BIOINFORMATICS ORIGINAL PAPER
Vol. 21 no. 7 2005, pages 869–879
doi:10.1093/bioinformatics/bti107

Genome analysis

Theoretical and practical advances in genome halving

Peng Yin* and Alexander J. Hartemink*

Department of Computer Science, Duke University, Box 90129, Durham, NC 27708-0129, USA

Received on September 29, 2003; revised on May 31, 2004; accepted on July 28, 2004
Advance Access publication October 28, 2004

ABSTRACT

Motivation: Duplication of an organism's entire genome is a rare but spectacular event, enabling the rapid emergence of multiple new gene functions. Over time, the parallel linkage of duplicated genes across chromosomes may be disrupted by reciprocal translocations, while the intra-chromosomal order of genes may be shuffled by inversions and transpositions. Some duplicate genes may evolve unrecognizably or be deleted. As a consequence, the only detectable signature of an ancient duplication event in a modern genome may be the presence of various chromosomal segments containing parallel paralogous genes, with each segment appearing exactly twice in the genome. The problem of reconstructing the linkage structure of an ancestral genome before duplication is known as *genome halving with unordered chromosomes*.

Results: In this paper, we derive a new upper bound on the genome halving distance that is tighter than the best known, and a new lower bound that is almost always tighter than the best known. We also define the notion of genome halving diameter, and obtain both upper and lower bounds for it. Our tighter bounds on genome halving distance yield a new algorithm for reconstructing an ancestral duplicated genome. We create a software package *GenomeHalving* based on this new algorithm and test it on the yeast genome, identifying a sequence of translocations for halving the yeast genome that is shorter than previously conjectured possible.

Availability: *GenomeHalving* is available upon email request.

Contact: py@cs.duke.edu; amink@cs.duke.edu

1 INTRODUCTION

1.1 Biological motivation

In the course of evolution, gene duplications are extremely significant events, enabling the emergence of new gene functions (Ohno, 1970). The presence of one copy of each gene is normally sufficient for the survival of the species, allowing other (redundant) copies to evolve with less selective pressure. Beyond the duplication or multiplication of individual genes, it is possible for the entire genome of a species to be duplicated in a process known as *tetraploidization*. Although tetraploidization is normally lethal, in rare cases a tetraploid can become a stabilized diploid with two sets of identical chromosomes. The functionalities of the genes in one set are usually preserved, while the genes in the other set are now free to evolve into novel functional units, presenting the species with a tremendous opportunity for new evolutionary possibilities. A potentially more important consequence of whole-genome duplication is the combinatorial number of possibilities for the co-evolution of a group of genes in concert (Fryxell, 1996).

Evidence supporting the occurrence of whole-genome duplication has been adduced in numerous plant genomes (Ahn and Tanksley, 1993; Gaut and Doebley, 1997; Moore *et al*., 1995; Scheffler *et al*., 1997; Shoemaker *et al*., 1996; Paterson *et al*., 1996), as well as in vertebrate genomes (Nadeau, 1991; Lundin, 1993; Gibson and Spring, 2000; Gu *et al*., 2002; McLysaght *et al*., 2002). A particularly convincing example of whole-genome duplication is found in the yeast genome. Wolfe and Shields (1997) provided early strong evidence that the genome of *Saccharomyces cerevisiae* is the product of an ancient tetraploidization, which has been further supported by subsequent studies (Vision and Brown, 2000; Seoighe *et al*., 2000; Langkjær *et al*., 2003; Dietrich *et al*., 2004; Kellis *et al*., 2004). However, we note that there exist alternative views on whole-genome duplication in yeast (Mewes *et al*., 1997; Coissac *et al*., 1997; Llorente *et al*., 2000a,b) and that it remains a somewhat controversial issue. Evidence also exists to suggest the flowering plant *Arabidopsis* may have undergone whole-genome duplication, but this is not conclusive (Ku *et al*., 2000; Paterson *et al*., 2000; Lynch and Conery, 2000). For surveys on whole-genome duplication, see Wolfe (2001) and Durand (2003).

During the course of evolution subsequent to genome duplication, the parallel linkage of genes across chromosomes may be disrupted by reciprocal translocations, while the intra-chromosomal order of genes may be modified by inversions and transpositions. Some duplicate genes may evolve unrecognizably or be deleted. As a consequence, sometimes the only extant evidence of an ancient duplication in a modern genome is the presence of various duplicate chromosomal segments containing parallel paralogous genes dispersed throughout the genome.

The *genome halving* problem is to construct a (minimal) sequence of operations—translocations, inversions or transpositions—that transform an ancestral genome immediately after a genome duplication event into a modern genome; or conversely but equivalently, a minimal sequence of operations that transform a modern genome G into an ancestral duplicated genome G'. In the latter interpretation of the problem, the modern genome G is said to be *halved* by these transformations, since G' consists of two identical copies of each chromosome, representing the ancestral genome immediately after duplication.

El-Mabrouk *et al*. (1998) propose two formulations of the genome halving problem. The problem of *genome halving with ordered chromosomes* considers a chromosome as an ordered sequence of gene blocks, and aims to construct a sequence of operations that transform an ancient duplicated genome to that of a modern species via translocations and intra-chromosomal operations, like inversions

*To whom correspondence should be addressed.

and transpositions. Seoighe and Wolfe (1998) study this problem using a computer simulation and a heuristic analytical method. El-Mabrouk *et al*. propose an exact algorithm to solve this problem (El-Mabrouk *et al*., 1999; El-Mabrouk, 2000; El-Mabrouk and Sankoff, 2002).

We are here interested in the related problem of *genome halving with unordered chromosomes*, which considers a chromosome as an unordered collection of gene blocks, and aims to construct a sequence of only translocations that transform the synteny or linkage structure of the genome of an ancestral duplicated genome to that of a modern species. Both the ordered and unordered problems can provide insight to understanding the possible evolutionary path that leads from the ancestral duplicated genome to that of a modern species. However, one aspect of the comparative importance of the unordered version of the genome halving problem resides in the possibility that intra-chromosomal operations, such as inversions and transpositions, alter the order of gene blocks within a chromosome repeatedly between translocations, as suggested by El-Mabrouk *et al*. (1998). In such a context, the intra-chromosomal order of the gene blocks and the intra-chromosomal operations are of only marginal significance in exploring the possible optimal sequence of translocation events that transform an ancient genome to its current state; as a result, the unordered formulation of the problem as discussed in this paper is of greater relevance. Furthermore, a potential practical constraint on the application of genome halving with ordered chromosomes in some cases might be the unavailability of data on the intra-chromosomal order of gene blocks for a species.

El-Mabrouk *et al*. (1998) provide both upper and lower bounds for the problem of genome halving with unordered chromosomes, and give a heuristic algorithm for computing the ancestral genome. We improve both of their bounds, and then design and implement an algorithm for reconstructing an ancestral duplicated genome. We create a software package *GenomeHalving* and apply it to the yeast genome to obtain a shorter halving path than was previously conjectured possible. In addition, we define the notion of genome halving diameter, and offer an upper bound and a lower bound that almost always match for genomes with a realistic number of chromosomes.

1.2 Definitions and notation

For the remainder of the paper, we refer to the problem of genome halving with unordered chromosomes as simply genome halving, for brevity. In this formulation of the problem, a genome G is a set of chromosomes and a chromosome S_i is a collection of gene blocks, or blocks. Since we are interested in studying the translocation history of the ancient duplicated genome, we can ignore gene blocks that occur only once in the genome (due to subsequent gene deletion or mutation) because they contribute no useful information in reconstructing the translocation history of the genome. Thus, we restrict our attention to gene blocks that appear exactly twice in the genome. If a gene block happens to appear twice in the same chromosome, it is called a 2-*block*. A genome $G = \{S_1, S_2, \ldots, S_n\}$ can be represented by an equivalent *intersection graph* as follows (El-Mabrouk *et al*., 1998). Create a vertex v_i for each chromosome S_i; connect v_i and v_j with an undirected edge $e(v_i, v_j)$ if and only if $S_i \cap S_j \neq \emptyset$ and $i \neq j$; connect v_i to itself with a loop if and only if S_i contains a 2-block. A vertex with (without) a loop to itself is called a *loop-vertex* (*non-loop-vertex*). Note that a loop-vertex is adjacent to itself. Denote by $h(v)$ the number of vertices adjacent to v, including v itself. A vertex v with $h(v) = k$ will sometimes be called a *k-vertex*. A pair of adjacent (non-loop) 1-vertices are referred to as a *perfectly matched vertex pair*. A graph consisting of only perfectly matched vertex pairs is called a *perfect matching graph*. A duplicated genome with two identical sets of chromosomes corresponds to a *perfect* matching graph. To simplify notation, we use the symbol G interchangeably to denote both a genome and the intersection graph derived from that genome.

The basic operation allowed in the genome halving problem is *translocation*, or the exchange of gene blocks between two chromosomes. We represent a translocation between v_i and v_j with the quadruplet $\delta = (v_i, v_j, B_i, B_j)$, where $B_i \subseteq S_i$ and $B_j \subseteq S_j$, indicating the movement of block set B_i from S_i to S_j and of block set B_j from S_j to S_i. In the above formulation, neither fission nor fusion of chromosomes are allowed: $S_i \neq \emptyset$; $S_j \neq \emptyset$; when $B_i = \emptyset$, we require that $B_j \neq S_j$; when $B_j = \emptyset$, we require that $B_i \neq S_i$. Sometimes, we omit B_i and B_j and just write $\delta(v_i, v_j)$. After the translocation $\delta(v_i, v_j)$, vertex v_i is denoted by δv_i and vertex v_j is denoted by δv_j. A vertex v_k that is adjacent to v_i or v_j before $\delta(v_i, v_j)$ must be adjacent to δv_i or δv_j, provided $i \neq k$ and $j \neq k$. If v_i is adjacent to v_j before $\delta(v_i, v_j)$, either or both of δv_i and δv_j may be loop-vertices. To make these notions more concrete, Figure 1 shows an example of a sequence of translocations that transform a particular genome into an ancestral genome immediately after duplication.

