencies.

Since the methods for unsigned and signed genomes are essentially the same, we treat the two cases at the same time. For either case, suppose there are n markers, χ_l linear chromosomes and χ_c circular chromosomes in the reference genome and n genes, χ'_l linear chromosomes and χ'_c circular chromosomes in the random genome.

Let γ_i be the event that marker g_i and $g_i + 1$ form an adjacency, in the form of either $(g_i, g_i + 1)$ or $(g_i + 1, g_i)$. (In the signed case, $(g_i, g_i + 1)$ or $(-g_i - 1, -g_i)$.)

Denote Λ as the set of adjacencies in the reference genome, i.e., markers i and $i + 1$ where i is not the end of a chromosome. Clearly $|\Lambda| = n - \chi_l$.

Let Γ_i^u (Γ_i^s for signed cases) be the indicator random variable for the event γ_i, i.e. Γ_i^u (or Γ_i^s) counts 1 when γ_i occurs, 0 otherwise. In the random genome, let p^u (or p^s) be the probability of event γ_i, where i takes any value from set Λ.

Lemma 2. *In the random genome, the probability of the event γ_i, where $i \in \Lambda$, is $p^u = \frac{2(n-\chi'_l)}{n(n-1)}$ for the unsigned case and $p^s = \frac{n-\chi'_l}{2n(n-1)}$ for the signed case.*

Proof. In the random genome, marker g_i can be located at the end of some linear chromosome, with probability $\frac{2\chi'_l}{n}$. When this happens, for unsigned genomes, there

is only one possible position for $g_i + 1$ to form an adjacency with g_i, which gives the probability $\frac{1}{n-1}$. For signed genomes, γ_i happens when g_i, $g_i + 1$ is located at the left end of the chromosome or $-g_i - 1$, $-g_i$ is located at the right end of the chromosome. Either of the two cases gives the probability $\frac{1}{2}\frac{1}{2(n-1)}$.

Gene g_i can also be placed in the interior of chromosomes with probability $\frac{n-2\chi'_l}{n}$. For unsigned genomes, two possible positions are available for $g_i + 1$ to form an adjacency with g_i, with total probability $\frac{2}{n-1}$. While for signed genomes, one possible position is available depending on the sign of g_i, with probability $\frac{1}{2(n-1)}$.

Summing up we have,

$$\begin{aligned} p^u &= \frac{2\chi'_l}{n}\frac{1}{n-1} + \frac{n-2\chi'_l}{n}\frac{2}{n-1} = \frac{2(n-\chi'_l)}{n(n-1)} \\ p^s &= \frac{2\chi'_l}{n}\frac{1}{4(n-1)} + \frac{n-2\chi'_l}{n}\frac{1}{2(n-1)} = \frac{n-\chi'_l}{2n(n-1)}. \end{aligned} \tag{3.1}$$

□

Theorem 6. *The expected number of adjacencies is*

$$\mathbf{E}[X] = \begin{cases} 2 - \frac{2(\chi_l+\chi'_l-1)}{n} + O(\frac{1}{n^2}), & \textit{unsigned genome} \\ \frac{1}{2} - \frac{\chi_l+\chi'_l-1}{2n} + O(\frac{1}{n^2}), & \textit{signed genome.} \end{cases} \tag{3.2}$$

Proof. Let X^u (X^s) be the number of adjacencies for unsigned genomes (signed genomes), which is just the summation of Γ_i^u's (Γ_i^s's) for all i in Λ.

$$X^u = \textstyle\sum_{i\in\Lambda} \Gamma_i^u \qquad \text{and} \qquad X^s = \textstyle\sum_{i\in\Lambda} \Gamma_i^s \ .$$

The expectations are easily derived:

$$\mathbf{E}[X^u] = \sum_{i\in\Lambda} \mathbf{E}[\Gamma_i^u] = \sum_{i\in\Lambda} p^u = \frac{2(n-\chi_l)(n-\chi_l')}{n(n-1)}$$

$$= 2 - \frac{2(\chi_l + \chi_l' - 1)}{n} + O(\frac{1}{n^2}) \tag{3.3}$$

$$\mathbf{E}[X^s] = \sum_{i\in\Lambda} \mathbf{E}[\Gamma_i^s] = \sum_{i\in\Lambda} p^s = \frac{1}{2}\frac{(n-\chi_l)(n-\chi_l')}{n(n-1)}$$

$$= \frac{1}{2} - \frac{\chi_l + \chi_l' - 1}{2n} + O(\frac{1}{n^2}). \tag{3.4}$$

□

Theorem 7. *The variance of the number of adjacencies is*

$$\mathbf{V}[X] = \begin{cases} 2 - \frac{2(\chi_l+\chi_l'+1)}{n} + O(\frac{1}{n^2}), & \textit{unsigned genome} \\ \frac{1}{2} - \frac{\chi_l+\chi_l'-1}{2n} + O(\frac{1}{n^2}), & \textit{signed genome.} \end{cases} \tag{3.5}$$

Proof. $\mathbf{V}[X] = \mathbf{E}[X^2] - \mathbf{E}^2[X]$, and we first calculate the non-centered second moment $\mathbf{E}[X^2]$, which can be expressed as the following summation.

$$\mathbf{E}[X^2] = \mathbf{E}[(\sum_{u\in\Lambda} \Gamma_i)^2] \tag{3.6}$$

$$= \sum_{|i-j|>1} \mathbf{E}[\Gamma_i\Gamma_j] + \sum_{|i-j|=1} \mathbf{E}[\Gamma_i\Gamma_j] + \sum_i \mathbf{E}[\Gamma_i^2].$$

In the last expression, there are $(n-\chi_l)(n-\chi_l-3)+2\chi_l$, $2(n-2\chi_l)$ and $n-\chi_l$ summands in the three summations correspondingly. For the present version of this paper, we will not detail the case-by-case calculation of these summands, which results in the quantities in the statement of this theorem.

□

Theorem 8. *The limiting probability distribution is* Poisson[2] *for unsigned genomes and* Poisson$[\frac{1}{2}]$ *for signed genomes.*

Proof. We first prove kth factorial moment converges to 2^k for unsigned genomes and 2^{-k} for signed genomes. Then by Theorem 2, we get the above conclusion.

Since Γ_i is the indicator random variable of value 0 or 1, the k-th factorial moment can be written as the following summation, where the k index runs over all k-tuples on the set Λ and no two indices take on the same value.

$$\mathbf{E}[X(X-1)\dots(X-k+1)] = \sum_{i_1,i_2,\dots,i_k \in \Lambda} \mathbf{E}[\Gamma_{i_1}\Gamma_{i_2}\dots\Gamma_{i_k}]. \tag{3.7}$$

Since the value of conditional expection depends on indices $i_1, i_2, \dots, i_k$, the summation on the right hand side of (3.7) is split into two summations:

$$\Sigma_1 = \sum_{\substack{i_1,i_2,\dots,i_k \in \Lambda \\ |i_l - i_m|>1,\ \forall l,m}} \mathbf{E}[\Gamma_{i_1}\Gamma_{i_2}\dots\Gamma_{i_k}] \tag{3.8}$$

$$\Sigma_2 = \sum_{\substack{i_1,i_2,\dots,i_k \in \Lambda \\ |i_l - i_m|=1,\ \exists l,m}} \mathbf{E}[\Gamma_{i_1}\Gamma_{i_2}\dots\Gamma_{i_k}]. \tag{3.9}$$

Denote $\mathbf{E}[\Gamma_{i_1}\Gamma_{i_2}\dots\Gamma_{i_k}|g_{i_1}, g_{i_2}, \dots, g_{i_k}]$ as the conditional expectation for given indices $\{i_j : 1 \le j \le k\}$, when the i_l-th element on the random genome is g_{i_l} for all $1 \le l \le k$. Then the unconditional expectation can be expressed as:

$$\begin{aligned} &\mathbf{E}[\Gamma_{i_1}\Gamma_{i_2}\dots\Gamma_{i_k}] \\ &= \frac{1}{\binom{n}{k}} \sum_{g_{i_1},g_{i_2},\dots,g_{i_k}} \mathbf{E}[\Gamma_{i_1}\Gamma_{i_2}\dots\Gamma_{i_k}|g_{i_1}, g_{i_2}, \dots, g_{i_k}]. \end{aligned} \tag{3.10}$$

The set $\{g_{i_1}, g_{i_2}, \ldots, g_{i_k} : i_1, i_2, \ldots, i_k \in \Lambda\}$ can be split into:

$$\mathcal{A}_1 = \{g_{i_1}, g_{i_2}, \ldots, g_{i_k} : |g_{i_l} - g_{i_m}| > 2,\ \forall l, m\}$$
$$\mathcal{A}_2 = \{g_{i_1}, g_{i_2}, \ldots, g_{i_k} : |g_{i_l} - g_{i_m}| \leq 2,\ \exists l, m\}.$$

Then the expection $\mathbf{E}' = \mathbf{E}[\Gamma_{i_1}\Gamma_{i_2}\ldots\Gamma_{i_k}]$ on the right hand side of (3.8) becomes:

$$\begin{aligned}\mathbf{E}' =& \mathbf{P}(\mathcal{A}_1)\mathbf{E}[\Gamma_{i_1}\Gamma_{i_2}\ldots\Gamma_{i_k}|\mathcal{A}_1] \\ &+\mathbf{P}(\mathcal{A}_1)\mathbf{E}[\Gamma_{i_1}\Gamma_{i_2}\ldots\Gamma_{i_k}|\mathcal{A}_2]\end{aligned}$$

$$\begin{aligned}\mathbf{P}(\mathcal{A}_1) \geq& \frac{(n-2\chi_l')(n-2\chi_l'-5)\ldots(n-2\chi_l'-5k+5)}{n(n-1)\ldots(n-k+1)} \\ =& 1 - \frac{2k(\chi_l'+k-1)}{n} + O(\frac{1}{n^2}) \qquad (3.11)\\ \mathbf{P}(\mathcal{A}_2) =& 1 - \mathbf{P}(\mathcal{A}_1) \\ \leq& \frac{2k(\chi_l'+k-1)}{n} + O(\frac{1}{n^2}). \qquad (3.12)\end{aligned}$$

Since $\mathcal{A}_1$ asymptotically occurs with probability 1, as n goes to ∞, then $\mathbf{E}' \overset{n\to\infty}{\longrightarrow} \mathbf{E}[\Gamma_{i_1}\Gamma_{i_2}\ldots\Gamma_{i_k}|\mathcal{A}_1]$, which takes the maximum value among all conditional expectations as:

$$\mathbf{E}[\Gamma_{i_1}\Gamma_{i_2}\ldots\Gamma_{i_k}|\mathcal{A}_1] = \begin{cases}\frac{2^k}{(n-k)(n-k-1)\ldots(n-2k+1)}, & \text{unsigned genomes} \\ \frac{1}{2^k(n-k)(n-k-1)\ldots(n-2k+1)}, & \text{signed genomes.}\end{cases} \qquad (3.13)$$

Since the number of summands in Σ_1 is at least $n(n-3)(n-6)\ldots(n-3k+3) = n^k + O(n^{k-1})$, the number of summands in Σ_2 is in the order $O(n^{k-1})$ and the unconditional

expectaion in Σ_2 is no larger than $\mathbf{E}[\Gamma_{i_1}\Gamma_{i_2}\ldots\Gamma_{i_k}|\mathcal{A}_1]$, then

$$\begin{aligned}\Sigma_1 &= \left(\sum_{\substack{i_1,i_2,\ldots,i_k\in\Lambda\\ |i_l-i_m|>1,\ \forall l,m}} \mathbf{E}[\Gamma_{i_1}\Gamma_{i_2}\ldots\Gamma_{i_k}|\mathcal{A}_1]\right)\cdot\left(1+O(\frac{1}{n})\right)\\ &= \left(n^k+O(n^{k-1})\right)\cdot O(n^{-k}) = O(1) \qquad (3.14)\\ \Sigma_2 &= O(n^{k-1})\cdot O(n^{-k}) = O(\frac{1}{n}). \qquad (3.15)\end{aligned}$$

So Σ_2 is at least one order of magnitude smaller than Σ_2.

$$\begin{aligned}&\mathbf{E}[X(X-1)\ldots(X-k+1)] = \Sigma_1\cdot\left(1+O(\frac{1}{n})\right)\\ &= \left(n^k+O(n^{k-1})\right)\cdot\mathbf{E}[\Gamma_{i_1}\Gamma_{i_2}\ldots\Gamma_{i_k}|\mathcal{A}_1]\\ &= \begin{cases}2^k+O(\frac{1}{n}), & \text{unsigned genome}\\ 2^{-k}+O(\frac{1}{n}), & \text{signed genome.}\end{cases} \qquad (3.16)\end{aligned}$$

From the convergence of the factorial moments, we have the convergence of the probability distribution: the limiting probability distribution of number of adjacencies is Poisson[2] for unsigned genomes and Poisson[$\frac{1}{2}$] for signed genomes. □

Remark 1. *Using the methods of Theorem 6, we can calculate the covariance between* Γ_i *and* Γ_j *as* $Cov(\Gamma_i,\Gamma_j) = \mathbf{E}[\Gamma_i\Gamma_j] - \mathbf{E}[\Gamma_i]\mathbf{E}[\Gamma_j]$*:*

$$\begin{aligned}Cov(\Gamma_i^u,\Gamma_j^u) &= \frac{4}{n^3}+O(\frac{1}{n^4}), && |i-j|>1\\ Cov(\Gamma_i^u,\Gamma_j^u) &= -\frac{2}{n^2}+\frac{4\chi_l'-2}{n^3}+O(\frac{1}{n^4}), && |i-j|=1\\ Cov(\Gamma_i^s,\Gamma_j^s) &= \frac{1}{4n^3}+O(\frac{1}{n^4}), && \textit{for all cases.} \qquad (3.17)\end{aligned}$$

Since $\mathbf{V}[\Gamma_i^u] = \frac{2}{n} + O(\frac{1}{n^3})$ *and* $\mathbf{V}[\Gamma_i^s] = \frac{1}{2n} + O(\frac{1}{n^3})$, *the covariances are at least one order of magnitude smaller than the variance. The* Γ_i*'s can be treated as independent identical random variables under a mild approximation, which leads to Poisson distributions.*

4 Conclusion

In this paper, we used a combinatorial approach to find the generating functions counting the number of genomes with given numbers of markers and adjacencies for genomes with only one chromosome. We used probabilistic methods to calculate the

number of adjacencies	p-value for unsigned case	p-value for signed case
0	1	1
1	0.8647	0.3935
2	0.5940	0.0902
3	0.3233	0.0144
4	0.1429	0.0018
5	0.0527	0.00017
6	0.0166	0.000014
7	0.0045	1.00×10^{-6}
8	0.0011	6.23×10^{-8}
9	2.37×10^{-4}	3.50×10^{-9}
10	4.64×10^{-5}	4.10×10^{-10}

Table 1: p-values for given number of adjacencies when n is large.

exact values for random expectations and variances of the number of adjacencies for genomes with any number of linear and circular chromosomes. The overall conclusion is that the limiting probability distribution is Poisson[2] for the unsigned case and Poisson[$\frac{1}{2}$] for the signed case.

