Weighted Minimum-Length Rearrangement Scenarios

Pijus Simonaitis[1]
LIRMM – Université Montpellier, France
pijus.simonaitis@lirmm.fr

Annie Chateau
LIRMM – Université Montpellier, France

Krister M. Swenson
LIRMM, CNRS – Université Montpellier, France
swenson@lirmm.fr

Abstract

We present the first known model of genome rearrangement with an arbitrary real-valued weight function on the rearrangements. It is based on the dominant model for the mathematical and algorithmic study of genome rearrangement, Double Cut and Join (DCJ). Our objective function is the sum or product of the weights of the DCJs in an evolutionary scenario, and the function can be minimized or maximized. If the likelihood of observing an independent DCJ was estimated based on biological conditions, for example, then this objective function could be the likelihood of observing the independent DCJs together in a scenario. We present an $O(n^4)$-time dynamic programming algorithm solving the Minimum Cost Parsimonious Scenario (MCPS) problem for co-tailed genomes with n genes (or syntenic blocks). Combining this with our previous work on MCPS yields a polynomial-time algorithm for general genomes. The key theoretical contribution is a novel link between the parsimonious DCJ (or 2-break) scenarios and quadrangulations of a regular polygon. To demonstrate that our algorithm is fast enough to treat biological data, we run it on syntenic blocks constructed for Human paired with Chimpanzee, Gibbon, Mouse, and Chicken. We argue that the Human and Gibbon pair is a particularly interesting model for the study of weighted genome rearrangements.

2012 ACM Subject Classification Applied computing → Bioinformatics

Keywords and phrases Weighted genome rearrangement, Double cut and join (DCJ), Edge switch, Minimum-weight quadrangulation

Digital Object Identifier 10.4230/LIPIcs.WABI.2019.13

Supplement Material Code available at `bitbucket.org/pijus_simonaitis/mcps_wabi2019/`.

Funding This work is partially supported by the Labex NUMEV flagship project GEM, and by the CNRS project Osez l'Interdisciplinarité.

1 Introduction

Since its introduction in 2005, Double Cut and Join (DCJ) has become the foundational model for mathematical and algorithmic study of genome rearrangement. Its power to simulate a diverse combination of chromosomal events along with its relatively simple mathematical structure has facilitated a diverse expansion and use of DCJ. Grounded on primary work that outlines how to compute DCJ scenarios between pairs of genomes [42, 9], many active research topics have blossomed including the sampling of scenarios [30], whole

[1] Corresponding author

genome duplication [41, 4], orthology assignment and evolution under gene duplication and indels [33, 44], family-free genomic distances [29], cancer genomics [43], ancestral genome reconstruction [3, 2], and phylogenetic inference [28, 3].

Yet to further the utility of DCJ, there is a potential for the addition of biological constraints to the model. This article, when combined with our previous work [35], represents the first known algorithm – for any known model of genome rearrangement – where an arbitrary real-valued weight function on rearrangements can be given as part of the input. The objective function can be the sum or product of the weights of the DCJs in an evolutionary scenario, and the function can be minimized or maximized. Say, for example, that one was to weight a DCJ based on the likelihood of observing it given particular biological conditions. Then our objective function could compute the maximum likelihood scenario over all minimum-length scenarios.

Our dynamic programming algorithm is possible thanks to the existence of equivalence classes on weighted parsimonious scenarios.

1.1 Background

Most research on weighted genome rearrangement has only considered constraints on the gene sequences, or the types of rearrangements. Length-weighted reversals and centered reversals are such examples [7]. Other examples weight the types of rearrangements relative to each other, possible types being inversion, transposition, inverse transposition, insertion/deletion (indel), and tandem duplication random loss [13, 17, 25]. The preservation of common intervals or groups of co-localized genes can also be enforced. Hartman, Middendorf, and Bernt present a summary of this work in [24]. The same authors have combined these approaches by studying type-weighted rearrangement scenarios preserving common intervals [23].

Over the last few years research has emerged that incorporates external biological information for weighting genome rearrangements. This includes methods that consider the sizes of intergenic regions [12, 20, 14], or that minimize the number of DCJs cutting regions that are distant in physical space [39, 36]. In [32], we computed scenarios that maximize the Hi-C contacts between breakpoints of rearrangements using a greedy algorithm for weighted pairs of gene adjacencies. It was left an open question as to whether an exact polynomial-time algorithm to do this exists. This article answers a similar question, by describing an exact polynomial-time algorithm for weighted pairs of gene extremities (rather than adjacencies).

We base our article on the model called *Double Cut and Join* (DCJ) [42, 9]. Each chromosome is represented by an ordering of directed genes (or syntenic blocks), and each gene is represented by two extremities. Extremities that are adjacent in this ordering are paired, and transformations of these pairs occur by swapping extremities of two pairs. DCJ can naturally be interpreted as a graph edit model with the use of the *breakpoint graph*, where there is an edge between gene extremities a and b for each adjacent pair. A DCJ operation replaces an edge pair $\big\{\{a,b\},\{c,d\}\big\}$ of the graph by $\big\{\{a,c\},\{b,d\}\big\}$ or $\big\{\{a,d\},\{b,c\}\big\}$.