1.3 Problem definition

The genome halving problem requires finding the minimum number $d(G)$ of translocations that are sufficient to transform a given genome G into an ancestral duplicated genome G' containing two identical sets of chromosomes. We call $d(G)$ the *genome halving distance* of G. Let $|G|$ be the size of genome G. Since $|G'|$ is even and $|G| = |G'|$, we require that $|G|$ be even.

We define the *genome halving diameter* for genomes of size n, $D(n)$, as the maximum value of the genome halving distance for any genome with n chromosomes:

$$D(n) = \max_{|G|=n} d(G)$$

The genome halving diameter problem is to find $D(n)$ for even n.

The rest of the paper is organized as follows. We first give an upper and a lower bound for the genome halving diameter problem in Section 2. Then in Section 3, we give a new upper bound for the genome halving distance $d(G)$ that is tighter than the best known, and a new lower bound that is almost always tighter than the best known. Based on the insight obtained in the analysis of these tighter bounds for genome halving distance, we report a novel algorithm to reconstruct ancestral duplicated genomes in Section 4. In Section 5, we analyze the yeast genome with a software package *GenomeHalving* we have developed to implement our algorithm, and identify a sequence of translocations for halving the yeast genome that is of shorter length than was previously conjectured possible. We close with a discussion of our results.

2 GENOME HALVING DIAMETER

In this section, we obtain an upper bound and a lower bound for the genome halving diameter problem. For genomes with a realistic number of chromosomes, the upper bound almost always matches the lower bound.

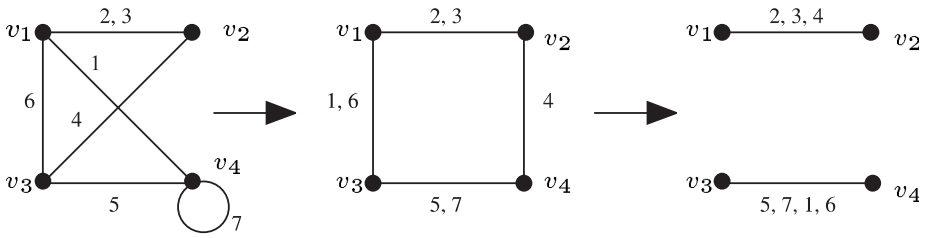

Fig. 1. Two translocations are sufficient to transform the intersection graph on the left representing a genome with four chromosomes into a perfect matching graph. In this example, v_1 corresponds to $S_1 = \{1, 2, 3, 6\}$, v_2 to $S_2 = \{2, 3, 4\}$, v_3 to $S_3 = \{4, 5, 6\}$ and v_4 to $S_4 = \{1, 5, 7, 7\}$. In the first translocation, v_3 exchanges block 4 with blocks 1 and 7 from v_4. In the second translocation, v_1 exchanges blocks 1 and 6 with block 4 from v_4.

2.1 Genome halving diameter: upper bound

El-Mabrouk *et al*. (1998) studied the diameter problem in which chromosomes can be merged and split, and offered an upper bound of n to construct a 'trivial' duplicated genome. Their construction simply merges all chromosomes into one big chromosome using $n-1$ fusion translocations, and then divides the resultant chromosome into two identical chromosomes using a fission translocation. By examining this problem with a bit more scrutiny, we are able to derive a tighter upper bound without resorting to either fusions or fissions.

For ease of exposition, we introduce a little more notation. Denote by $I(v_i, v_j)$ one copy of each of the blocks shared by vertices v_i and v_j; note that if v_i is a loop-vertex we permit $v_i = v_j$, in which case $I(v_i, v_i)$ contains only one copy of each 2-block contained in v_i. Denote by $\mathcal{I}(v)$ the collection of blocks shared between v and all its adjacent vertices, including itself. Note that $\mathcal{I}(v_i)$ is just S_i. Finally, a genome with n chromosomes that has either n or $n-1$ loop-vertices is defined to be a *loopy genome*.

THEOREM 2.1. *$D(n) \leq n - 1$; if we restrict our attention to non-loopy genomes, we have $D(n) \leq n - 2$.*

PROOF. We give a constructive proof. Color all perfectly matched vertices black, and color the remaining vertices white. Now select a vertex pair (v_1, v_2) as follows.

- If there exists a white loop-vertex, select it as v_1. If there exists another white loop-vertex, select it as v_2; otherwise select an arbitrary white vertex as v_2.
- If there is no white loop-vertex, since each white vertex must have least one white neighbor, we can arbitrarily select a pair of neighboring white vertices as v_1 and v_2.

Color v_1 and v_2 black. Then perform translocation $\delta_1(v_1, v_2, B_1, \emptyset)$ where $B_1 = \mathcal{I}(v_1) \setminus (I(v_1, v_2) \cup I(v_1, v_1))$. Note that one of $I(v_1, v_1)$ or $I(v_1, v_2)$ could be empty, but not both. Also note that if v_1 contains a 2-block, B_1 will contain only *one* copy of that 2-block. After translocation δ_1, vertex $\delta_1 v_1$ is a non-loop 1-vertex whose only neighbor is $\delta_1 v_2$.

Next, select another white loop-vertex, if it exists, as v_3; otherwise choose an arbitrary white vertex as v_3. Perform translocation $\delta_2(\delta_1 v_2, v_3, B_2, \emptyset)$ where $B_2 = \mathcal{I}(\delta_1 v_2) \setminus I(\delta_1 v_1, \delta_1 v_2)$. Note that both copies of the 2-blocks in $\delta_1 v_2$, if they exist, will be passed to v_3. Now, after two translocations, we have produced a perfectly matched vertex pair, $(\delta_1 v_1, \delta_2 \delta_1 v_2)$, and each is newly colored black.

Repeat the above operations until we are left with only four white vertices. Since every two translocations generate a pair of perfectly matched vertices, we have performed at most $n - 4$ translocations to this point. It is easy to verify that three more translocations are sufficient to transform any set of four vertices into two perfectly matched vertex pairs. Hence, the total number of translocations needed to halve any genome with n chromosomes is at most $n - 1$.

Now we restrict our attention to non-loopy genomes and show that $D(n) \leq n - 2$. We discuss two cases.

(1) If there exist perfectly matched vertices in the initial graph, we observe that these perfectly matched vertices require no translocations and hence we must have $D(n) \leq (n-1) - 2 \leq n - 2$.

(2) If there are no perfectly matched vertices in the initial graph, we observe that the final four white vertices must contain at least two white non-loop-vertices, since we start with at least two non-loop-vertices by definition, and take care to exhaust all the white loop-vertices before considering any white non-loop-vertex. In such a case, it is easy to verify that two more translocations are sufficient to transform the remaining four vertices into two perfectly matched pairs and hence we must have $D(n) \leq n - 2$.

We have thus shown that the total number of translocations needed to halve a non-loopy genome with n chromosomes is at most $n - 2$. □

2.2 Genome halving diameter: lower bound

Before proceeding to this section, we note that obtaining a lower bound on genome halving diameter is the most technically challenging problem addressed in this paper. As a result, the proof is unavoidably more involved, and may at certain points be tedious. We preface the proof with an overview intuition, but readers uninterested in the details can safely skip ahead to the statement of the lower bound itself in Section 2.2.5.

2.2.1 Proof intuition and overview To derive a lower bound on the number of translocations required to transform an arbitrary graph into a perfect matching graph, it is easier to study the reverse process in which a perfect matching graph is transformed into an arbitrary graph. More specifically, we study the special case of transforming a perfect matching graph $G(V, E)$ into a clique $K(V)$, a graph whose vertices are all pairwise adjacent.

One critical observation is that if a vertex v is adjacent to either v_1 or v_2 after a translocation $\delta(v_1, v_2)$, then v must have been adjacent

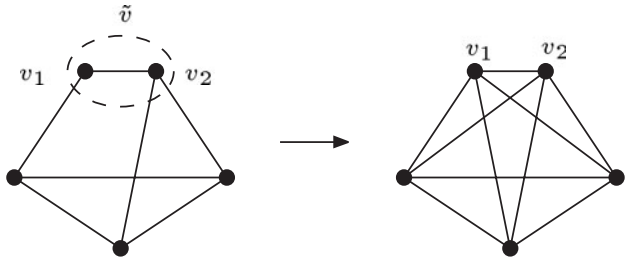

Fig. 2. Imagine that translocation $\delta(v_1, v_2)$ transforms the graph on the left to the 5-clique on the right. If we consider v_1 and v_2 as one vertex $\tilde{v}$, the graph on the left can be viewed as a 4-clique.

