Mathematical Analysis of Evolution by Duplicated Regions in Saccharomyces cerevisiae

AZIZI SOMAYYEH
Dept. of applied physics
Graduate School of Engineering, Osaka University
Osaka, Japan
somayyeh_azizi@ap.eng.osaka-u.ac.jp

KABOLI SAEED
Dept. of biotechnology
Graduate School of Engineering, Osaka University
Osaka, Japan
kaboli.saeed@bio.eng.osaka-u.ac.jp

YAGI ATSUSHI
Dept. of applied physics
Graduate School of Engineering, Osaka University
Osaka, Japan
yagi@ap.eng.osaka-u.ac.jp

Abstract— Since many genomes such as Saccharomyces cerevisiae genome have been shown to be resulted from an ancestral doubling, many approaches have been developed for reconstruction of its ancestral genome. In this paper, we consider a genome halving issue, aiming at evolutionary distance in S. cerevisiae. We adapt the genome halving algorithm of El-Mabrouk and Sankoff to find the evolutionary distance in S. cerevisiae by using 55 pairs of ancestral duplicated regions existing in current-genome. We describe that halving distance is accessible by using natural graph drawn for those 55 evolutionary paralog regions originated from WGD. From a bioinformatics prospective, using the duplicated blocks of current genome, we infer that present genomic organization of S. cerevisiae has the potential to be valuable detailed information about its ancestry genome and provides novel insights into genome evolution of this reference organism.

Keywords— Whole genome duplication; evolutionary distance; Saccharomyces cerevisiae.

I. INTRODUCTION

Evolutionary-induced genomic distance is studied to be explained at various levels in eukaryotes [1-4]. In Saccharomyces cerevisiae genome, evolution initiated by whole genome duplication (WGD) which doubles the gene content of a genome and result in a duplicated genome that contains two identical copies of each chromosome [5]. After this event to date, set of rearrangements have shuffled the genes within a chromosome or among chromosomes, causing evolutionary distance between current genome and its ancestor. Zheng et al. [6] provided an algorithm which is able to propose the organization of a set of genes along the history of yeast genomes. Afterward, many methods have been developed to explain the structural differences among evolutionary different yeast genomes including S. cerevisiae. For instance, Gordon et al. [4] mapped structure of an ancestor genome in S. cerevisiae before it underwent whole genome duplication. Their study presented rearrangements from ancestor to current genome and finally argued regarding evolutionary retained gene clusters. In another study, Byrne et al. [7] used Yeast Gene Order Browser (YGOB) for computational analysis of synteny relationship between yeasts. Among these approaches, in this paper we use genome halving distance which was first discussed by El-Mabrouk and Sankoff [8]. Genome halving problem is to reconstruct the ancestral genome on the basis of a decomposition of the present-day genome into a set of apparently duplicated blocks of genes or DNA sequence dispersed among the chromosomes. In this paper, we are interested in using updated map of duplicated regions in the yeast genome studied in [9] implying that perfect duplicated genome of S. cerevisiae contains 55 different pairs of syntenic chromosomal regions originated from WGD, which structurally is more understandable and more accessible in Saccharomyces Genome Database (SGD) than other evolutionary genes or regions which have been used by other studies. Therefore, we first found the genomic location of each of 55 pairs by using SGD database and http://acer.gen.tcd.ie/~khwolfe/yeast [9]. Thus, based on the genome halving theorem, perfect duplicated genome of S. cerevisiae consist of regions R1,..., R55 where every region R is duplicated to R + R. In the next section, we make a partial graph and subsequently natural graph for 55 duplicated pairs to calculate the distance.

II. MATERIAL AND METHODS

Genes and Genomes

A gene “a” represents an oriented sequence of DNA whose two extremities are its tail a (a) and its head a (a). The adjacency of two consecutive genes a, and b is denoted by an unordered set, either {a, b}, {a, b}, {a, b}, depending on the order and orientation of a, and b. An extremity that is not adjacent to any other genes is called a telomere and is represented by a singleton set (a) or (a).

A genome consists of linear or circular chromosomes that are composed of genomic markers. Markers are represented by signed integers such that the sign indicates...
the orientations of markers in chromosomes. By convention, -x = x. A linear chromosome is represented by an ordered sequence of signed integers surrounded by the unsigned marker o at each end indicating the telomeres. A circular chromosome is represented by a circularly ordered sequence of signed integers. For example, (1 2 -3) (o 4 -5 o) is a genome composed of one circular and one linear chromosome. Thus, a duplicated genome has two identical copies of each gene that are called paralogs. For a gene a, we denote its copies by \( a_1 \) and \( a_2 \) and the paralogs extremities \( a_1^{\text{t}} \) and \( a_2^{\text{t}} \).