This edge edit operation on a graph is called a *2-break*. 2-breaks – also known as edge swaps, switches, rewirings, or flips [21] – have been studied in the restricted setting of genome rearrangement [5, 9] and sorting strings by mathematical transpositions [1, 18], but also in more general settings of generating random networks [21] and network design [11, 19].

1.2 The Problem

The foundational problem is that of finding a minimum length (or *parsimonious*) 2-break scenario transforming an arbitrary multi-graph into another one having the same degree sequence [11, 9, 30]. For an arbitrary real valued cost function ω defined on the set of

2-breaks, we treat the problem of finding a parsimonious 2-break scenario minimizing the sum of the ω-costs of its 2-breaks, the ω-Minimum Cost Parsimonious Scenario problem (see Section 5 for a more precise definition of our cost function).

In this article we present an $O(n^4)$-time algorithm solving ω-MCPS for the perfect matchings on n vertices. These graphs represent *co-tailed* pairs of genomes, which are the pairs of genomes that share the same telomeric adjacencies [31].

Consider the ω-MCPS problem for general genomes.

▶ **Problem 1** (ω-Minimum Cost Parsimonious Scenario or ω-MCPS).

INPUT: *A pair of genomes (A, B), and a real valued cost function ω defined for the set of DCJ rearrangements on the gene extremities of A and B.*

OUTPUT: *A minimum length DCJ scenario transforming A into B that minimizes the sum of the ω-costs of its operations.*

Our previous work guarantees that our solution for ω-MCPS on co-tailed genomes can be extended to ω-MCPS on general genomes, with an overhead that is linear in the number of linear chromosomes [35]. Note that ω-MCPS asks for a parsimonious scenario minimizing the sum of costs. Our algorithm from Section 5 can be easily modified to find a parsimonious scenario minimizing the product of costs, or maximizing the sum or the product of costs. For the sake of simplicity we only treat the case of minimizing the sum of costs.

Experiments presented in Section 6 show that our algorithm is efficient enough for species as distant as Human and Chicken. We construct breakpoint graphs between Human and four other species including Chimpanzee, Gibbon, Mouse, and Chicken, and demonstrate that our algorithm can be used in practice despite its elevated time complexity.

Designing an informative cost function remains a challenging open problem. In Section 7 we provide a short overview of the recent results linking evolution by genome rearrangement to various chromatin features. We also propose the use of *sure* 2-breaks as a testing ground for potential cost functions. These are the rearrangements that are present in every parsimonious scenario transforming one genome into another. We observe that such rearrangements comprise at least 7% of the parsimonious scenarios for the studied species.

2 Genomes, Breakpoint Graphs and DCJ Scenarios

A *genome* consists of *chromosomes* that are linear or circular orders of genes separated by potential *breakpoint* regions. In Figure 1 the tail of an arrow represents the *tail extremity*, and the head of an arrow represents the *head extremity* of a gene. A set of *adjacencies* between the gene extremities can uniquely represent a genome. An *adjacency* is either *internal*: an unordered pair of the extremities that are adjacent on a chromosome, or *external*: a single extremity adjacent to one of the two ends of a linear chromosome. We will suppose that two genomes are partitioned into n genes or syntenic blocks each occurring exactly once in each genome. We use the *breakpoint graph* to represent a pair of genomes, and our goal is to transform one genome into another using a sequence of *DCJ* operations.

▶ **Definition 1** (co-tailed genomes). *Two genomes are co-tailed if their sets of external adjacencies are equal.*

▶ **Definition 2** (breakpoint graph [5]). *$G(A, B)$ for genomes A and B is an 2-edge-colored Eulerian undirected multi-graph. Its vertices are the $2n$ gene extremities and an additional vertex $\circ$. For every internal adjacency $\{a, b\} \in A$ (respectively $\{a, b\} \in B$) there is a*

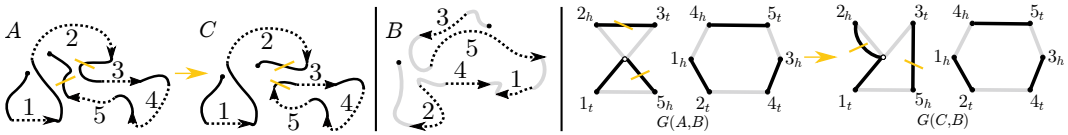

Figure 1 Three genomes A, B and C are depicted together with the breakpoint graphs $G(A, B)$ and $G(C, B)$. A and B consists of a single linear chromosome, while C consists of a linear and a circular chromosome. The three genomes contain the five genes depicted as dashed arrows. The adjacencies of A are $\{\{1_t\}, \{1_h, 2_t\}, \{2_h, 3_t\}, \{3_h, 4_t\}, \{4_h, 5_t\}\{5_h\}\}$. A DCJ $\{2_h, 3_t\}, \{5_h\} \to \{5_h, 3_t\}, \{2_h\}$ transforms A into C. The corresponding graph transformation $G(A, B) \to G(C, B)$ is a 2-break transforming one simple alternating cycle into two.