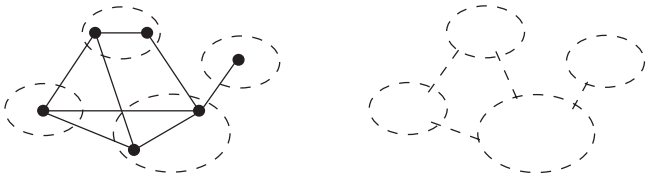

Fig. 3. A pseudo-graph $\tilde{G}$ on the right can be derived from an intersection graph G on the left. Vertices and edges in graph G are depicted as solid circles and lines, respectively; pseudo-nodes and pseudo-edges in pseudo-graph $\tilde{G}$ are depicted as dashed circles and lines, respectively.

to either v_1 or v_2 before the translocation. Let $\delta(v_1, v_2)$ be the last of any series of translocations that transform a graph with vertex set $V = \{v_1, v_2, \ldots, v_n\}$ into an n-vertex clique. We observe that if we view v_1 and v_2 as one vertex $\tilde{v}$ such that $\tilde{v}$'s neighbors are the union of the neighbors of v_1 and v_2, then the $n-1$ vertices $\tilde{v}, v_3, \ldots, v_n$ are pairwise adjacent before translocation $\delta(v_1, v_2)$. In other words, the graph before translocation $\delta(v_1, v_2)$ can be viewed as an $(n-1)$-clique, if v_1 and v_2 are considered as a single vertex. For an example illustration, see Figure 2.

Based on this observation, an induction proof is constructed along the following lines. We first introduce the concept of a *pseudo-clique* and show that the size of the largest pseudo-clique in a graph can be increased by at most one with each translocation, providing us with an inductional device (Lemma 2.4). Then by analyzing the base case to find the largest pseudo-clique in a perfect matching graph (Lemma 2.6), we have a proof by induction to obtain a lower bound for the halving diameter (Corollary 2.8).

2.2.2 Additional notation and definitions A central device used in this section is the *pseudo-graph*, which is derived from an intersection graph and provides an alternative view thereof. Given an intersection graph $G(V, E)$, a *pseudo-node* $\tilde{v}_i$ is defined as a non-empty subset of the vertices V. Two pseudo-nodes are *disjoint* if their intersection is empty. Given two disjoint pseudo-nodes $\tilde{v}_i$ and $\tilde{v}_j$, if no vertex in $\tilde{v}_i$ has an adjacent vertex in $\tilde{v}_j$, then $\tilde{v}_i$ and $\tilde{v}_j$ are *non-adjacent*; otherwise, they are *adjacent*. Given a particular set of disjoint pseudo-nodes, we can connect each pair of adjacent pseudo-nodes with a *pseudo-edge* and get a pseudo-graph $\tilde{G}$ (see Fig. 3). For readability, we sometimes omit the pseudo description for a pseudo-edge when it is clear from context. We emphasize a pseudo-graph $\tilde{G}$ exists only in the context of an underlying intersection graph G, and the adjacencies in $\tilde{G}$ are completely determined by the adjacencies in G, given a particular set of pseudo-nodes. In this sense, $\tilde{G}$ is said to be *derived* from G. In particular, if translocations performed on an underlying graph G change the adjacencies in G, the adjacencies in the derived graph $\tilde{G}$ may change correspondingly. We also note that multiple pseudo-graphs can be derived from the same underlying intersection graph G by choosing different sets of vertices to be the pseudo-nodes.

We define adjacency rules between a vertex and a pseudo-node in an analogous manner: given a vertex v_i and a pseudo-node $\tilde{v}_j$, where $v_i \notin \tilde{v}_j$, if vertex v_i has no adjacent vertex in pseudo-node $\tilde{v}_j$, then vertex v_i and pseudo-node $\tilde{v}_j$ are non-adjacent; otherwise, they are adjacent.

A pseudo-graph is *complete* if all the pseudo-nodes in it are pairwise adjacent. Now we provide two definitions that apply to complete pseudo-graphs, *expanding vertex pair* and *pseudo-clique*, which will be important later in the proofs.

DEFINITION 2.2. *Given a complete pseudo-graph $\tilde{G}$ with a set of k disjoint pseudo-nodes $\tilde{V}$, and a pseudo-node $\tilde{v} \in \tilde{V}$ containing a vertex pair (v_i, v_j), the vertex pair (v_i, v_j) is called an* expanding vertex pair *for pseudo-graph $\tilde{G}$ if there is a translocation $\delta(v_i, v_j)$ such that the vertices contained in $\tilde{v}$ can be split into two new disjoint pseudo-nodes $\tilde{v}_i \ni v_i$ and $\tilde{v}_j \ni v_j$ satisfying the following two conditions:*

(1) *the $k+1$ pseudo-nodes in $\{\tilde{v}_i\} \cup \{\tilde{v}_j\} \cup (\tilde{V} \setminus \{\tilde{v}\})$ and their induced edges after translocation $\delta(v_i, v_j)$ form a complete pseudo-graph $\tilde{G}'$.*

(2) *either $|\tilde{v}_i| = 1$ or $\tilde{v}_i$ contains an expanding vertex pair for the newly formed complete pseudo-graph $\tilde{G}'$, and the same holds for $\tilde{v}_j$.*

The translocation $\delta(v_i, v_j)$ together with the split of $\tilde{v}$ into $\tilde{v}_i \ni v_i$ and $\tilde{v}_j \ni v_j$ is referred to as an expansion *and is denoted by $\epsilon(v_i, v_j)$.*

DEFINITION 2.3. *A complete pseudo-graph $\tilde{G}$ with a set of k disjoint pseudo-nodes $\tilde{V}$ is called a* pseudo-clique *if for all $\tilde{v} \in \tilde{V}$ either $|\tilde{v}| = 1$ or $\tilde{v}$ contains an expanding vertex pair for pseudo-graph $\tilde{G}$.*

2.2.3 Inductional device We now prove a lower bound on $D(n)$ by induction. We begin with the following lemma, which shows that each translocation can increase the size of the largest pseudo-clique in a pseudo-graph by at most one.

LEMMA 2.4. *Given an intersection graph G, if translocation δ results in a new intersection graph G' and a k-pseudo-clique can be derived from G', then a $(k-1)$-pseudo-clique can be derived from G.*

PROOF. Denote the k-pseudo-clique derived from G' as $\tilde{K}'$ and let its pseudo-nodes be $\tilde{V}' = \{\tilde{v}_1, \tilde{v}_2, \ldots, \tilde{v}_k\}$. For concreteness, suppose translocation δ is between vertices v_i and v_j. We study two cases.

(1) If v_i and v_j belong to two distinct pseudo-nodes in $\tilde{V}'$, suppose w.l.o.g. that $v_i \in \tilde{v}_{k-1}$ and $v_j \in \tilde{v}_k$. Define a new pseudo-node $\tilde{v} = \tilde{v}_{k-1} \cup \tilde{v}_k$, and let $\tilde{V} = \{\tilde{v}_1, \tilde{v}_2, \ldots, \tilde{v}_{k-2}, \tilde{v}\}$. We claim that the $k-1$ pseudo-nodes in $\tilde{V}$ together with the set of induced edges connecting them form a $(k-1)$-pseudo-clique $\tilde{K}$ that can be derived from G.

Indeed, we have that the pseudo-nodes in $\tilde{V}$ are pairwise disjoint, which follows immediately from the fact that the pseudo-nodes in $\tilde{V}'$ are disjoint. Furthermore, any translocation between v_i and v_j adds no new edge between

pseudo-nodes in $\tilde{V} \setminus \{\tilde{v}\}$ and does not affect the connectivity between $\tilde{v}$ and any pseudo-node in $\tilde{V} \setminus \{\tilde{v}\}$. Therefore, pseudo-nodes in $\tilde{V}$ must all be pairwise adjacent before performing $\delta(v_i, v_j)$, showing that $\tilde{K}$ is complete. We still need to show that each pseudo-node in $\tilde{V}$ either has cardinality 1 or contains an expanding vertex pair.

By definition, $\tilde{v}$ contains an expanding vertex pair (v_i, v_j). For any other pseudo-node $\tilde{v}_t \in \tilde{V}$ with $|\tilde{v}_t| > 1$, since $\tilde{V}'$ and its induced edges in G' form a k-pseudo-clique $\tilde{K}'$, pseudo-node $\tilde{v}_t$ contains an expanding vertex pair, say (v_a, v_b), for $\tilde{K}'$. For any vertex $v_t \in \tilde{v}_t$, if vertex v_t is adjacent (non-adjacent) to any pseudo-node $\tilde{v}_s$ in $\tilde{V} \setminus \{\tilde{v}_t\}$ in G', it must be adjacent (non-adjacent) to $\tilde{v}_s$ in G. Furthermore, if v_t is a loop-vertex (non-loop-vertex) in G', then it is a loop-vertex (non-loop-vertex) in G. Therefore, by Definition 2.2, it is straightforward to verify that (v_a, v_b) is an expanding vertex pair for the pseudo-graph that is derived from G by $\tilde{V}$, together with its induced edges.

(2) If, on the other hand, v_i and v_j belong to at most one pseudo-node in $\tilde{V}$, there exists a set $\tilde{V}$ of $k-1$ pseudo-nodes in $\tilde{V}'$ containing neither v_i nor v_j and thus unaffected by translocation δ. More precisely, if vertex $v_s \in \tilde{v}_s \in \tilde{V}$ and vertex $v_t \in \tilde{v}_t \in \tilde{V}$ are adjacent (non-adjacent) in G', then they are adjacent (non-adjacent) in G; if v_s is a loop-vertex (non-loop-vertex) in G', then it is a loop-vertex (non-loop-vertex) in G, and the same is true for v_t. Therefore, the pseudo-nodes in $\tilde{V}$ and the induced edges connecting them form a $(k-1)$-pseudo-clique $\tilde{K}$ that can be derived from G.

