Genome Halving under DCJ Revisited

Julia Mixtacki

International NRW Graduate School in Bioinformatics and Genome Research,
Universität Bielefeld, Germany
julia.mixtacki@uni-bielefeld.de

Abstract. The Genome Halving Problem is the following: Given a rearranged duplicated genome, find a perfectly duplicated genome such that the rearrangement distance between these genomes is minimal with respect to a particular model of genome rearrangement. Recently, Warren and Sankoff studied this problem under the general DCJ model where the pre-duplicated genome contains both, linear and circular chromosomes. In this paper, we revisit the Genome Halving Problem for the DCJ distance and we propose a genome model such that constraints for linear genomes, as well as the ones for circular genomes are taken into account. Moreover, we correct an error in the original paper.

1 Introduction

Besides genome rearrangements, another important source for genome evolution is whole genome duplication. In the early 1970s, Susumu Ohno [13] came up with the hypothesis that whole genome duplication has occurred in mammalian evolution. Not without controversy, this question has been addressed several times in the last three decades, both in the biological and in the computational literature.

In fact, there is biological evidence for genome duplication among several eukaryotes. An outstanding example was the duplication in the yeast genome that was recently confirmed [12]. Even two rounds of duplication are found in vertebrates [5]. Duplication is a particularly common event in plants [1],[10] where most of the common crops have polyploid genomes.

The combinatorial problem, called the Genome Halving Problem, was first introduced in [8]: Assuming that a genome is duplicated and then rearranged over time, can we reconstruct an ancestral genome from the gene order that we observe today? The key to the solution of this question is the structure of the genome right after duplication: It must have been *perfect*, i.e. each chromosome has existed in two identical copies. Of course, there exist many perfectly duplicated genomes that could have been the ancestral genome. Therefore, we want to reconstruct one genome such that its *distance*, defined as the minimum number of rearrangements needed to transform it into the observed genome, is minimal.

Clearly, solutions to this problem depend on the underlying genome model and also on the rearrangement operations that are allowed. The most common genome rearrangement operations are *translocations*, *fusions*, *fissions*, *inversions*

X. Hu and J. Wang (Eds.): COCOON 2008, LNCS 5092, pp. 276–286, 2008.

and *block interchanges*. It is remarkable that all these operations can be modelled by a single one, called *double cut and join* (DCJ) operation [15]. As shown in [4], the DCJ operation applies for genomes with a mixture of linear and circular chromosomes. In contrast, in the Hannenhalli-Pevzner (HP) theory [11] it is assumed that the genomes only consist of linear chromosomes and only translocations, fusions, fissions and inversions are considered.

El-Mabrouk and Sankoff [9] solved the Genome Halving Problem under the HP distance. Their algorithm for the reconstruction of doubled genomes is far from being trivial and is the final result of a whole series of papers [8],[7],[6]. In addition to the well-known *breakpoint graph*, they introduce further graphs, called *natural graph* and *signature graph*. Later, Alekseyev and Pevzner gave an alternative approach based on the notion of *contracted breakpoint graph* [2] and corrected in [3] an error in the El-Mabrouk-Sankoff analysis.

Very recently, Warren and Sankoff [14] studied the Genome Halving Problem under the more general DCJ model. This generalization yields a simplified problem since some of the complicated components of the breakpoint graph, such as *hurdles* and *knots*, can be ignored. Unfortunately, their solution still relies on the complex concepts introduced by El-Mabrouk and Sankoff. Indeed, as we will see in this paper, the problem can be solved by working directly on the natural graph.

In the following, we will revisit the Genome Halving Problem under the double cut and join operation where the ancestral genome may contain linear and circular chromosomes. Therefore, in our genome model, we take into account both, the constraints usually required for genomes with only linear chromosomes, as well as the ones for genomes with only circular chromosomes. Compared to the more general model studied in [14], these requirements on the ancestral genome increase the distance between the genomes. This yields a new proof and a simple algorithm for reconstructing an ancestral genome. Moreover, by our results, we will also correct an error in the Warren-Sankoff analysis.

The structure of this paper is as follows. We begin by formalizing the problem in the next section. Then, in Section 3, we study the effect of a DCJ operation on the natural graph. In Section 4 we present our distance formula and a linear-time algorithm to reconstruct an ancestral genome with the minimum number of DCJ operations. Finally, we will discuss the Warren-Sankoff formula in Section 5. The last section summarizes our results and addresses some open questions.