black (respectively gray) edge $\{a, b\}$ *in* $G(A, B)$ *and for every external adjacency* $\{a\} \in A$ *(respectively* $\{a\} \in B$*) there is a black (respectively gray) edge* $\{a, \circ\}$ *in* $G(A, B)$*. There is also a number of black and gray loops* $\{\circ, \circ\}$ *ensuring that the black and gray degrees of* $\circ$ *are equal.*

▶ **Definition 3** (double cut and join [42])**.** *A DCJ cuts one or two adjacencies of a genome and joins the resulting ends of the chromosomes back in one of the four following ways:* $\{a, b\}, \{c, d\} \to \{a, c\}, \{b, d\}$*;* $\{a, b\}, \{c\} \to \{a, c\}, \{b\}$*;* $\{a, b\} \to \{a\}, \{b\}$*; and* $\{a\}, \{b\} \to \{a, b\}$*.*

3 Parsimonious 2-break Scenarios for 2-edge-colored Graphs

The problem of transforming a genome A into B using a sequence of DCJ operations can be reduced to the problem of transforming the breakpoint graph $G(A, B)$ into $G(B, B)$ using 2-breaks on its black edges (see Figure 1) [35].

▶ **Definition 4** (2-break)**.** *A 2-break* $\{\{u, v\}, \{q, s\}\} \to \{\{u, q\}, \{v, s\}\}$ *is a graph transformation replacing edges* $\{u, v\}$ *and* $\{q, s\}$*, with edges* $\{u, q\}$ *and* $\{v, s\}$*. The* vertices *and the* edges *of this 2-break are respectively* $\{u, v, q, s\}$ *and* $\{\{u, v\}, \{q, s\}, \{u, q\}, \{v, s\}\}$*.*

The problem of transforming a breakpoint graph using 2-breaks can be formulated in a more general setting of transforming an arbitrary multi-graph H_1 into another H_2. A 2-break preserves the degrees of the vertices, thus H_1 can only be transformed into a graph H_2 having the same degree sequence as H_1. Given H_1 and H_2 with equal degree sequences, one could combine them into an 2-edge-colored multi-graph using a bijection between the equal degree vertices, coloring the edges of H_1 in black and the edges of H_2 in gray. Such an Eulerian 2-edge-colored graph admits a decomposition into edge-disjoint alternating cycles. An alternating cycle is *simple*, if it cannot be further decomposed into edge-disjoint alternating cycles. A simple alternating cycle is a *circle*, if the black and gray degrees of its vertices are equal to 1. The *size* of a simple cycle is its number of vertices. The breakpoint graph $G(A, B)$ in Figure 1 has two simple cycles, but only one of them is a circle.

▶ **Definition 5** (2-break scenario and terminal graph)**.** *A 2-break* scenario *for an 2-edge-colored multi-graph* G *is a sequence of 2-breaks transforming the black edges of* G *into its gray edges. We call an 2-edge-colored graph with equal multisets of black and gray edges a* terminal *graph.*

See Figure 2 (a) for an example of a 2-break scenario and a terminal graph. The problem of finding a minimum length (or *parsimonious*) 2-break scenario is closely related to the problem of finding a Maximum Alternating Edge-disjoint Cycle Decomposition

(MAECD) of a graph [9, 11, 30, 15]. The key observation is that a 2-break in a parsimonious scenario increases the size of a MAECD of a graph by 1. Theorem 6 leads to a couple of easy corollaries that we will use in our proofs.

▶ **Theorem 6** (Bienstock and Günlük in [11]). *The minimum length of a 2-break scenario for an Eulerian 2-edge-colored multi-graph is equal to the number of its vertices divided by two minus the size of its* MAECD.

▶ **Corollary 7.** *The first 2-break τ in a parsimonious scenario ρ for a circle transforms it into a union of two vertex-disjoint circles. The rest of ρ does not contain τ.*

▶ **Corollary 8.** *A parsimonious scenario for a graph having two connected components can be partitioned into two sub-sequences that are parsimonious scenarios for these components. If parsimonious scenarios ρ_1 and ρ_2 for these components are given, then their* shuffle*, a sequence that can be partitioned into two sub-sequences equal to ρ_1 and ρ_2, is a parsimonious scenario for the graph.*

4 Equivalence Classes of the Parsimonious Scenarios for a Circle

In Section 5 we concern ourselves with the cost of a scenario, that is, the sum of the costs of its constituent 2-breaks. This way two scenarios consisting of the same 2-breaks, but possibly in a different order, have the same cost. This motivates the following definition.

▶ **Definition 9** (equivalent scenarios). *2-break scenarios are* equivalent *if they consist of the same 2-breaks.*

A *Scenario graph* of a parsimonious scenario ρ for a circle is introduced in Section 4.1, as a graph whose edges are exactly the edges of the 2-breaks in ρ. Two parsimonious scenarios for a circle are then proved to be equivalent if and only if their scenario graphs are equal in Section 4.4. In Section 4.1 we show that the scenario graph of a parsimonious scenario is planar and provide an embedding of this graph that is a partition of a regular even polygon into quadrilaterals, also known as a "quadrangulation" or a "complete quadrillage". In Section 4.4 we show that for any *quadrangulation of a circle* there exists an equal scenario graph, thus establishing a bijection between the quadrangulations and the equivalence classes of the parsimonious scenarios for a circle. A bijection between the 2-breaks in a scenario and the faces of its scenario graph is established in Section 4.2. This link enables us to partition a parsimonious scenario for a circle into the subsequences that are parsimonious scenarios for its *sub-circles* in Section 4.3, and ultimately allows us to efficiently search the space of all the parsimonious scenarios in Section 5.