Putting everything together proves the lemma. □

2.2.4 Base case for the induction Lemma 2.4 provides us with an inductional device to derive a lower bound. We next study the base case by finding the largest pseudo-clique that can be derived from a perfect matching graph, but this requires some further machinery. Given a vertex v (pseudo-node $\tilde{v}$), the *pseudo-degree* of v ($\tilde{v}$) is the number of pseudo-nodes adjacent to v ($\tilde{v}$) and is denoted by $\tilde{h}(v)$ ($\tilde{h}(\tilde{v})$). Given a pseudo-graph $\tilde{G}$, if a pseudo-node $\tilde{v}$ only contains vertices of pseudo-degree 0 or 1, $\tilde{v}$ is referred to as a *singly-adjacent-pseudo-node*. Note that though a singly-adjacent-pseudo-node $\tilde{v}$ contains only vertices of pseudo-degree 0 or 1, the pseudo-degree of $\tilde{v}$ itself may be greater than 1.

LEMMA 2.5. *Given a singly-adjacent-pseudo-node $\tilde{v}$ in a complete pseudo-graph $\tilde{G}$, if $\tilde{v}$ contains an expanding vertex pair, we must have*

$$|\tilde{v}| \geq 2^{\tilde{h}(\tilde{v})-1}$$

PROOF. We prove by induction. When $\tilde{h}(\tilde{v}) = 1$, the lemma is trivially true. Now suppose that the lemma holds for $\tilde{h}(\tilde{v}) = k$; we show that it also holds for $k+1$.

Denote the $k+1$ pseudo-nodes adjacent to $\tilde{v}$ by $\tilde{v}_1, \tilde{v}_2, \ldots, \tilde{v}_{k+1}$. For concreteness, let (v_a, v_b) be the expanding vertex pair contained in $\tilde{v}$. Since $\tilde{v}$ is a singly-adjacent-pseudo-node, the pseudo-degree of v_a and v_b is at most 1. Assume w.l.o.g. that $\tilde{v}_{k+1}$ is the pseudo-node adjacent to v_b, if such a pseudo-node exists. After expansion $\epsilon(v_a, v_b)$, $\tilde{v}$ is split into two new pseudo-nodes, $\tilde{v}_a \ni v_a$ and $\tilde{v}_b \ni v_b$, in the newly formed complete pseudo-graph $\tilde{G}'$. The $k+1$ pseudo-nodes $\tilde{v}_a, \tilde{v}_1, \tilde{v}_2, \ldots, \tilde{v}_k$ together with their induced edges also form a complete pseudo-graph $\tilde{G}_s$. We claim that $\tilde{v}_a$ is a singly-adjacent-pseudo-node in $\tilde{G}_s$ (though it may not be a singly-adjacent-pseudo-node in $\tilde{G}'$). The claim follows from the fact that expansion $\epsilon(v_a, v_b)$ cannot connect any vertex in $\tilde{v}_a$ to any of the pseudo-nodes $\tilde{v}_1, \tilde{v}_2, \ldots, \tilde{v}_k$, though such expansion might connect a vertex in $\tilde{v}_a$ to $\tilde{v}_b$ or $\tilde{v}_{k+1}$.

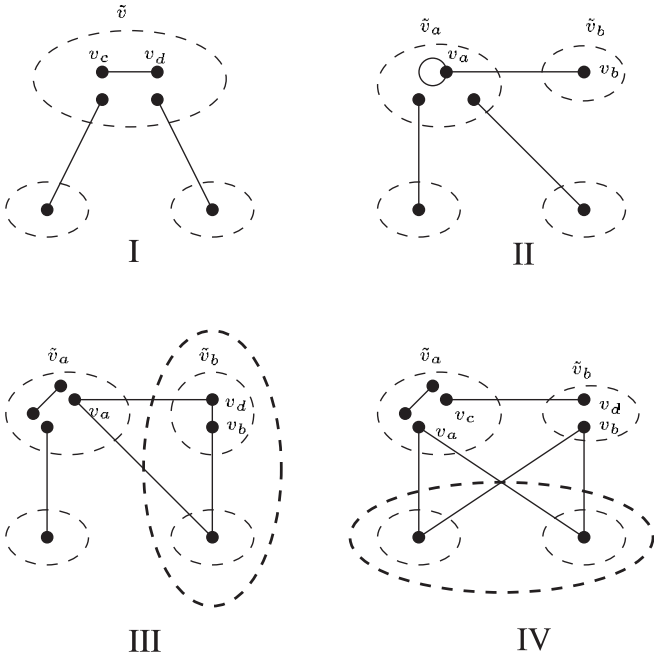

Fig. 4. Panel I depicts pseudo-node $\tilde{v}$ before expansion. Pseudo-node $\tilde{v}$ contains a perfectly matched vertex pair (v_c, v_d). Panels II, III and IV illustrate cases 1, 2 and 3 discussed in the proof of Lemma 2.6, respectively. In panels III and IV, when pseudo-nodes are merged as discussed in the proof, the resultant pseudo-nodes are represented in bold.

According to the definition of expansion, $\tilde{v}_a$ must contain an expanding vertex pair for $\tilde{G}'$ and such a vertex pair is necessarily an expanding vertex pair for $\tilde{G}_s$. Therefore, $\tilde{v}_a$ is a singly-adjacent-pseudo-node with pseudo-degree k in the complete pseudo-graph $\tilde{G}_s$ and it contains an expanding vertex pair. According to the induction hypothesis, we have $|\tilde{v}_a| \geq 2^{k-1}$. Symmetrically, we have $|\tilde{v}_b| \geq 2^{k-1}$. Thus we have $|\tilde{v}| = |\tilde{v}_a| + |\tilde{v}_b| \geq 2^k$. This completes the proof. □

LEMMA 2.6. *Let $\tilde{G}$ be a k-pseudo-clique derived from a perfect matching graph G. For $k > 2$, we must have $|G| \geq k \times 2^k$.*

PROOF. We prove the lemma by showing that each of the k pseudo-nodes in $\tilde{G}$ contains at least 2^k vertices. Consider any such pseudo-node $\tilde{v}$. Because $\tilde{G}$ is derived from a perfect matching graph, when $k > 2$, each pseudo-node must contain an expanding vertex pair. For concreteness, let (v_a, v_b) be the expanding vertex pair in $\tilde{v}$. After expansion, pseudo-node $\tilde{v}$ is split into pseudo-node $\tilde{v}_a \ni v_a$ and pseudo-node $\tilde{v}_b \ni v_b$. Denote the resulting pseudo-graph by $\tilde{G}_s$. Since there is no loop-vertex in a perfect matching graph, $\tilde{v}$ must contain a perfectly matched vertex pair (v_c, v_d), which connects $\tilde{v}_a$ to $\tilde{v}_b$ in $\tilde{G}_s$. Assume w.l.o.g. $v_c \in \tilde{v}_a$ and $v_d \in \tilde{v}_b$.

We discuss three possible expansion cases as depicted in Figure 4 and show in each case that $|\tilde{v}| \geq 2^k$.

(1) $|\{v_c, v_d\} \cap \{v_a, v_b\}| = 2$. In other words, $v_a = v_c$ and $v_b = v_d$. After the expansion, $\tilde{v}_a$ is a singly-adjacent-pseudo-node with pseudo-degree k in the resulting pseudo-graph $\tilde{G}_s$. According to Lemma 2.5, we have $|\tilde{v}_a| \geq 2^{k-1}$. Symmetrically, $|\tilde{v}_b| \geq 2^{k-1}$. Hence $|\tilde{v}| \geq 2^k$.

(2) $|\{v_c, v_d\} \cap \{v_a, v_b\}| = 1$. Assume w.l.o.g. $v_a = v_c$. After the expansion, it is possible that v_a has pseudo-degree 2 in the resulting graph $\tilde{G}_s$. To reduce its pseudo-degree to 1, we merge the pseudo-node(s) adjacent to v_a into one single pseudo-node, and obtain a pseudo-graph $\tilde{G}_m$. Now $\tilde{v}_a$ is a singly-adjacent-pseudo-node in $\tilde{G}_m$ with pseudo-degree at least $k-1$. Since there is no loop vertex in $\tilde{v}_a$ and $\tilde{v}_a$ contains an expanding vertex pair, $\tilde{v}_a$ must contain a perfectly matched vertex pair v_{aa} and v_{ab}. By expanding $\tilde{v}_a$ and merging the pseudo-node(s) adjacent to v_{aa} into a single pseudo-node as before, we can derive a complete pseudo-graph in which $\tilde{v}_{aa}$ is a singly-adjacent-pseudo-node with pseudo-degree $k-1$ that contains an expanding vertex pair for the pseudo-graph. According to Lemma 2.5, we have $|\tilde{v}_{aa}| \geq 2^{k-2}$. Hence $|\tilde{v}_a| \geq 2^{k-1}$. Similarly, we can show $|\tilde{v}_b| \geq 2^{k-1}$. Thus we have $|\tilde{v}| \geq 2^k$.

(3) $|\{v_c, v_d\} \cap \{v_a, v_d\}| = 0$. By an argument similar to that of case 2, we can show $|\tilde{v}| \geq 2^k$.

This completes the proof. □

We note that for $|G| < 24$, the largest pseudo-clique that can be derived from a perfect matching graph G is a 2-pseudo-clique.

2.2.5 Statement of the lower bound Lemmas 2.4 and 2.6 complete the induction and lead to the following theorem and corollary, the straightforward proofs of which are omitted for brevity.

THEOREM 2.7. *Given a perfect matching graph G with n vertices, it takes at least $n-k$ translocations to transform G into an n-clique, where $k = 2$ when $n < 24$; when $n \geq 24$, k is the largest integer that satisfies $k \times 2^k \leq n$.*

COROLLARY 2.8. *$D(n) \geq n-k$, where $k = 2$ when $n < 24$; when $n \geq 24$, k is the largest integer that satisfies $k \times 2^k \leq n$.*

3 GENOME HALVING DISTANCE

While the diameter problem attempts to find the maximum halving distance for all genomes of size n, the distance problem attempts to find the halving distance for a particular genome of size n. By definition, the halving distance for a particular genome is less than or equal to the diameter.