**Example 1:** Consider the following genome defined on the set of genes \( \{a, b, c, d\} \): \( \{(d_1^a),(d_2^a,d_2^b),(d_1^c,c_1^b),(c_1^b,b_2^b),(b_1^b,d_2^a),(c_1^c,c_1^d)\} \). 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 paralog extremity “Fig. 1”.

**Duplicated genes and genomes**

A duplicated gene is a couple of homologous oriented sequences of DNA nucleotides, identified by two tails \( a_1^{\text{t}} \) and \( a_2^{\text{t}} \) and two heads \( a_1^{\text{h}} \) and \( a_2^{\text{h}} \). An all-duplicates genome is a set of adjacencies on a set of duplicated genes, where each gene extremity is contained in at most one adjacency.

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

For a chromosome \( C \), we define \( \overline{C} = \{|x| \text{ x is an adjacency or telomere of } C\} \). This notation is useful to describe the different notions of a duplicated genome that can be found in the literature, for linear genomes in [8] and for circular genomes in [10]. By bringing this together for genomes with a mixture of linear and circular chromosomes, we have:

**Definition** A duplicated genome \( A \) consisting of chromosomes \( C_1, \ldots, C_K \) is:

- Linearly perfectly duplicated, if for each linear chromosome \( C_i \) we have \( C_i = \overline{C_j} \) for some \( j \in \{1, \ldots, K\} \) \{i\};
- Circularly perfectly duplicated, if for each circular chromosome \( C_i \) either we have \( C_i = \overline{C_j} \) for some \( j \in \{1, \ldots, K\} \) \{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 is linear and circular chromosomes.

Note that this definition does not depend on the assignment of the paralogs. Two examples of perfectly duplicated genomes are given in “Fig. 2”, from the 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 this 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.

**Problem definition**

The Genome Halving Problem, Given a rearranged duplicated genome \( A \), find a perfectly duplicated genome \( B \) such that distance between \( A \) and \( B \) is minimal. The genome halving problem aims at constructing possible pre-duplication configuration of genomes which have undergone a whole duplication in the course of their histories. It was solved in the most complicated case where only linear chromosomes are allowed by El-Mabrouk and Sankoff [8] resulting in a rather complicated algorithm. Alekseyev and Pevzner discuss and solved the same problem on uni-chromosomal genomes [11]. To solve this problem, we will construct other graphs in the next section. The graph is defined on the adjacencies and telomeres of \( A \), it represents the relation between paralogous extremities.

**Partial graph**

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** the partial graph \( G(V, A) \) associated with \( G \) has the edge set \( A \) of black edges linking adjacent terms (other than the obverse) in \( G \). The partial graph associated with the genome \( G \) of Example 2 is shown in “Fig. 3”. To differentiate the two occurrences of each gene \( x \), one is subscripted “1,” and its counterpart is “2.”

**Example 2** Let \( B = \{a, b, c, d, e, f, g, h\} \) be a set of \( 8 \) genes, and let \( G \) be a genome consisting of four chromosomes:

1. \( +a + b - c + b - d \);
2. \( -c - a + f \);
3. \( -e + g - f - d \);
4. \( +h + e - g + h \).

\( G \) is a rearranged duplicated genome. Each gene appears exactly twice in the set of chromosomes; e.g., gene \( b \).
appears twice in chromosome 1. Signs represent gene orientation.

1: \begin{align*}
& a_1 \quad d_1 \\
& \alpha_1 \quad b_1 \\
& \beta_1 \quad c_1 \\
& \gamma_1 \quad h_1 \\
& \delta_1 \quad \eta_1 \\
& \theta_1 \quad \iota_1 \\
& \lambda_1 \quad \nu_1 \\
& \xi_1 \quad \zeta_1 \\
& \omega_1 \end{align*}

2: \begin{align*}
& a_2 \\
& \alpha_2 \\
& \beta_2 \\
& \gamma_2 \\
& \delta_2 \\
& \theta_2 \\
& \lambda_2 \\
& \xi_2 \\
& \omega_2 \\
& d_2 \\
& \eta_2 \\
& \iota_2 \\
& \kappa_2 \\
& \lambda_2 \\
& \nu_2 \\
& \xi_2 \\
& \zeta_2 \\
& \omega_2
\end{align*}

3: \begin{align*}
& a_3 \\
& \alpha_3 \\
& \beta_3 \\
& \gamma_3 \\
& \delta_3 \\
& \theta_3 \\
& \lambda_3 \\
& \xi_3 \\
& \omega_3 \\
& e_3 \\
& f_3 \\
& g_3 \\
& h_3 \\
& i_3 \\
& j_3 \\
& k_3 \\
& \iota_3 \\
& \kappa_3 \\
& \nu_3 \\
& \xi_3 \\
& \zeta_3 \\
& \omega_3
\end{align*}