▶ **Definition 10** (sub-circle). *Take an odd length path in a circle. If the edges incident to its endpoints are black (respectively gray), then add a gray (respectively black) edge incident to the endpoints of this path to obtain a circle. We say that the added edge* delimits *the* sub-circle*. See Figure 2 (b) for an example of sub-circles.*

4.1 A Scenario Graph is a Quadrangulation of a Circle

We show that a scenario graph for a parsimonious scenario of a circle is planar, and provide an embedding of this graph that is a partition of that circle into quadrilaterals (i.e. cycles of length four).

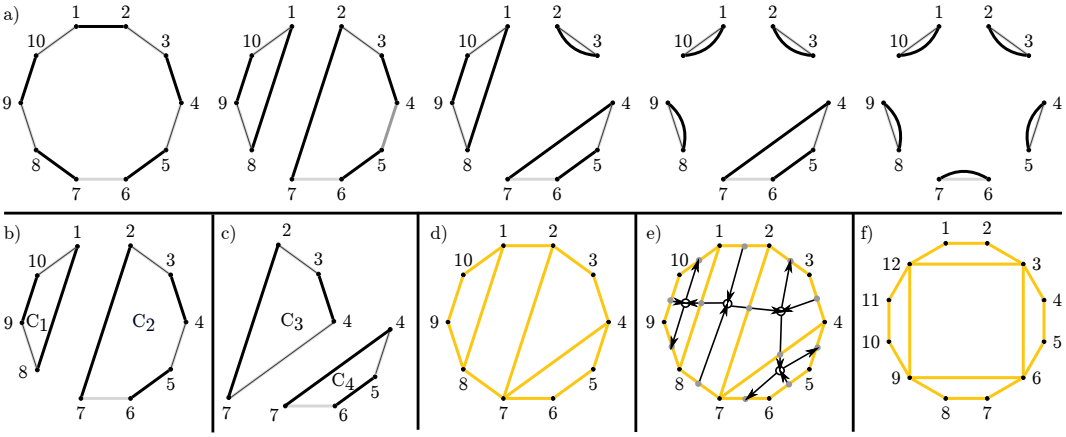

Figure 2 An example of a parsimonious scenario for a circle, its sub-circles and the scenario graph. a) A circular straight-line embedding Σ_C of a circle C and a parsimonious 2-break scenario ρ of length 4 transforming this circle into a terminal graph. b) The first 2-break of ρ transforms C into a union of two vertex-disjoint circles C_1 and C_2, these are the sub-circles of C delimited by the edges $\{1, 8\}$ and $\{2, 7\}$ respectively. c) The edge $\{4, 7\}$ delimits the sub-circles C_3 and C_4 of C_2. C_4 is also a sub-circle of C, but C_3 is not. d) A circular straight-line drawing $\Sigma_{\mathcal{S}(C,\rho)}$ of the scenario graph $\mathcal{S}(C, \rho)$. e) We draw an embedding of the trajectory graph [34] $\mathcal{T}$ of ρ on top of the scenario graph. *Operation* nodes of $\mathcal{T}$ (in white) correspond to the faces of $\mathcal{S}(C, \rho)$, and *adjacency* nodes of $\mathcal{T}$ (in gray) correspond to the edges of $\mathcal{S}(C, \rho)$. The edges incident to the black (respectively gray) edges of the circle are directed outwards (respectively inwards). The rest of the edges are directed in such a way as to guarantee that the in and out degrees of the rest of the vertices of $\mathcal{T}$ are equal. f) An example of a scenario graph with a face that is not incident to any of the edges of the circle.

▶ **Definition 11** (scenario graph). *For a circle C and its parsimonious 2-break scenario ρ, define an undirected simple graph $\mathcal{S}(C, \rho)$ called a* scenario graph. *The vertices of $\mathcal{S}(C, \rho)$ are the vertices of C. If C has more than two vertices, then the edges of $\mathcal{S}(C, \rho)$ are the edges of the 2-breaks in ρ. Otherwise, $\mathcal{S}(C, \rho)$ contains a single edge incident to its two vertices.*

See Figure 2 (d) for an example of a *circular straight-line drawing* of a scenario graph.