3.1 Genome halving distance: upper bound

We can obtain a tighter upper bound on the genome halving distance by analyzing the algorithm presented in the proof for Theorem 2.1 more closely. In the worst case, it may take two translocations to obtain each perfectly matched pair of vertices. However, if the intersection graph contains a non-loop 1-vertex, a perfectly matched vertex pair can be produced using just one translocation. In some sense, the existence of a non-loop 1-vertex has the potential to save one translocation in transforming the intersection graph to a perfect matching graph. This observation leads to the following lemma.

LEMMA 3.1. *Given a genome G of size n, $d(G) \leq n-2+\gamma(G)-\min\{s, (n-4)/2\}$, where s is the number of non-loop 1-vertices in G; $\gamma(G) = 1$ if G is a loopy genome, and $\gamma(G) = 0$ otherwise.*

PROOF. During the transformation of the final four vertices, the existence of a non-loop 1-vertex does not necessarily help to save translocations; for example, two translocations are still required to transform a star graph with four vertices to a perfect matching graph (a star graph is one in which one central vertex is adjacent to all the other 1-vertices). In contrast, when the number of remaining white vertices is greater than 4, the existence of a non-loop 1-vertex can always save one translocation. However, to achieve the potential savings of a non-loop 1-vertex, we need to *consume* two vertices. More precisely, after one translocation, the non-loop 1-vertex and its neighbor become a perfectly matched pair and thus cannot be used in future translocations. The claim then follows. □

By extending the intuition behind Lemma 3.1, we can get an even better upper bound on $d(G)$. For readability, in the remainder of Section 3.1, we sometimes omit the non-loop description for a vertex when it is clear from context.

Given a graph $G(V, E)$, a *well-separated vertex set* $W \subset V$ is a set of non-loop-vertices such that:

(1) for any $v_i, v_j \in W$, v_i and v_j are not adjacent and share no common neighbor if $h(v_i) > 1$ and $h(v_j) > 1$; and

(2) $\sum_{v_i \in W} h(v_i) \leq (n-4)/2$.

THEOREM 3.2. *Given a genome G of size n, $d(G) \leq n - 2 + \gamma(G) - |W^\star|$, where $W^\star$ denotes the maximum well-separated vertex set contained in G, and $\gamma(G)$ is defined as in Lemma 3.1.*

PROOF. Observe that we can create two 1-vertices by performing a translocation between the two neighbors of a 2-vertex. For example, consider the case depicted in Figure 5 in which v_1 is a 2-vertex: after the translocation $\delta(v_2, v_3, \emptyset, B_3)$ where $B_3 = \mathcal{I}(v_3) \setminus I(v_3, v_4)$, we obtain two 1-vertices, δv_1 and δv_3. Therefore by Lemma 3.1, we have that the existence of a 2-vertex can also save one translocation. However, four vertices (v_1, v_2, v_3 and v_4) are consumed to achieve the potential savings of a 2-vertex. In general, we can create a $(k-1)$-vertex from a k-vertex with one translocation. By applying the above procedure recursively, any k-vertex has the potential to save one translocation at a cost of consuming $2k$ vertices.

In addition, to realize the potential savings of a non-loop 1-vertex v_1 whose only neighbor is v_2, we first find a vertex $v' \in V \setminus W^\star$ that will not be consumed during the processing of the vertices in $W^\star$. Note that the existence of v' is guaranteed by the well-separatedness of $W^\star$: the total number of vertices that will be consumed will be at most $2 \times \sum_{v_i \in W^\star} h(v_i) \leq n - 4 < n$. Then perform translocation $\delta(v_2, v', B_2, \emptyset)$ where $B_2 = \mathcal{I}(v_2) \setminus I(v_1, v_2)$. This leaves $(v_1, \delta v_2)$ as a perfectly matched pair.

Label the vertices in $W^\star$ as $\alpha_1, \alpha_2, \ldots, \alpha_{|W^\star|}$ such that $h(\alpha_i) \leq h(\alpha_j)$ for all $i < j$. If $h(\alpha_j)$ increases, we say α_j is *destroyed*. We now show that if we process the vertices $\alpha_1, \alpha_2, \ldots, \alpha_{|W^\star|}$ in order, then we neither consume nor destroy any $\alpha_j \in W^\star$ while processing $\alpha_i \in W^\star$.

- If $h(\alpha_i) = 1$, we realize the potential savings of α_i by touching only its neighbor and v'. By the well-separatedness of $W^\star$, no $\alpha_j \in W^\star$ is consumed or destroyed.

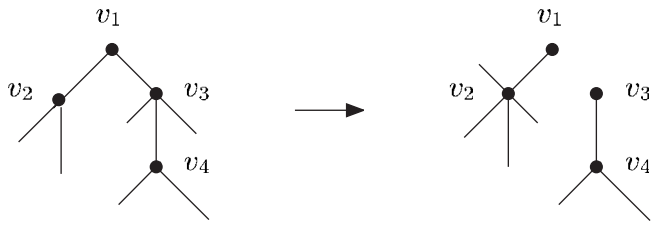

Fig. 5. A translocation between two neighbors of a 2-vertex, v_1, can produce two 1-vertices (in this case v_1 and v_3).

- If $h(\alpha_i) > 1$, realizing the potential savings of α_i may affect α_i, its neighbors, v', and possibly some vertices adjacent to α_i's neighbors. But by this point, all the potential savings of 1-vertices must have already been realized (since the vertices are processed in order), and any $\alpha_j \in W^\star$ with $h(\alpha_j) > 1$ shares no neighbor with α_i by the well-separatedness of $W^\star$. Therefore, no $\alpha_j \in W^\star$ is consumed or destroyed.

Thus, we can fully realize the potential savings of all the $\alpha_i \in W^\star$, resulting in the upper bound as claimed. □

3.2 Genome halving distance: lower bound

By studying the so-called *fan structure* of the intersection graph induced by genome G, El-Mabrouk *et al*. obtain a lower bound of $\lceil \log_2([(e-n/2)/p]+1)\rceil$ for $d(G)$, where n is the number of chromosomes in G, e is the number of edges in the intersection graph representing G, and p is the largest number of neighbors shared by any two vertices in the intersection graph. Their strategy is to count the maximum number of edges that can be reduced with one translocation. This strategy is also at the core of their greedy algorithm to find the optimal number of translocations. In this section, we use a different strategy to derive a lower bound that is almost always tighter than the above lower bound; some experimental evidence for this claim comes in the analysis of the yeast genome later in the paper.

We have the following lemma.

LEMMA 3.3. *$d(G) \geq \lceil \frac{h(v^\star)}{2} \rceil$, for $h(v^\star) > 1$, where $v^\star$ is the vertex with the maximum degree in G.*

PROOF. When $h(v^\star) > 1$, label edges initially incident to $v^\star$ as *bad*. A bad edge can disappear by being merged into another bad edge. Alternatively, it can become an edge connecting the two vertices of a perfectly matched vertex pair, in which case we say the edge becomes *good*. Let $b(G)$ be the number of bad edges in G. Initially, $b(G) = h(v^\star)$. Since there are no bad edges in the final perfect matching graph, we must remove $b(G)$ bad edges to arrive at the final graph. We enumerate below all possible types of translocations and their influence upon $b(G)$.

- A translocation between $v^\star$ and a neighbor v_1 decreases $b(G)$ by at most two. Such a translocation can happen when $v^\star$ is a loop-vertex with a 1-vertex neighbor, v_2. A translocation between $v^\star$ and v_1 turns the edge $e(v^\star, v_2)$ into a good edge, and merges edges $e(v^\star, v^\star)$ and $e(v^\star, v_1)$ into one bad edge. See Figure 6 for an illustration.
- A translocation between $v^\star$ and a vertex v_1 not adjacent to $v^\star$ decreases $b(G)$ by at most two. Such a translocation can happen when $v^\star$ is a 2-vertex, v_1 is a 1-vertex, and $v^\star$ and v_1 have a common neighbor 2-vertex, v_2, as illustrated in Figure 7. A translocation between $v^\star$ and v_1 turns both of the bad edges incident to $v^\star$ into good edges.
- A translocation between two neighbors of $v^\star$ decreases $b(G)$ by at most two. This case can happen when $v^\star$ is a 2-vertex with a neighboring 1-vertex, as illustrated in Figure 8. A translocation between v_1 and v_2 merges the two bad edges incident to $v^\star$ into one good edge.
- A translocation between a neighbor of $v^\star$ and a vertex not adjacent to $v^\star$ or between two vertices neither of which is adjacent to $v^\star$ does not decrease $b(G)$ (though it may increase $b(G)$ by one).

As a single translocation decreases $b(G)$ by at most two, the total number of translocations required is at least $\lceil h(v^\star)/2 \rceil$. □

Note that when $h(v^\star) = 1$, our lower bound is simply $d(G) \geq 0$, since in a perfect matching graph, $h(v^\star) = 1$ and $d(G) = 0$.

By extending Lemma 3.3, we get the following theorem.