2 Problem Formulation

As usual, a *gene* is represented by a directed identifier where the direction is indicated by a *head* and a *tail*. These are called the *extremities* of the gene. The tail of a gene a is denoted by a^t, and its head is denoted by a^h.

An *adjacency* of two consecutive genes a and b can be of four different types:

$$\{a^h, b^t\}, \{a^h, b^h\}, \{a^t, b^t\}, \{a^t, b^h\}.$$

An extremity that is not adjacent to any other gene is called a *telomere*, represented by a singleton set $\{a^h\}$ or $\{a^t\}$.

Definition 1. *A* duplicated genome *A is a set of adjacencies and telomeres such that the head and the tail of every gene appears exactly twice.*

Thus, a duplicated genome has two identical copies of each gene that are called *paralogs* and we distinguish them by a subscript, called an *assignment of the paralogs*. For a gene a, we denote its copies by a_1 and a_2 and the *paralogous extremities* by a_1^t, a_2^t and a_1^h, a_2^h.

Example 1. Consider the following genome A_1 defined on the set of genes $\{a, b, c, d\}$:

$$\{\{d_2^h\}, \{d_2^t, a_2^t\}, \{a_2^h, d_1^h\}, \{d_1^t, c_2^h\}, \{c_2^t, b_2^t\}, \{b_2^h\}, \{b_1^h\}, \{b_1^t, c_1^t\}, \{c_1^h, a_1^t\}, \{a_1^h\}\}$$

A genome can be represented as a graph, called the *genome graph*, with vertices corresponding to the adjacencies and telomeres and edges joining the head and the tail of each paralogous extremity. Thus, we have:

Suppose that the genome graph consists of N components C_1 to C_N. A *chromosome* is a set of adjacencies and telomeres that belong to the same component. Note that, by definition, each vertex in the genome graph has degree one or two, and thus the components of the genome graph are either *linear* or *circular*. We call a genome *linear* if all its chromosomes are linear. Similarly, a genome is *circular* if all its chromosomes are circular. For example, the above genome graph is a linear genome consisting of two linear chromosomes.

For paralogous extremities, we also use the following notation: if p is an extremity, then $\overline{p}$ is its corresponding paralogous extremity. By elevating this notation to sets of extremities, we can apply it to adjacencies and telomeres. For example, for an adjacency $x = \{a_1^h, b_2^t\}$, we have $\overline{x} = \{a_2^h, b_1^t\}$.

For a chromosome C, we define $\overline{C} = \{\overline{x} \mid x \text{ is an adjacency or telomere of } C\}$. This notation is useful to describe the different notions of a duplicated genome that can be found in the literature, for linear genomes in [9] and for circular genomes in [3]. By bringing this together for genomes with a mixture of linear and circular chromosomes, we have:

Definition 2. *A duplicated genome A consisting of chromosomes $C_1, \ldots, C_N$ is*

- linear-perfectly duplicated*, if for each linear chromosome C_i, we have $C_i = \overline{C_j}$ for some $j \in \{1, \ldots, N\} \backslash \{i\}$;*
- circular-perfectly duplicated*, if for each circular chromosome C_i, either we have $C_i = \overline{C_j}$ for some $j \in \{1, \ldots, N\} \backslash \{i\}$ or $C_i = C \cup \overline{C}$, where each adjacency of C_i occurs either in C or in $\overline{C}$, but not in both;*
- perfectly duplicated*, if it is linear- and circular-perfectly duplicated.*

Note that this definition does not depend on the assignment of the paralogs. Two examples of perfectly duplicated genomes are given in Fig. 1. From the

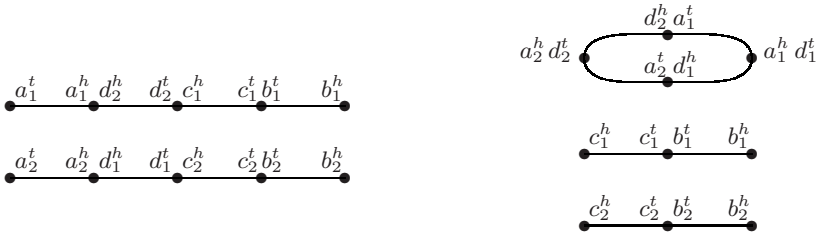

Fig. 1. Two perfectly duplicated genomes

right genome in that figure, we also see that the number of chromosomes of a perfectly duplicated genome is not necessarily even.