Based on the limiting Poisson distribution, we can devise a statistical test for whether the two genomes contain a significant evolutionary signal, when the number

of markers is not too small. For unsigned genomes with number of adjacencies a, the p-value is calculated by $p^u(a) = 1 - \sum_{i=0}^{a-1} e^{-2}\frac{2^i}{i!}$. For signed genomes with number of adjacencies a , the p-value is calculated by $p^s(a) = 1 - \sum_{i=0}^{a-1} e^{-\frac{1}{2}}\frac{2^{-i}}{i!}$. Based on Table 1, when the unsigned distance is larger than 5 or the signed adjacency distance is larger than 2, a statistical test with a critical region of 5% will reject the null hypothesis of randomness and accept that there is a significant evolutionary signal between the two genomes involved.

Acknowledgements

We thank Glenn Tesler for discussing his previous work on this topic with us, and Daniel Panario for his suggestions and encouragement. Research supported in part by a grant to DS from the Natural Sciences and Engineering Research Council of Canada (NSERC). DS holds the Canada Research Chair in Mathematical Genomics.

References

[1] M. Abramson and W. Moser. Combinations, successions and the n-kings problem. *Mathematics Magazine*, 39: 269–273, 1966.

[2] P. Billingsley. *Probability and Measure*, third edition. Wiley InterScience, 1995.

[3] P. Flajolet and R. Sedgewick. *Analytic Combinatorics.* Cambridge University Press, 2008.

[4] J.H. Nadeau, J.H. and B.A. Taylor. Lengths of chromosomal segments conserved since divergence of man and mouse. *Proceedings of the National Academy of Sciences (U.S.A.)*, 81: 814–818, 1984

[5] J. Riordan. 1965. A recurrence for permutations without rising or falling successions. *Annals of Mathematical Statistics*, 36: 708–710, 1995.

[6] D.P. Robbins. The probability that neighbors remain neighbors after random rearrangements. *The American Mathematical Monthly*, 87: 122–124, 1980.

[7] D. Sankoff and M. Blanchette. Multiple genome rearrangement and breakpoint phylogeny. *Journal of Computational Biology*, 5: 555–570, 1998.

[8] G. Tesler. Decomposition of permutations into rising and falling subsequences. unpublished manuscript, 2005.

Chapter 6

Decompositions of multiple breakpoint graphs and rapid exact solutions to the median problem

Andrew Wei Xu and David Sankoff. Proceedings of the Workshop on Algorithms in Bioinformatics, WABI 2008, Lecture Notes in Bioinformatics 5251, Springer. 2008.
Work of Andrew Wei Xu. Part of the research based on suggestions by D. Sankoff.

Abstract

The median genome problem reduces to a search for the vertex matching in the multiple breakpoint graph (MBG) that maximizes the number of alternating colour cycles formed with the matches representing the given genomes. We describe a class of "adequate" subgraphs of MBGs that allow a decomposition of an MBG into smaller, more easily solved graphs. We enumerate all of these graphs up to a certain size and incorporate the search for them into an exhaustive algorithm for the median problem. This enables a dramatic speedup in most randomly generated instances with hundreds or even thousands of vertices, as long as the ratio of genome rearrangements to genome size is not too large.

1 Introduction

The median problem underlies one approach to phylogenetics based on genomic distance. The idea, illustrated in Figure 1, is to optimize each ancestral node of an unrooted phylogeny in terms of its three or more immediate neighbours, modern or ancestral, and to iterate across the tree until convergence of the objective function (to a local optimum) at all nodes. This approach to the "small phylogeny" problem (i.e., the graph structure of the tree is given and does not need to be inferred, in contrast to the "big phylogeny problem") has a decade of history in the study of genome rearrangement [7, 6, 2, 1], though its use in sequence-based phylogenetics dates to the 1970s [8].

In the study of genome rearrangement, genomes are treated as signed permutations on $1, \ldots, n$, either circular or linear, sometimes fragmented into chromosomes. The metric d on the set of genomes is an edit distance that counts the minimum number of operations required to transform one genome into another. The allowed

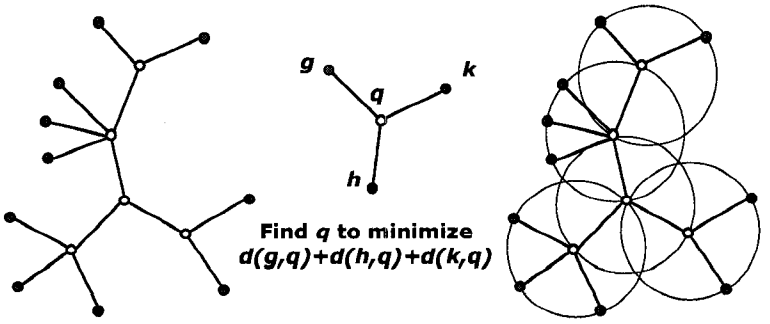


Figure 1: Left: unrooted phylogeny with open dots representing ancestral genomes to be inferred. Middle: median problem with three given genomes g, h and k and median q to be inferred. Right: decomposition of phylogeny into overlapping median problems.

operations may include the reversal of a contiguous chromosomal fragment, which also switches the sign on each term in the scope of the reversal; translocation, which involves the exchange of suffixes or prefixes of two chromosomes; transposition, or the excision of a contiguous chromosomal fragment and its re-insertion elsewhere on the chromosome; and a limited number of other operations. While distances involving reversals and translocations only can be calculated in time linear in n [4, 10], the complexity of allowing transpositions in the distance calculation, either alone or in combination with reversals and translocations, is unknown. Recently, by generalizing the operation of transposition to that of block interchange [12], it became possible to include transpositions with reversals and translocations in genomic distance calculations, within a framework known as "double cut and join" (DCJ). Moreover, the DCJ framework allows for substantial mathematical simplification of the distance calculation.

The median problem for genomic rearrangement distances in NP-hard [3, 9]. Algorithms have been developed to find exact solutions for small instances [3, 6] and there are rapid heuristics of varying degrees of efficiency and accuracy [2, 1, 5]. In

the present paper, we explore the hypothesis that although there are no worst-case guarantees, it is worthwhile to develop methods to rapidly detect instances which are easily solved exactly.

Because of its simple structure, we choose to work with DCJ distance d as most likely to yield non-trivial mathematical results. We require genomes to consist of one or more circular chromosomes, but this is for simplicity of presentation, and our results could fairly easily be extended to genomes with multiple linear chromosomes. Then the median problem is to find a genome q with the smallest total distance $\sum_{g \in G} d(q, g)$, for a given set of genomes $\mathcal{G}$.

The mathematical analysis of genomic distances generally invokes the *breakpoint graph*, which we will describe in Section 2. For DCJ, we have $d(g, h) = n - c$, where n is the number of genes in genomes g and h, and c is the number of cycles in the breakpoint graph. We define *adequate* subgraphs of the breakpoint graph, and key graph transformations in Section 2, and we demonstrate in Section 3 how to decompose large instances of median problems into smaller instances. This effectively reduces the search space of the median problem and makes it possible to design algorithms applicable to most instances of interest to biologists. In Sections 4 and 5, we sketch some of the considerations involved in these algorithms and describe the results of simulations on various data sets.

2 Graph and subgraph structures

2.1 Breakpoint graph

We construct the breakpoint graph of two genomes as in Figure 2 by representing each gene by an ordered pair of vertices, adding coloured edges to represent the adjacencies

between two genes, red edges for one genome and blue for the other.

In a genome, every gene has two adjacencies, one incident to each of its two endpoints, since it appears exactly once in that genome. Then in the breakpoint graph, every vertex is incident to one red edge and one blue one. Thus the breakpoint graph is a 2-regular graph which automatically decomposes into a set of alternating-colour cycles.

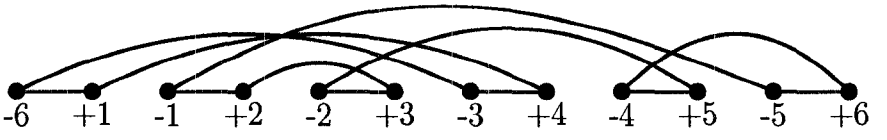


Figure 2: Breakpoint graph for blue genome 1 -5 -2 3 -6 -4 and red genome 1 2 3 4 5 6.

The edges of one colour form a perfect matching of the breakpoint graph, which we will simply refer to as a *matching*, unless otherwise specified. By the red matching, we mean the matching consisting of all the red edges.

The size for breakpoint graphs, multiple breakpoint graphs and median graphs is defined as half the number of vertices in it, which also equals to the number of genes in each genome and the size of either perfect matching.

2.2 Multiple breakpoint graph and median graph

The breakpoint graph extends naturally to a multiple breakpoint graph (MBG), representing a set $\mathcal{G}$ of three or more genomes. The number of genomes $N_{\mathcal{G}} \geq 3$ in $\mathcal{G}$ is also the chromatic number of the MBG. The colours assigned to the genomes are labeled by the integers from 1 to $N_{\mathcal{G}}$. We will use $B(\mathcal{G})$ or B throughout to refer to the MBG of the genomes $\mathcal{G}$.

For a given distance d, the median problem for $\mathcal{G} = \{g_1, \ldots, g_{N_\mathcal{G}}\}$ is to find a genome q which minimizes $\sum_{i=1}^{N_\mathcal{G}} d(g_i, q)$. For a candidate median genome, we use a different colour for its matching E, namely colour 0. Adding E to the MBG $B(\mathcal{G})$ results in the *median graph* $M_E(\mathcal{G}) = B(\mathcal{G}) \cup E$.

The set of all possible candidate matchings is denoted by $\mathcal{E}$. The set of all possible median graphs is $\mathcal{M}(\mathcal{G}) = \{M = B(\mathcal{G}) \cup E : E \in \mathcal{E}\}$.

The 0-i cycles in a median graph with matching E, numbering $c(0, i)$ in all, are the cycles where 0-edges and i edges alternate. Let $c_E(B) = \sum_{i=1}^{N_\mathcal{G}} c(0, i)$. Then $c_{\max}(B) = \max\{c_E(B) : E \in \mathcal{E}\}$ is the maximum number of cycles that can be constructed from B.

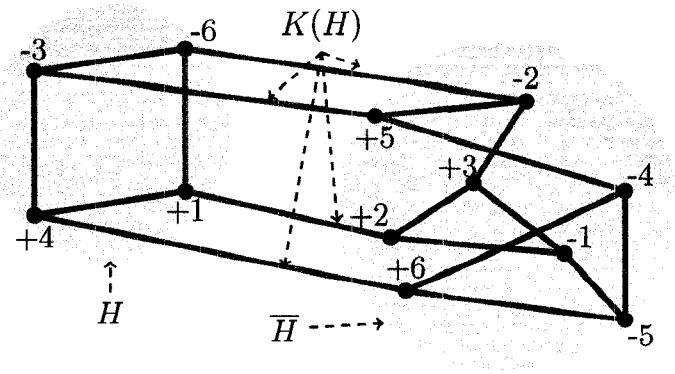
Minimizing the total distance in the median problem is equivalent to finding an optimal matching E, i.e., with $c_E(B) = c_{\max}(B)$. Let $\mathcal{E}^\star(B)$ be the set of all optimal matchings.

2.3 MBG subgraphs and connecting edges

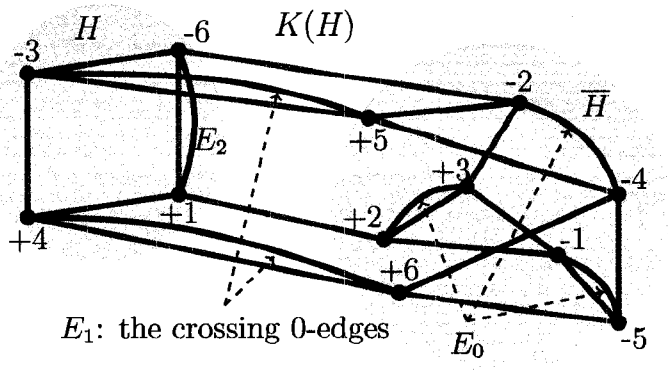
Let $\mathbf{V}(G)$ and $\mathbf{E}(G)$ be the sets of vertices and edges of a regular graph G. A *proper subgraph* H of G is one where $\mathbf{V}(H) = \mathbf{V}(G)$ and $\mathbf{E}(H) = \mathbf{E}(G)$ do not both hold at the same time. An *induced subgraph* H of G is the subgraph which satisfies the property that if $x, y \in \mathbf{V}(H)$ and $(x, y) \in \mathbf{E}(G)$, then $(x, y) \in \mathbf{E}(H)$.

