BMC Bioinformatics

Research article

Analysis of circular genome rearrangement by fusions, fissions and block-interchanges

Chin Lung Lu*[1], Yen Lin Huang[2], Tsui Ching Wang[1] and Hsien-Tai Chiu*[1]

Address: [1]Department of Biological Science and Technology, National Chiao Tung University, Hsinchu 300, ROC, Taiwan and [2]Department of Computer Science, National Tsing Hua University, Hsinchu 300, ROC, Taiwan

Email: Chin Lung Lu* - cllu@mail.nctu.edu.tw; Yen Lin Huang - slippers.bi92g@nctu.edu.tw; Tsui Ching Wang - jingjing.bi92g@nctu.edu.tw; Hsien-Tai Chiu* - chiu@cc.nctu.edu.tw

* Corresponding authors

Published: 12 June 2006

BMC Bioinformatics 2006, **7**:295 doi:10.1186/1471-2105-7-295

Received: 03 December 2005
Accepted: 12 June 2006

This article is available from: http://www.biomedcentral.com/1471-2105/7/295

Abstract

Background: Analysis of genomes evolving via block-interchange events leads to a combinatorial problem of sorting by block-interchanges, which has been studied recently to evaluate the evolutionary relationship in distance between two biological species since block-interchange can be considered as a generalization of transposition. However, for genomes consisting of multiple chromosomes, their evolutionary history should also include events of chromosome fusions and fissions, where fusion merges two chromosomes into one and fission splits a chromosome into two.

Results: In this paper, we study the problem of genome rearrangement between two genomes of circular and multiple chromosomes by considering fusion, fission and block-interchange events altogether. By use of permutation groups in algebra, we propose an $O(n^2)$ time algorithm to efficiently compute and obtain a minimum series of fusions, fissions and block-interchanges required to transform one circular multi-chromosomal genome into another, where n is the number of genes shared by the two studied genomes. In addition, we have implemented this algorithm as a web server, called FFBI, and have also applied it to analyzing by gene orders the whole genomes of three human *Vibrio* pathogens, each with multiple and circular chromosomes, to infer their evolutionary relationships. Consequently, our experimental results coincide well with our previous results obtained using the chromosome-by-chromosome comparisons by landmark orders between any two *Vibrio* chromosomal sequences as well as using the traditional comparative analysis of 16S rRNA sequences.

Conclusion: FFBI is a useful tool for the bioinformatics analysis of circular and multiple genome rearrangement by fusions, fissions and block-interchanges.

Background

For the past two decades, genome rearrangements have been studied and can be modelled to learn more about the evolution of mitochondrial, chloroplast, viral, bacterial and mammalian genomes [1]. To evaluate the evolutionary distance between two related genomes in gene

order, various rearrangement events acting on genes within or among chromosomes have been proposed, such as reversals (also known as inversions) [1-10], transpositions [11,12], block-interchanges [13-15], translocations [16,17], and fusions and fissions [18,19]. Most genome rearrangement studies in computation involve the issue of solving the combinatorial problem to find an optimal series of rearrangements required to transform one genome into another.

Recently, the study on the genome rearrangement by block-interchanges has increasingly drawn great attention, since the block-interchange event is a generalization of transposition and, currently, its computational models measuring the genetic distance are more tractable than those modeled by transposition. Christie [13] first introduced the concept of block-interchange, affecting a chromosome by swapping two non-intersecting blocks containing any number of consecutive genes. Block-interchange can be considered as a generalization of transposition, since any exchanged blocks via transposition must be contiguous in a chromosome, whereas those via block-interchange need not be. As a matter of fact, the occurrence of an exchange of two non-contiguous blocks has been suggested in the previous studies related to the biological processes of bacterial replication [[20], and references therein]. Christie also proposed an $O(n^2)$ time algorithm, where n is the number of genes, to solve the so-called *block-interchange distance problem* that is to find a minimum series of block-interchanges for transforming one linear chromosome into another. Later, we [14] designed a simpler algorithm for solving the block-interchange problem on linear or circular chromosomes with time-complexity of $O(\delta n)$, where δ is the the minimum number of block-interchanges required for the transformation and can be calculated in $O(n)$ time in advance. We also demonstrated that block-interchange events play a significant role in the genetic evolution of bacterial (*Vibrio*) species. Very recently, based on this algorithm, we have further implemented a tool, called ROBIN, for analyzing the rearrangements of gene orders via block-interchanges between two linear/circular chromosomal genomes [15]. Not only gene-order data but also sequence data are allowed to be input into the ROBIN system. If the input is the sequence data, ROBIN can automatically search for the common homologous/conserved regions shared by all input sequences.

It should be noted that the above block-interchange studies were dedicated to genomes containing only one chromosome (i.e., uni-chromosomal genomes) for evaluating their evolutionary relationships. However, for biological species with different numbers of chromosomes, the evolutionary history must also consider events of chromosomal fusions and fissions. A *fusion* occurs when two chromosomes merge into one and a *fission* takes place when a chromosome splits into two. The reason is that different chromosomes may as well exchange their genetic material with each other and, moreover, this exchange can only be achieved via inter-chromosomal operations such as fusions and fissions, instead of intra-chromosomal operations like block-interchanges. Hence, it is worthwhile to study genome rearrangements considering fusions, fissions and block-interchanges altogether. In this paper, we solve such a genome rearrangement problem by designing an efficient algorithm to compute and obtain a minimum series of all the events involving fusions, fissions and block-interchanges that are required to transform one circular multi-chromosomal genome into another, when both have the same set of genes without repeats. Although most eukaryotic genomes are linear, most prokaryotic (e.g., bacterial) genomes are circular and some of them consist of multiple circular chromosomes and large plasmids[1]. For example, some important bacterial pathogens like *Brucella*, *Burkholderia*, *Leptospira* and *Vibrio* species fall into this category. Notably, our approach is based on permutation group in algebra, instead of breakpoint graph, a commonly used approach in the study of genome rearrangement.

Recently, Yancopoulos *et al*. [21] used breakpoint graph to design an algorithm to solve a genome rearrangement problem in which the considered reversals, translocations (including fusions and fissions) and block-interchanges were given different weights. Unfortunately, their algorithm cannot be applied to solving our problem in which the events we considered are unweighted, because a series of weighted events with minimum weights in total may not be a minimum series of unweighted events, provided the events are given different weights.

Results and discussion

Based on Algorithm Sorting-by-ffbi developed in this study, we have implemented a web server, called FFBI [22], in which biologists or scientists in genomics can conduct comprehensive analyses of circular genome rearrangements by fusions, fissions and block-interchanges for their scientific interests and needs. Furthermore, we used this web server to conduct the rearrangement analyses on the whole genomes of three pathogenic *Vibrio* species, including *V. vulnificus*, *V. parahaemolyticus* and *V. cholerae*, to infer their evolutionary relationships.

Each of these three *Vibrio* pathogens consists of two circular chromosomes, and all their genomic sequences have

Table 1: The sequence information of three pathogenic *Vibrio* species, each with two circular chromosomes.

Accession NO.	Species	Chromosome	Size (Mbps)
[GenBank:NC_005139]	*V. vulnificus* YJ016	1	3.4
[GenBank:NC_005140]	*V. vulnificus* YJ016	2	1.9
[GenBank:NC_004603]	*V. parahaemolyticus* RIMD 2210633	1	3.3
[GenBank:NC_004605]	*V. parahaemolyticus* RIMD 2210633	2	1.9
[GenBank:NC_002505]	*V. cholerae* El Tor N16961	1	3.0
[GenBank:NC_002506]	*V. cholerae* El Tor N16961	2	1.0

recently been reported in GenBank with protein-coding genes annotated (see Table 1 for their sequence information). As annotated in GenBank (as of April 2006), the genomes of *V. vulnificus*, *V. parahaemolyticus* and *V. cholerae* contain 5098, 4992 and 4008 genes, respectively. From these protein-coding genes, we identified a total of 2393 (one-to-one) orthologous genes that are physically located in different positions on the chromosomes (see the Method section for construction of orthologous genes). Inevitably, there can be a high possibility that some genes with mis-annotated or uncertain protein functions are included in the genome annotation data. We therefore used only those authentic genes whose protein functions are not annotated as hypothetical or putative proteins, or are conserved and not poorly characterized (e.g., not those genes with only general function prediction or unknown function) in the NCBI COGs [23] database of orthologous genes. As a result, there are 1274 authentic orthologous genes in total remained for the further study of genome rearrangement. The relative orders of these orthologous genes along chromosomes, as well as the annotated COGs of their coding proteins, are detailed in the web site of our server.