Alternatively to the formulation on the level of chromosomes, a perfectly duplicated genome can also be characterized locally, as stated by the next lemma.

Lemma 1. *A genome A is perfectly duplicated if and only if*

- *for each adjacency $\{u, v\}$ in A, also $\{\overline{u}, \overline{v}\}$ is in A and $u \neq \overline{v}$, and*
- *for each telomere $\{u\}$ in A, also $\{\overline{u}\}$ is in A.*

Now, let us consider rearrangement operations. Generally speaking, such an operation applied to two adjacencies or telomeres of a genome disconnects the incident edges of the genome graph, and reconnects them in one of the possible other ways. More formally, given a graph with vertices of degree one (*external* vertices) or degree two (*internal* vertices), we have:

Definition 3 ([4]). *The* double cut and join *(DCJ) operation acts on two vertices u and v of a graph with vertices of degree one or two in one of the following three ways:*

(a) If both $u = \{p, q\}$ and $v = \{r, s\}$ are internal vertices, these are replaced by the two vertices $\{p, r\}$ and $\{s, q\}$ or by the two vertices $\{p, s\}$ and $\{q, r\}$.
(b) If $u = \{p, q\}$ is internal and $v = \{r\}$ is external, these are replaced by $\{p, r\}$ and $\{q\}$ or by $\{q, r\}$ and $\{p\}$.
(c) If both $u = \{q\}$ and $v = \{r\}$ are external, these are replaced by $\{q, r\}$.

In addition, as an inverse of case (c), a single internal vertex $\{q, r\}$ can be replaced by two external vertices $\{q\}$ and $\{r\}$.

Given two genomes A and B, the *DCJ distance* denoted by $d_{DCJ}(A, B)$ is the minimum number of DCJ operations necessary to transform genome A into genome B. Thus, we can formulate the following problem:

The Genome Halving Problem. Given a rearranged duplicated genome A, find a perfectly duplicated genome B such that the DCJ distance between A and B is minimal.

To solve this problem, we will construct another graph in the next section. Again, the graph is defined on the adjacencies and telomeres of A, but this time it represents the relation between paralogous extremities.

3 Natural Graphs

Let us consider a duplicated genome A with n genes each present in two copies. Assume that the two paralogs of every gene are assigned arbitrarily.

Definition 4. *The* natural graph $NG(A)$ *is a graph whose vertices are the adjacencies and telomeres of A and, for each extremity, the two paralogous extremities are connected by an edge, i.e. two vertices u and v are connected if $p \in u$ and $\overline{p} \in v$.*

Observe that the total number of edges in the graph equals two times the number of genes. The natural graph of genome A_1 from Example 1 is given in Fig. 2.

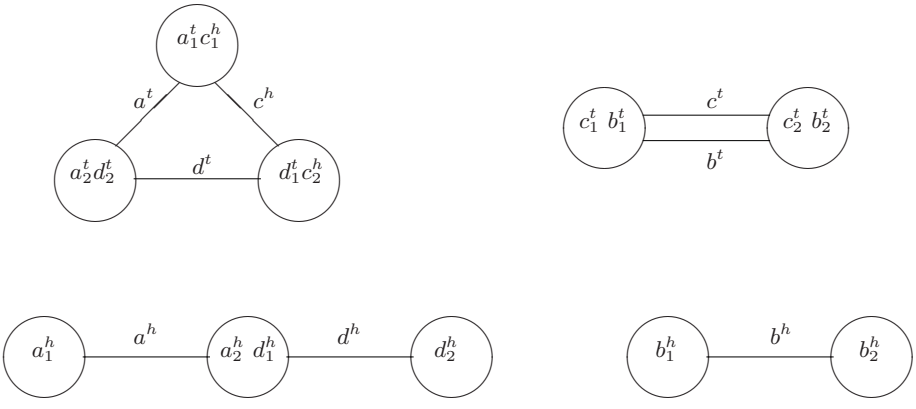

Fig. 2. Natural graph $N(A_1)$ of genome A_1 of Example 1

In a natural graph, by definition, every vertex has degree one or two. Thus, the natural graph consists only of cycles and paths.