In this paper, we will focus on the induced proper subgraphs, with an even number of vertices, of an MBG. Half of the number of these vertices is defined as the size of the subgraph H, denoted by m. $\mathcal{E}(H)$ is the set of all perfect 0-matchings $E(H)$, the cycle number determined by H and $E(H)$ is $c_{E(H)}(H)$, and $c_{\max}(H)$ is the maximum number of cycles that can be constructed from H by adding some $E(H)$. A 0-matching $E^\star(H)$ with $c_{E^\star(H)}(H) = c_{\max}(H)$ is called an optimal local matching, and $\mathcal{E}^\star(H)$ is the set of such matchings.

The *connecting edges* of a subgraph H in an MBG $B(\mathcal{G})$ are the edges of $B(\mathcal{G})$ incident to H exactly once, and are denoted by $K(H)$. The complementary induced subgraph of H in $B(\mathcal{G})$, denoted as $\overline{H}$, is the subgraph of $B(\mathcal{G})$ induced by $\mathbf{V}(B) - \mathbf{V}(H)$. Note that $B(\mathcal{G}) = H + K(H) + \overline{H}$, as illustrated in Figure 3.



(a)



(b)

Figure 3: MBG and median graph. Red, blue, green and black denote colours 1, 2, 3 and 0. (a) An MBG of based on three genomes, red (1 2 3 4 5 6), blue (1 -5 -2 3 -6 -4) and green (1 3 5 -4 6 -2). A subgraph H, the connecting edge set $K(H)$ and the complementary subgraph $\overline{H}$ are illustrated. (b) A median graph. The candidate matching is divided into three 0-edge sets: E_0, E_1 and E_2.

2.4 Crossing edges and decomposers

For an MBG B and a subgraph H, a potential 0-edge would be *H-crossing* if it connected a vertex in $\mathbf{V}(H)$ to a vertex in $\mathbf{V}(\overline{H})$. A candidate matching containing one or more H-crossing 0-edges is an H-crossing candidate. A MBG subgraph H is called a *decomposer* if for any MBG containing it, there is an optimal matching that is not H-crossing. It is a *strong decomposer* if for any MBG containing it, all the optimal matchings are not H-crossing.

For an MBG B, the search space for an optimal matching is $\mathcal{E}$, which is of size $(2n-1)!! = \frac{(2n)!}{2^n n!}$. If B contains a (strong) decomposer H of size m, then the search can be limited to the smaller space $\mathcal{E}(H) \times \mathcal{E}(\overline{H}) = \{E = E_H \cup E_{\overline{H}} : E_H \in \mathcal{E}(H), E_{\overline{H}} \in \mathcal{E}(\overline{H})\}$, which is of size $(2m-1)!! \cdot (2n-2m-1)!!$.

2.5 Adequate and strongly adequate subgraphs

In an MBG for a set of genomes $\mathcal{G}$, a connected subgraph H of size m is an *adequate subgraph* if $c_{\max}(H) \geq \frac{1}{2}mN_{\mathcal{G}}$; it is *strongly adequate* if $c_{\max}(H) > \frac{1}{2}mN_{\mathcal{G}}$.

A (strongly) adequate subgraph H is *simple* if it does not contain another (strongly) adequate subgraph as an induced subgraph; deleting any vertex from H will destroy its adequacy. In addition, a simple (strong) adequate subgraph H is *minimal* if we cannot even delete any edges without destroying its adequacy, i.e., for any edge $e \in \mathbf{E}(H)$, $c_{\max}(H-e) < \frac{1}{2}mN_{\mathcal{G}}$ ($c_{\max}(H-e) \leq \frac{1}{2}mN_{\mathcal{G}}$).

2.6 Edge shrinking, expansion and contraction

To shrink an edge e in a graph B, delete its two end vertices and any edges (including e) parallel to e, then for the edges incident to the deleted vertices, replace each pair of edges of same colour by a single edge of that colour, producing a new graph $B \circ e$,

as illustrated by Fig 4(a)–(c). To shrink a set of edges A, shrink the edges in A one by one in any order, producing $B \circ A$.

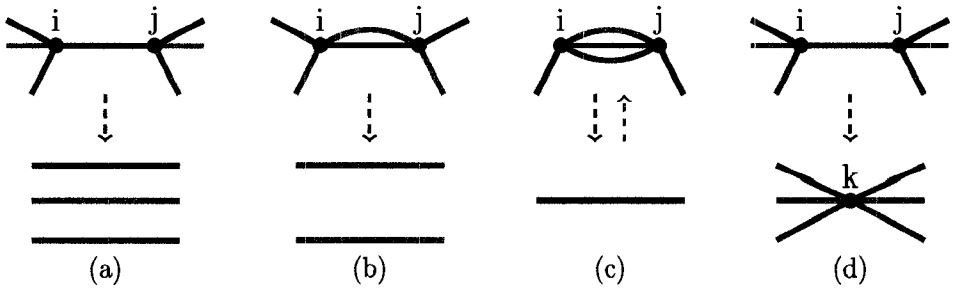


Figure 4: Edge shrinking, expansion and contraction in a median graph based on 3 genomes: the downward arrows in (a), (b) and (c) illustrate edge shrinking in various situations; (c) the upward arrow illustrates an expansion of a black edge; (d) illustrates a contraction of a black edge.

To expand a 0-edge (a, b) in a graph B, remove that edge, add two new vertices i and j to the graph, connect i and j by $N_{\mathcal{G}}$ edges with colours ranging from 1 to $N_{\mathcal{G}}$, and add 0-edges (a, i) and (b, j), as illustrated by Fig 4(c) following the direction of the red arrow.zx

Proposition 1. *If median graph M' is obtained from another median graph M by expanding some 0-edge, then they contain the same number of cycles, i.e. $c(M') = c(M)$.*

To contract a 0-edge e from a graph G, delete e and merge its two end vertices, resulting in the graph G/e, as illustrated by Fig 4(d).

3 An adequate subgraph is a decomposer

In this section, we prove our main result: every (strongly) adequate subgraph is a (strong) decomposer. The general idea of the proof is that if H is a (strongly) adequate subgraph of MBG $B(\mathcal{G})$, for any H-crossing candidate matching E, we can

always find another candidate matching E' that is not crossing, with $c_{E'}(B) \geq c_E(B)$ (or $c_{E'}(B) > c_E(B)$).

We partition the 0-edges in E among three sets: E_0, the set of 0-edges not incident to H; E_1, those incident to H exactly once; and E_2, those incident to H twice. In the median graph $M = B \cup E$, we shrink the 0-edge set E_0 and expand each 0-edge in E_2. The resultant median graph illustrated by Fig 5(a) is called the *twin median graph*, denoted by $\overset{\circ\bullet}{M} = \overset{\circ\bullet}{B} \cup \overset{\circ\bullet}{E}$.

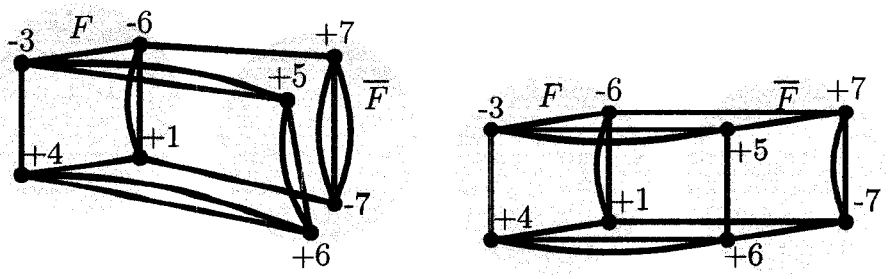


Figure 5: Twin median graph and symmetrical median graph. (a) The twin median graph is obtained from the median graph in Figure 3b by shrinking the 0-edge set E_0 and expanding the 0-edge set E_2. (b) is the corresponding symmetric graph, with the left part mirror-symmetric to the right part.

If the 0-edges of a cycle in M are all in E_0, then after shrinking all 0-edges in E_0, this cycle does not appear in $\overset{\circ\bullet}{M}$. If a cycle in M contains 0-edges in E_1 or E_2, then with only part of the cycle being shrunk, this cycle does appear in $\overset{\circ\bullet}{M}$. Denote $c_{E_0}(B)$ as the number of cycles formed by B and 0-edges in E_0 only. Then

Proposition 2.

$$c_E(B) = c_{E_0}(B) + c_{\overset{\circ\bullet}{E}}(\overset{\circ\bullet}{B}) \tag{3.1}$$

Since E_0 is not incident to the subgraph H, shrinking E_0 does not affect H. So H remains in $\overset{\circ\bullet}{M}$. Denote the subgraph in $\overset{\circ\bullet}{M}$ induced by $\mathbf{V}(H)$ as F. If a pair of connecting edges with colour i in M, is connected by a 0-i alternating colour path,

with all 0-edges in E_0, then after shrinking E_0, this pair of i-edges are merged into a new i-edge e, with both ends incident to $\mathbf{V}(H)$. Edges like e are contained in F but not in H. Thus

Proposition 3. *Suppose $\overset{\circ}{B}^{\bullet}$ is a twin MBG constructed from B based on a subgraph H of size m, and F is the subgraph in $\overset{\circ}{B}^{\bullet}$ induced by $\mathbf{V}(H)$. Then F is of size m and $F \supseteq H$. If H is a (strongly) adequate subgraph, then so is F.*

Suppose the number of connecting edges in $K(F)$ of the twin MBG $\overset{\circ}{B}^{\bullet}$ is $2k$. The 0-edges in $\overset{\circ}{M}^{\bullet}$ denoted by $\overset{\circ}{E}^{\bullet}$ are either from E_1 or the new added ones when expanding E_2. All of them are incident to F exactly once, so each 0-edge in $\overset{\circ}{E}^{\bullet}$ is F-crossing. Then F and $\overline{F}$ must be of the same size.

The 0-edges in $\overset{\circ}{E}^{\bullet}$ can be viewed as a mapping from the vertex set $\mathbf{V}(F)$ to $\mathbf{V}(\overline{F})$. If under this mapping, F is isomorphic to $\overline{F}$, as illustrated by Fig 5(b) then we call the twin median graph a *symmetrical median graph*, and we denote it by $\overset{\circ}{M}^{\circ}$.

In any twin median graph, the size of an alternating colour cycle is at least 1, which is only possible when a 0-edge is parallel to a connecting edge. All other cycles have minimum size 2. We have

Proposition 4. *If in a twin median graph $\overset{\circ}{M}^{\bullet}$, any cycle containing a connecting edge is of size 1 and any other cycle is of size 2, then $\overset{\circ}{M}^{\bullet}$ contains the largest possible number of cycles among all twin median graphs formed from $\overset{\circ}{B}^{\bullet}$. The maximum cycle number is $mN_G + k$. This can be achieved only when $\overset{\circ}{M}^{\bullet}$ is a symmetrical median graph $\overset{\circ}{M}^{\circ}$.*

Proof. Since there are $2k$ connecting edges, the number of cycles of size 1 must be $2k$. Then the number of remaining non-0 edges is $2mN_G - 2k$. Hence there are $mN_G - k$ cycles of size 2. The maximum total number of cycles is $mN_G + k$. Because of the symmetry of $\overset{\circ}{M}^{\circ}$, the other cycles can only be of size 2. Hence $\overset{\circ}{M}^{\circ}$ is the only twin median graph containing the maximum number of cycles. □

Next we investigate the difference between a twin median graph $\overset{\circ\bullet}{M}$ and a symmetric median graph $\overset{\circ\circ}{M}$, in terms of the number of DCJ operations needed to transform one into another.

Lemma 1. *If $\overset{\circ\bullet}{M}$ is a twin median graph and $\overset{\circ\circ}{M}$ is the symmetric median graph, then we can transform one into the other by exactly $mN_G + k - c(\overset{\circ\bullet}{M})$ DCJ operations on non 0-edges.*

Proof. We construct the *contracted graph*, illustrated in Figure 6, by contracting 0-edges of a median graph $\overset{\circ\bullet}{M}$, where edges in $\overline{F}$ are represented by dashed lines and the connecting edges are represented by half-dashed, half-solid lines with the solid end incident to F and the dashed end incident to $\overline{F}$. For conciseness, when we say *solid edges* (*dashed edges*), we mean the solid (dashed) edges contained by F ($\overline{F}$) or the solid (dashed) ends of connecting edges. The contracted graph for $\overset{\circ\bullet}{M}$ is denoted by $\overset{\bullet\bullet}{M}$ and the contracted graph for $\overset{\circ\circ}{M}$ is denoted by $\overset{\infty}{M}$.

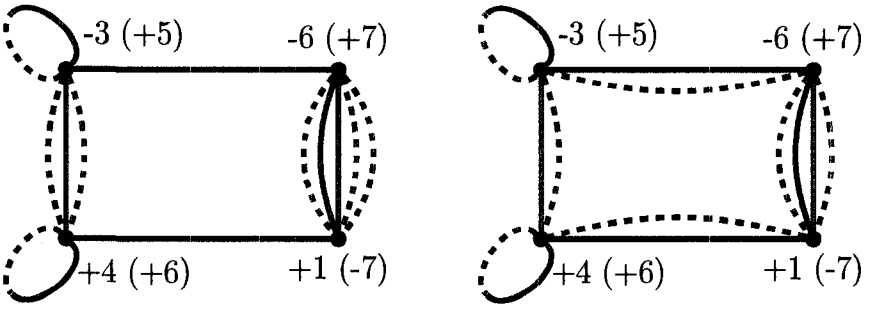


Figure 6: The contracted twin graph (a) and contracted symmetric graph (b). The contracted graphs are generated from a twin median graph by contracting 0-edges. Dashed edges are from the complementary subgraphs and the half-solid-half-dashed ones are the connecting edges.