For each pair of these pathogenic *Vibrio* species, the variation in their gene orders has suggested that the genome rearrangement events have occurred and their genomes are closely related in evolution. To evaluate the contribution of fusions, fissions and block-interchanges to these observed rearrangements, we used the server developed in this study to compute the rearrangement distance between the gene orders of any two *Vibrio* genomes. Consequently, as shown in Table 2, the calculated rearrangement distance between *V. vulnificus* and *V. parahaemolyticus* is smaller than that between *V. vulnificus* and *V. cholerae* and that between *V. parahaemolyticus* and *V. cholerae*, suggesting that *V. vulnificus* is closer to *V. parahaemolyticus* than to *V. cholerae* in evolutionary relationship. Intriguingly, this result of genome-wide experiment well coincides with those we obtained in the previous chromosome-wide experiments [14,15]. Recall that our previous experiments were conducted in a chromosome-by-chromosome style of computing the block-interchange distance between the landmark orders of any two large/small *Vibrio* chromosomes, where the used landmarks are the maximal unique matches (MUMs) or the locally collinear blocks (LCBs) that are commonly shared by three large/small *Vibrio* chromosomes.

In fact, the evolutionary relationships of the three pathogenic *Vibrio* species revealed in our experiment of analyzing their genome rearrangements also confirms that obtained by the biological community on the basis of the traditional comparative analysis of 16S rRNA gene sequences [24-26]. For confirmation, we here repeated this comparative analysis as follows. The 16S rRNA gene sequences of three Vibrios were first aligned using the Clustal W program [27], from which the distance matrix (as shown in Table 3) was then estimated by the algorithm of Kimura's two-parameter model in PHYLIP package [28].

Conclusion

In this paper, we studied the genome rearrangement problem between circular genomes with multiple chromosomes by simultaneously considering fusion, fission and block-interchange events. We have shown in the Method section that an optimal series of events required to transform one genome into another can be obtained in a canonical order such that all fusions come before all

Table 2: The calculated rearrangement distances among *V. vulnificus*, *V. parahaemolyticus* and *V. cholerae* by fusions, fissions and block-interchanges.

Species Compared	*V. vulnificus*	*V. parahaemolyticus*	*V. cholerae*
V. vulnificus	0	174	364
V. parahaemolyticus	174	0	391
V. cholerae	364	391	0

Table 3: The calculated distances among *V. vulnificus* [GenBank:X76333], *V. parahaemolyticus* [GenBank:X56580] and *V. cholerae* [GenBank:X76337] by the traditional comparative analysis of their 16S rRNA gene sequences (accession numbers of 16S rRNAs are given in square brackets).

Species Compared	*V. vulnificus*	*V. parahaemolyticus*	*V. cholerae*
V. vulnificus	0.000000	0.034524	0.050261
V. parahaemolyticus	0.034524	0.000000	0.076739
V. cholerae	0.050261	0.076739	0.000000

block-interchanges, which come before all fissions. Based on this property as well as the concept of permutation groups in algebra, we have successfully designed an $O(n^2)$ time algorithm to obtain the minimum number of fusion, fission and block-interchange events for the transformation and also to generate an optimal scenario of the required rearrangement events. In addition, we have practically implemented this algorithm as a web server and applied it to analyzing by gene orders the whole genomes of three human *Vibrio* pathogens to infer their evolutionary relationships. As a consequence, our experimental results well coincide with the previous results obtained using the chromosome-by-chromosome comparisons by landmark orders between any two *Vibrio* chromosomal sequences as well as using the traditional comparative analysis of 16S rRNA sequences. The algorithm, however, should not be applied on linear multi-chromosomal genomes, because as mentioned in the Method section, it is not always possible to have an optimal scenario in a canonical order for linear genomes. Further studies in genome rearrangement can still be pursued to solve the problem for linear multi-chromosomal genomes.

Methods

Permutations versus genome rearrangements

In group theory, a *permutation* is defined to be a one-to-one mapping from a set $E = \{1, 2, ..., n\}$ into itself, where n is some positive integer. For example, we may define a permutation α of the set $\{1, 2, 3, 4, 5, 6, 7\}$ by specifying $\alpha(1) = 4$, $\alpha(2) = 3$, $\alpha(3) = 1$, $\alpha(4) = 2$, $\alpha(5) = 7$, $\alpha(7) = 6$ and $\alpha(6) = 5$. The above mapping can be expressed using a *cycle notation* as illustrated in Figure 1 and simply denoted by $\alpha = (1, 4, 2, 3)\ (5, 7, 6)$. A cycle of length k, say $(a_1, a_2, ..., a_k)$, is simply called *k-cycle* and can be rewritten as $(a_i, a_{i+1}, ..., a_k, a_1, ..., a_{i-1})$, where $2 \le i < k$, or $(a_k, a_1, a_2, ..., a_{k-1})$. Any two cycles are said to be *disjoint* if they have no element in common. In fact, any permutation, say α, can be written in a unique way as the product of disjoint cycles, which is called the *cycle decomposition* of α, if we ignore the order of the cycles in the product [29]. Usually, a cycle of length one in α is not explicitly written and its element, say x, is said to be *fixed* by α since $\alpha(x) = x$. Especially, the permutation whose elements are all fixed is called an *identity permutation* and is denoted by **1** (i.e., $\mathbf{1} = (1)\ (2)\ ...\ (n)$).

Given two permutations α and β of E, the *composition* (or *product*) of α and β, denoted by $\alpha\beta$, is defined to be a permutation of E with $\alpha\beta(x) = \alpha(\beta(x))$ for all $x \in E$. For instance, if we let $E = \{1, 2, 3, 4, 5, 6\}$, $\alpha = (2, 3)$ and $\beta = (2, 1, 5, 3, 6, 4)$, then $\alpha\beta = (2, 1, 5)\ (3, 6, 4)$. If α and β are disjoint cycles, then $\alpha\beta = \beta\alpha$. The *inverse* of α is defined to be a permutation, denoted by α^{-1}, such that $\alpha\alpha^{-1} = \alpha^{-1}\alpha = \mathbf{1}$. If a permutation is expressed by the product of disjoint cycles, then its inverse can be obtained by just reversing the order of the elements in each cycle. For example, if $\alpha = (2, 1, 5)\ (3, 6, 4)$, then $\alpha^{-1} = (5, 1, 2)\ (4, 6, 3)$. Clearly, $\alpha^{-1} = \alpha$ if α is a 2-cycle.

Meidanis and Dias [19,30] first noted that each cycle of a permutation may represent a circular chromosome of a genome with each element of the cycle corresponding to a gene, and the order of the cycle corresponding to the gene order of the chromosome. Figure 1, for example, shows a genome with two circular chromosomes, one represented by (1, 4, 2, 3) and the other by (5, 7, 6). Moreover, they observed that global evolutionary events, such as fusions and fissions (respectively, transpositions), correspond to the composition of a 2-cycle (respectively, 3-cycles) and the permutation representing a genome. For instance, let α be any permutation whose cycle decomposition is $\alpha_1\alpha_2 ... \alpha_r$. If $\rho = (x, y)$ is a 2-cycle and x and y are in the different cycles of α, say $\alpha_p = (a_1 \equiv x, a_2, ..., a_i)$ and

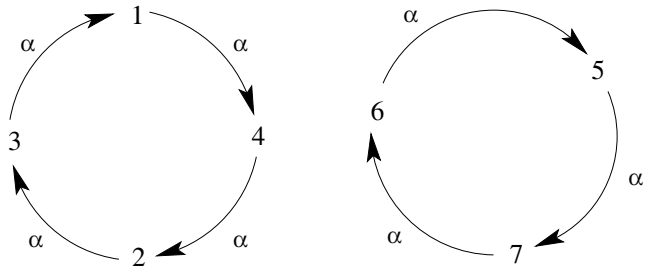

Figure 1
The illustration of a permutation $\alpha = (1, 4, 2, 3)\ (5, 7, 6)$ meaning that $\alpha(1) = 4$, $\alpha(2) = 3$, $\alpha(3) = 1$, $\alpha(4) = 2$, $\alpha(5) = 7$, $\alpha(7) = 6$ and $\alpha(6) = 5$.

$\alpha_q = (b_1 \equiv \gamma, b_2, ..., b_j)$ where $1 \le p, q \le r$, then in the composition $\rho\alpha$, α_p and α_q are joined into a cycle $(a_1, a_2, ..., a_i, b_1, b_2, ..., b_j)$, i.e., ρ is a fusion event affecting on α (and also called a *join operation* of α here). If $\rho = (x, \gamma)$ is a 2-cycle and x and γ are in the same cycle of α, say $\alpha_p = (a_1 \equiv x, a_2, ..., a_i \equiv \gamma, a_{i+1}, ..., a_j)$ where $1 \le p \le r$, then in the composition $\rho\alpha$, this cycle α_p is broken into two disjoint cycles $(a_1, a_2, ..., a_{i-1})$ and $(a_i, a_{i+1}, ..., a_j)$, i.e., ρ is a fission event affecting on α (and also called a *split operation* of α here). If $\rho = (x, \gamma, z)$ is a 3-cycle and x, γ and z are in the same cycle of α, say $\alpha_p = (a_1 \equiv x, a_2, ..., a_i, b_1 \equiv \gamma, b_2, ..., b_j, c_1 \equiv z, c_2, ..., c_k)$ where $1 \le p \le r$, then in the composition $\rho\alpha$, the cycle α_p becomes $(a_1, a_2, ..., a_i, c_1, c_2, ..., c_k, b_1, b_2, ..., b_j)$, i.e., ρ is a transposition event affecting on α.