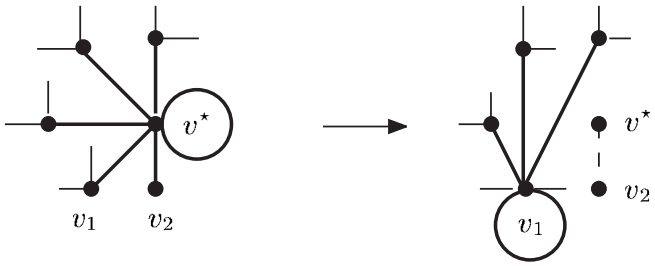

Fig. 6. There exists a translocation between $v^\star$ and v_1 that decreases the number of bad edges by two. Bad edges are depicted as bold solid segments; good edges as dashed segments.

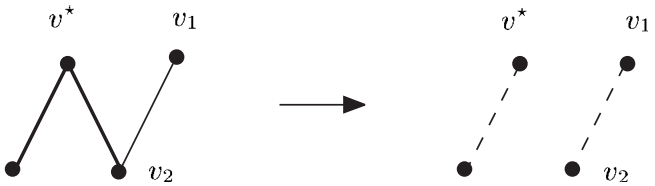

Fig. 7. There exists a translocation between $v^\star$ and v_1 that decreases the number of bad edges by two. Bad edges are depicted as bold solid segments; good edges as dashed segments.

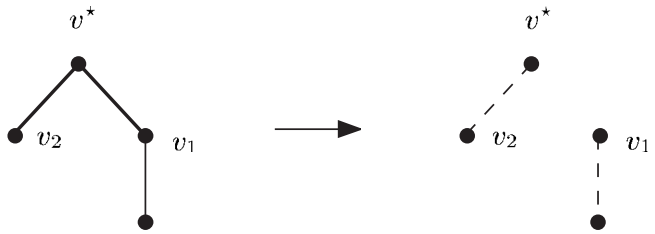

Fig. 8. There exists a translocation between v_1 and v_2 that decreases the number of bad edges by two. Bad edges are depicted as bold solid segments; good edges as dashed segments.

THEOREM 3.4. *Given a graph $G(V, E)$, let V_1 and V_2 be two disjoint subsets of V, such that no vertex in $V_1 \cup V_2$ is part of any perfectly matched vertex pair, and every vertex in V_1 has a neighbor in V_2 and vice versa. We have*

$$d(G) \geq \max_{V_1, V_2 \subset V} \left\lceil \frac{\max(|V_1|, |V_2|)}{2} \right\rceil$$

PROOF. Assume w.l.o.g. $|V_1| \geq |V_2|$. Let us redefine the initial selection criteria for bad edges from before. For each $v \in V_1$, choose an arbitrary edge incident to v and label it as a *bad* edge. Again, since there are no bad edges in the final perfect matching graph, the total change in the number of bad edges must be $|V_1|$. Similar to the analysis in Lemma 3.3, we can show that any translocation decreases the number of bad edges by at most two. This proves the theorem. □

4 ALGORITHM TO RECONSTRUCT ANCESTRAL DUPLICATED GENOMES

We now present an algorithm to reconstruct ancestral duplicated genomes based on the intuition behind the proof of Theorem 3.2. The algorithm GENOME-HALVING first colors perfectly matched vertices black and other vertices white. Then it processes the white vertices until either (1) there are only white loop-vertices left, at which point it calls procedure LOOP-VERTICES; or (2) there are at most four white vertices left, at which point it calls procedure 4-VERTICES.

We describe the algorithm in detail below. We first present the main routine GENOME-HALVING and then describe the procedures LOOP-VERTICES and 4-VERTICES called by GENOME-HALVING. Finally, we describe a routine 1-VERTEX that is called by both GENOME-HALVING and LOOP-VERTICES.

The main algorithm GENOME-HALVING($G(V, E)$) is presented below.

(1) Color all perfectly matched vertices in V black, and color the remaining vertices white.

(2) If no white non-loop-vertex exists, call the procedure LOOP-VERTICES(V), which will terminate.

(3) If the number of white vertices is less than or equal to four, call the procedure 4-VERTICES(V), which will terminate.

(4) Find a white non-loop-vertex v_1 of the smallest degree. If $h(v_1) = 1$, call the procedure 1-VERTEX(v_1, V). Otherwise, label any two of its neighbors as v_2 and v_3. Since vertex v_2 is not a non-loop 1-vertex, it must have another neighbor different from v_1. Label it as v_4. Note that it is possible $v_4 = v_2$ or $v_4 = v_3$. Perform translocation $\delta(v_2, v_3, B_2, \emptyset)$ where $B_2 = \mathcal{I}(v_2) \setminus I(v_2, v_4)$. Then v_2 becomes a 1-vertex, and $h(v_1)$ is decreased by one. Call the procedure 1-VERTEX(δv_2, δV). Note that at this point, δV contains at least four white vertices.

(5) Repeat Steps 2, 3 and 4 until termination.

For a vertex set V whose white vertices are all loop-vertices, we define the procedure LOOP-VERTICES(V) as follows:

(1) Arbitrarily select a white vertex v_1 and a white vertex v_2. Perform translocation $\delta(v_1, v_2, B_1, B_2)$ where $B_1 = \mathcal{I}(v_1) \setminus (I(v_1, v_1) \cup I(v_1, v_2))$ and $B_2 = I(v_2, v_2)$. Then v_1 becomes a non-loop 1-vertex.

(2) If v_2 also becomes a non-loop 1-vertex, color v_1 and v_2 black and go to Step 4.

(3) If, on the other hand, v_2 is not a non-loop 1-vertex, call the procedure 1-VERTEX(δv_1, δV). Note that at this point, δV contains at least four white vertices.

(4) If no white vertex remains, terminate; otherwise repeat Steps 1, 2 and 3.

For a vertex set V that contains at most four white vertices, the procedure 4-VERTICES(V) transforms the white vertices in V into perfectly matched vertex pairs with two or three translocations and terminates. We omit the details here for brevity.

For any non-loop 1-vertex $v_1 \in V$, where V contains at least four white vertices, we define the procedure 1-VERTEX(v_1, V) as follows:

(1) Label the neighbor of v_1 as v_2.

(2) Find the white vertex with the maximum degree in $V \setminus \{v_1, v_2\}$ and label it as v_3. Note that the existence of v_3 is guaranteed by the fact that V has at least four white vertices, including v_1 and v_2.

(3) Perform translocation $\delta(v_2, v_3, B_2, \emptyset)$ where $B_2 = \mathcal{I}(v_2) \setminus I(v_1, v_2)$. Color v_1 and v_2 black.

We have implemented the above algorithm in Java. A user-friendly graphical interface is provided for illustrating the sequence of translocations used to reconstruct an ancestral duplicated genome. For example, the result of halving a genome represented by an 8-clique graph with our program is shown in Figure 9.

5 GENOME HALVING DISTANCE AND ANCESTRAL GENOME FOR YEAST

To compare the results of our bounds analysis and of our algorithm with those reported by El-Mabrouk *et al*. (1998), we use the same yeast genome data set. The data was initially drawn from Wolfe and Shields (1997), and is reproduced here in Table 1.

The analysis of El-Mabrouk *et al*. (1998) gives the bounds, $3 \leq d(G) \leq 16$. Their program reconstructs a duplicated yeast genome using thirteen translocations, and they conjectured this value to be optimal based on a series of experiments they performed to find a lower halving distance. In comparison, our analysis yields the bounds, $6 \leq d(G) \leq 12$, and our algorithm halves the yeast genome using only eleven translocations, as shown in Figure 10 and the Appendix. In Table 2, we present the allocation of gene blocks among the chromosomes of one possible ancestral yeast genome immediately after genome duplication.

6 CONCLUSIONS AND FUTURE DIRECTIONS

In this paper, we define the concept of genome halving diameter $D(n)$ and obtain both upper and lower bounds for it. We also derive a tighter upper bound for the genome halving distance $d(G)$ than the best known. In addition, we develop a new strategy for computing a lower bound for genome halving distance; the lower bound we get is almost always tighter than the best known, and in particular, is tighter for the yeast genome halving problem. Furthermore, we design and implement a software package *GenomeHalving* to reconstruct possible ancestral duplicated genomes. The same yeast data set used by El-Mabrouk *et al*. (1998) is analyzed with our bound formulae, and tested with our

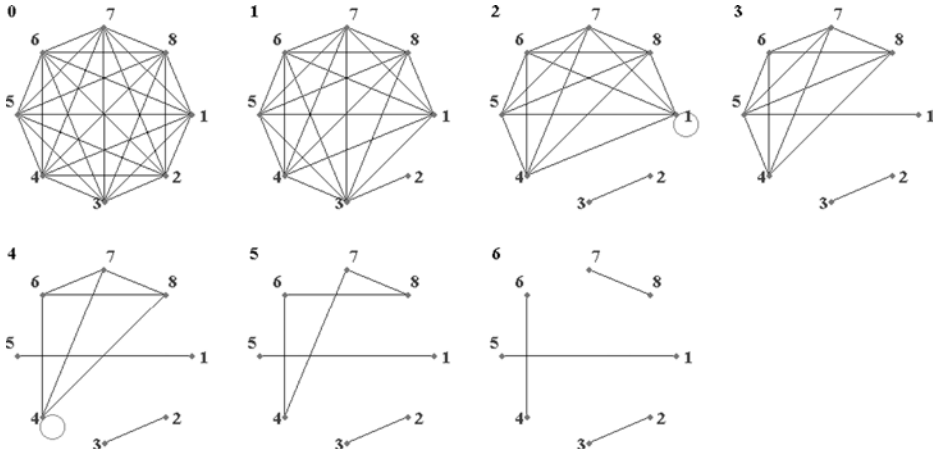

Fig. 9. One sequence of six translocations that transform an 8-clique graph into a perfect matching graph, provided as graphical output of our Java software package *GenomeHalving*.