Comparing the median graph $\overset{\circ\bullet}{M}$ and the contracted graph $\overset{\bullet\bullet}{M}$, it easy to see that each vertex in $\overset{\bullet\bullet}{M}$ has degree $2N_G$, incident to N_G solid edges and N_G dashed edges.

The 0-i alternating colour cycle in $\overset{\circ\bullet}{M}$ becomes the alternating pattern (solid/dashed) cycle with colour i. The number of alternating pattern cycles is equal to the number of alternating colour cycles. Thus there are $c(\overset{\circ\bullet}{M})$ pattern alternating cycles in $\overset{\circ\bullet}{M}$ and $mN_{\mathcal{G}} + k$ cycles in $\overset{\infty}{M}$.

To transform $\overset{\bullet\bullet}{M}$ to $\overset{\infty}{M}$, we can show that there always exists a DCJ operation on two dashed edges with the same colour that increases the cycle number by one. When a connecting edge does not form a loop, apply a DCJ operation to loop it. Then arbitrarily select a solid edge from a cycle with size more than 2, apply a DCJ operation to make a dashed edge parallel to it. Thus with a number $mN_{\mathcal{G}} + k - c(\overset{\circ\bullet}{M})$ DCJ operations, we can transform $\overset{\circ\bullet}{M}$ to $\overset{\circ\circ}{M}$ or *vice versa*.

□

Proposition 5. *An arbitrary DCJ operation on non-0 edges in a median graph changes the cycle number by 1, 0, or -1.*

Proof. If the two edges belong to one cycle, it will either split into two cycles or remain as a single cycle. If the two edges belong to two cycles, then they will be joined into one cycle. □

Theorem 1. *If H is a (strongly) adequate subgraph of MBG B and E is a H-crossing candidate matching, then there is a candidate matching E' which is not H-crossing, with $c_{E'}(B) \geq c_E(B)$ (or $c_{E'}(B) > c_E(B)$).*

Proof. 1. From the median graph $M = B \cup E$, construct the twin median graph $\overset{\circ\bullet}{M}$ and twin MBG $\overset{\circ\bullet}{B}$ by shrinking 0-edges not incident to H (E_0) and expanding 0-edges incident to H twice (E_2). Denote the subgraph of $\overset{\circ\bullet}{M}$ induced by $\mathbf{V}(H)$ as F. Then $c_E(B) = c_{E_0}(B) + c_{\circ\bullet}(\overset{\circ\bullet}{B})$.

2. Construct the symmetrical median graph $\overset{\circ\circ}{M}$ with $F = \overline{F}$ and F also a (strongly) adequate subgraph.

3. Since F is a (strongly) adequate subgraph, there exists a 0-matching D of F satisfying $c_D(F) \geq \frac{1}{2}mN_{\mathcal{G}}$ (or $c_D(F) > \frac{1}{2}mN_{\mathcal{G}}$).

4. Replace the 0-matching in $\overset{\circ\circ}{M}$ by two copies of D, one on F and one on $\overline{F}$. Denote the 0-matching as $2D$ and denote the resultant median graph as $\overset{\circ\circ}{B} \cup 2D$, with $c_{2D}(\overset{\circ\circ}{B}) \geq mN_{\mathcal{G}}$ (or $> mN_{\mathcal{G}}$).

5. Transform $\overset{\circ\circ}{B}$ to $\overset{\circ\bullet}{B}$ by $mN_{\mathcal{G}} + k - c(\overset{\circ\bullet}{M})$ DCJ operations on $\overline{F}$ in $\overset{\circ\circ}{B}$. So $c_{2D}(\overset{\circ\bullet}{B}) \geq c_{\overset{\circ\bullet}{E}}(\overset{\circ\bullet}{B})$ (or $c_{2D}(\overset{\circ\bullet}{B}) > c_{\overset{\circ\bullet}{E}}(\overset{\circ\bullet}{B})$).

6. Shrink the newly added sets of $N_{\mathcal{G}}$ parallel edges in $\overset{\circ\bullet}{B}$ and reverse the shrinking operations on E_0 in step 1, to recover the MBG B. Then the 0-matching $2D$ becomes the candidate matching E' and the new median graph becomes $M' = B \cup E'$. Then $c_{E'}(B) = c_{2D}\overset{\circ\bullet}{B} + c_{E_0}(B)$. Thus $c_{E'}(B) \geq c_E(B)$ (or $c_{E'}(B) > c_E(B)$).

□

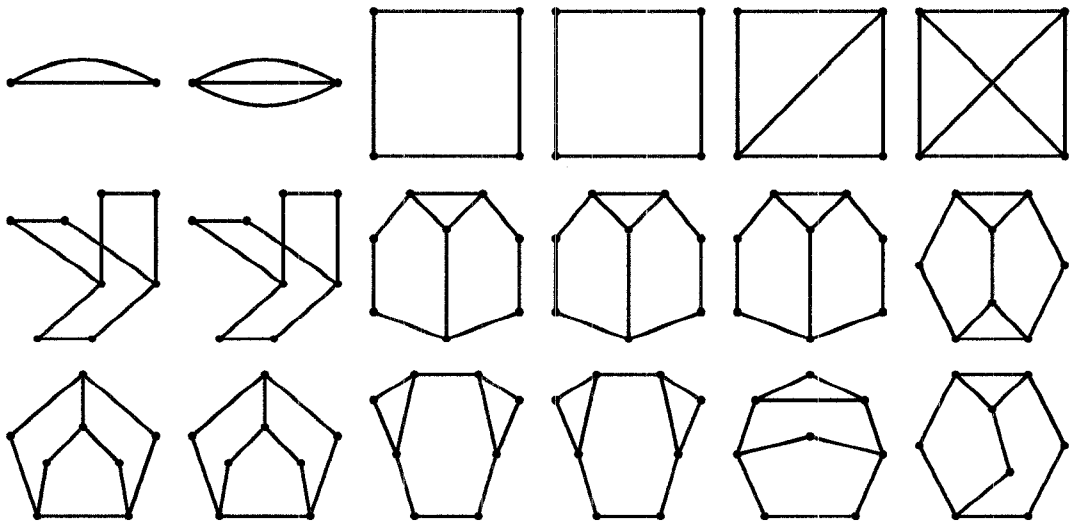


Figure 7: Simple adequate subgraphs of size 1, 2 and 4 for MBGs on three genomes.

Theorem 2. *Any adequate subgraph is a decomposer. A strongly adequate subgraph is a strong decomposer.*

Proof. For an adequate subgraph there must be a optimal matching that is not crossing. Otherwise by Theorem 1, from the optimal crossing matching, we can construct a candidate matching that is not crossing and has at least as many cycles. Thus the adequate subgraph is a decomposer.

For a strongly adequate subgraph, the non-crossing candidate matchings are always better than the corresponding crossing candidate matchings. Then the optimal matchings cannot be crossing matchings. The strongly adequate subgraph is thus a strong decomposer. □

4 Median calculation incorporating MBG decomposition

As adequate subgraphs are the key to decompose the median problems, we need to inventory them before making use of them. It turns out that it is most useful to limit this project to simple adequate graphs. Non-simple adequate graphs are both harder to enumerate and harder to use, and are likely to have simple ones embedded in them, which serve the same general purpose [11]. By exhaustive search, we have found all simple adequate graphs of size < 6; these are depicted in Figure 7. Though we have some of size 6, it would be a massive undertaking to compile the complete set with current methods.

Our basic algorithm for solving the median problem is a branch and bound, where edges of colour 0 are added at each step; we omit the details of procedures we use to increase the effectiveness of the bounds. To make use of the adequate subgraph

ρ/n n	10	20	30	40	50	60	80	100	200	300	500	1000	2000	5000
0.1	10	10	10	10	10	10	10	10	10	10	10	10	10	10
0.2	10	10	10	10	10	10	10	10	10	10	10	10	10	10
0.3	10	10	10	10	10	10	10	10	10	10	10	10	1	
0.4	10	10	10	10	10	10	10	10	0	0				
0.5	10	10	10	10	10	10	4	0						
0.6	10	10	10	10	9	6								
0.7	10	10	10	10	6									
0.8	10	10	10	10	6									
0.9	10	10	10	10	4									
1.0	10	10	10	8	2									

Table 1: The number of runs, out of ten, where the median was found in less than 10 minutes on a MacBook, 2.16GHz, on one CPU.

theory we have developed, at each step we search for such an inventoried subgraph before adding edges, and if one is found, we carry out a decomposition and then solve the resulting smaller problem(s) [11].

5 Experimental results

To see how useful our method is on a range of genomes, we undertook experiments on sets of three random genomes. Our JAVA program included a search for adequate subgraphs followed by decomposition at each step of a branch and bound algorithm to find the maximum number of cycles. We varied the parameters n and $\pi = \rho/n$, where ρ was the number of random reversals applied to the ancestor $I = 1, \ldots, n$ independently to derive three different genomes.

run	speedup factor	run time		number of edges			
		with AS	no AS	AS1	AS2	AS4	AS0
1	41,407	4.5×10^{-2}	1.9×10^{3}	53	39	8	0
2	85,702	3.0×10^{-2}	2.9×10^{3}	53	34	12	1
3	2,542	5.4×10^{0}	1.4×10^{4}	56	26	16	2
4	16,588	3.9×10^{-2}	6.5×10^{2}	58	42	0	0
5	$> 10^{6}$	5.9×10^{2}	stopped	52	41	4	3
6	199,076	6.0×10^{-3}	1.2×10^{3}	56	44	0	0
7	6,991	2.9×10^{-1}	2.1×10^{3}	54	33	12	1
8	$> 10^{6}$	4.2×10^{1}	stopped	57	38	0	5
9	1,734	8.7×10^{0}	1.5×10^{4}	65	22	8	5
10	855	2.1×10^{0}	1.8×10^{3}	52	38	8	2

Table 2: Speedup due to adequate subgraph (AS) discovery. Three genomes are generated from the identity genome with $n = 100$ by 40 random reversals. Time is measured in seconds. Runs were halted after 10 hours. AS1, AS2, AS4, AS0 are the numbers of edges in the solution median constructed consequent to the detection of adequate subgraphs of sizes 1, 2, 4 and at steps where no adequate subgraphs were found, respectively.

5.1 The effects of n and $\pi = \rho/n$ on the proportion of rapidly solvable instances

Table 1 shows that relatively large instances can be solved if ρ/n remains at 0.3 or less. It also shows that for small n, the median is easy to find even if ρ/n is large enough to effectively scramble the genomes.

5.2 The effect of adequate subgraph discovery on speed-up

Table 2 shows how the occurrence of adequate subgraphs can dramatically speed up the solution to the median problem, generally from more than a half an hour to a fraction of a second.

5.3 Time to solution

Our results in Section 5.1 suggest a rather abrupt cut-off in performance as n or ρ/n become large. We explore this in more detail by focusing on the particular parameter values $n = 1000$ and $\rho/n = .31$. Figure 8 shows how the instances are divided into a rapidly solvable fraction and a relatively intractable fraction, with very few cases in between.

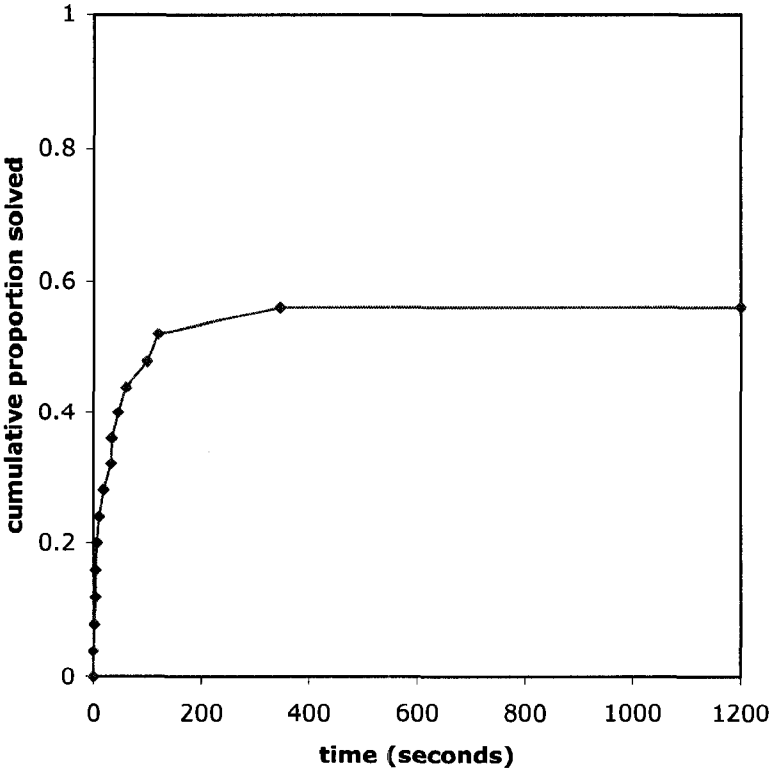


Figure 8: Cumulative proportion of instances solved, by run time. $n = 1000$, $\rho/n = .31$. More than half are solved in less than 2 minutes; almost half take more than 20 minutes.