Recently, Lin *et al.* [14] further observed that a block-interchange event affecting on α corresponds to the composition of two 2-cycles, say ρ_1 and ρ_2, and α under the condition that ρ_1 is a split operation of α and ρ_2 is a join operation of $\rho_1\alpha$. More clearly, let $\alpha_p = (a_1, a_2,..., a_k)$ be a cycle of α, $\rho_1 = (a_1, a_i)$ and $\rho_2 = (a_h, a_j)$, where $1 < i \le k, 1 \le h \le i - 1$ and $i \le j \le k$. Then $\rho_2\rho_1\alpha$ is the resulting permutation by exchanging the blocks $[a_h, a_{i-1}]$ and $[a_j, a_k]$ of α_p. That is, $\rho_2\rho_1$ is a block-interchange event affecting on α by swapping $[a_h, a_{i-1}]$ and $[a_j, a_k]$, two non-intersecting blocks in α.

As discussed above, any series of fusions, fissions and block-interchanges required to transform one circular multi-chromosomal genome α into another I can be expressed by a product of 2-cycles, say $\rho_k\rho_{k-1}...\rho_1$, such that $\rho_k\rho_{k-1}...\rho_1\alpha = I$ (hence, $\rho_k\rho_{k-1}...\rho_1 = I\alpha^{-1}$). This property implies that $I\alpha^{-1}$ contains all information that can be utilized to derive $\rho_1, \rho_2,...,\rho_k$ for transforming α into I.

It is well known that every permutation can be written as a product of 2-cycles. For example, (1, 2, 3, 4) = (1, 4) (1, 3) (1, 2). However, there are many ways of expressing a permutation α as a product of 2-cycles [29]. Given a permutation α, let $f(\alpha)$ denote the number of the disjoint cycles in the cycle decomposition of α. Notice that $f(\alpha)$ counts also the non-expressed cycles of length one. For example, if α = (1, 5) (2, 4) is a permutation of $E = \{1, 2,..., 5\}$, then $f(\alpha) = 3$, instead of $f(\alpha) = 2$, since α = (1, 5) (2, 4) (3). Then the following lemma shows the lower bound of the number of 2-cycles in any product of 2-cycles of a permutation.

Lemma 1 *Let α be an arbitrary permutation of $E = \{1, 2,..., n\}$. If α can be expressed as a product of m 2-cycles, say $\alpha = \alpha_1\alpha_2...\alpha_m$ with each α_i being 2-cycle, then $m \ge n - f(\alpha)$.*

Proof. We prove this lemma by induction on m. The lemma is true if $m = 0$, since $\alpha = \mathbf{1}$ then, meaning that $f(\alpha) = n$, and hence $m = n - f(\alpha) = 0$. Suppose now that the lemma holds for any permutation that can expressed as a product of less than m 2-cycles, where $m > 0$. Let $\alpha' = \alpha_2\alpha_3, ..., \alpha_m$. Then by the induction hypothesis, we have $m - 1 \ge n - f(\alpha')$. Since $\alpha = \alpha_1\alpha'$ and α_1 is a 2-cycle, α_1 operates on α' either as a fusion by joining two cycles of α' into one cycle (i.e., $f(\alpha) = f(\alpha') - 1$) or as a fission by splitting one cycle of α' into two cycles (i.e., $f(\alpha) = f(\alpha') + 1$). Whichever α_1 operates on α', we have $f(\alpha) \ge f(\alpha') - 1$. As a result, $m = (m - 1) + 1 \ge n - f(\alpha') + 1 = n - (f(\alpha') - 1) \ge n - f(\alpha)$.

Optimal scenario in canonical order

As mentioned previously, each circular multi-chromosomal genome with n genes can be expressed by a permutation of $E = \{1, 2,..., n\}$. Given two such genomes G_1 and G_2 over the same gene set E, the *genome rearrangement distance* between G_1 and G_2, denoted by $d(G_1,G_2)$, is defined to be the minimum number of events needed to transform G_1 into G_2, where the events allowed to take place are fusions, fissions and block-interchanges. In this section, we shall show that there is an optimal series of events required to transform G_1 into G_2 such that all fusions come prior to all block-interchanges, which come before all fissions. Here, such an optimal scenario of genome rearrangements is referred as in *canonical order*.

Lemma 2 $d(G_1, G_2) = d(G_2, G_1)$.

Proof. Let $\Phi = <\sigma_1, \sigma_2,...,\sigma_\delta>$ be an optimal series of events required to transform G_1 into G_2. Clearly, $\Phi' = <\sigma_\delta, \sigma_{\delta-1},...,\sigma_1>$ is an optimal series of events for transforming G_2 into G_1 by reversing the role of every event σ_i, where $1 \le i \le \delta$, such that σ_i is a fission (respectively, fusion) in Φ' if σ_i is a fusion (respectively, fission) in Φ.

Lemma 3 *There is an optimal series of events required to transform G_1 into G_2 such that every fission occurs after every fusion and block-interchange.*

Proof. Let $\Phi = <\sigma_1, \sigma_2,...,\sigma_\delta>$ be an optimal series of events needed to transform G_1 into G_2. Of course, if every fission occurs after every fusion and block-interchange in Φ, then the proof is done. Now, suppose that not every fission occurs after every fusion or block-interchange in Φ. Then let i be the largest index in Φ such that σ_i is a fission preceding σ_{i+1} that is either a fusion or a block-interchange. We can then obtain a new optimal series $\Phi' = <\sigma_1,...,\sigma_{i-1}, \sigma'_i, \sigma'_{i+1}, \sigma_{i+2},...,\sigma_\delta>$ to transform G_1 into G_2 such that σ'_i is a fusion or a block-interchange and σ'_{i+1} is a fission, as discussed below. Suppose that σ_i splits a chromosome α into α_1 and α_2. If σ_{i+1} is a fusion, then we assume that it joins two chromosomes β_1 and β_2 into β; otherwise, if σ_{i+1}

is a block-interchange, then assume that it affects β_1 such that β_1 becomes β through a block-interchange. Clearly, if neither β_1 nor β_2 is created by σ_i, then the desired series Φ' is obtained by swapping σ_i and σ_{i+1} in Φ (i.e., $\sigma'_i = \sigma_{i+1}$ and $\sigma'_{i+1} = \sigma_i$). If both β_1 and β_2 are created by σ_i, then the net rearrangement of σ_i (a split operation) followed by σ_{i+1} (a joint operation) either has no effect on α or becomes a block-interchange affecting α. By removing σ_i and σ_{i+1} from Φ or replacing them with an extra block-interchange, we thus obtain a new optimal series of the events transforming G_1 into G_2 with strictly less than δ events, a contradiction. Hence, we assume that only one of β_1 and β_2 is created by σ_i and without loss of generality, let $\beta_1 = \alpha_1$. Now, we consider the following two cases.

Case 1: σ_{i+1} is a fusion. For simplicity of discussion, we let $\alpha = (1, 2,..., x - 1, x,..., \gamma - 1)$, $\beta_2 = (\gamma, \gamma + 1,..., z)$, $\sigma_i = (1, x)$ and $\sigma_{i+1} = (1, \gamma)$, where $1 < x < \gamma - 1$ and $\gamma < z$. Then the net rearrangement caused by σ_i and σ_{i+1} is to transform α and β into $\alpha_2 = (x,x+1,...,\gamma - 1)$ and $\beta = (1, 2,..., x - 1, \gamma, \gamma + 1,..., z)$. In fact, this rearrangement can also be done by first joining α and β_2 into $(1, 2,..., z)$ via σ_{i+1} and then splitting $(1, 2,..., z)$ into α_2 and β via (x, γ). In other words, we can obtain Φ' by letting $\sigma'_i = \sigma_{i+1}$ and $\sigma'_{i+1} = (x, \gamma)$.

Case 2: σ_{i+1} is a block-interchange. Clearly, the net rearrangement caused by σ_i and σ_{i+1} is to transform α into β and α_2, which is equivalent to the rearrangement by first applying $\sigma_{i+1}$to α and then further splitting it into β and α_2 via σ_i. Then Φ' is obtained by swapping σ_i and σ_{i+1} in Φ.

In other words, we can always obtain Φ' from Φ according to the method described above. Repeating this process on the resulting Φ', we can finally obtain an optimal series of events that are required to transform G_1 into G_2 such that all fissions come after all fusions and block-interchanges.