Definition 5. *A cycle (or a path) with k edges, is a* k-cycle *(or* k-path*). If k is even, the cycle (or path) is called* even*, otherwise* odd*.*

Note that an adjacency $\{p, \overline{p}\}$ consisting of two paralogous extremities is a 1-cycle. The set of components of the natural graph can be partitioned into the following four disjoint subsets:

- EC := set of even cycles
- EP := set of even paths
- OC := set of odd cycles
- OP := set of odd paths

The following lemma is an immediate consequence of Lemma 1:

Lemma 2. *A genome A is perfectly duplicated if and only if all cycles in $NG(A)$ are 2-cycles and all paths in $NG(A)$ are 1-paths, i.e. $n = |EC| + |OP|/2$.*

4 Reconstructing an Ancestral Genome

In this section, we solve the genome halving problem by applying DCJ operations to the natural graph. This allows us to reconstruct a perfectly duplicated genome. We will first present our distance formula in Section 4.1 and then a linear time algorithm in Section 4.2.

4.1 Distance Formula

Consider a rearranged duplicated genome A. When a DCJ operation is applied to genome A, it acts on the adjacencies and telomeres of genome A. The same DCJ operation acts also on the natural graph $NG(A)$ since the adjacencies and telomeres of genome A are the vertices of this graph. Because the natural graph is a union of cycles and paths, all the properties of DCJ operations apply here as well, for instance: A DCJ operation can change the number of components only by one, as shown in [4]. Thus, we get a lower bound on the distance:

Lemma 3. *For a given genome A and any perfectly duplicated genome B over the same set of n genes, we have that*

$$d_{DCJ}(A, B) \geq n - (|EC| + \left\lfloor \frac{|OP|}{2} \right\rfloor).$$

In fact, there always exists a DCJ operation that increases either the number of even cycles or the number of odd paths. Thus, the distance decreases and the lower bound is strict as we see in the next theorem.

Theorem 1. *Let A be a rearranged duplicated genome with n genes each present in two copies, then the minimal distance between A and any perfectly duplicated genome B equals*

$$\min_B d_{DCJ}(A, B) = n - (|EC| + \left\lfloor \frac{|OP|}{2} \right\rfloor)$$

where $|EC|$ is the number of even cycles and $|OP|$ is the number of odd paths in the natural graph $NG(A)$.

Proof. We explain how to find a sequence of DCJ operations that achieves the lower bound of Lemma 3.

Let J, K, L and M be the total number of edges in all even cycles, even paths, odd cycles and odd paths, respectively. Note that the number of genes equals half of the total number of edges in $NG(A)$, i. e. $n = (J + K + L + M)/2$.

Consider a connected component G of $NG(A)$.

1. If G is an even j-cycle, we can create $\frac{j}{2}$ 2-cycles with $\frac{j}{2} - 1$ DCJ operations. Thus, for $|EC|$ even cycles with J edges in total, we need $\frac{J}{2} - |EC|$ DCJ operations to create $\frac{J}{2}$ 2-cycles.

2. If G is an even k-path, we can create $\frac{k}{2}$ 2-cycles with $\frac{k}{2}$ DCJ operations. Thus, for $|EP|$ even paths with K edges in total, we need $\frac{K}{2}$ DCJ operations to create $\frac{K}{2}$ 2-cycles.
3. If $|OP|$ is even, then $|OC|$ is also even.
   (a) If G is an odd l-cycle, we can create $\frac{l-1}{2}$ 2-cycles and one 1-cycle with $\frac{l-1}{2}$ DCJ operations. Thus, for $|OC|$ odd cycles with L edges in total, we need $\frac{L-|OC|}{2}$ DCJ operations to create $\frac{L-|OC|}{2}$ 2-cycles and $|OC|$ 1-cycles. We can choose two 1-cycles and create one 2-cycle. Since $|OC|$ is even, we can thus create $\frac{|OC|}{2}$ 2-cycles with $\frac{|OC|}{2}$ DCJ operations. Thus, in total we need $\frac{L-|OC|}{2} + \frac{|OC|}{2} = \frac{L}{2}$ DCJ operations.
   (b) If G is an odd m-path, we can create $\frac{m-1}{2}$ 2-cycles and one 1-path with $\frac{m-1}{2}$ DCJ operations. Thus, for $|OP|$ odd paths with M edges in total, we need $\frac{M-|OP|}{2}$ DCJ operations to create $\frac{M-|OP|}{2}$ 2-cycles and $|OP|$ 1-paths.