▶ **Definition 12** (circular straight-line drawing of a scenario graph). *A circular straight-line drawing of a graph is a drawing on a plane with the vertices of the graph arranged on a circle and the edges drawn as straight lines. If the edges in the drawing do not cross, then the drawing is an* embedding. *Fix a circular straight-line embedding Σ_C of a circle C. The sub-circles of C inherit their circular straight-line embeddings from Σ_C. The scenario graph $\mathcal{S}(C, \rho)$ for a parsimonious scenario ρ for C inherits a circular straight-line drawing $\Sigma_{\mathcal{S}(C,\rho)}$ from Σ_C.*

▶ **Theorem 13.** *For every circle C and its parsimonious scenario ρ, the scenario graph $\mathcal{S}(C, \rho)$ is planar, $\Sigma_{\mathcal{S}(C,\rho)}$ is its embedding and all the internal faces of $\Sigma_{\mathcal{S}(C,\rho)}$ are quadrilaterals.*

Proof. C has an even number of vertices by construction. If C has two vertices, then its scenario graph has a single edge and no internal faces. Suppose that the statement is true for every circle having at most $2k - 2 \geq 2$ vertices and take a circle C with $2k$ vertices together with its parsimonious scenario ρ.

Due to Corollary 7, the first 2-break τ of ρ transforms C into a union of two vertex disjoint circles. Denote these circles by C_1 and C_2 (see Figure 2 (b)). Due to Corollary 8, the rest of ρ can be partitioned into ρ_1 and ρ_2, that are scenarios for C_1 and C_2 respectively.

If $\tau = \{\{u, v\}, \{q, s\}\} \to \{\{u, q\}, \{v, s\}\}$, then the edges of $\Sigma_{\mathcal{S}(C,\rho)}$ can be obtained by adding the edges $\{u, v\}$ and $\{q, s\}$ to the union of the edges of $\mathcal{S}(C_1, \rho_1)$ and $\mathcal{S}(C_2, \rho_2)$ (see Figure 2 (d). $\Sigma_{\mathcal{S}(C_1,\rho_1)}$ and $\Sigma_{\mathcal{S}(C_2,\rho_2)}$ satisfy the inductive hypothesis, so their edges do not cross. $\{u, v\}$ and $\{q, s\}$ do not cross the other edges and together with $\{u, q\}$ and $\{v, s\}$ bound the only face of $\Sigma_{\mathcal{S}(C,\rho)}$ not belonging to $\Sigma_{\mathcal{S}(C_1,\rho_1)}$ or $\Sigma_{\mathcal{S}(C_2,\rho_2)}$. ◀

▶ **Definition 14** (Dual graph). *The* dual graph *of an embedding of a planar graph G is a graph that has a vertex for each face of G. The dual graph has an edge whenever two faces of G are separated from each other by an edge.*

The dual graph of an embedding of a scenario graph, up to some minor modifications, is a *trajectory graph* introduced by Shao, Lin, and Moret (see Figure 2 (e) for an example). In [34] these authors show that the trajectory graph of a parsimonious scenario for a circle is a tree, which could be used to prove Theorem 13. By fixing an embedding of a scenario graph, we are not only able to describe the equivalence classes of the parsimonious scenarios for a circle, but also to search them efficiently.

4.2 A Bijection Between the 2-breaks in a Scenario and the Faces of a Scenario Graph

▶ **Lemma 15** (Proven in Appendix A.1). *For an embedding of a connected graph $G = (V, E)$ with $|V|$ edges incident to the outer face and all the inner faces being quadrilaterals, the number of internal faces $|F_{int}|$ is $\frac{|V|}{2} - 1$.*

▶ **Proposition 16.** *For a circle C and a parsimonious scenario ρ for C, there exists a bijection between the 2-breaks in ρ and the internal faces of $\Sigma_{\mathcal{S}(C,\rho)}$ that associates to every 2-break in ρ the face bounded by its edges.*

Proof. Take a circle $C = (V, E)$ and its parsimonious scenario ρ. There are $\frac{|V|}{2} - 1$ 2-breaks in ρ due to Theorem 6 and $\frac{|V|}{2} - 1$ internal faces in $\Sigma_{\mathcal{S}(C,\rho)}$ due to Lemma 15. Take a 2-break τ in ρ. By construction, the edges of τ form a cycle of length 4 in $\mathcal{S}(C, \rho)$. $\Sigma_{\mathcal{S}(C,\rho)}$ is an embedding of $\mathcal{S}(C, \rho)$ and in the part of the plane bounded by this cycle there are no vertices. This means that the edges of τ bound a face in $\Sigma_{\mathcal{S}(C,\rho)}$. All the 2-breaks in ρ are unique due to Corollary 7, thus we obtain an injection between the 2-breaks in ρ and the faces in $\Sigma_{\mathcal{S}(C,\rho)}$. As the sizes of these two sets are equal, the function is bijection. ◀

▶ **Corollary 17** (Proven in Appendix A.2). *Consider a circle C, a parsimonious scenario ρ for C, and an internal face f of $\Sigma_{\mathcal{S}(C,\rho)}$ bounded by the edges $\{u, v\}, \{v, s\}, \{s, q\}$ and $\{q, u\}$, where $\{u, v\}$ is a black edge of C. Then ρ contains a 2-break $\{\{u, v\}, \{q, s\}\} \to \{\{u, q\}, \{v, s\}\}$.*

4.3 Partitioning Scenario into Scenarios for Sub-circles

In this section we demonstrate how a parsimonious scenario for a circle can be partitioned into scenarios for its sub-circles. Theorem 19 is at the heart of the dynamic programming algorithm for MCPS presented in Section 5.