Lemma 4 *There is an optimal series of events required to transform G_1 into G_2 in a canonical order such that all fusions come before all block-interchanges, which come before all fissions.*

Proof. Let $\Phi = <\sigma_1, \sigma_2,..., \sigma_\delta>$ be an optimal series of events required to transform G_1 into G_2. If there are no fusions or block-interchanges, then the proof is completed. If not, according to Lemma 3, we may assume that all fusions and block-interchanges occur earlier than all fissions. Let i be the index of the last non-fission in Φ and also let G' be the resulting genome after all $\sigma_1, \sigma_2,...\sigma_i$ have affected G_1. Since Φ is optimal, it is straightforward to see that $\Phi_i = <\sigma_1, \sigma_2,...,\sigma_i>$ is an optimal series of fusions and block-interchanges needed to transform G_1 into G'. As discussed in the proof of Lemma 2, $\Phi'_i = <\sigma_i, \sigma_{i-1},...,\sigma_1>$ is an optimal series of fissions and block-interchanges for transforming G' into G_1. Moreover, by Lemma 3, we can obtain $\Phi''_i = <\sigma'_i, \sigma'_{i-1},..., \sigma'_1>$ from Φ'_i for transforming G' into G_1 such that all block- interchanges in Φ''_i occur prior to all fissions. Consequently, $<\sigma'_1, \sigma'_2,..., \sigma'_i>$ is an optimal series of fusions and block-interchanges needed to transform G_1 into G', and all its fusions occur before all its block-interchanges. Therefore, there is an optimal series of events needed to transform G_1 into G_2 such that all fusions come earlier than all block-interchanges, which come before all fissions.

It is worth mentioning that an optimal scenario in a canonical order does not necessarily exist for linear multi-chromosomal genomes. For example, suppose that G_1 and G_2 are two given linear multi-chromosomal genomes, where $G_1 = (1, 4, 5)\ (2, 3)$ and $G_2 = (1, 2, 3)\ (4, 5)$. Then the optimal scenario between them is a fission, splitting $(1, 4, 5)$ into $(1)\ (4, 5)$, followed by a fusion, joining (1) and $(2, 3)$ to $(1, 2, 3)$. However, this optimal scenario can not be transformed into another in the canonical order according to the steps as described in Lemmas 3 and 4. Actually, there is no an optimal scenario between such two linear genomes using any two rearrangement events that begin with a fusion.

Algorithm

Let α and I be two given circular multi-chromosomal genomes over the same gene set $E = \{1, 2,...,n\}$. Here, we assume that the genes in I are sorted in the order of increasing and consecutive numbers, and that gene $i + 1$ is on the right side of gene i within the same chromosome, where $1 \le i \le n - 1$. For example, $I = (1, 2)\ (3, 4, 5)\ (6, 7, 8, 9)$ if I has three circular chromosomes with two, three and four genes, respectively. In this case, the computation of $d(\alpha,I)$ and its corresponding optimal scenario can be considered as a problem of *sorting* α using the minimum set of operations, including fusions, fissions and block-interchanges.

Suppose that $\rho_\lambda\rho_{\lambda-1} ... \rho_1$ is a product of 2-cycles that corresponds to an optimal series Φ of fusions, fissions and block-interchanges for transforming α into I. Then

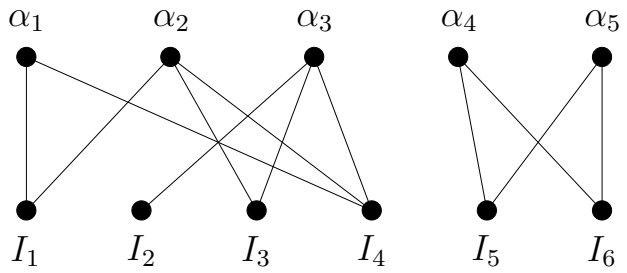

Figure 2

The induced bipartite graph $\mathcal{G}(\alpha, I)$ with two connected components.

$\Phi = \rho_\lambda \rho_{\lambda-1} \ldots \rho_1 = I\alpha^{-1}$ and by Lemma 1, $\lambda \geq n - f(I\alpha^{-1})$. If $\Phi' = \rho'_\lambda \rho'_{\lambda-1} \ldots \rho'_1$ is another product of λ 2-cycles with $\Phi' = I\alpha^{-1}$, then the number of 2-cycles in Φ' that function as fusions or fissions must be greater than or equal to that in Φ; otherwise, the total number of fusions, fissions and block-interchanges in Φ' for transforming α into I must be less than that in Φ, a contradiction. The reason is that a fusion or fission requires only one 2-cycle for rearrangement, whereas a block-interchange requires two 2-cycles. In other words, the number of 2-cycles serving as the fusions and fissions is minimum in any optimal series of events.

Based on the above observation as well as Lemma 4, below we design an efficient algorithm for computing $d(\alpha, I)$ and its optimal scenario of rearrangement events in a canonical order. Let $\chi(\alpha)$ and $\chi(I)$ denote the numbers of chromosomes in α and I, respectively, and let $\alpha = \alpha_1 \alpha_2 \ldots \alpha_{\chi(\alpha)}$ and $I = I_1 I_2 \ldots I_{\chi(I)}$. Then, an undirected graph $\mathcal{G}(\alpha, I) = (\mathcal{V}_\alpha, \mathcal{V}_I, \mathcal{E})$ is constructed from α and I as follows.

- $\mathcal{V}_\alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_{\chi(\alpha)}\}$.
- $\mathcal{V}_I = \{I_1, I_2, \ldots, I_{\chi(I)}\}$.
- $\mathcal{E} = \{(\alpha_i, I_j) \mid 1 \leq i \leq \chi(\alpha), 1 \leq j \leq \chi(I)$, and α_i and I_j have at least a common gene$\}$.

For instance, suppose that $\alpha = \alpha_1 \alpha_2 \ldots \alpha_5 =$ (1, 2, 10) (11, 8, 9, 3, 6) (7, 4, 5,12) (13, 15) (14, 16) and $I = I_1 I_2 \ldots I_6 =$ (1, 2, 3) (4, 5) (6, 7, 8) (9, 10, 11, 12) (13, 14) (15, 16). Then the induced graph $\mathcal{G}(\alpha, I)$ is shown in Figure 2, which is a bipartite graph since $\mathcal{V}_\alpha$ and $\mathcal{V}_I$ are independent sets in $\mathcal{G}(\alpha, I)$ (i.e., no edge between any two vertices in $\mathcal{V}_\alpha$ or $\mathcal{V}_I$). A *connected component* of $\mathcal{G}(\alpha, I)$ is defined to be a maximal subgraph of $\mathcal{G}(\alpha, I)$ such that there exists a path between any pair of vertices in this subgraph. For example, the induced $\mathcal{G}(\alpha, I)$ as shown in Figure 2 has two connected components. Notice that if in a chromosome I_k of I there are two genes that appear in two different chromosomes α_i and α_j of α, then $(\alpha_i, I_k) \in \mathcal{E}$ and $(\alpha_j, I_k) \in \mathcal{E}$, and hence both α_i and α_j belong to the same connected component in $\mathcal{G}(\alpha, I)$.

Let $\{C_1, C_2, \ldots, C_\omega\}$ denote the collection of all connected components in $\mathcal{G}(\alpha, I)$. For each $1 \leq i \leq \omega$, let β_i and J_i denote the chromosomes in α and I, respectively, whose corresponding vertices belong to C_i in $\mathcal{G}(\alpha, I)$. Let gene(β_i) and gene(J_i) be the collections of the genes in all chromosomes of β_i and J_i, respectively. Then gene(β_i) = gene(J_i) and gene(β_i) $\cap$ gene(β_j) = $\varnothing$ for any $1 \leq j \neq i \leq \omega$. Let n_i be the number of genes in gene(β_i). Clearly, $n = n_1 + n_2 + \ldots + n_\omega$. In addition, it can be verified that $I\alpha^{-1} = (J_1 \beta_1^{-1})(J_2 \beta_2^{-1}) \ldots (J_\omega \beta_\omega^{-1})$ and $f(I\alpha^{-1}) = f(J_1 \beta_1^{-1}) + f(J_2 \beta_2^{-1}) + \ldots + f(J_\omega \beta_\omega^{-1})$.

According to the properties above, we then find a product Φ of 2-cycles so that $\Phi\alpha = I$ as follows. We first find a product Φ_i of 2-cycles that corresponds to an optimal series of the rearrangement events required to transform β_i into J_i (i.e., $\Phi_i \beta_i = J_i$ and hence $\Phi_i = J_i \beta_j^{-1}$) and then let $\Phi = \Phi_\omega \Phi_{\omega-1} \ldots \Phi_1$. Clearly, $\Phi = I\alpha^{-1}$ and hence Φ corresponds to a feasible series of events for transforming α into I. Actually, we shall show later that the number of 2-cycles in each Φ_i is $n_i - f(J_i \beta_i^{-1})$, and in Φ_i the number of 2-cycles functioning as the fusions and fissions is minimum. This causes that the number of 2-cycles in Φ equals to $\sum_{i=1}^{\omega} n_i - f(J_i \beta_i^{-1}) = n - f(I\alpha^{-1})$, in which the number of 2-cycles serving as the fusions and fissions is minimum. As a result, Φ is an optimal series of events for transforming α into I. The above description indicates that the original problem can be conquered by independently solving the same problem on the smaller instance whose induced bipartite graph is a connected component of $\mathcal{G}(\alpha, I)$.