6 Conclusion

In this paper we have demonstrated the potential of adequate subgraphs for greatly speeding up the solution of realistic instances of the median problem. Many improvements seem possible, but questions remain. If we could inventory non-simple adequate graphs, or all simple adequate graphs of size 6 or more, could we achieve significant improvement in running time? It may well be that the computational costs of identifying larger adequate graphs within MBGs would nullify any gains due to the additional decompositions they provided.

Acknowledgements

We thank the reviewers for their comments and suggestions. Research supported in part by a grant to DS from the Natural Sciences and Engineering Research Council of Canada (NSERC). DS holds the Canada Research Chair in Mathematical Genomics.

References

[1] Adam Z, Sankoff D. 2008. The ABCs of MGR with DCJ. *Evol Bioinform* **4**, 69–74.

[2] Bourque G, Pevzner PA. 2002. Genome-scale evolution: Reconstructing gene orders in the ancestral species. *Genome Res* **12**, 26–36.

[3] Caprara A. 2003. The reversal median problem. *INFORMS J Comput* **15** 93–113.

[4] Hannenhalli S, Pevzner PA. 1999. Transforming cabbage into turnip: Polynomial algorithm for sorting signed permutations by reversals. *JACM* **46**, 1–27.

[5] Lenne R, Solnon C, Stützle T, Tannier E, Birattari M. 2008. Reactive stochastic local search algorithms for the genomic median problem. *EvoCOP 2008* LNCS **4972**, 266–276.

[6] Moret BME, Siepel AC, Tang J, Liu T. 2002. Inversion medians outperform breakpoint medians in phylogeny reconstruction from gene-order data. *WABI 2002*, LNCS **2452**.

[7] Sankoff D, Blanchette M. 1998. Multiple genome rearrangement and breakpoint phylogeny. *J Comput Biol* **5**, 555–570.

[8] Sankoff D, Morel C, Cedergren RJ. 1973. Evolution of 5S RNA and the non-randomness of base replacement. *Nature New Biol* **245**, 232–234.

[9] Tannier E, Zheng C, Sankoff D. 2008. Multichromosomal median and halving problems. *WABI 2008.*

[10] Tesler G. 2002. Efficient algorithms for multichromosomal genome rearrangements. *JCSS* **65**, 587–609.

[11] Xu, AW. 2008. A fast and exact algorithm for the median of three problem—a graph decomposition approach. *RECOMB CG 2008* Submitted.

[12] Yancopoulos S, Attie O, Friedberg R. 2005. Efficient sorting of genomic permutations by translocation, inversion and block interchange. *Bioinform* **21**, 3340–3346.

Chapter 7

A fast and exact algorithm for the median of three problem—a graph decomposition approach

Submitted for publication. Work of Andrew Wei Xu.

Abstract

In a previous paper, we have shown that adequate subgraphs can be used to decompose multiple breakpoint graphs, achieving a dramatic speedup in solving the median problem. In this paper, focusing on the median of three problem, we prove more important properties about adequate subgraphs with rank 3 and discuss the algorithms inventorying simple adequate subgraphs. After finding simple adequate subgraphs of small sizes, we incorporate them into ASMedian, an algorithm to solve the median of three problem. Results on simulated data show dramatic speedup so that many instances can be solved very quickly, even ones containing hundreds or thousands of genes.

1 Introduction

The median problem[3, 7, 8, 6, 2, 1] for genomic rearrangement distances is NP-hard [4, 9]. Algorithms have been developed to find exact solutions for small instances [4, 6] and there are rapid heuristics of varying degrees of efficiency and accuracy [2, 1, 5]. In a previous paper [10], with the aim of finding a decomposition method to reduce the size of the problem, we introduced the notion of adequate subgraph and showed how they lead to such a decomposition. By applying this method recursively, the size of the problem is effectively reduced. In this paper, we focus on the median of three problem, which is to find a genome q with smallest total distance $\sum_{1\leq i\leq 3} d(q, g_i)$ for any given three genomes g_1, g_2, g_3.

Because of its simple structure, we choose to work with DCJ distance[11] $d = n-c$ as most likely to yield non-trivial mathematical results, where n is the number of genes in each genome (assuming that they have the same gene content) and c is the number of cycles in the breakpoint graph. We require genomes to consist of one or more

circular chromosomes, but our results could be extended to genomes with multiple linear chromosomes.

In Section 2 several related concepts are defined, such as *breakpoint graph* and *adequate subgraph*. In Section 3 some important properties about adequate subgraphs of rank 3 are proved. We discuss the problem of inventorying simple adequate subgraphs in Section 4. Then in Section 5, we give an algorithm ASMedian to solve the median problem. Results on simulated data are given and discussed in Section 6.

2 Graphs, subgraphs and more

2.1 Breakpoint graph

We construct the breakpoint graph of two genomes by representing each gene by an ordered pair of vertices, adding coloured edges to represent the adjacencies between two genes, red edges for one genome and black for the other.

In a genome, every gene has two adjacencies, one incident to each of its two endpoints, since it appears exactly once in that genome. Then in the breakpoint graph, every vertex is incident to one red edge and one black one. Thus the breakpoint graph is a 2-regular graph which automatically decomposes into a set of alternating-colour cycles.

The edges of one colour form a perfect matching of the breakpoint graph, which we will simply refer to as a *matching*, unless otherwise specified. By the red matching, we mean the matching consisting of all the red edges.

The **size** of the breakpoint graph is defined as half the number of vertices it contains, which equals to the size of its matchings and the number of gens in each genome.

2.2 Multiple breakpoint graph and median graph

The breakpoint graph extends naturally to a multiple breakpoint graph (MBG), representing a set $\mathcal{G}$ of three or more genomes. The number of genomes[1] $N_{\mathcal{G}} \geq 3$ in $\mathcal{G}$ is called the **rank** of the MBG, which is also its edge chromatic number. The colours assigned to the genomes are labeled by the integers from 1 to $N_{\mathcal{G}}$. The **size** of an MBG or its subgraph is also defined as half the number of vertices it contains.

For a candidate median genome, we use a different colour for its matching E, namely colour 0. Adding E to the MBG results in the **median graph**. The set of all possible candidate matchings is denoted by $\mathcal{E}$.

The 0-i **cycles** in a median graph with matching E, numbering $c(0,i)$ in all, are the cycles where 0-edges and i edges alternate. Let $c_E = \sum_{1 \leq i \leq 3} c(0,i)$. Then $c_{\max} = \max\{c_E : E \in \mathcal{E}\}$ is the maximum number of cycles that can be formed from the MBG. *Minimizing the total DCJ distance in the median problem is equivalent to finding an optimal 0-matching E, i.e., with $c_E = c_{\max}$.*

2.3 Subgraphs

Let $\mathbf{V}(G)$ and $\mathbf{E}(G)$ be the sets of vertices and edges of a regular graph G. A **proper subgraph** H of G is one where $\mathbf{V}(H) = \mathbf{V}(G)$ and $\mathbf{E}(H) = \mathbf{E}(G)$ do not both hold at the same time. An **induced subgraph** H of G is the subgraph which satisfies the property that if $x, y \in \mathbf{V}(H)$ and $(x,y) \in \mathbf{E}(G)$, then $(x,y) \in \mathbf{E}(H)$.

In this paper, we will focus on the induced proper subgraphs of MBGs, with even numbers of vertices. Through this paper, the size of a subgraph is denoted by m. For a proper induced subgraph H, $\mathcal{E}(H)$ is the set of all its perfect 0-matchings $E(H)$. The number of cycles determined by H and $E(H)$ is $c_{E(H)}(H)$, and $c_{\max}(H)$ is the

[1]for the median of three problem, this number is just 3

maximum number of cycles that can be formed from H. A 0-matching $E^\star(H)$ with $c_{E^\star(H)}(H) = c_{\max}(H)$ is called an optimal partial 0-matching, and $\mathcal{E}^\star(H)$ is the set of such 0-matchings.

2.4 Non-crossing 0-matchings and decomposers

For a subgraph H of an MBG G, a potential 0-edge would be **H-crossing** if it connected a vertex in $\mathbf{V}(H)$ to a vertex in $\mathbf{V}(G) - \mathbf{V}(H)$. A candidate matching containing one or more H-crossing 0-edges is an H-crossing.

An MBG subgraph H is called a **decomposer** if for any MBG containing it, there is an optimal matching that is not H-crossing. It is a **strong decomposer** if for any MBG containing it, all the optimal matchings are not H-crossing.

2.5 Adequate and strongly adequate subgraphs

A connected MBG subgraph H of size m is an **adequate subgraph** if $c_{\max}(H) \geq \frac{1}{2}mN_{\mathcal{G}}$; it is **strongly adequate** if $c_{\max}(H) > \frac{1}{2}mN_{\mathcal{G}}$. For the median of three problem, an adequate subgraph of rank 3 is a subgraph with $c_{\max}(H) \geq \frac{3m}{2}$ and a strongly adequate subgraph of rank 3 is one with $c_{\max}(H) > \frac{3m}{2}$.

A (strongly) adequate subgraph H is **simple** if it does not contain another (strongly) adequate subgraph as an induced subgraph; deleting any vertex from H will destroy its adequacy.

Adequate subgraphs enable us to decompose the MBG into a set of smaller ones, as in the next theorem.

Theorem 1. *[10] Any adequate subgraph is a decomposer. Any strongly adequate subgraph is a strong decomposer.*

3 The properties of simple adequate subgraphs of rank 3

In this section, we prove other important properties about simple adequate subgraphs of rank 3. Multiple edges in MBGs are the simple adequate subgraphs of size one, which are the only exceptions to many of the properties stated below.

3.1 More properties about adequate subgraphs of rank 3

Lemma 1. *The vertices of simple adequate subgraphs of rank 3 have degrees either 2 or 3.*

Proof. Since the MBG for the median of three problem is 3-regular, the vertex degrees of its induced subgraphs can only be 1, 2 or 3.

The lemma is true for parallel edges (the smallest simple adequate subgraphs), where the vertex degrees are 2 or 3. For simple adequate subgraphs of size two or more, we prove by contradiction that they can not contain vertices of degree 1.

Assume there is a simple adequate subgraph H of size m containing a vertex x of degree 1. In one of the optimal 0-matchings of H, x is connected to vertex y by a 0-edge e, and e appears only in one of the colour-alternating cycles. By deleting edge e and vertices x, y, only that cycle is destroyed. Because of its adequacy, the maximum number of cycles formed with H is at least $\frac{3m}{2}$. So for the resultant subgraph F of size $m-1$, the maximum number of cycles can be formed is at least $\frac{3m}{2} - 1 = \frac{3(m-1)}{2} + \frac{1}{2} > \frac{3(m-1)}{2}$. Therefore F, as a subgraph of H, is also an adequate subgraph, which contradicts the assumption that H is simple.

So the vertex degrees in a simple adequate subgraph can only be 2 or 3. □

Lemma 2. *Except for multiple edges, the size of a simple adequate subgraph of rank 3 is even.*

Proof. Suppose there is an odd-sized simple adequate subgraph H of size $2k+1$. Because of its adequacy, the maximum number of cycles formed with H is at least $\left\lceil \frac{3(2k+1)}{2} \right\rceil = 3k+2$. Since H is an proper subgraph, there exists a vertex x with degree 2. Suppose 0-edge e is incident to x in one of H's optimal 0-matchings. By deleting e and the corresponding vertices, two colour-alternating cycles are destroyed. Then for the resultant subgraph F of size $2k$, the maximum number of cycles formed with F is at least $3k = \frac{3}{2} \times 2k$. Hence F, as a subgraph of H, is also an adequate subgraph, which contradicts the simplicity of H. □

Lemma 3. *Except for multiple edges, the maximum number of cycles of a simple adequate subgraph of rank 3 is exactly $\frac{3m}{2}$, where m is its size.*

Proof. Because of Lemma 2, we only need to consider even-sized simple adequate subgraphs. Suppose H is a simple adequate subgraph of size $2k$, with which the maximum number of cycles formed is at least $3k+1$. Then by deleting a 0-edge connecting to a degree 2 vertex, the size of the subgraph decreases by 1 and the number of cycles decreases by 2. So H contains another adequate subgraph of size $2k-1$ whose maximum number of cycles is at least $3k-1 = \lceil \frac{3}{2}(2k-1) \rceil$, which contradicts the simplicity assumption for H. □

3.2 There are infinite many simple adequate subgraphs

In this subsection we show that there are infinitely many adequate subgraphs, by proving the number of simple adequate subgraphs is infinite, which follows from the infinite size of a special family of simple adequate subgraphs—the mirrored-tree graphs.

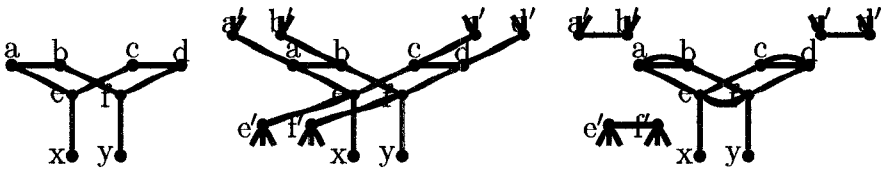


Figure 1: (a) Illustration of a double-Y end and it is connected to the remaining subgraph only through vertices x and y; (b) shows a 0-matching not containing 0-edges $(a, b), (c, d), (e, f)$; (c) shows another 0-matching obtained by applying 3 DCJ operations to the 0-matching in (b), that does contain those three 0-edges and forms more colour-alternating cycles.