   Since L and M are even, summing up (a) and (b) gives us in total $\frac{L+M}{2} - \frac{|OP|}{2}$ DCJ operations.
4. If $|OP|$ is odd, then $|OC|$ is also odd.
   (a) If G is an odd l-cycle, we can create $\frac{l-1}{2}$ 2-cycles and one 1-cycle with $\frac{l-1}{2}$ DCJ operations. Thus, for $|OC|$ odd cycles with L edges in total, we need $\frac{L-|OC|}{2}$ DCJ operations to create $\frac{L-|OC|}{2}$ 2-cycles and $|OC|$ 1-cycles. We can choose two 1-cycles and create one 2-cycle. Since $|OC|$ is odd, there is one remaining 1-cycle that can be transformed into a 1-path by one extra DCJ operation. Thus, in total we need $\frac{L-|OC|}{2} + \frac{|OC|-1}{2} + 1 = \frac{L+1}{2}$ DCJ operations.
   (b) If G is an odd m-path, we can create $\frac{m-1}{2}$ 2-cycles and one 1-path with $\frac{m-1}{2}$ DCJ operations. Thus, for $|OP|$ odd paths with M edges in total, we need $\frac{M-|OP|}{2}$ DCJ operations to create $\frac{M-|OP|}{2}$ 2-cycles and $|OP|$ 1-paths.

   Since L and M are odd, summing up (a) and (b) gives us in total $\frac{L+1}{2} + \frac{M-|OP|}{2} = \frac{L+M}{2} - \frac{|OP|-1}{2}$ DCJ operations.

By bringing together the results, the distance formula follows. □

4.2 Algorithm

In this section, we show how the distance computation as well as an algorithm for reconstructing an ancestral genome can be implemented in linear time. Based on the proof of Theorem 1, our strategy for reconstructing a perfectly duplicated genome is the following:

1. Construct the natural graph.
2. Maximize the number of even cycles and odd paths in the natural graph.
3. Reconstruct the perfectly duplicated genome from the resulting natural graph.

The natural graph can easily be constructed in $O(n)$ time and $O(n)$ space if we store the information about the adjacencies and the telomeres in two tables. The first table represents the vertices of the natural graph. Each of its entries contains one or two extremities, depending whether it represents an adjacency or a telomere. The edges can be obtained from the second table that stores for each paralogous extremity the index of the vertex that contains it. The two tables for genome A_1 of Example 1 are given in Tables 1 and 2. Thus, the natural graph $NG(A_1)$ has 10 vertices and 8 edges, for example one edge joining vertex 10 with vertex 3, another edge joining vertex 9 with vertex 2, and so on.

Table 1. Table storing the adjacencies and telomeres of genome A_1. Adjacencies have two entries, telomeres just one.

	1	2	3	4	5	6	7	8	9	10
first	d_2^h	d_2^t	a_2^h	d_1^t	c_2^t	b_2^h	b_1^h	b_1^t	c_1^h	a_1^h
second	–	a_2^t	d_1^h	c_2^h	b_2^t	–	–	c_1^t	a_1^t	–

Table 2. Table storing for each gene in A_1 the location of its head and its tail in Table 1

	a_1	a_2	b_1	b_2	c_1	c_2	d_1	d_2
head	10	3	7	6	9	4	3	1
tail	9	2	8	5	8	5	4	2

Using these tables, the connected components can be computed in linear time, and thus the distance as given by Theorem 1.