To simplify our discussion, throughout the rest of this section we assume that the induced $\mathcal{G}(\alpha, I)$ of a given instance α and I has exactly one connected component. Let $\Phi = <\sigma_1, \sigma_2, ..., \sigma_\delta>$ be an optimal series of events for transforming α into I in which all fusions precede all block-interchanges that further precede all fissions. Let n_{fu}, n_{bi} and n_{fi} denote the numbers of fusions, block-interchanges and fissions, respectively, in Φ. Then $\delta = n_{fu} + n_{bi} + n_{fi}$. In the following, we shall show that Φ can be expressed by a product of $n - f(I\alpha^{-1})$ 2-cycles in which the number of 2-cycles functioning as the fusions and fissions is minimum.

It should be noticed that the chromosomes considered here are disjoint (i.e., without gene duplication). Hence, for any two chromosomes α_i and α_j in α with $(\alpha_i I_k) \in \mathcal{E}$ and $(\alpha_j, I_k) \in \mathcal{E}$, there must exist a fusion in Φ that joins α_i and α_j to one chromosome; otherwise, I_k can not be formed from α by a fission later. Since all needed fusions come together in the beginning of Φ, $n_{fu} = \chi(\alpha) - 1$, which is the lower bound of the number of fusions required in any optimal series of events for transforming α into I. After these n_{fu} fusions, the resulting α becomes only one chromosome. Since the next n_{bi} block-interchanges are intra-chromosomal mutations, we have $n_{fi} = \chi(I) - 1$. Actually, $\chi(I) - 1$ is the minimum number of the required fissions in any optimal series of events for transforming α into I, since it is the minimum number of the fusions used in the corresponding optimal series of events to transform I into α.

Given any cycle ρ, we use $x \in \rho$ to denote that x is a number in ρ. For any two $x \in \rho$ and $y \in \rho$, they are said to be *adjacent* in ρ if $\rho(x) = y$ or $\rho(y) = x$. Next, we show a way to derive n_{fu} 2-cycles from $I\alpha^{-1}$ such that these 2-cycles function as the fusions that join all chromosomes of α to a single one, if α has multiple chromosomes, where $n_{fu} = \chi(\alpha) - 1$. For simplicity, later in the text we use "cycle in $I\alpha^{-1}$" to represent " cycle in the cycle decomposition of $I\alpha^{-1}$" in meaning, unless a possible confusion may arise.

Lemma 5 *Let α_i and α_j be any two disjoint cycles in α. Then there must exist a cycle in $I\alpha^{-1}$ that contains two numbers x and y such that $x \in \alpha_i$ and $y \in \alpha_j$.*

Proof. Since we assume that the induced $\mathcal{G}(\alpha, I)$ contains exactly and only one connected component, and α_i and α_j contain some numbers u and v, respectively, such that both u and v are in a cycle I_k of I. Notice that $u \notin \alpha_j$ and $v \notin \alpha_i$. Suppose that there is no cycle in $I\alpha^{-1}$ that contains two numbers x and y such that they are in these two different cycles of α, say $x \in \alpha_i$ and $y \in \alpha_j$. Then all numbers in any cycle of $I\alpha^{-1}$ are contained in some cycle of α. Without loss of generality, let $I_k = (u \equiv a_1, a_2, ..., a_p \equiv v, ..., a_q)$ and let $p < q$ for simplifying the discussion. For each $1 \le x \le p$, let $\alpha(a_x) = b_x$. Then we have $I\alpha^{-1}(b_x) = a_{x+1}$ (since $I\alpha^{-1}\alpha = I$), which means that both b_x and a_{x+1} are in the same cycle of $I\alpha^{-1}$ and hence they are also in the same cycle of α. If a_x is in α_i, then b_x is also in α_i, which further leads to $a_{x+1} \in \alpha_i$. Since $u = a_1$ is in α_i, all $a_2, a_3, ..., a_p$ are in α_i. As a result, both of u and v are in α_i, a contradiction. Hence, there exists a cycle in $I\alpha^{-1}$ that contains x and y such that $x \in \alpha_i$ and $y \in \alpha_j$.

The following lemma can be easily verified.

Lemma 6 $(a_1, a_2,..., a_i,..., a_j) = (a_1, a_2,..., a_i)\ (a_{i+1},..., a_j)\ (a_i, a_j)$, *where* $1 \le i < j$.

According to Lemma 5, for any two cycles α_i and α_j of α, we can find two numbers x and y in a cycle of $I\alpha^{-1}$, say β, such that $x \in \alpha_i$ and $y \in \alpha_j$. Let $\beta = (a_1, a_2,..., a_q)$, where $q \ge 2$. Then we consider the following two cases. Case 1: x and y are adjacent in β. For simplicity, let $x = a_{q-1}$ and $y = a_q$. Then by Lemma 6, $\beta = (a_1, a_2,..., a_{q-1})\ (a_q)\ (x, y)$. Case 2: x and y are not adjacent in β. Let $x = a_p$ and $y = a_q$, where $1 \le p < q - 1$. Then $\beta = (a_1, a_2,..., a_p)\ (a_{p+1},..., a_q)\ (x,y)$ according to Lemma 6. In other words, we can derive a 2-cycle (x, y) from β such that it can join α_i, and α_j to one cycle. After α_i and α_j are joined together via (x, y), the number of the cycles (including 1-cycles) in the resulting $I\alpha^{-1}$ increases by one. Repeatedly based on the procedure above, we can derive consecutive n_{fu} 2-cycles from $I\alpha^{-1}$, say $\phi_1, \phi_2,..., \phi_{n_{fu}}$, that can join $\chi(\alpha)$ cycles in α to a single one, where $n_{fu} = \chi(\alpha) - 1$. In other words, $\phi_1, \phi_2,..., \phi_{n_{fu}}$ function as $\chi(\alpha) - 1$ fusions that transform genome α with $\chi(\alpha)$ chromosomes into a genome, denoted by α', with a single chromosome. Clearly, we have $\alpha' = \phi_{n_{fu}}\ \phi_{n_{fu}-1} ... \phi_1 \alpha$, $I\alpha'^{-1} = I\alpha^{-1}\ \phi_1\phi_2 ...\ \phi_{n_{fu}}$, and $f(I\alpha'^{-}$

$^{1}) = f(I\alpha^{-1}) + n_{fu}$. Hence, we can immediately claim the following.

Claim 1 $\alpha' = \phi_{n_{fu}}\ \phi_{n_{fu}-1} \ldots \phi_1 \alpha$, $I\alpha'^{-1} = I\alpha^{-1}\ \phi_1\phi_2 \ldots\ \phi_{n_{fu}}$, and $f(I\alpha'^{-1}) = f(I\alpha^{-1}) + n_{fu}$, where $n_{fu} = \chi(\alpha) - 1$.

Without loss of generality, we now suppose that $\chi(I) > 1$. Similarly as the discussion above, we can derive consecutive n_{fi} 2-cycles from $\alpha' I^{-1}$, say ψ_1, ψ_2, ..., $\psi_{n_{fi}}$, such that they serve as the fusions to transform I with $\chi(I)$ chromosomes into a genome, denoted by I', with only one chromosome, where $n_{fi} = \chi(I) - 1$ and $\alpha' I^{-1}$ is the inverse of $I\alpha'^{-1}$ (i.e., $\alpha' I^{-1} = (I\alpha'^{-1})^{-1}$). Then we have $I' = \psi_{n_{fi}}\ \psi_{n_{fi}-1} \ldots \psi_1 I$ (hence $\psi_1\psi_2 \ldots\ \psi_{n_{fi}} I' = I$), $\alpha' I'^{-1} = \alpha' I^{-1} \psi_1\psi_2 \ldots\ \psi_{n_{fi}}$ and $f(\alpha' I'^{-1}) = f(\alpha' I^{-1}) + n_{fi}$. Conversely, we can use $\psi_{n_{fi}}$, $\psi_{n_{fi}-1} \ldots$, ψ_1 as fissions to split I' with one chromosome into I with n_{fi} chromosomes. Since $\alpha' I^{-1}$ is the inverse of $I\alpha'^{-1}$, it can be easily obtained from $I\alpha'^{-1}$ by just reversing the order of the numbers in each cycle of $I\alpha'^{-1}$ and hence $f(\alpha' I^{-1}) = f(I\alpha'^{-1})$, which leads to $f(\alpha' I'^{-1}) = f(I\alpha'^{-1}) + n_{fi}$. As a result, we have $f(I'\alpha'^{-1}) = f(\alpha' I^{-1}) = f(I\alpha'^{-1}) + n_{fi}$ since $I'\alpha'^{-1} = (\alpha' I^{-1})^{-1}$. Therefore, the following claim can be obtained.

Claim 2 $\psi_1\psi_2 \ldots\ \psi_{n_{fi}} I' = I$, $\alpha' I'^{-1} = \alpha' I^{-1} \psi_1\psi_2 \ldots\ \psi_{n_{fi}}$ *and* $f(I'\alpha'^{-1}) = f(I\alpha'^{-1}) + n_{fi}$, *where* $n_{fi} = \chi(I) - 1$.