▶ **Lemma 18** (Proven in Appendix A.3). *For every circle C, a parsimonious scenario ρ for C, and an edge e of $\Sigma_{\mathcal{S}(C,\rho)}$, ρ can be partitioned into two (possibly empty) parsimonious scenarios, one for each sub-circle delimited by e.*

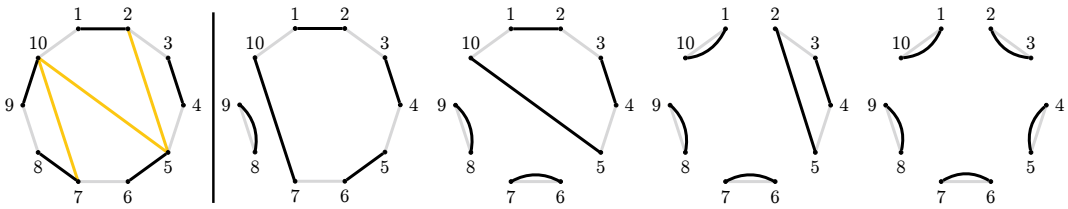

Figure 3 An illustration of how to find a parsimonious scenario whose scenario graph is equal to a given qadrangulation of a circle. On the left, the inner edges of a quadrangulation are depicted on top of a circle. To the right, a parsimonious scenario realizing this quadrangulation is given.

▶ **Theorem 19.** *For every circle C having at least four vertices, a parsimonious scenario ρ for C, and an internal face f of $\Sigma_{\mathcal{S}(C,\rho)}$, ρ can be partitioned into four (possibly empty) parsimonious scenarios, one for each sub-circle delimited by the edges bounding f, plus the 2-break whose edges bound f.*

Proof. Proposition 16 provides a 2-break τ in ρ whose edges bound f. An edge bounding f delimits two sub-circles of C, denote the one containing only two vertices of τ to be *external* (the other one contains all four vertices of τ). Repeat Lemma 18 for the four edges bounding f to obtain parsimonious scenarios for the four external sub-circles. See Figure 2 (f) for an example, where all four external sub-circles are of size larger than 2. By construction, these scenarios do not intersect and together with τ partition ρ. ◀

4.4 A Bijection Between the Equivalence Classes of Scenarios and Quadrangulations

The results in this subsection are presented for the sake of completeness. They establish bijections between the equivalence classes of the parsimonious scenarios for a circle, their scenario graphs and the quadrangulations of that circle, however they are not explicitly used in the later sections.

▶ **Theorem 20** (Proven in Appendix A.4)**.** *For every circle C and its parsimonious scenarios ρ and μ, their scenario graphs are equal if and only if these scenarios are equivalent.*

▶ **Definition 21** (quadrangulation of a circle)**.** *Take a circular straight-line embedding of a circle C and forget the colors of its edges. If C is has two vertices then keep a single edge joining them to obtain an embedding Σ_S. Otherwise add straight-line edges to obtain an embedding Σ_S with all the internal faces being quadrilaterals.*

▶ **Proposition 22** (Proven in Appendix A.5)**.** *For every circle C and its quadrangulation Σ_S there exists a parsimonious scenario ρ such that $\Sigma_{\mathcal{S}(C,\rho)} = \Sigma_S$.*

Quadrangulations of an even regular polygon, also known as complete quadrillages, correspond to rooted ternary trees just as triangulations of a polygon correspond to rooted binary trees [6]. This means that for a circle with $2n+2$ vertices there are $\frac{1}{2n+1}\binom{3n}{n} \approx \frac{1}{n^{3/2}}(\frac{27}{4})^n$ equivalence classes. For comparison, the number of the parsimonious scenarios for such a circle is $(n+1)^{(n-1)}$ [31].

5 Minimum Cost Parsimonious Scenario for a Circle

Consider a circle C, the 2-breaks that act on the vertices of C, and a cost function ω mapping these 2-breaks to real numbers. In other words, every 2-break $\tau = \{\{u,v\},\{q,s\}\} \to \{\{u,q\},\{v,s\}\}$, with u,v,q,s being vertices of C, gets assigned a ω-cost $\omega(\tau)$. The ω-cost

$\omega(\rho)$ of a scenario ρ is the sum of the costs of its constituent 2-breaks. The ω-cost $\omega(C)$ of a circle C is the minimum cost of a parsimonious scenario for C. Without loss of generality we suppose the vertices of C are named $\{1, \ldots, |V|\}$ while respecting their clockwise order on the circle, and that $\{1, 2\}$ is a black edge as in Figure 2. For the vertices $i, j \in V$ with $i + j$ being odd, define $C[i, j]$ to be the sub-circle of C containing the path in C going clockwise from i to j. We denote its ω-cost by $\omega[i, j]$.