4: \begin{align*}
& a_4 \\
& \alpha_4 \\
& \beta_4 \\
& \gamma_4 \\
& \delta_4 \\
& \theta_4 \\
& \lambda_4 \\
& \xi_4 \\
& \omega_4 \\
& e_4 \\
& f_4 \\
& g_4 \\
& h_4 \\
& i_4 \\
& j_4 \\
& k_4 \\
& \iota_4 \\
& \kappa_4 \\
& \nu_4 \\
& \xi_4 \\
& \zeta_4 \\
& \omega_4
\end{align*}

Figure 3: The partial graph G (V, A) corresponding to Example 2

Natural graph
We define the set NG of natural graphs of g (V, A) as follows:

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

Observe that the total number of edges in the graph equals two times the number of genes. The natural graph of genome A from Example 2 is given in “Fig. 4”. In a natural graph, by definition, every vertex has degree one or two. Thus, the natural graph consists only of cycles and paths.

\[
\begin{align*}
& a_1 \quad d_1 \\
& \alpha_1 \quad b_1 \\
& \beta_1 \quad c_1 \\
& \gamma_1 \quad h_1 \\
& \delta_1 \quad \eta_1 \\
& \theta_1 \quad \iota_1 \\
& \lambda_1 \quad \nu_1 \\
& \xi_1 \quad \zeta_1 \\
& \omega_1 \\
& a_2 \\
& \alpha_2 \\
& \beta_2 \\
& \gamma_2 \\
& \delta_2 \\
& \theta_2 \\
& \lambda_2 \\
& \xi_2 \\
& \omega_2 \\
& e_3 \\
& f_3 \\
& g_3 \\
& h_3 \\
& i_3 \\
& j_3 \\
& k_3 \\
& \iota_3 \\
& \kappa_3 \\
& \nu_3 \\
& \xi_3 \\
& \zeta_3 \\
& \omega_3 \\
& a_4 \\
& \alpha_4 \\
& \beta_4 \\
& \gamma_4 \\
& \delta_4 \\
& \theta_4 \\
& \lambda_4 \\
& \xi_4 \\
& \omega_4 \\
& e_4 \\
& f_4 \\
& g_4 \\
& h_4 \\
& i_4 \\
& j_4 \\
& k_4 \\
& \iota_4 \\
& \kappa_4 \\
& \nu_4 \\
& \xi_4 \\
& \zeta_4 \\
& \omega_4
\end{align*}
\]

Figure 4: Natural graph of the partial graph G (V, A) in Figure 3

**Definition** 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 \( \{p, \bar{p}\} \) 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** 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., \( N = |EC| + |OP|/2 \).

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

\[
d(A, B) = n - |EC| - \frac{1}{2} |OP| + \frac{1}{2} |OC|
\]

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

**Proof:** Let J, K, L and M is 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 \( \frac{1}{2} \) 2-cycles with \( \frac{1}{2} \) DCJ operations. Thus, for |EC| even cycles with J edges in total, we need \( \frac{1}{2} \) = \( \frac{1}{2} |EC| \) DCJ operations to create 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 \( \frac{1}{2} \) = \( \frac{1}{2} |EP| \) DCJ operations to create 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. Since, for |OC| odd cycles with L edges in total, we need \( L - |OC| \) 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 \( \frac{2}{2} |OC| \) 2-cycles with \( \frac{2}{2} |OC| \) DCJ operations. Thus, in total we need \( \frac{L - |OC|}{2} + \frac{1}{2} |OC| \) = \( \frac{L}{2} \) 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 \( M - |OP| \) 2 DCJ operations to create \( \frac{M - |OP|}{2} \) 2-cycles and |OP| 1-paths. Since L and M are even, summarizing up (a) and (b) gives us in total \( \frac{L + M - |OP|}{2} \) 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 \( L - |OC| \) 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 \( \frac{L - |OC|}{2} + \frac{1}{2} |OC| + 1 = \frac{L + 1}{2} \) 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 \( M - |OP| \) 2 DCJ operations to create \( \frac{M - |OP|}{2} \) 2-cycles and |OP| 1-paths.

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

**Algorithm**
Based on [12], in this section we show how the distance computation as well as an algorithm for reconstructing an ancestral genome can be implemented to run in linear time. Based on the proof of Theorem, 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 paralogs extremity the index of the vertex that contains it. The two tables for genome A of Example 1 are given in Tables I and II. Thus, the natural graph NG(A) 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.