Notice that both α' and I' now are the genomes with only one chromosome. Then based on the algorithm proposed by Lin *et al*. [14], we can find $n_{bi} = \dfrac{n - f(I'\alpha'^{-1})}{2}$ block-interchanges from $I'\alpha'^{-1}$ to transform α' into I'. Certainly, these n_{bi} block-interchanges can be further expressed by a product of $2n_{bi}$ 2-cycles, say $\tau_1^1, \tau_1^2, \tau_2^1, \tau_2^2, \ldots, \tau_{n_{bi}}^1, \tau_{n_{bi}}^2$, such that every two consecutive 2-cycles act as a block-interchange in the process of transforming α' into I', where $I'\alpha'^{-1} = \tau_{n_{bi}}^2 \tau_{n_{bi}}^1 \tau_{n_{bi}-1}^2 \tau_{n_{bi}-1}^1 \ldots \tau_1^2 \tau_1^1$. Hence, we have the following claim immediately.

Claim 3 $I' = \tau_{n_{bi}}^2\ \tau_{n_{bi}}^1 \tau_{n_{bi}-1}^2 \tau_{n_{bi}-1}^1 \ldots \tau_1^2 \tau_1^1\ \alpha'$.

Now we let $\Phi = \psi_1\psi_2 \ldots \psi_{n_{fi}} \tau_{n_{bi}}^2\ \tau_{n_{bi}}^1 \ldots \tau_1^2 \tau_1^1 \phi_{n_{fu}} \phi_{n_{fu}-1} \ldots \phi_1$. Then the result of $\Phi\alpha = I$ (hence $\Phi = I\alpha^{-1}$) can be easily verified by Claims 1, 2 and 3 as follows.

$$\begin{aligned}
\Phi\alpha &= \psi_1\psi_2 \ldots \psi_{n_{fi}} \tau_{n_{bi}}^2 \tau_{n_{bi}}^1 \ldots \tau_1^2 \tau_1^1 \phi_{n_{fu}} \phi_{n_{fu}-1} \ldots \phi_1 \alpha \\
&= \psi_1\psi_2 \ldots \psi_{n_{fi}} \tau_{n_{bi}}^2 \tau_{n_{bi}}^1 \ldots \tau_1^2 \tau_1^1 \alpha' && \text{(by Claim 1)} \\
&= \psi_1\psi_2 \ldots \psi_{n_{fi}} I' && \text{(by Claim 3)} \\
&= I && \text{(by Claim 2)}
\end{aligned}$$

In other words, Φ is a product of $(n_{fu} + (n - f(I'\alpha'^{-1})) + n_{fi})$ 2-cycles that can transform α into I. More clearly, Φ first uses $\phi_1, \phi_2, \ldots, \phi_{n_{fu}}$ (acting as n_{fu} fusions) to transform α into α', then uses $\tau_1^1, \tau_1^2, \ldots, \tau_{n_{bi}}^1, \tau_{n_{bi}}^2$ (acting as n_{bi} block-interchanges) to transform α' into I', and finally uses $\psi_{n_{fi}}$, $\psi_{n_{fi}-1}, \ldots, \psi_1$ (acting as n_{fi} fissions) to transform I' into I. By Claims 1 and 2, we can show that $(n_{fu} + (n - f(I'\alpha'^{-1})) + n_{fi} = n - f(I\alpha^{-1})$ as follows.

$$\begin{aligned}
& n_{fu} + (n - f(I'\alpha'^{-1})) + n_{fi} \\
&= n_{fu} + (n - f(I\alpha'^{-1}) + n_{fi}) + n_{fi} && \text{(by Claim 2)} \\
&= n_{fu} + (n - f(I\alpha^{-1}) + n_{fu} - n_{fi}) + n_{fi} && \text{(by Claim 1)} \\
&= n - f(I\alpha^{-1})
\end{aligned}$$

As mentioned before, $\chi(\alpha) - 1$ and $\chi(I) - 1$ are the lower bounds of the numbers of fusions and fissions, respectively, required in any optimal series of rearrangement events for transforming α into I. Hence, the number of 2-cycles in Φ that function as the fusions and fissions is minimum. Along with that $\Phi = I\alpha^{-1}$ can be expressed as a product of $n - f(I\alpha^{-1})$ 2-cycles, we thus conclude that Φ is an optimal series of the events that transform α into I with first n_{fu} fusions, then n_{bi} block-interchanges and finally n_{fi} fissions, where $n_{fu} = \chi(\alpha) - 1$, $n_{bi} = \dfrac{n - f(I'\alpha'^{-1})}{2} = \dfrac{n - f(I\alpha^{-1}) - n_{fu} - n_{fi}}{2}$, and $n_{fi} = \chi(I) - 1$.

Lemma 7 *There is an optimal series of the events needed to transform α into I in a canonical order such that all n_{fu} fusions come before all n_{bi} block-interchanges that come before all n_{fi} fissions, where $n_{fu} = \chi(\alpha) - 1$,*

$$n_{bi} = \frac{n - f(I\alpha^{-1}) - n_{fu} - n_{fi}}{2} \text{ and } n_{fi} = \chi(I) - 1.$$

Let us take α = (1, 2,10) (11, 8, 9, 3, 6) (7, 4, 5, 12) and I = (1, 2, 3) (4, 5) (6, 7, 8) (9, 10, 11, 12) for an example. It should be straightforward to see that $\mathcal{G}(\alpha,I)$ is a connected bipartite graph with $\chi(\alpha) = 3$ and $\chi(I) = 4$, and $I\alpha^{-1}$ = (1, 11, 7, 9, 6) (3, 10) (4, 8, 12) and hence $f(I\alpha^{-2}) = 5$, since two 1-cycles (i.e., (2) and (5)) are not explicitly shown. First, we are to find two 2-cycles ϕ_1 and ϕ_2 (since $n_{fu} = \chi(\alpha)$ -1 = 2) from $I\alpha^{-1}$ to transform genome α with three chromosomes into genome α' with exactly one chromosome. To this purpose, we let ϕ_1 = (3,10) and ϕ_2 = (4,8), since $I\alpha^{-1}$ = (1, 11, 7, 9, 6) (4, 12) (4, 8) (3, 10). Then by Claim 1, $\alpha' = \phi_2\phi_1\alpha$ = (4, 5, 12, 7, 8, 9, 10, 1, 2, 3, 6, 11) and $I\alpha'^{-1} = I\alpha^{-1}\phi_1\phi_2$ = (1, 11, 7, 9, 6) (4, 12). Next, we need to find three 2-cycles ψ_1, ψ_2 and ψ_3 (since $n_{fi} = \chi(I) - 1 = 3$) from $\alpha' I^{-1}$, which is equal to $(I\alpha'^{-1})^{-1}$ = (6, 9, 7, 11, 1) (12, 4) = (1, 7, 11) (1, 9) (1, 6) (12, 4), to transform I into I' with only one chromosome. By letting ψ_1 = (12, 4), ψ_2 = (1, 6) and ψ_3 = (1, 9), we have $I' = \psi_3\psi_2\psi_1$ = (1, 2, 3, 6, 7, 8, 9, 10, 11, 4, 5, 12) and $\alpha' I'^{-1} = \alpha' I^{-1}\psi_1\psi_2\psi_3$ = (1, 7, 11) according to Claim 2. Finally, we will find two 2-cycles τ_1^1 and τ_1^2 (since $n - f(I\alpha^{-1}) - n_{fu} - n_{fi}$ = 12 - 5 - 2 - 3 = 2) from $I'\alpha'^{-1}$ that act as a block-interchange to transform α' into I', where $I'\alpha'^{-1} = (\alpha' I'^{-1})^{-1}$ = (11, 7, 1) = (11, 1) (11, 7). By letting τ_1^1 = (11, 7) and τ_1^2 = (11, 1), we have $\tau_1^2\,\tau_1^1\,\alpha'$ = (11, 1) (11, 7) (4, 5, 12, 7, 8, 9, 10, 1, 2, 3, 6, 11) = (11, 4, 5, 12, 1, 2, 3, 6, 7, 8, 9, 10), which indeed equals I'. Consequently, we find an optimal series of events $\Phi = \psi_1\psi_2\psi_3\,\tau_1^2\,\tau_1^1\,\phi_2\phi_1$ that transform α into I (i.e., $\Phi\alpha = I$).

Based on the idea above, we have designed Algorithm Sorting-by-ffbi (meaning sorting by fusions, fissions and block-interchanges) to compute the genome rearrangement distance $d(\alpha, I)$ between two given circular multi-chromosomal genomes α and I, and also to generate an optimal scenario of the required rearrangement events in a canonical order. In Algorithm Sorting-by-ffbi, the purpose of Step 2.3.3 (respectively, Step 2.4.4) is to find two numbers x and y that are both in some cycle of $\gamma = J_i\beta_i^{-1}$ (respectively, $\gamma = \beta_i' J_i'^{-1}$), but in different cycles in β_i. By Lemma 5, such x and y exist. In fact, they can be found using the following simple approach. For simplicity, let $\gamma_k = (a_k^1, a_k^2, \ldots, a_k^{l_k})$ be a cycle in γ that contains two numbers x and y such that they are in different cycles in β_i. Then we only need to check whether a_k^1 and a_k^j, where $2 \le j \le l_k$, are in different cycles in β_i or not. If so, we let $x = a_k^1$ and $y = a_k^j$. The reason is as follows. Suppose that both a_k^1 and a_k^j for all $2 \le j \le l_k$ are in the same cycle in β_i. Then all of numbers $a_k^1, a_k^2, \ldots, a_k^{l_k}$ in γ_k are in the same cycle in β_i, which contradicts the above assumption that γ_k contains x and y that are in different cycles in β_i.