A quadrilateral is *valid for a circle* if it appears in some quadrangulation of that circle. Take vertices $1 \leq i < r < s < j \leq |V|$. They are the vertices of a valid quadrilateral for C if and only if $i + r$, $r + s$, $s + j$, $i + j$ are all odd (see Figure 2). Due to Theorem 19, such a quadrilateral partitions a parsimonious scenario for $C[i, j]$ into parsimonious scenarios for $C[i, r]$, $C[r, s]$, $C[s, j]$, along with the 2-break acting on the edges bounding the quadrilateral. In Lemma 23 we show how $\omega[i, j]$ can be found by iterating through the valid quadrilaterals for $C[i, j]$ containing the edge $\{i, j\}$. In Theorem 24 we conclude that $\omega(C)$ can be computed by iterating once through all the valid quadrilaterals of C.

▶ **Lemma 23** (Proven in Appendix A.6). *For a sub-circle $C[i, j]$ of a circle C with $i < j$, its cost $\omega[i, j]$ is 0 if it has two vertices, otherwise*

$$\omega[i, j] = min_{r,s}\big(\omega[i, r] + \omega[r, s] + \omega[s, j] + \omega(\tau)\big)$$

for vertices $i < r < s < j$ of a valid quadrilateral for $C[i, j]$, where $\tau = \big\{\{i, r\}, \{s, j\}\big\} \to \big\{\{i, j\}, \{r, s\}\big\}$ if i is odd, and $\tau = \big\{\{i, j\}, \{r, s\}\big\} \to \big\{\{i, r\}, \{j, s\}\big\}$ if i is even.

▶ **Theorem 24.** *If the ω-cost of a 2-break can be computed in constant time, then the ω-cost of a circle can be computed in $O(|V|^4)$ time using dynamic programming.*

Proof. Take a circle C. The ω-costs of its size 2 sub-circles can be initialized in $O(|V|)$ time. Suppose the ω-costs of its sub-circles of size at most $2k - 2 > 0$ to be precomputed. We will show that the ω-costs of its sub-circles of size $2k$ can be computed in $O(|V|k^2)$ time. There are $|V| - 2k + 1$ sub-circles of C of size $2k$. Due to Lemma 23, the ω-cost of such a sub-circle is equal to the minimum of the set of size $\Theta(k^2)$. Every element of this set can be computed in constant time, as the ω-costs of the smaller sub-circles are precomputed and the ω-cost of a 2-break can be found in constant time. This means that the time complexity of computing the ω-costs for all the sub-circles of C of size $2k$ is $\Theta((|V| - k)k^2)$. This leads to an $O(|V|^4)$-time dynamic programming algorithm for computing $\omega(C)$. ◀

In [35] we demonstrated how the ω-cost of a breakpoint graph can be found using an algorithm finding the ω-cost of a circle. This leads directly to Theorem 25.

▶ **Theorem 25.** *Take two genomes having m linear chromosomes and sharing n syntenic blocks. If the ω-cost of a 2-break can be computed in constant time, then the ω-cost of their breakpoint graph can be computed in $O(mn^4)$ time.*

6 Experiments

We use OrthoCluster [45] to construct syntenic blocks between Human and four other species including Chimpanzee, Gibbon, Mouse, and Chicken. See Appendix B for details regarding this process. Several runs of OrthoCluster are performed for the species while setting the input parameter of the minimum number of genes in a syntenic block to be $\ell \in \{1, 2, 4, 8\}$. Details concerning these runs are summarized in Table 1. Ideally we would like to use the blocks with $\ell = 1$, however the blocks containing a small number of genes could be

less reliable due to annotation and assembly errors. For every set of syntenic blocks we constructed a breakpoint graph, computed the size of its largest simple cycle, that we call the *circumference*, and the length of a parsimonious scenario. Choosing an informative cost function is out of the scope of this work, however, we test the running time of our algorithm with a cost function ω_p that is equal to 1 for every 2-break (the full dynamic program ran despite all parsimonious scenarios having the same cost). If a breakpoint graph G has a circle C of size four, then any parsimonious scenario for G must contain a 2-break transforming C into a terminal graph. We call such 2-breaks *sure* and report the ratio of the number of sure 2-breaks to the length of a parsimonious scenario in Table 1.

6.1 The Elevated Time Complexity of Computing MCPS is not Prohibitive and Sure 2-breaks are Abundant

While the breakpoint graphs obtained with Chimpanzee have low circumference and could still be analyzed by hand, this is impossible for Gibbon and even more so for Mouse and Chicken. Computing the ω_p-cost took seconds for Chimpanzee and Gibbon, less than 3 minutes for Mouse, and less than 10 minutes for Chicken (see Table 1).

The parsimonious scenarios in all cases include a significant share of sure 2-breaks. We can test a given cost function on these 2-breaks, knowing that they must appear in any parsimonious scenario. This approach has been used before, for example, when ape/human deletions were recently studied [26, 22], however we are not aware of such a study for inversions or DCJs.

7 Biological Constraints

Recent studies revealed the role played by the structure of topologically associating domains (TADs), active chromatin, and 3D co-localization of the breakpoint regions in evolution by genome rearrangements. DNA breaks have been shown to occur in active chromatin regions coming into 3D contact in the nucleus [40, 10, 38]. Gibbon rearrangements were shown to occur at TAD boundaries, with most TADs maintained as intact modules during and after rearrangement [27]. A genome-wide depletion of deletions in active chromatin, and at TAD boundaries was observed across primate evolution [22]. Deletions causing TAD fusion were shown to be rare and under negative selection [26].