Definition 1. ***An mirrored-tree graph:*** *two identical 3-edge-coloured binary trees with corresponding pairs of leaf vertices connected by simple edges. Being an MBG subgraph, the size of an mirrored-tree graph is defined as half the number of its vertices, which also is the number of vertices contained in each tree.*

Proposition 1. *1. Any binary tree containing more than one vertex must have even size;*

2. for a binary tree with m vertices, there are $\frac{m}{2}+1$ leaf vertices (with degree 1) and $\frac{m}{2}-1$ inner vertices (with degree 3).

3. the total number of edges is $m-1$.

Proposition 2. *For a mirrored-tree graph of size m, there are $\frac{5m}{2}-1$ edges in total; $\frac{m}{2}+1$ of them connect the two binary trees and $2m-2$ of them lie in the trees.*

Definition 2. ***Double-Y end:*** *A mirrored-tree graph of size 4, with one connecting edge missing, as illustrated by Figure 1(a). Being a part of an MBG subgraph, it is connected to the remaining graph through the two vertices of degree one.*

Lemma 4. *If a double-Y end appears in an MBG subgraph H, then the 0-edges (a, b), (c, d) and (e, f) connecting corresponding vertices of the two identical trees, must exist in any optimal 0-matching of H, as illustrated by Figure 1(c).*

Proof. In an optimal 0-matching of H, if any of the three 0-edges (a, b), (c, d) and (e, f) appears, the other two 0-edges must also exist. Then only one case is left to disprove—that none of these 0-edges appears in some optimal 0-matching, as illustrated by Figure 1(b).

Figure 1(c) is obtained by three DCJ operations on the 0-edges of Figure 1(b), creating 0-edges (a, b), (c, d) and (e, f). By comparison we can see that: a′, b′ are connected by a green-black alternating path and c′, d′ are connected by a blue-black alternating path in both figures. So they are involved in the same number of cycles in both figures.

Apart from these, Figure 1(b) contains another 6 paths, which can form at most 6 colour-alternating cycles; Figure 1(c) contains 4 cycles as well as another 6 paths of 3 different colours which will form at least 3 cycles, summing up to a total of 7 cycles or more. So Figure 1(c) forms more cycles than Figure 1(b).

For the cases where vertices a', b', c', d', e', f' are incident to different set of edges, the same result still holds.

Since 0-matchings of H containing 0-edges (a, b), (c, d) and (e, f) have more cycles than 0-matchings not containing them, these three 0-edges must exist in any optimal 0-matchings. □

Theorem 2. *With a mirrored-tree graph of size m, we can form a maximum of $\frac{3m}{2}$ colour alternating cycles, hence it is an adequate subgraph; Furthermore, it does not contain any smaller adequate subgraphs, so it is a simple adequate subgraph.*

Proof. We first prove that there is a 0-matching of the mirrored-tree graph, forming

$\frac{3m}{2}$ colour alternating cycles. This is just the set of 0-edges connecting the corresponding vertices of the two trees. With this 0-matching, each non-0 edge connecting two trees makes a cycle by itself; and the edges on the tree form cycles of size 2 with the corresponding edges on the other tree. From Proposition 2, there are $\frac{m}{2}+1$ edges connecting trees and $2m-2$ edges on the trees, the total number of cycles is $\frac{3m}{2}$.

Next we show that is the only optimal 0-matching. For any binary tree, since the number of leaf vertices is larger than the number of inner vertices by 2, there is always an inner vertex being connected to two leaf vertices. In the corresponding mirrored-tree graph, this gives a double-Y end.

For a mirrored-tree graph H, we add two 0-edges parallel to the connecting edges of its double-Y end as 0-edges (a, b) and (c, d) in Figure 1(c). Then by shrinking them, H becomes a quasi mirrored-tree graph[2] of smaller size, containing double-Y ends or quasi double-Y ends[3]. By applying this procedure of adding and shrinking 0-edges to the (quasi) double-Y ends recursively, H finally becomes a three-parallel edge. Since in each step the new added 0-edges must appear in all optimal 0-matchings, the resultant perfect 0-matching is the only optimal 0-matching of H.

The symmetrical structure of mirrored-tree graphs leads to colour-alternating cycles of smallest sizes—1 and 2 only. In detecting whether a mirrored-tree graph H contains any smaller adequate subgraphs, it is sufficient to only consider its subgraphs with symmetrical structures. Using reasoning similar to the above paragraphs, it can be shown that the optimal 0-matchings for these symmetrical subgraphs of H are the subsets of the 0-edges in the optimal 0-matching of H. However none of these symmetrical subgraphs of H can form cycles of $\frac{3}{2}$ times their sizes. So the mirrored-tree graphs are simple adequate subgraphs. □

[2]in which, there may be multiple edges connecting the two identical trees.

[3]the ones whose connecting edges might be multiple edges. Obviously the conclusion in Lemma 4 also applies to quasi double-Y ends.

Theorem 3. *There are infinitely many simple adequate subgraphs.*

Proof. Since there are binary trees of arbitrary large size, which give mirrored-tree graphs with arbitrary large (even) size. Also because mirrored-tree graphs are simple adequate subgraphs, there exist simple adequate subgraphs with arbitrary large (even) size. □

4 Inventorying Simple Adequate Subgraphs

4.1 It is practical to use simple adequate subgraphs of small sizes

Before using the adequate subgraphs to reduce the search space for finding an optimal 0-matching, we need to inventory the adequate subgraphs. Theorem 3.2 states that there are infinitely many simple adequate subgraphs, hence infinitely many adequate subgraphs, so it is impossible to inventory all of them and use them to decompose the median problems. However, it is practical to work on simple adequate subgraphs of small sizes, as justified by the following:

1. There are much fewer simple adequate subgraphs. And many non-simple adequate subgraphs can be decomposed into several simple adequate subgraphs embedded in each other. Hence many non-simple ones can be detected through the constituent simple ones.

2. The algorithms to inventory simple adequate subgraphs for a given size require more than exponential time in their size.

3. The total number of simple adequate subgraphs increases dramatically as the size increases. The complexity of the algorithm to detect the existence of a

given simple adequate subgraph also increases accordingly. Combining these two factors, we conclude that it is prohibitively expensive to detect the existence of simple adequate subgraphs of large sizes.

4. Simple adequate subgraphs of small sizes exist with much higher probability than subgraphs of greater size on random MBGs. The details will be given in the full version of this paper.

4.2 Algorithms to inventory simple adequate subgraphs

To enumerate the simple adequate subgraphs, we need to search among all the MBG subgraphs, which consist of (perfect or non-perfect) matchings of three colours. In order to count the number of cycles, the perfect 0-matchings must also be enumerated. So the algorithms need to work on graphs consisting of 4 matchings, hence the problem is computationally costly.

Our simple adequate subgraph inventorying algorithm uses a depth-first search method. The graph grows by adding an edge at each step. It is backtracked whenever the current graph contains a smaller simple adequate subgraph and then restrained on another path to grow the graph until all subgraphs have been searched.

To speed up the algorithm, we adopt several useful methods and techniques:

1. Only inventory simple adequate subgraphs of even sizes, as a result of Lemma 2.

2. Fix the 0-matching. Any median subgraph is isomorphic to $\frac{(2m)!}{2^m m!} - 1$ other median subgraphs by permuting the 0-edges.

3. Only allow the graphs whose number of 1-edges is no less than the number of 2-edges and the number of 2-edges is no less than the number of 3-edges,

because of the isomorphism associated with the permutation of colours.

4. Every vertex must be incident to 2 or 3 non-0-edges, because of Lemma 1.

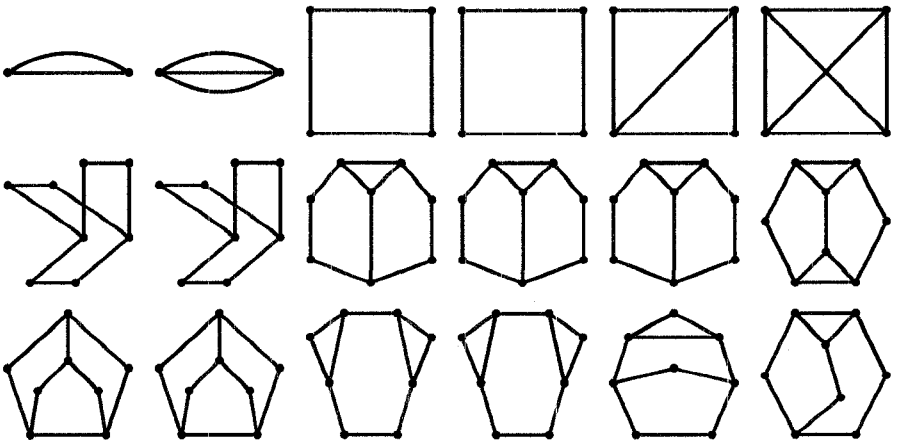


Figure 2: Simple adequate subgraphs of size 1, 2 and 4 for MBGs on three genomes.

4.3 Simple adequate subgraph enumerated

In Figure 2, the simple adequate subgraphs of size 1, 2 and 4 are listed. Each subgraph represent a class of subgraphs isomorphic under the permutations of vertices and colours.[4]

5 Solving the median of three problem by recursively detecting simple adequate subgraphs

Our algorithm using adequate subgraphs to decompose the median problems is called **ASMedian**. It adopts a branch-and-bound method to find an optimal 0-matching for any given MBG. At any intermediate step during the branch-and-bound search,

[4]Strictly speaking, Figures (a) and (f) are not proper subgraphs of a connected MBG.

an intermediate configuration (IC for short) is constructed, containing a partial 0-matching and an intermediate MBG (iMBG for short) resulted by a process of edge-shrinking [10] of that partial 0-matching from the original MBG. The algorithm keeps a list of unexamined ICs $\mathcal{L}$, initially just consisting of the original MBG.

At each step, from $\mathcal{L}$ an unexamined IC with the largest upper bound is selected to examine. According to whether an inventoried simple adequate subgraph exists in that iMBG and what simple adequate subgraph it is, a number of new ICs are generated, containing smaller and non-empty iMBGs and expanded partial 0-matchings. Then we update U the largest upper bound of all unexamined ICs and c^* the largest cycle number encountered so far. We prune the ICs whose upper bounds are no larger than c^*. The algorithm stops when $c^* \geq U$ or no unexamined ICs remain. Then c^* is the maximum cycle number for the original MBG, and the corresponding 0-matching is an optimal 0-matching.

5.1 Examining the intermediate MBGs

Definition 3. ***The inquiry set*** *is the set of simple adequate subgraphs (of small sizes) for which the ASMedian algorithm looks on the intermediate MBGs. For a specific algorithm, the inquiry set is given as a parameter.*

The iMBG of the selected IC is examined to see the existence of any simple adequate subgraph in the inquiry set. If one of such subgraphs H exists, then we know there is an optimal 0-matchings of iMBG which is non-H-crossing. This 0-matching can be divided into two parts: a 0-matching of H and a partial 0-matching of the remaining intermediate MBG.

A **major set** of H is the minimal set of 0-matchings of H, which guarantees that at least one of them must appear in an optimal 0-matchings of the MBG, without the

Algorithm 1: ASMedian

Input: three genomes containing any number of circular chromosomes
Output: the median genome and the maximum number of cycles c

construct the MBG, assign its upper bound u to U and its lower bound to c^* and push it into the unexamined list $\mathcal{L}$;
while $U > c^*$ **and** $\mathcal{L}$ **is not empty do**
  pop out an IC with $u = U$ from $\mathcal{L}$;
  if an adequate subgraph H **is found in the iMBG of that IC then**
    set the major set as the one of H;
  else
    select the vertex with smallest label and set the major set as the set containing all 0-edges incident to that vertex;
  generate a set of new ICs with their partial 0-matchings are expanded to include a 0-matching in the major set and their iMBGs as the resultant graphs of shrinking these partial 0-matchings;
  update U and c^*;
  if c^* **gets updated then** Remove all the ICs with $u \leq c^*$ in $\mathcal{L}$;
  push the new generated ICs with $u > c^*$ into $\mathcal{L}$;
// `the maximum cycle number has been found;`
set c as c^* and construct the median genome from the optimal 0-matching obtained;
return c and the median genome;

knowledge of the remaining part of the MBG (as will be shown else where). The size of the major set is denoted by μ. Since the inquiry set is given in advance, the major sets for the simple adequate subgraphs are also known in advance. Then according to this major set, μ new ICs will be generated with smaller iMBGs, each resulting from the shrinking of a 0-matching of H in the major set from the iMBG of the currently selected IC.

When no simple adequate subgraph in the inquiry set exists, the vertex v with the smallest remaining label is selected. The nominal major set of size $2\tilde{n} - 1$ is constructed, where $\tilde{n}$ is the size of the iMBG — just the set of 0-edges incident to

v (here each partial 0-matching is just a 0-edge). Then $2\tilde{n} - 1$ new ICs are created accordingly.

If the inquiry set is chosen as all simple adequate subgraphs of sizes 1, 2 and 4, then the sizes of their major sets are just one, except for one case where it is 2. It can be seen that whenever a simple adequate subgraph is detected, the search space is roughly reduced by a factor of $\tilde{n}$, $\tilde{n}^2$ or $\tilde{n}^4$.

5.2 The lower bound and the upper bound

For each intermediate configuration, the ASMedian algorithm calculates its upper bound and prunes it if the value is no larger than c^*—the maximum number of cycles encountered so far.

Because of the search schema we use (see next subsection), it takes a while for the algorithm to reach any perfect 0-matching. Due to the fact that the number of cycles formed by partial 0-matchings are small, to calculate c^* from them will make the pruning procedure very inefficient. Instead, for each intermediate configuration, a tight lower bound is calculated and c^* takes the maximum of these lower bounds and the encountered cycle numbers.