Algorithm sorting-by-ffbi

Input: Two circular multi-chromosomal genomes α and I.

Output: $d(\alpha,I)$ and a minimum series Φ of events required to transform α into I.

1: Find all connected components $C_1, C_2, \ldots, C_\omega$ in graph $\mathcal{G}(\alpha,I)$;

/* Denote by n_i the number of genes in C_i. */

2: **for** each C_i, $1 \le i \le \omega$, **do**

/* Denote by β_i (resp. J_i) the collection of chromosomes in α (resp. I) whose corresponding vertices are in C_i. */

2.1: Compute $J_i\beta_i^{-1}$ and let $\gamma = J_i\beta_i^{-1}$;

2.2: $n_{fu} = \chi(\beta_i) - 1$, $n_{fi} = \chi(J_i) - 1$, $n_{bi} = \frac{n_i - f(\gamma) - n_{fu} - n_{fi}}{2}$ and $\delta_i = n_{fu} + n_{bi} + n_{fi}$;

2.3: **if** $\chi(\beta_i) > 1$ **then** /* To compute $\phi_1, \phi_2, \ldots, \phi_{n_{fu}}$ */

2.3.1: **for** each cycle of β_i **do**

Create a set to contain all the numbers in this cycle;

endfor

2.3.2: /* Let $\gamma_1\gamma_2 \ldots \gamma_p$ be the cycle decomposition of the current γ and

let $\gamma_q = (a_q^1, a_q^2, \ldots, a_q^{l_q})$, where $1 \le q \le p$ and $l_q \ge 2$ */

$k = 1$ and $h = 2$;

2.3.3: **for** $j = 1$ to n_{fu} **do**

S = find-set (a_k^1);

while S = find-set(a_k^h)) **do**

if $h < l_k$ **then** $h = h + 1$; **else** $k = k + 1$, $h = 2$ and S = findset (a_k^1);

endwhile

$x = a_k^1$ and $y = a_k^h$;

$\phi_j = (x, y)$ and union(x, y);

endfor

2.3.4: $\beta_i' = \phi_{n_{fu}} \phi_{n_{fu}-1} \ldots \phi_1 \beta_i$ and $\gamma = \gamma\phi_1\phi_2 \ldots \phi_{n_{fu}}$; /* Currently, γ is $J_i \beta_i'^{-1}$ */

endif

2.4: **if** $\chi(J_i) > 1$ **then**/* To compute $\psi_1, \psi_2, \ldots, \psi_{n_{fi}}$ */

2.4.1: $\gamma = \gamma^1$/* New γ becomes $\beta_i' J_i'^{-1}$ */

2.4.2: **for** each cycle of J_i **do**

Create a set to contain all the numbers in this cycle;

endfor

2.4.3: /* Let $\gamma_1\gamma_2 \ldots \gamma_p$ be the cycle decomposition of the current γ and

let $\gamma_q = (a_q^1, a_q^2, \ldots, a_q^{l_q})$, where $1 \le q \le p$ and $l_q \ge 2$ */

$k = 1$ and $h = 2$;

2.4.4: **for** $j = 1$ to n_{fi} **do**

S = find-set(a_k^1);

while (S = find-set(a_k^h)) **do**

if $h < I_k$ **then** $h = h + 1$; **else** $k = k + 1$, $h = 2$ and S = find-set(a_k^1);

endwhile

$x = a_k^1$ and $y = a_k^h$;

$\psi_j = (x, y)$ and union(x, y);

endfor

2.4.5: $J_i'^{-1} = \psi_{n_{fi}} \psi_{n_{fi}-1} \ldots \psi_1 J_i$ and $\gamma = \gamma\psi_1\psi_2 \ldots \psi_{n_{fi}}$; /* Currently, γ is $\beta_i' J_i'^{-1}$ */

endif

2.5: /* To compute $\tau_1^1, \tau_1^2, \ldots, \tau_{n_{bi}}^1, \tau_{n_{bi}}^2$ */

2.5.1: $\gamma = \gamma^1$;/* New γ becomes $J_i'^{-1}\ \beta_i'^{-1}$ */

2.5.2: $n_{bi} = \dfrac{n_i - f(\gamma)}{2}$;

2.5.3: **for** $j = 1$ to n_{bi} **do**

Arbitrarily choose two adjacent elements x and y in γ;

/* Let $\beta_i' = (a_1, a_2 \ldots, a_{n_i})$ */

Circularly shift $(a_1, a_2, \ldots, a_{n_i})$ such that $a_1 = x$ and assume $y = a_k$;

$\tau_j^1 = (x, y)$;

for $h = 1$ to n_i **do**

index $(a_h) = h$;

end for

Find two adjacent elements u and v in $\gamma(x, y)$ such that index$(u) \le k$ - 1 and index$(v) \ge k$;

$\tau_j^2 = (u, v)$;

$\beta_i' = \tau_j^2\ \tau_j^1\ \beta_i'$ and $\gamma = \gamma\,\tau_j^1\ \tau_j^2$;

endfor

3: Let $\Phi_i = \psi_1 \ldots \psi_{n_{fi}} \tau^2_{n_{bi}}\ \tau^1_{n_{bi}} \ldots \tau_1^2 \tau_1^1 \phi_{n_{fu}} \ldots \phi_1$ for each $1 \le i \le \omega$.

4: Output $d(\alpha, I) = \sum_{i=1}^{\omega} \delta_i$ and $\Phi = \Phi_1 \Phi_2 \ldots \Phi_\omega$;

Theorem 1 *Given two circular multi-chromosomal genomes α and I over the same gene set $E = \{1,2,\ldots, n\}$, the problem of computing the genome rearrangement distance between α and I using fusions, fissions and block-interchanges can be solved and an optimal series of such events in a canonical order can be obtained in $O(n^2)$ time.*

Proof. As discussed above, Algorithm Sorting-by-ffbi transforms α into I using the minimum number of fusions, fissions and block-interchanges. Next, we follow to analyze its time-complexity. Notice that given an undirected graph with p vertices and q edges, all the connected components in this graph can be found in $O(p + q)$ time using depth-first search or breadth-first search [31]. As a result, Step 1 can be done in $O(n^2)$ time for computing the connected components in the induced bipartite graph $\mathcal{G}(\alpha, I)$, since in the worst case, the number of edges in $\mathcal{G}(\alpha,I)$ is $\chi(\alpha) \times \chi(I)$ and $\chi(\alpha) = O(n)$ and $\chi(I) = O(n)$. In Step 2, there are ω outer iterations, each computing the minimum series of events needed to transform β_i into J_i, where $1 \le i \le \omega$. Clearly, Steps 2.1 costs $O(n_i)$ time for computing $J_i \beta_i^{-1}$ and Step 2.2 takes only a constant time. The time cost of Step 2.3 is mostly contributed from that of Step 2.3.3. There are n_{fu} inner iterations in Step 2.3, each with the purpose of finding two numbers x and y that are both in the same cycle in γ and, however, in the different cycles in β_i. In the worst case, Step 2.3 needs n_i find-set operations and n_{fu} union operations to finish its overall process. Note that Step 2.3.1 can be implemented by initially creating a set for each number in gene(β_i) and then performing n_i - $\chi(\beta_i)$ union operations to generate $\chi(\beta_i)$ sets with each corresponding to a cycle in β_i, where $\chi(\beta_i) = n_{fu} + 1$. Hence, the total number of union operations is n_i - 1 in Step 2.3. In fact, these find-set and union operations can be implemented in $O(n_i)$ time using the so-called "static disjoint set union and find" algorithm proposed by Gabow and Tarjan [32]. In other words, Step 2.3 can cost only $O(n_i)$ time. By the same principle, it can be verified that the time cost of Step 2.4 is $O(n_i)$. As for Step 2.5, adopted from our previous work [14], it takes $O(n_{bi}n_i)$ time, where $n_{bi} = \frac{n_i - f(J_i' \beta_i^{-1})}{2} = O\,n_i)$. As a result, the time cost of Step 2 is $O(\zeta n)$, where ζ is the maximum n_{bi} among all iterations in Step 2 and $\zeta < n$. Clearly, Steps 3 and 4 cost constant time. Therefore, the total time-complexity of Algorithm Sorting-by-ffbi is $O(n^2)$.