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

<table>
<thead>
<tr>
<th>TABLE I: TABLE SORTING FOR THE ADJACENCIES AND TELOMERS OF GENOME A.</th>
</tr>
</thead>
<tbody>
<tr>
<td>first</td>
</tr>
<tr>
<td>second</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE II: TABLE SORTING FOR EACH IN A THE LOCATION OF ITS HEAD AND ITS TAIL IN TABLE I.</th>
</tr>
</thead>
<tbody>
<tr>
<td>head</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>9</td>
</tr>
</tbody>
</table>

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 “Fig. 5”, following the idea used in the proof of theorem.

1: Construct NG(A), the natural graph of genome A
2: while there exists a k-path with k > 1 do
3: Create a 2-cycle (and a \((k-2)\)-path if \(k > 2\))
4: end while
(* all remaining paths have length 1 *)
5: while there exists a k-cycle with \(k > 2\) do
6: Create a 2-cycle and a \((k-2)\)-cycle
7: end while
(* all remaining cycles have length 1 or 2 *)
8: while there exists a 1-cycle do
9: if there exists another 1-cycle then
10: Create a 2-cycle
11: else
12: Create a 1-path
13: end if
14: end while

In each time an unmarked adjacency is detected, all adjacencies on its path or cycle are marked and transformed into 2-cycles and 1-paths. 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 \(u^x\).

Thus, the overall running time of the algorithm for reconstructing a perfectly duplicated genome is linear.

III. RESULT

Location of 55 duplicated regions in the yeast genome

A whole duplicated genome of *S. cerevisiae* included in 16 linear chromosomes rearranged over evolutionary time to construct perfect ancestral genome with cluttered gene order. An overview of location of the 55 duplicated regain on 16 the chromosomes is depicted in "Fig. 6". Post-WGD information of *S. cerevisiae* helped us to draw partial graph and natural graph of the present –day genome of budding yeast with 55 pairs of chromosomal markers in order to reconstruct its perfect duplicated genome as depicted in the next section.

In order to reconstruct perfectly duplicated genome, we maximize the number of even cycles and odd paths in the natural graph. This is done by Algorithm “Fig. 5”, following the idea used in the proof of theorem.

<table>
<thead>
<tr>
<th>Fig. 6: Locations of the 55 duplicated regions on the 16 chromosomes</th>
</tr>
</thead>
</table>

Reconstructing the perfectly ancestral yeast genome

In this section, we first solve the genome halving problem to construct the partial and natural graph of *S. cerevisiae*, this allows us to reconstruct perfectly duplicated genome of the yeast. Then we use distance formula (Theorem) to calculate the halving distance between current genome of *S. cerevisiae* and its perfectly duplicated genome.

Partial graph in the yeast genome

Given a genome of *S. cerevisiae* with 55 duplicated markers, the partial graph associated with this genome has the edge set of black edges linking adjacent terms in genome. The partial graph associated with the genome of *S. cerevisiae* is shown in "Fig. 7". To differentiate the two occurrences of each gene x, one is subscripted “1,” and its counterpart is “2.”
Natural graph in the yeast genome

Given a partial graph of *S. cerevisiae* with 55 duplicated markers and 16 chromosomes, its natural graph according definition is shown in "Fig. 8". The set of components of the natural graph of *S. cerevisiae* can be partitioned into the following 21 disjoint subsets:

<table>
<thead>
<tr>
<th>Component Set</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EP</td>
<td>Even Path</td>
</tr>
<tr>
<td>OP</td>
<td>Odd Path</td>
</tr>
<tr>
<td>OC</td>
<td>Odd Cycle</td>
</tr>
<tr>
<td>EC</td>
<td>Even Cycle</td>
</tr>
</tbody>
</table>

Then the next step we use our distance formula (Theorem) to calculate the halving distance between current genome of *S. cerevisiae* and its perfectly duplicated genome:

\[
d(A, B) = n - |EC| - |OP| - \left\lfloor \frac{n}{2} \right\rfloor
\]

Where \(|EC|\) is the number of even cycles and \(|OP|\) is the number of odd paths in the natural graph of *S. cerevisiae*.

Figure 8: The natural graph associated with the genome of *S. cerevisiae*
IV. CONCLUSION

In this paper we used more accessible data originated from WGD to calculate and understand halving distance in \textit{S. cerevisiae}. From a bioinformatics prospective, using duplicated blocks of current genome in \textit{S. cerevisiae}, we have inferred that how to generalize the natural graph followed by the partial graph for whole genome of this organism which is valuable for deep understanding of genomic evolution and its induced distance since WGD to date. Our calculation shows 49 evolutionary distances between current genome and perfectly duplicated genome indicating that at least 49 duplication events are needed to transform those genomes to each other.

REFERENCES