Since the DCJ distance is a metric measure, for any median of three problem, there is an associated lower bound for the total distance. Assume the three known genomes are labeled as 1, 2, 3, and $d_{1,2}, d_{1,3}, d_{2,3}$ denote the pairwise distances and $c_{1,2}, c_{1,3}, c_{2,3}$ denote the cycle numbers between any two pairs. The lower bound for the total distance is $d \geq \frac{d_{1,2}+d_{1,3}+d_{2,3}}{2}$. Because $d_{i,j} = n - c_{i,j}$, then we get an upper bound for the total cycle number,

$$c \leq \frac{3n}{2} + \frac{c_{1,2} + c_{1,3} + c_{2,3}}{2}. \tag{5.1}$$

To find a lower bound for the total cycle number, we can set the 0-matching to any of the matchings representing the three known genomes and take largest total cycle number of the three as the lower bound, so that

$$c \geq c_{1,2} + c_{1,3} + c_{2,3} - \min\{c_{1,2}, c_{1,3}, c_{2,3}\}. \tag{5.2}$$

For any IC, by adding $\tilde{c}$, the number of cycles formed by its partial 0-matching, to the lower bound and upper bound of its intermediate MBG, we get the upper bound and lower bound of this IC, denoted by u and l correspondingly.

$$u = \tilde{c} + \frac{3\tilde{n}}{2} + \frac{\tilde{c}_{1,2} + \tilde{c}_{1,3} + \tilde{c}_{2,3}}{2} \tag{5.3}$$

$$l = \tilde{c} + \tilde{c}_{1,2} + \tilde{c}_{1,3} + \tilde{c}_{2,3} - \min\{\tilde{c}_{1,2}, \tilde{c}_{1,3}, \tilde{c}_{2,3}\}. \tag{5.4}$$

The IC and all ICs derived from it, are referred as the **parent** IC and the **child** ICs. A non-increasing property holds between the upper bounds of the parent IC and the child ICs.

Lemma 5. *The upper bounds of the child ICs are never larger than the upper bound of their parent IC.*

Proof. Suppose a child IC is obtained from the parent IC by adding a 0-edge e. We first inspect the possible effects on $\tilde{c}$ and $\tilde{c}_{1,2}$ of adding e to the iMBG of the parent IC.

a If e connects two 1-2 cycles, then the two cycles will be merged into one. Then $\tilde{c}$ remains the same and $\tilde{c}_{1,2}$ decreases by 1;

b if e parallels a 1-edge (or a 2-edge), then one 0-1 cycle of size 1 is formed and the 1-2 cycle containing that edge becomes a shorter one. So that $\tilde{c}$ increases

by 1 and $\tilde{c}_{1,2}$ remains the same;

c if e connects two vertices of the same 1-2 cycle, not paralleling any edges, then this 1-2 cycle may be split into two or remain with a smaller size. Therefore $\tilde{c}$ remains the same and $\tilde{c}_{1,2}$ increases by 0 or 1.

Since the size of the iMBG in the child IC decreases by 1, $\frac{3\tilde{n}}{2}$ decreases by $\frac{3}{2}$. As long as $\tilde{c} + \frac{\tilde{c}_{1,2}+\tilde{c}_{1,3}+\tilde{c}_{2,3}}{2}$ does not increase more than $\frac{3}{2}$, u never increases.

1 If e does not parallel any edge, then $\tilde{c}$ remains the same, and each of $\tilde{c}_{1,2}, \tilde{c}_{1,3}, \tilde{c}_{2,3}$ increases at most by 1. So u does not increase;

2 if e parallels one edge, $\tilde{c}$ increases by 1 and only one of $\tilde{c}_{1,2}, \tilde{c}_{1,3}, \tilde{c}_{2,3}$ increases at most by 1. So u does not increase;

3 if e parallels two edges, $\tilde{c}$ increases by 2 and the cycle formed by the parallel edges is destroyed and the other two terms of $\tilde{c}_{1,2}, \tilde{c}_{1,3}, \tilde{c}_{2,3}$ remain the same. So u does not change;

4 e parallels three edges, $\tilde{c}$ increases by 3 and the three cycles formed by the parallel edges are destroyed. So u remains the same.

So the upper bound of the child ICs are never larger than the upper bound of their parent IC.

□

The algorithm maintains an overall upper bound U which is the maximum upper bound of all unexamined ICs. Another global variable, as mentioned before, is the largest total cycle number or lower bound c^* found so far. Obviously the maximum total cycle number c of the original MBG lies between c^* and U.

5.3 The optimistic search schema

Our algorithm is neither a strict depth-first nor a strict breadth-first search schema, but follows an "optimistic" search strategy . From the list of all unexamined ICs, we select the one with the largest upper bound. The intuition behind this is, the ICs with larger upper bounds are more likely to lead to perfect 0-matchings with larger cycle numbers. Beside the intuitive aspect, we can prove that this optimistic search schema has a smallest search space in terms of the number of ICs it examines.

Theorem 4. *The set of ICs the optimistic search schema examines includes all ICs with $u > c$, plus a subset of ICs with $u = c$. Further more, since the search space of every branch-and-bound method includes all ICs with $u > c$, the optimistic search schema has the smallest search space possible.*

Proof. Obviously every IC with $u > c$ should be examined by the algorithm, otherwise, the possibility of having a maximum total cycle number with $c+1$ or more can not be eliminated.

Because of Lemma 5, for any IC with $u \geq c$, all the ICs lying on the path from the original of the search to this IC have their upper bounds larger than or equal to c. So the algorithm with optimistic search schema never needs to examine any IC with $u < c$ to find the ones with $u \geq c$, i.e., this algorithm finds all ICs with $u \geq c$ without examining any ones with smaller upper bounds. By the time that the ones with $u \geq c$ have been examined, an optimal 0-matching with c cycles has been found and the algorithm stops. And the search space for the optimistic schema includes all the ICs with $u > c$ and a subset of the ICs with $u = c$.

Hence the optimistic search schema has the smallest search space possible. □

The exact algorithm in [4] consists of cascaded runs of depth-first branch-and-bound search, with the first run seeking a solution whose cycle number is equal to

the upper bound of the original MBG and the subsequent runs seeking solutions with one cycle less than the previous ones, until a solution is found. The cascaded branch-band-bound algorithm and our optimistic branch-and-bound algorithm are similar in terms of the search spaces. The intermediate configurations may be examined more than once in the former algorithm. In our optimistic algorithm, some intermediate configurations with smaller upper bounds need to be stored temporarily. Although storing huge amount of these intermediate configurations can be a challenge to physical memories or even hard disks, the problem is dramatically improved with the adequate subgraph decomposition method and it can be further improved by finding better pruning methods, such as finding a better lower bound or running a heuristic before the main exact algorithm starts.

6 Results on simulated data

ASMedian algorithm is implemented in Java and runs on a MacBook, using only one 2.16GHz CPU. Sets of data are simulated with varying parameters n and $\pi = \rho/n$, where n is the number of gene in each genome and ρ is the number of random reversals applied to the ancestor $I = 1, \ldots, n$ independently to derive each of the three different genomes. n ranges among 10, 20, 30, 40, 50, 60, 80, 100, 200, 300, 500, 1000, 2000, 5000 and π starts from 0.1 and increases by intervals of 0.1. For each data set, 10 instances are generated.

6.1 The running time for simulated data sets with varying n and $\pi = \rho/n$

Table 1 shows the average running time in seconds for all data sets whose 10 instances can all be solved within one hour or the number of solved instances in parenthesis for

n	ρ/n	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
10		$4\,10^{-4}$	$1\,10^{-4}$	$2\,10^{-4}$	$8\,10^{-4}$	$4\,10^{-4}$	$2\,10^{-3}$	$2\,10^{-3}$	$8\,10^{-4}$	$4\,10^{-4}$	$5\,10^{-4}$
20		$2\,10^{-4}$	$2\,10^{-4}$	$3\,10^{-4}$	$6\,10^{-4}$	$9\,10^{-4}$	$6\,10^{-4}$	$2\,10^{-2}$	$7\,10^{-3}$	$2\,10^{-2}$	$5\,10^{-3}$
30		$2\,10^{-3}$	$2\,10^{-3}$	$3\,10^{-4}$	$7\,10^{-4}$	$5\,10^{-3}$	$3\,10^{-2}$	$5\,10^{-2}$	$1\,10^{-1}$	$4\,10^{-1}$	1
40		$1\,10^{-4}$	$2\,10^{-4}$	$3\,10^{-4}$	$6\,10^{-4}$	$4\,10^{-2}$	1	6	$6\,10^{1}$	$6\,10^{1}$	$5\,10^{1}$
50		0	$4\,10^{-4}$	$5\,10^{-4}$	$2\,10^{-3}$	$7\,10^{-2}$	$7\,10^{1}$	(9)	(7)	(7)	
60		$2\,10^{-3}$	$1\,10^{-3}$	$5\,10^{-3}$	$3\,10^{-2}$	$5\,10^{1}$	(7)				
80		$3\,10^{-4}$	$4\,10^{-4}$	$7\,10^{-4}$	$8\,10^{-2}$	(9)					
100		$3\,10^{-4}$	$7\,10^{-4}$	$1\,10^{-3}$	$7\,10^{1}$	(1)					
200		$7\,10^{-3}$	$1\,10^{-2}$	$3\,10^{-2}$	(0)						
300		$5\,10^{-3}$	$5\,10^{-3}$	$2\,10^{-2}$	(0)						
500		$2\,10^{-2}$	$2\,10^{-2}$	$1\,10^{-1}$	(0)						
1000		$9\,10^{-2}$	$7\,10^{-2}$	$8\,10^{1}$	(0)						
2000		$9\,10^{-2}$	$3\,10^{-1}$	(3)							
5000		2	2	(0)							

Table 1: For each data set, if its ten instances all finish in 1 hour, then their average running time is shown in seconds; otherwise the number of finished instances is shown with parenthesis.

the remaining data sets. It can be seen that relatively large instances can be solved if ρ/n remains at 0.3 or less. It also shows that for small n, the median is easy to find even if ρ/n is large enough to effectively scramble the genomes.

6.2 The effect of adequate subgraph discovery on speed-up

Table 2 shows how the occurrence of adequate subgraphs can dramatically speed up the solution to the median problem, generally from more than a half an hour to a fraction of a second.

7 Conclusion

In this paper, several important properties about the adequate subgraphs of rank 3 are proved. We show that there are infinitely many adequate subgraphs, hence

run	speedup factor	run time with AS	run time no AS	number of edges AS1	AS2	AS4	AS0
1	41,407	4.5×10^{-2}	1.9×10^{3}	53	39	8	0
2	85,702	3.0×10^{-2}	2.9×10^{3}	53	34	12	1
3	2,542	5.4×10^{0}	1.4×10^{4}	56	26	16	2
4	16,588	3.9×10^{-2}	6.5×10^{2}	58	42	0	0
5	$> 10^{6}$	5.9×10^{2}	stopped	52	41	4	3
6	199,076	6.0×10^{-3}	1.2×10^{3}	56	44	0	0
7	6,991	2.9×10^{-1}	2.1×10^{3}	54	33	12	1
8	$> 10^{6}$	4.2×10^{1}	stopped	57	38	0	5
9	1,734	8.7×10^{0}	1.5×10^{4}	65	22	8	5
10	855	2.1×10^{0}	1.8×10^{3}	52	38	8	2

Table 2: Speedup due to adequate subgraph (AS) discovery. Three genomes are generated from the identity genome with $n = 100$ by 40 random reversals. Time is measured in seconds. Runs were halted after 10 hours. AS1, AS2, AS4, AS0 are the numbers of edges in the solution median constructed consequent to the detection of adequate subgraphs of sizes 1, 2, 4 and at steps where no adequate subgraphs were found, respectively.

it is not possible to list all these subgraphs. By showing that the simple adequate subgraphs of small sizes have the largest occurrence probability on random MBGs and the algorithms of detecting them are simple and fast, it is practical and efficient to solve the median of three problem by only using simple adequate subgraphs of small sizes. This is confirmed by the dramatic speedup shown in the results on simulated data. Whether it is worth exploring simple adequate subgraphs of size 6 is not clear. It depends on many factors, such as the size of the problem (number of genes genomes contained) and the algorithms for detecting subgraphs and their implementations.

References

[1] Adam Z, Sankoff D. 2008. The ABCs of MGR with DCJ. *Evol Bioinform* **4**, 69–74.

[2] Bourque G, Pevzner PA. 2002. Genome-scale evolution: Reconstructing gene orders in the ancestral species. *Genome Res* **12**, 26–36.

[3] Bryant D. 1998. The complexity of the breakpoint median problem. TR CRM-2579. Centre de recherches mathématiques, Université de Montréal.

[4] Caprara A. 2003. The reversal median problem. *INFORMS J Comput* **15** 93–113.

[5] Lenne R, Solnon C, Stützle T, Tannier E, Birattari M. 2008. Reactive stochastic local search algorithms for the genomic median problem. *EvoCOP 2008* LNCS **4972**, 266–276.

[6] Moret BME, Siepel AC, Tang J, Liu T. 2002. Inversion medians outperform breakpoint medians in phylogeny reconstruction from gene-order data. *WABI 2002*, LNCS **2452**.

[7] Pe'er I, Shamir R. 1998. The median problems for breakpoints are NP-complete. *the Electronic Col loquium of Computational Complexity Report*, number TR98-071, 1998

[8] Sankoff D, Blanchette M. 1998. Multiple genome rearrangement and breakpoint phylogeny. *J Comput Biol* **5**, 555–570.