In order to reconstruct a perfectly duplicated genome, we maximize the number of even cycles and odd paths in the natural graph. This is done by Algorithm 1, following the idea used in the proof of Theorem 1. By marking each adjacency of Table 1, Algorithm 1 can be implemented in linear time. The adjacencies are processed in left-to-right order and each time an unmarked adjacency is detected, all adjacencies on its path or cycle are marked and transformed into 2-cycles and 1-paths by successively applying DCJ operations. Note that, by applying a DCJ operation, at most 4 entries in each of the two tables have to be updated. Eventually, all cycles are 2-cycles and all paths are 1-paths and a perfectly duplicated genome can be obtained as follows: By ignoring the assignment of the paralogs, each 2-cycle consists of two adjacencies of the form $\{u^x, v^y\}$, where $x, y \in \{t, h\}$, and each 1-path connects two telomeres of the form u^x, where $x \in \{t, h\}$. Thus, a perfectly duplicated genome can be reconstructed by replacing each 2-cycle by the adjacency $\{u^x, v^y\}$ and each 1-path by the telomere u^x. So, the overall running time of the algorithm for reconstructing a perfectly duplicated genome is linear.

5 A Note on the Warren-Sankoff Formula

In [14], Warren and Sankoff consider a more general genome model where the ancestral genome has to be neither circular-perfectly duplicated, nor linear-perfectly duplicated. Therefore, we will use the notion *general-perfectly duplicated* in order to distinguish it from our definition of a perfectly duplicated genome. More precisely, a genome is general-perfectly duplicated if and only if for each adjacency $\{u, v\}$ in A, also $\{\overline{u}, \overline{v}\}$ is in A, and for each telomere $\{u\}$ in A,

Algorithm 1. Reconstruction of a perfectly duplicated genome

```
Construct NG(A), the natural graph of genome A
while there exists a k-path with k > 1 do
    Create a 2-cycle (and a (k − 2)-path if k > 2)
end while
/* all remaining paths have length 1 */
while there exists a k-cycle with k > 2 do
    Create a 2-cycle and a (k − 2)-cycle
end while
/* all remaining cycles have length 1 or 2 */
while there exists a 1-cycle do
    if there exists another 1-cycle then
        Create a 2-cycle
    else
        Create a 1-path
    end if
end while
```

also $\{\overline{u}\}$ is in A. Observe that, in contrast to our definition, a general-perfectly duplicated genome can have adjacencies of the type $\{u, \overline{u}\}$. For example, the following genome is general-perfectly duplicated, but not perfectly duplicated:

Now, let us denote by $d_{DCJ}^{general}(A, B)$ the minimum number of DCJ operations needed to transform a rearranged duplicated genome A into a general-perfectly duplicated genome B. By showing an upper and a lower bound, Warren and Sankoff finally claim that

$$\min_B d_{DCJ}^{general}(A, B) = n - (|EC| + |OP| + \left\lfloor \frac{|OC|}{2} \right\rfloor).$$

As a counterexample, consider a genome with just one gene a. Assume that the genome has two linear chromosomes, each consisting of one paralog a_1 and a_2. Note that the genome is general-perfectly duplicated and the natural graph has two paths of length one. Thus, the distance should be zero, but the above formula gives us

$$n - |OP| = 1 - 2 = -1.$$

Even though their distance formula is formulated in terms also defined in the natural graph, Warren and Sankoff follow a different approach. Therefore, instead of using their techniques, we will present in the following a correction of their result by modifying our algorithm.

As mentioned above, the difference is that a general-perfectly duplicated genome may have adjacencies that correspond to 1-cycles in the natural graph. Thus, we have:

Lemma 4. *A genome A is general-perfectly duplicated if and only if all cycles in $NG(A)$ are 2-cycles or 1-cycles, and all paths in $NG(A)$ are 1-paths.*

As a consequence of this lemma, we do not have to apply DCJ operations in order to get rid of 1-cycles in the natural graph as in our genome model. Since there are at most $\lceil |OC|/2 \rceil$ such DCJ operations, one can easily show that

$$\min_B d_{DCJ}(A, B) = \min_B d_{DCJ}^{general}(A, B) + \left\lceil \frac{|OC|}{2} \right\rceil .$$

By this fundamental relation, one can derive the distance formula for the general DCJ model studied by Warren and Sankoff in [14]:

Theorem 2. *Let A be a rearranged duplicated genome with $2n$ genes, then the minimal distance between A and any perfectly duplicated genome B equals*

$$\min_B d_{DCJ}^{general}(A, B) = n - (|EC| + \frac{|OP| + |OC|}{2})$$

where $|EC|$ is the number of even cycles, $|OC|$ the number of odd cycles and $|OP|$ the number of odd paths in the natural graph $NG(A)$.