Construction of orthologous genes

To analyze the rearrangement of three *Vibrio* genomes, we identified and constructed a table of orthologous genes that are putatively not involved in horizontal gene transfer (HGT) events by adopting the so-called symmetrical best hits (SymBets for short). In principle, two genes match and give SymBets if they are more similar to each other than they are to any other genes from the compared genomes [33,34]. Detection of such SymBet genes is, yet arguably, the simplest and most suitable method for identification of probable orthologs for closely related genomes [33,34]. Particularly, this prediction of orthologs holds to be applicable even when sequence similarity between the compared proteins is relatively low [33,34]. For the purpose of excluding paralogous genes derived from lineage-specific gene duplications, we here considered only one-to-one orthologous genes, which actually have been demonstrated as a major pattern in prokaryotic genome evolution [34]. Therefore, we used the following steps to identify an HGT-free table of one-to-one orthologous genes from the three complete *Vibrio* genomes. First, the GenePlot [35] program offered by NCBI was utilized to find and construct a table of SymBet genes between each pair of *Vibrio* genomes. Next, after removing all one-to-many or many-to-many SymBets, the three resulting tables of SymBet genes were joined to give a new one of one-to-one orthologous genes for all the three *Vibrio* genomes by using a rule as follows. If genes a (from genome A) and b (from genome B),b and c (from genome

C), and *c* and *a* are all one-to-one SymBet pairs, then a, *b* and *c* are considered as one-to-one orthologous genes for the genomes *A,B* and *C*. In other words, the SymBet relationships among a, *b* and *c* result in a triangle. Finally, those genes that were involved in the putative HGT events detected and available in the Horizontal Gene Transfer Database [36] were then deleted from the table of one-to-one orthologous genes.

Authors' contributions

CLL and HTC contributed equally to this work. CLL conceived of the study, participated in the design and analysis of algorithm and drafted the manuscript. YLH participated in the algorithm design and software development, and carried out the bioinformatics experiments. TCW participated in the software development. HTC participated in the design and coordination of this study as well as in drafting the manuscript.

Note

[1]The Institute for Genome Research (TIGR) offers a website of Comprehensive Microbial Resource (CMR) that provides information on all of the publicly available and complete bacterial genomes.

Acknowledgements

The authors would like to thank anonymous referees for many constructive comments in the presentation of this paper. This work was supported in part by National Science Council of Republic of China under grants NSC 94-2213-E-009-141 (C.L. Lu) and NSC 94-2113-M-009-007 (H.-T. Chiu).

References

1. Hannenhalli S, Pevzner PA: **Transforming cabbage into turnip: polynomial algorithm for sorting signed permutations by reversals.** *J Assoc Comput Mach* 1999, **46:**1-27.
2. Bader DA, Yan M, Moret BMW: **A linear-time algorithm for computing inversion distance between signed permutations with an experimental study.** *J Comput Biol* 2001, **8:**483-491.
3. Bafna V, Pevzner PA: **Genome rearrangements and sorting by reversals.** *SIAM J Comput* 1996, **25:**272-289.
4. Berman P, Hannenhalli S: **Fast sorting by reversal.** In *Proceedings of the 7th Annual Symposium on Combinatorial Pattern Matching (CPM1996), Volume 1075 of Lecture Notes in Computer Science* Edited by: Hirschberg DS, Myers E. Springer-Verlag; 1996:168-185.
5. Berman P, Hannenhalli S, Karpinski M: **1.375-approximation algorithm for sorting by reversals.** In *Proceedings of the 10th Annual European Symposium on Algorithms (ESA2002), Volume 2461 of Lecture Notes in Computer Science* Edited by: Mohring RH, Raman R. Springer-Verlag; 2002:200-210.
6. Caprara A: **Sorting by reversal is difficult.** In *Proceedings of the 1th Annual International Conference on Research in Computational Molecular Biology (RECOMB1997)* ACM Press; 1997:75-83.
7. Caprara A: **Sorting permutations by reversals and *Eulerian* cycle decompositions.** *SIAM J Discrete Math* 1999, **12:**91-110.
8. Christie DA: **A 3/2-approximation algorithm for sorting by reversals.** In *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA1998), ACM/SIAM* 1998:244-252.
9. Kaplan H, Shamir R, Tarjan RE: **Faster and simpler algorithm for sorting signed permutations by reversals.** *SIAM J Comput* 2000, **29:**880-892.
10. Kececioglu JD, Sankoff D: **Exact and approximation algorithms for the inversion distance between two permutations.** In *Proceedings of the 4th Annual Symposium on Combinatorial Pattern Matching (CPM1993), Volume 684 of Lecture Notes on Computer Science* Edited by: Apostolico A, Crochemore M, Galil Z, Manber U. Springer-Verlag; 1993:87-105.
11. Bafna V, Pevzner PA: **Sorting by transpositions.** *SIAM J Discrete Math* 1998, **11:**221-240.
12. Walter MEMT, Dias Z, Meidanis J: **Reversal and transposition distance of linear chromosomes.** In *Proceedings of String Processing and Information Retrieval (SPIRE1998)* IEEE Computer Society; 1998:96-102.
13. Christie DA: **Sorting by block-interchanges.** *Inform Process Lett* 1996, **60:**165-169.
14. Lin YC, Lu CL, Chang HY, Tang CY: **An efficient algorithm for sorting by block- interchanges and its application to the evolution of *Vibrio* species.** *J Comput Biol* 2005, **12:**102-112.
15. Lu CL, Wang TC, Lin YC, Tang CY: **ROBIN: a tool for genome rearrangement of block-interchanges.** *Bioinformatics* 2005, **21:**2780-2782.
16. Hannenhalli S: **Polynomial algorithm for computing translocation distance between genomes.** *Discrete Appl Math* 1996, **71:**137-151.
17. Kececioglu JD, Ravi R: **Of mice and men: algorithms for evolutionary distances between genomes with translocation.** In *Proceedings of the 6th ACM-SIAM Symposium on Discrete Algorithms (SODA1995)* ACM/SIAM, San Francisco; 1995:604-613.
18. Hannenhalli S, Pevzner PA: **Transforming men into mice (polynomial algorithm for genomic distance problem).** In *Proceedings of the 36th IEEE Symposium on Foundations of Computer Science (FOCS1995)* IEEE Computer Society; 1995:581-592.
19. Meidanis J, Bias Z: **Genome rearrangements distance by fusion, fission, and transposition is easy.** In *Proceedings of the 8th International Symposium on String Processing and Information Retrieval (SPIRE2001)* Edited by: Navarro G. IEEE Computer Society; 2001:250-253.
20. Tillier ER, Collins RA: **Genome rearrangement by replication-directed translocation.** *Nat Genet* 2000, **26:**195-197.
21. Yancopoulos S, Attie O, Friedberg R: **Efficient sorting of genomic permutations by translocation, inversion and block interchange.** *Bioinformatics* 2005, **21:**3340-3346.
22. **FFBI: a tool of circular genome rearrangement by fusions, fissions and block-interchanges** [http://genome.life.nctu.edu.tw/FFBI/]
23. **The COG database** [http://www.ncbi.nlm.nih.gov/COG/]
24. Dorsch M, Lane D, Stackebrandt E: **Towards a phylogeny of the genus *Vibrio* based on 16S rRNA sequences.** *Int J Syst Bacterial* 1992, **42:**58-63.
25. Kita-Tsukamoto K, Oyaizu H, Nanba K, Simidu U: **Phylogenetic relationships of marine bacteria, mainly members of the family *Vibrionaceae* ,determined on the basis of 16S rRNA sequences.** *Int J Syst Bacteriol* 1993, **43:**8-19.
26. Okada K, Iida T, Kita-Tsukamoto K, Honda T: **Vibrios commonly possess two chromosomes.** *J Bacteriol* 2005, **187:**752-757.
27. Thompson JD, Higgins DG, Gibson TJ: **CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice.** *Nucleic Acids Res* 1994, **22:**4673-4680.
28. Felsenstein J: **PHYLIP: phylogeny inference package (version 3.2).** *Cladistics* 1989, **5:**164-166.
29. Fraleigh JB: *A First Course in Abstract Algebra* 7th edition. Addison-Wesley; 2003.
30. Meidanis J, Dias Z: **An alternative algebraic formalism for genome rearrangements.** In *Comparative Genomics: Empirical and Analytical Approaches to Gene Order Dynamics, Map Alignment and Evolution of Gene Families* Edited by: Sankoff D, Nadeau JH. Kluwer Academic Publisher; 2000:213-223.
31. Cormen TH, Leiserson CE, Rivest RL, Stein C: *Introduction to Algorithms* 2nd edition. The MIT Press; 2001.
32. Gabow HN, Tarjan RE: **A linear-time algorithm for a special case of disjoint set union.** *J Comput Syst Sci* 1985, **30:**209-221.
33. Tatusov RL, Koonin EV, Lipman DJ: **A genomic perspective on protein families.** *Science* 1997, **278:**631-637.
34. Koonin EV: **Orthologs, paralogs, and evolutionary genomics.** *Annu Rev Genet* 2005, **39:**309-338.
35. **GenePlot** [http://www.ncbi.nlm.nih.gov/sutils/geneplot.cgi]
36. Garcia-Vallve S, Guzman E, Montero MA, Romeu A: **HGT-DB: a database of putative horizontally transferred genes in prokaryotic complete genomes.** *Nucleic Acids Res* 2003, **31:**187-189 [http://www.fut.es/~debb/HGT/].