[9] Tannier E, Zheng C, Sankoff D. 2008. Multichromosomal median and halving problems. *Proceedings of the Workshop on Algorithms in Bioinformatics, WABI 2008, Lecture Notes in Bioinformatics* **5251**, *Springer.*

[10] Xu A.W, Sankoff D. Decompositions of multiple breakpoint graphs and rapid exact solutions to the median problem. *Proceedings of the Workshop on Algorithms in Bioinformatics, WABI 2008, Lecture Notes in Bioinformatics* **5251**, *Springer.*

[11] Yancopoulos S, Attie O, Friedberg R. 2005. Efficient sorting of genomic permutations by translocation, inversion and block interchange. *Bioinform* **21**, 3340–3346.

Chapter 8

Discussion and conclusion

8.1 The tests of remaining evolutionary signal

By using combinatorial and probability methods, we derived the asymptotic probability distributions for breakpoint distance and DCJ distance in terms of the number of common adjacencies and number of cycles in the breakpoint graph,

- – For unsigned genomes, the number of common adjacencies is asymptotically Poisson distributed with parameter 2;
  - – For signed genomes the number of common adjacencies is asymptotically Poisson distributed with parameter $\frac{1}{2}$;
- – For signed circular genomes, the number of cycles is normal distributed with mean $\log 2 + \frac{1}{2}(\log n + \gamma)$ and variance $\log 2 + \frac{1}{2}(\log n + \gamma) - \frac{\pi^2}{8} + \frac{1}{4n}$ asymptotically in n;
  - – For signed genomes containing linear chromosomes, the number of cycles is normal distributed with mean $\max(\chi_r, \chi_b) + \frac{2\chi_r\chi_b}{2\chi_r+2\chi_b-1} + \frac{1}{2}\log\left(\frac{n+\max(\chi_r,\chi_b)}{\chi_r+\chi_b}\right)$ and variance $\frac{8\chi_r^2\chi_b^2}{(2\chi_r+2\chi_b-1)^2(2\chi_r+2\chi_b-3)} + \frac{1}{2}\log\left(\frac{n+\max(\chi_r,\chi_b)}{\chi_r+\chi_b}\right) - \frac{1}{4(\chi_r+\chi_b)} + \frac{1}{4(n+\max(\chi_r,\chi_b))}$ asymptotically in n, χ_r and χ_b.

number of adjacencies	p-value for unsigned case	p-value for signed case
0	1	1
1	0.8647	0.3935
2	0.5940	0.0902
3	0.3233	0.0144
4	0.1429	0.0018
5	0.0527	0.00017
6	0.0166	0.000014
7	0.0045	1.00×10^{-6}
8	0.0011	6.23×10^{-8}
9	2.37×10^{-4}	3.50×10^{-9}
10	4.64×10^{-5}	4.10×10^{-10}

Table 8.1: p-values for given number of common adjacencies when n is sufficiently large.

where n denotes the number of genes in each genome, and χ_r, χ_b denote the number of linear chromosomes.

Now we can design statistical tests to see whether there is significant signal remaining between pairs of gene orders. The null hypothesis is that the given pair of genomes is randomly generated; and the alternative hypothesis is there is strong signal on the gene orders showing the phylogeny relationship. The significance of a test is the probability for events generated by the null hypothesis but rejected by the test.

significance	critical region			
level	$n = 15, \chi = 0$	$n = 37, \chi = 0$	$n = 10000, \chi = 0$	$n = 10000, \chi = 23$
0.10	$c \geq 3.7$	$c \geq 4.4$	$c \geq 8.3$	$c \geq 40.4$
0.05	$c \geq 4.1$	$c \geq 4.8$	$c \geq 9.0$	$c \geq 41.3$
0.02	$c \geq 4.5$	$c \geq 5.4$	$c \geq 9.9$	$c \geq 42.2$
0.01	$c \geq 4.8$	$c \geq 5.7$	$c \geq 10.4$	$c \geq 42.9$

Table 8.2: Statistical tests of different significance levels based on the cycle numbers in the breakpoint graphs for various cases. The critical region is the region where the null hypothesis is rejected.

By using adjacency information for pairs of signed genomes sharing 4 or more common adjacencies, or for pairs of unsigned genomes sharing 7 or more common adjacencies, a test with significance 1% rejects the null hypothesis, implying that these pairs of genomes containing sinificant evolutionary signal in their gene orders (see Table 8.1).

For the corresponding test for DCJ distances or the cycle numbers, the critical region depends on the details of the concerned genomes, i.e., the number of genes and the number of linear chromosomes. In Table 8.2 four cases are given: i) circular genomes with 15 genes (a typical profile of mitochondria genomes excluding tRNAs); ii) circular genomes with 37 genes (a typical profile of mitochondria genomes including tRNAs); iii) circular genomes with 10,000 genes and vi) genomes with 10,000 genes distributed among 23 linear chromosomes (a typical profile of human genome). A test with 1% significance will imply the existence of significant signal when the DCJ distance is no more than 10, 31, 9989, 9980 for the corresponding four cases.

8.2 Adequate subgraphs and the median problem

The DCJ median problem is NP-hard. And from the definition of NP-hardness, even if an optimal solution is presented, there is no algorithm to verify its optima in time polynomial in the size of the problem. Hence it is not surprising to find that the DCJ median problem together with its equivalent problem of finding the optimum matching of the MBG maximizing the total number of cycles, lacks a simple mathematical characterization. Due to the lack of mathematical characterization, the properties about these problems are hard to discover and can be even harder to prove.

The idea of using adequate subgraphs to decompose MBGs was motivated by the discovery of corresponding property for multiple edges—any optimal 0-matching must contain 0-edges parallel to the multiple edges on the MBG. This property of multiple edges on MBG of rank 3 (the ones modeling the median of three problems) can be easily proved. (Despite the salience of this property, it has not been remarked upon previously.) By thinking the multiple edges as the smallest nontrivial subgraphs, this nice property then can be interpreted as a rudimentary theory of decomposing MBGs into (the smallest) subgraphs. I was motivated to find the method to decompose the MBGs into larger subgraphs, hence the theory of adequate subgraphs.

The key of the theory lies at the fact that a discovery of an adequate subgraph H one an MBG B guarantees at least on optimal 0-matching is composed of a 0-matching of H and a 0-matching of its complementary induced subgraph $\bar{H}$. By using further improving methods, the theory guarantees an optimal solution can be found among a limited number of smaller problems on each discovery of an adequate subgraph.

Each of such discoveries enables a significant search space reduction by a factor in the order $(2n)^m$, where n, m are the sizes of the MBG and the discovered sub-

graph. But this never means that the median problem can be solved in polynomial time. Up to now several problems remain unknown. First of all, it is not known whether each MBG contains at least an adequate subgraph, even the number of all possible adequate subgraphs are infinite (at least the ones of rank 3). Secondly, even if each MBG contain an adequate subgraph, for many cases the decompositions will reduce the problem into more than one smaller problem. In this case, the worst case running time for those problems is still at least exponential, though the exponent term might be smaller than n. Thirdly inventorying adequate subgraphs of large size is computational prohibiting and also to discover them on the MBGs is very costly.

Besides the problems mentioned above, the decomposition approach by using adequate subgraphs works very well up to genomes containing hundreds or even thousands of genes as long as the ratio of distances to number of genes is not too high. For most biology problems, due to phylogeny relationships the genomic distances satisfy the above condition. With possible various improvements of the algorithms and the implementation of the program, this approach should lead to practical algorithms applicable to most biology instances.

Bibliography

[1] ABRAMSON, M., AND MOSER, W. Combinations, successions and the n-kings problem. *Mathematics Magazine 39*, 5 (1966), 269–273.

[2] ADAM, Z., AND SANKOFF, D. The abcs of mgr with dcj. *Evol Bioinform 4* (2008), 69–74.

[3] ANDERSON, S., BANKIER, A. T., BARRELL, B. G., DE BRUIJN, M. H., COULSON, A. R., DROUIN, J., EPERON, I. C., NIERLICH, D. P., ROE, B. A., SANGER, F., SCHREIER, P. H., SMITH, A. J., STADEN, R., AND YOUNG, I. G. Sequence and organization of the human mitochondrial genome. *Nature 290* (1981), 457–65.

[4] BOURQUE G, P. P. Genome-scale evolution: Reconstructing gene orders in the ancestral species. *Genome Res 12* (2002).

[5] CAPRARA, A. Sorting by reversals is difficult. *Proceedings of the first annual international conference on Computational molecular biology* (1998), 75–83.

[6] CAPRARA, A. The reversal median problem. *INFORMS J Comput 15* (2003), 93–113.

[7] CASPERSSON, T., ZECH, L., JOHANSSON, C., AND MODEST, E. J. Identification of human chromosomes by dna-binding fluorescent agents. *Chromosoma 30* (1970), 215–27.

[8] EL-MABROUK, N. Genome rearrangement by reversals and insertions/deletions of contiguous segments. In *Combinatorial Pattern Matching, 11th annual symposium* (2000), vol. 1848 of *LNCS*, pp. 222–234.

[9] FLEISCHMAN, R. D., ADAMS, M. D., WHITE, O., CLAYTON, R. A., KIRKNESS, E. F., KERLAVAGE, A. R., BULT, C. J., TOMB, J. F., DOUGHERTY, B. A., MERRICK, J. M., AND *et al.* Whole-genome random sequencing and assembly of *haemophilus influenzai rd. Science 269* (1995), 496–512.

[10] GOFFEAU, A., BARRELL, B., BUSSEY, H., DAVIS, R., DUJON, B., FELDMANN, H., GALIBERT, F., HOHEISEL, J., JACQ, C., JOHNSTON, M., LOUIS, E., MEWES, H., MURAKAMI, Y., PHILIPPSEN, P., TETTELIN, H., AND OLIVER, S. Life with 6000 genes. *Science*, 274 (1996), 546,563–567.

[11] GOSS, S. J., AND HARRIS, H. New method for mapping genes in human chromosomes. *Nature 255* (1975), 680.

[12] HANNENHALLI, S. Polynomial-time algorithm for computing translocation distance between genomes. *Discrte Applied Mathematics 71* (1996), 137–151.

[13] HANNENHALLI, S., AND PEVZNER, P. A. Transforming men into mice (polynomial algorithm for genomic distance problem). *Proceedings of the IEEE 3th Annual Symposium on Foundations of Computer Science.* (1995), 581–592.

[14] HANNENHALLI, S., AND PEVZNER, P. A. Transforming cabbage into turnip: Polynomial algorithm for sorting signed permutations by reversals. *JACM 46* (1999), 1–27.

[15] IHGC (INTERNATIONAL HUMAN GENOME SEQUENCING CONSORTIUM). Initial sequencing and analysis of the human genome. *Nature 409* (2001), 860–921.

[16] JEAN, G., AND NIKOLSKI, M. Genome rearrangements: a correct algorithm for optimal capping. *Inf. Process. Lett. 104*, 1 (2007), 14–20.

[17] KIM, J. H., AND WORMALD, N. C. Random matchings which induce hamilton cycles, and hamiltonian decompositions of random regular graphs. *Journal of Combinatorial Theory, Series B 81* (2001), 20–44.

[18] LENNE, R., SOLNON, C., STÜTZLE, T., TANNIER, E., AND BIRATTARI, M. Reactive stochastic local search algorithms for the genomic median problem. In *EvoCOP 2008* (2008), vol. 4972 of *LNCS*, pp. 266–276.

[19] MARRON, M., SWENSON, K. M., AND MORET, B. M. E. Genomic distances under deletions and insertions. *Theoretical Computer Science 325*, 3 (2004), 347–360.

[20] MORET, B. M. E., SIEPEL, A. C., TANG, J., AND LIU, T. Inversion medians outperform breakpoint medians in phylogeny reconstruction from gene-order data. In *WABI 2002* (2002), vol. 2452 of *LNCS*.

[21] OZERY-FLATO, M., AND SHAMIR, R. Two notes on genome rearrangements. *J. Bioinf. Comput. Biol. 1*, 1 (2003), 71–94.

[22] PAINTER, T. S. A new method for the study of chromosome rearrangements and the plotting of chromosome maps. *Science 78* (1933), 585–6.

[23] ROBBINS, D. P. The probability that neighbors remain neighbors after random rearrangements. *The American Mathematical Monthly 87*, 2 (1980), 122–124.

[24] SANGER, F., AIR, G. M., BARRELL, B. G., BROWN, N. L., COULSON, A. R., FIDDES, C. A., HUTCHISON, C. A., SLOCOMBE, P. M., AND SMITH, M. Nucleotide sequence of bacteriophage ϕx174 dna. *Nature 265* (1977), 687–95.

[25] SANKOFF, D. Genome rearrangement with gene families. *Bioinformatics 15* (1999), 909–917.

[26] SANKOFF, D., AND BLANCHETTE, M. Multiple genome rearrangement and breakpoint phylogeny. *J Comput Biol*, 5 (1998), 555–570.

[27] SANKOFF, D., AND HAQUE, L. The distribution of genomic distance between random genomes. *Journal of Computational Biology 13* (2006), 1005–1012.

[28] STURTEVANT, A. H. *A history of genetics.* New York: Harper and Row, 1965.

[29] TANNIER, E., ZHENG, C., AND SANKOFF, D. Multichromosomal median and halving problems. *WABI 2008* (2008).

[30] TESLER, G. Efficient algorithms for multichromosomal genome rearrangements. *JCSS 65* (2002), 587–609.

[31] TESLER, G. Decomposition of permutations into rising and falling subsequences. unpublished, 2005.

[32] VENTER, J. C., ADAMS, M. D., MYERS, E. W., AND *et al.* The sequence of the human genome. *Science 291* (2001), 1304–51.

[33] YANCOPOULOS, S., ATTIE, O., AND FRIEDBERG, R. Efficient sorting of genomic permutations by translocation, inversion and block interchange. *Bioinform 21* (2005), 3340–3346.