Table 1. Gene blocks in the 16 chromosomes of the modern yeast genome

Chromosome	Gene blocks
I	2 1
II	4 3 7 8 5 6
III	9 10 11
IV	20 12 12 54 15 21 3 13 16 17 24 22 14 23 19 18 9
V	28 25 27 4 26 13
VI	55 36
VII	36 25 26 32 6 33 5 30 34 31 29
VIII	35 14 37 29 1
IX	38 39 27
X	10 40 41 28 42
XI	42 40 43 35 41 52 38
XII	53 53 31 55 16 18 17 45 30 15 44
XIII	46 44 19 43 54 48 47 46
XIV	49 20 37 50 39 11
XV	49 21 22 52 50 23 45 51 47 2
XVI	48 32 33 51 8 24 7 34

software. We are able to compute better upper and lower bounds, and also identify a sequence of translocations to halve the yeast genome that is of shorter length than was previously conjectured possible.

The reconstructed ancestral yeast genome, together with the history of translocations leading to this reconstruction, should be interpreted with the right perspective. It would be risky to assume that the results obtained here reflect the actual translocation history of the yeast genome. Any such interpretation of our results, or of evolutionary biology results obtained by combinatorial optimization analysis in general, is likely to rest on the unsubstantiated assumption that a genome takes the most parsimonious path possible when changing from one state to another. In actuality, a modern genome is the result of a long course of evolution that is shaped by a host of factors, many of which are not easily traceable today. In light of this, we would suggest that the result of any combinatorial optimization procedure applied to a problem in evolutionary biology should be interpreted as constraining the set of possible paths rather than suggesting a single definitive path.

Our lower bound on genome halving distance is almost always tighter than or equal to the best known. As mentioned before, the analysis of El-Mabrouk *et al*. (1998) gives a lower bound of 3 for the yeast genome with 16 chromosomes, while our analysis yields a lower bound of 6. As another example, the method of El-Mabrouk *et al*. (1998) applied to the 8-clique graph of Figure 9 gives a lower bound of 3 while our analysis yields a lower bound of 4. In general, the method of El-Mabrouk *et al*. (1998) gives lower bounds of $\lceil \log_2((n/2)+1) \rceil$ and $\lceil \log_2(n/2) \rceil$ for an n-clique graph and an n-star graph, respectively, while our analysis yields tighter lower bounds of $n/2$ in both cases. Given a genome G with n chromosomes, the analysis of El-Mabrouk *et al*. (1998) always gives a lower bound less than $\lceil \log_2([n(n-2)/2p] + 1) \rceil$; in comparison, our method always yields a lower bound greater than $\lceil n/4 \rceil$. A detailed case-by-case analysis shows that only in some special cases when $n = 6$, 8 or 10 and $p = 1$ does the analysis of El-Mabrouk *et al*. (1998) yield a tighter lower bound; in all other cases, our lower bound is as tight or tighter. For genomes with $n \geq 20$, our lower bound is always strictly tighter.

Though we have not managed to derive an exact formula for the genome halving diameter $D(n)$, our upper and lower bound almost always match for genomes with a realistic number of chromosomes: for a non-loopy genome with fewer than $3 \times 2^3 = 24$ chromosomes, our lower bound equals our upper bound. For a non-loopy genome with chromosomal number between 24 and $4 \times 2^4 = 64$, or a loopy genome with fewer than 24 chromosomes, our upper bound differs from our lower bound by only one.

There is ample room for further research. A more careful analysis of the structure of the intersection graph might render insight into strategies for tightening the upper and lower bounds on genome halving diameter $D(n)$ as well as genome halving distance $d(G)$. Ideally, we would like to find exact formulae for both problems. Formulating an algorithm for calculating the lower bound on $d(G)$ encompasses an interesting graph theory problem, and we would like to find a way to solve it.

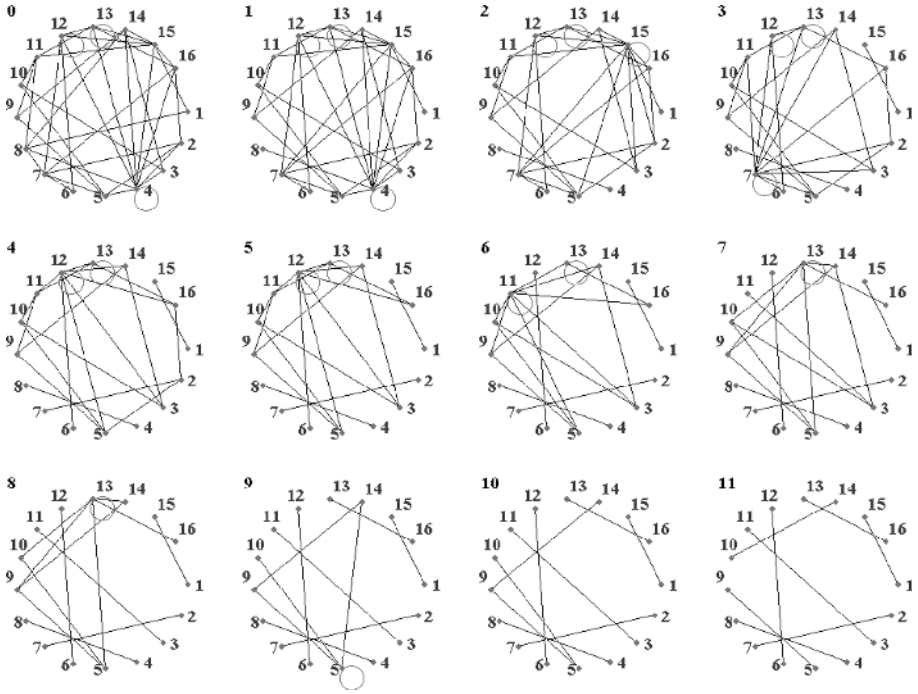

Fig. 10. One sequence of 11 translocations that transform the modern yeast genome into an ancestral duplicated genome, provided as graphical output of our Java software package *GenomeHalving*. Thirteen translocations was previously conjectured to be optimal. Details of the 11 translocations are given in the Appendix.

Table 2. Gene blocks in the 16 chromosomes of one possible ancestral yeast genome just after genome duplication

Chromosome	Gene blocks
I, XV	2 1
II, VII	6 5 3
III, XI	9
IV, VIII	14
V, IX	27 38 25 26 13 4 46 43 44 47 19 54 52 35 53 31 30 45 16 18 17 15 29 12 21 22 23
VI, XII	55 36
X, XIV	28 42 40 41 10 39 49 50 37 20 11
XIII, XVI	48 32 33 34 51 24 8 7

On the practical side, we would like to see how well our software package *GenomeHalving* performs for other genomes with evidence of ancient whole-genome duplication. However, this is pending more data from the genomics community. Though evidence for genome duplication is abundant in many species, the community has yet to reach a consensus view on whether whole-genome duplication occurred in these species. A particularly illustrative example is *Arabidopsis*. Though much of its genome is covered by paired chromosomal regions (AGI, 2000; Paterson *et al*., 2000), arguments have been made supporting a single whole-genome duplication event (Lynch and Conery, 2000) or multiple duplication events at different times (Vision *et al*., 2000). In light of this, we are cautious to restrict our attention to the yeast genome, the only genome in which the community has a widely accepted view of whole-genome duplication.

ACKNOWLEDGEMENTS

The authors are most grateful to Yusu Wang, Nabil Mustafa, and Pankaj K. Agarwal for fruitful discussions on the mathematical and algorithmic aspects of the work; to Zhenglong Gu for helpful comments on the biological aspects of the work; and to anonymous reviewers for their suggestions on further improving the paper. PH also wishes to thank the NSF for support on grant CCR-03-26157. A.J.H. wishes to thank the Alfred P Sloan foundation for a Sloan fellowship and the NSF for support under its CAREER award program.

REFERENCES

AGI: *Arabidopsis* Genome Initiative (2000) Analysis of the genome sequence of the flowering plant *Arabidopsis thaliana*. *Nature*, **408**, 796–815.

Ahn,S. and Tanksley,S.D. (1993) Comparative linkage maps of rice and maize genomes. *Proc. Natl Acad. Sci. USA*, **90**, 7980–7984.

Coissac,E., Maillier,E. and Netter,P. (1997) A comparative study of duplication in bacteria and eukaryotes: the importance of telomeres. *Mol. Biol. Evol.*, **14**, 1062–1074.

Dietrich,F.S., Voegeli,S., Brachat,S., Lerch,A., Gates,K., Steiner,S., Mohr,C., Pohlmann,R., Luedi,P., Choi,S. *et al*. (2004) The *Ashbya gossypii* genome as a tool for mapping the ancient *Saccharomyces cerevisiae* genome. *Science*, **304**, 304–307.

Durand,D. (2003) Vertebrate evolution: doubling and shuffling with a full deck. *Trends Genet.*, **19**, 2–5.

El-Mabrouk,N., Nadeau,J.H. and Sankoff,D. (1998) Genome halving. *Lect. Notes Comput. Sci.*, **1448**, 235–250.

El-Mabrouk,N., Bryant,D. and Sankoff,D. (1999) Reconstructing the pre-doubling genome. In Istrail,S. *et al*. (eds), *Proceedings of the Third Annual International*

Conference on Computational Molecular Biology (RECOMB99). ACM Press, New York, pp. 154–163.

El-Mabrouk,N. (2000) Recovery of ancestral tetraploids. In Sankoff,D. and Nadeau,J.H. (eds), *Comparative Genomics: Empirical and Analytical Approaches to Gene Order Dynamics, Map Alignment and the Evolution of Gene Families. Computational Biology Series*. Kluwer Academic Publishers, The Netherlands, Vol. 1, pp. 465–477.