It should be mentioned that an optimal algorithm for reconstructing a general-perfectly duplicated genome is obtained by just removing the last while-loop in our Algorithm 1.

6 Conclusion and Open Questions

In this paper, we solve the Genome Halving Problem for the DCJ distance under a general genome model with coexisting circular and linear chromosomes. Surprisingly, this can be done by working directly on the natural graph — all other graphs that are typically used in this context are bypassed. Moreover, our approach is also able to describe alternative genome models such as the one presented by Warren and Sankoff. Thus, our genome model represents a firm starting point for further studies and variants of the Genome Halving Problem.

One direction is to consider a more general set of rearrangement operations, the so-called *multi-break* rearrangements. By this generalization, a DCJ operation is equivalent to a 2-break operations and transpositions can be modelled by a 3-break operation instead of two DCJ operations as in our model. Therefore, the results of [3] can be extended to genomes with linear and circular chromosomes.

Finally, one can consider duplicated genomes with a higher multiplicity of each gene. This extension yields a natural graph with vertices of degree greater than two. It would have to be studied whether the DCJ operation can also be used on such a graph and how to partition the connected components.

Acknowledgments

The author would like to thank Jens Stoye and Robert Warren for helpful discussions. The anonymous reviewers gave valuable hints how to improve the paper.

References

1. Ahn, S., Tanksley, S.D.: Comparative Linkage Maps of Rice and Maize Genomes. Proc. Natl. Acad. Sci. 90(17), 7980–7984 (1993)
2. Alekseyev, M., Pevzner, P.: Whole Genome Duplications and Contracted Breakpoint Graphs. SIAM J. Comput. 36(6), 1748–1763 (2007)
3. Alekseyev, M., Pevzner, P.: Whole Genome Duplications, Multi-break Rearrangements, and Genome Halving Problem. In: Proceedings of SODA 2007, pp. 665–679 (2007)
4. Bergeron, A., Mixtacki, J., Stoye, J.: A Unifying View of Genome Rearrangements. In: Bücher, P., Moret, B.M.E. (eds.) WABI 2006. LNCS (LNBI), vol. 4175, pp. 163–173. Springer, Heidelberg (2006)
5. Dehal, P., Boore, J.L.: Two Rounds of Whole Genome Duplication in the Ancestral Vertebrate. PLoS Biology 3(10), 314 (2003)
6. El-Mabrouk, N.: Reconstructing an Ancestral Genome Using Minimum Segments Duplications and Reversals. J. Comput. Syst. Sci. 65(3), 442–464 (2002)
7. El-Mabrouk, N., Bryant, D., Sankoff, D.: Reconstructing the Pre-doubling Genome. In: Proceedings of RECOMB 1999, pp. 154–163 (1999)
8. El-Mabrouk, N., Nadeau, J., Sankoff, D.: Genome Halving. In: Farach-Colton, M. (ed.) CPM 1998. LNCS, vol. 1448, pp. 235–250. Springer, Heidelberg (1998)
9. El-Mabrouk, N., Sankoff, D.: The Reconstruction of Doubled Genomes. SIAM J. Comput. 32(3), 754–792 (2003)
10. Guyot, R., Keller, B.: Ancestral Genome Duplication in Rice. Genome 47, 610–614 (2004)
11. Hannenhalli, S., Pevzner, P.: Transforming Men into Mice (polynomial Algorithm for Genomic Distance Problem). In: Proceedings of FOCS 1995, pp. 581–592. IEEE Press, Los Alamitos (1995)
12. Kellis, M., Birren, B.W., Lander, E.S.: Proof and Evolutionary Analysis of Ancient Genome Duplication in the Yeast Saccharomyces Cerevisiae. Nature 428(6983), 617–624 (2004)
13. Ohno, S.: Ancient Linkage Group and Frozen Accidents. Nature 244, 259–262 (1973)
14. Warren, R., Sankoff, D.: Genome Halving with Double Cut and Join. In: Proceedings of APBC 2008, Series on Advances in Bioinformatics and Computational Biology, vol. 6 (2008)
15. Yancopoulos, S., Attie, O., Friedberg, R.: Efficient Sorting of Genomic Permutations by Translocation, Inversion and Block Interchange. Bioinformatics 21(16), 3340–3346 (2005)