El-Mabrouk,N. and Sankoff,D. (2002) The reconstruction of doubled genomes. *SIAM J. Comput.*, **32**, 754–792.

Fryxell,K.J. (1996) The co-evolutation of gene family trees. *Trends Genet.*, **12**, 364–369.

Gaut,B.S. and Doebley,J.F. (1997) DNA sequence evidence for the segmental allotetraploid origin of maize. *Proc. Natl Acad. Sci. USA*, **94**, 6809–6814.

Gibson,T.J. and Spring,J. (2000) Evidence in favor of ancient octaploidy in the vertebrate genome. *Biochem. Soc. Trans.*, **2**, 259–264.

Gu,X., Wang,Y. and Gu,J. (2002) Age distribution of human gene families shows significant roles of both large- and small-scale duplications in vertebrate evolution. *Nat. Genet.*, **31**, 205–209.

Kellis,M., Birren,B.W. and Lander,E.S. (2004) Proof and evolutionary analysis of ancient genome duplication in the yeast *Saccharomyces cerevisiae*. *Nature*, **428**, 617–624.

Ku,H.M., Vision,T., Liu,J. and Tanksley,S.D. (2000) Comparing sequence segments of the tomato and *Arabidopsis* genomes: large-scale duplication followed by selective gene loss creates a network of synteny. *Proc. Natl Acad. Sci. USA*, **97**, 9121–9126.

Langkjær,R.B., Cliften,P.F., Johnston,M. and Piskur,J. (2003) Yeast genome duplication was followed by asynchronous differentiation of duplicated genes. *Nature*, **421**, 848–852.

Llorente,B., Malpertuy,A., Neuveglise,C., de Montigny,J., Aigle,M., Artiguenave,F., Blandin,G., Bolotin-Fukuhara,M., Bon,E., Brottier,P. *et al*. (2000a) Genomic exploration of the hemiascomycetous yeasts 18. Comparative analysis of chromosome maps and synteny with *Saccharomyces cerevisiae*. *FEBS Lett.*, **487**, 101–112.

Llorente,B., Durrens,P., Malpertuy,A., Aigle,M., Artiguenave,F., Blandin,G., Bolotin-Fukuhara,M., Bon,E., Brottier,P., Casaregola,S. *et al*. (2000b) Genomic exploration of the hemiascomycetous yeasts 20. Evolution of gene redundancy compared to *Saccharomyces cerevisiae*. *FEBS Lett.*, **487**, 122–133.

Lundin,L.G. (1993) Evolution of vertebrate genome as reflected in paralogous chromosomal regions in man and in the house mouse. *Genomics*, **16**, 1–19.

Lynch,M. and Conery,J.S. (2000) The evolutionary fate and consequences of duplicated genes. *Science*, **290**, 1151–1155.

McLysaght,A., Hokamp,K. and Wolfe,K.H. (2002) Extensive genomic duplication during early chordate evolution. *Nat. Genet.*, **31**, 200–204.

Mewes,H.W., Albermann,K., Bahr,M., Frishman,D., Gleissner,A., Hani,J., Heumann,K., Kleine,K., Maierl,A., Oliver,S.G., Pfeiffer,F. and Zollner,A. (1997) Overview of the yeast genome. *Nature*, **387**, S7–S65.

Moore,G., Devos,K.M., Wang,Z. and Gale,M.D. (1995) Cereal genome evolution. Grasses, line up and form a circle. *Curr. Biol.*, **5**, 737–739.

Nadeau,J.H. (1991) Genome duplication and comparative mapping. In Adolph,K.W. (eds.), *Advanced Techniques in Chromosome Research*. Marcel Dekker Press, New York, pp. 269–296.

Ohno,S. (1970) *Evolution by Gene Duplication*. Springer-Verlag Press, New York.

Paterson,A.H., Lan,T.H., Reischmann,K.P., Chang,C., Lin,Y.R., Liu,S.C., Burow,M.D., Kowalski,S.P., Katsar,C.S., DelMonte,T.A. *et al*. (1996) Toward a unified genetic map of higher plants, transcending the monocot-dicot divergence. *Nat. Genet.*, **14**, 380–382.

Paterson,A.H., Bowers,J.E., Burow,M.D., Draye,X., Elsik,C.G., Jiang,C.X., Katsar,C.S., Lan,T.H., Lin,Y.R., Ming,R. and Wright,R.J. (2000) Comparative genomics of plant chromosomes. *Plant Cell*, **12**, 1523–1540.

Scheffler,J.A., Sharpe,A.G., Schmidt,H., Sperling,P., Parkin,I.A.P., Luhs,W., Lydiate,D.J. and Heinz,E. (1997) Desaturase multigene families of *Brassica napus* arose through genome duplication. *Theor. Appl. Genet.*, **94**, 583–591.

Seoighe,C., Federspiel,N., Jones,T., Hansen,N., Bivolarovic,V., Surzycki,R., Tamse,R., Komp,C., Huizar,L., Davis,R.W. *et al*. (2000) Prevalence of small inversions in yeast gene order evolution. *Proc. Natl Acad. Sci. USA*, **97**, 14 433–14 437.

Seoighe,C. and Wolfe,K.H. (1998) Extent of genomic rearrangement after genome duplication in yeast. *Proc. Natl Acad. Sci. USA*, **95**, 4447–4452.

Shoemaker,R.C., Polzin,K., Labate,J., Specht,J., Brummer,E.C., Olson,T., Young,N., Concibido,V., Wilcox,J., Tamulonis,J.P. *et al*. (1996) Genome duplication in soybean. *Genetics*, **144**, 329–338.

Vision,T.J. and Brown,D.G. (2000) In Sankoff,D. and Nadeau,J.H. (eds), *Comparative Genomics: Empirical and Analytical Approaches to Gene Order Dynamics, Map Alignment and the Evolution of Gene Families. Computational Biology Series*. Kluwer Academic Publishers, The Netherlands, Vol. 1, pp. 479–491.

Vision,T.J., Brown,D.G. and Tanksley,S.D. (2000) The origins of genomic duplications in *Arabidopsis*. *Science*, **290**, 2114–2117.

Wolfe,K.H. (2001) Yesterday's polyploids and the mystery of diploidization. *Nat. Rev. Genet.*, **2**, 33–41.

Wolfe,K.H. and Shields,D.C. (1997) Molecular evidence for an ancient duplication of the entire yeast genome. *Nature*, **387**, 708–713.

APPENDIX

The sequence of translocations shown in Figure 10:

$$\delta_1 = \{S_8, S_{15}, \{1, 29, 35, 37\}, \emptyset\}$$
$$\delta_2 = \{S_4, S_{15}, \{3, 9, 12, 12, 13, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 54\}, \emptyset\}$$
$$\delta_3 = \{S_{15}, S_7, \{9, 12, 12, 13, 15, 16, 17, 18, 19, 20, 21, 21, 22, 22, 23, 23, 24, 29, 3, 35, 37, 45, 47, 49, 50, 51, 52, 54\}, \emptyset\}$$
$$\delta_4 = \{S_7, S_{12}, \{9, 12, 12, 13, 15, 16, 17, 18, 19, 20, 21, 21, 22, 22, 23, 23, 24, 25, 26, 29, 29, 30, 31, 32, 33, 34, 35, 36, 37, 45, 47, 49, 50, 51, 52, 54\}, \emptyset\}$$
$$\delta_5 = \{S_2, S_{12}, \{4, 7, 8\}, \emptyset\}$$
$$\delta_6 = \{S_{12}, S_{11}, \{4, 7, 8, 9, 12, 12, 13, 15, 15, 16, 16, 17, 17, 18, 18, 19, 20, 21, 21, 22, 22, 23, 23, 24, 25, 26, 29, 29, 30, 30, 31, 31, 32, 33, 34, 35, 37, 44, 45, 45, 47, 49, 50, 51, 52, 53, 53, 54\}, \emptyset\}$$
$$\delta_7 = \{S_{11}, S_{13}, \{4, 7, 8, 12, 12, 13, 15, 15, 16, 16, 17, 17, 18, 18, 19, 20, 21, 21, 22, 22, 23, 23, 24, 25, 26, 29, 29, 30, 30, 31, 31, 32, 33, 34, 35, 35, 37, 38, 40, 41, 42, 43, 44, 45, 45, 47, 49, 50, 51, 52, 52, 53, 53, 54\}, \emptyset\}$$
$$\delta_8 = \{S_3, S_{13}, \{10, 11\}, \emptyset\}$$
$$\delta_9 = \{S_{13}, S_5, \{4, 10, 11, 12, 12, 13, 15, 15, 16, 16, 17, 17, 18, 18, 19, 19, 20, 21, 21, 22, 22, 23, 23, 25, 26, 29, 29, 30, 30, 31, 31, 35, 35, 37, 38, 40, 41, 42, 43, 43, 44, 44, 45, 45, 46, 46, 47, 47, 49, 50, 52, 52, 53, 53, 54, 54\}, \emptyset\}$$
$$\delta_{10} = \{S_5, S_9, \{4, 11, 12, 13, 15, 16, 17, 18, 19, 20, 21, 22, 23, 25, 26, 29, 30, 31, 35, 37, 43, 44, 45, 46, 47, 49, 50, 52, 53, 54\}, \emptyset\}$$
$$\delta_{11} = \{S_5, S_{14}, \{10, 28, 40, 41, 42\}, \{11, 20, 37, 39, 49, 50\}\}$$