Recently published high resolution maps of the breakpoint regions, Hi-C, and ChIP-seq data for the highly rearranged Gibbon genome [27] combined with all the biological data available for Human genome, make these two species a very interesting target for the study of weighted genome rearrangements. This is especially so, since their parsimonious DCJ scenario contains $> 24\%$ of sure 2-breaks. These can be studied in detail to better understand the mechanisms behind the individual rearrangements. This dataset complements the set of inversions detected in human individuals [37, 16].

The algorithm presented in this article could be used to find parsimonious scenarios maximizing the number of DNA breaks at active chromatin or TAD boundaries. Breakpoint co-locality is another parameter that has been maximized in a scenario [39, 36]. The ω-cost could consider information on chromatin state, TAD boundaries and co-locality simultaneously, meaning that all the features could be used at the same time.

MCPS could also be used with the DCJ scenarios preserving common intervals [8]. In our framework, we could assign costs to the DCJs according to how well they preserve the common intervals between the genomes.

Table 1 This table summarizes our analysis of the breakpoint graphs between Human and the four other species. ℓ is the minimum number of genes in a syntenic block. *Blocks* is the number of syntenic blocks. The *Coverage* is the proportion of the Human assembly covered by a syntenic block. *Scenario length* is the length of a parsimonious scenario. *Circumference* is the size of the largest simple cycle in a breakpoint graph. *Sure 2-breaks* is the proportion of a parsimonious scenarios that is composed of sure 2-breaks. *Time* is real time required to run our algorithm for ω_p-MCPS on Intel® Xeon® Processor E5-2650 v3 (25M Cache, 2.30 GHz).

ℓ	Blocks	Coverage	Scenario length	Circumference	Sure 2-breaks	Time
Chimpanzee						
1	202	90%	116	16	33%	<2s
2	100	90%	47	10	51%	<2s
4	68	90%	26	6	73%	<2s
8	50	89%	15	6	66%	<2s
Gibbon						
1	285	87%	195	80	24%	<7s
2	202	86%	130	56	28%	<3s
4	171	86%	105	60	32%	<3s
8	139	83%	82	36	37%	<3s
Mouse						
1	659	83%	523	116	17%	<3m
2	442	82%	366	94	15%	<1m
4	335	80%	287	94	11%	<1m
8	257	76%	224	72	9%	<20s
Chicken						
1	1697	75%	1429	210	12%	<10m
2	978	73%	851	98	10%	<2m
4	628	69%	558	82	7%	<2m
8	341	60%	305	60	8%	<15s

8 Conclusion and Future Work

Our recently introduced framework [35] deals with the φ-MCPS problem, which allows for costs on gene extremities and their adjacencies, extends results for co-tailed genomes to more general genome pairs. The ω-MCPS problem is a particular case of the φ-MCPS problem, as ω-costs depend only on gene extremities. Previous work provided polynomial-time algorithms for some particular φ-costs [39, 36, 14]. Finding a general polynomial-time algorithm for the φ-MCPS problem, however, remains an open problem.

Our characterization of parsimonious scenarios into equivalence classes could prove useful for tasks such as sampling weighted parsimonious scenarios. The problem of counting the number of the parsimonious scenarios within an equivalence class remains open, as does the problem of counting and sampling the ω-MCPS scenarios.

Our algorithm for ω-MCPS was shown to be efficient enough for species as distant as Human and Chicken, and we briefly discussed possibly informative biological constraints and cost functions. Mathematical modeling of these constraints and a choice of a particular cost function, however, are left for future work.

References

1. A. Amir and A. Levy. String rearrangement metrics: A survey. In *Algorithms and Applications*, pages 1–33. Springer, 2010.
2. Y. Anselmetti, W. Duchemin, E. Tannier, C. Chauve, and S. Bérard. Phylogenetic signal from rearrangements in 18 Anopheles species by joint scaffolding extant and ancestral genomes. *BMC genomics*, 19(2):96, 2018.
3. Y. Anselmetti, N. Luhmann, S. Bérard, E. Tannier, and C. Chauve. Comparative Methods for Reconstructing Ancient Genome Organization. In *Comparative Genomics*, pages 343–362. Springer, 2018.
4. P. Avdeyev, N. Alexeev, Y. Rong, and M.A. Alekseyev. A unified ILP framework for genome median, halving, and aliquoting problems under DCJ. In *RECOMB International Workshop on Comparative Genomics*, pages 156–178. Springer, 2017.
5. V. Bafna and P.A. Pevzner. Genome rearrangements and sorting by reversals. *SIAM Journal on Computing*, 25(2):272–289, 1996.
6. Y. Baryshnikov. On Stokes sets. In *New developments in singularity theory*, pages 65–86. Springer, 2001.
7. C. Baudet, U. Dias, and Z. Dias. Sorting by weighted inversions considering length and symmetry. *BMC bioinformatics*, 16(19):S3, 2015.
8. S. Bérard, A. Chateau, C. Chauve, C. Paul, and E. Tannier. Computation of perfect DCJ rearrangement scenarios with linear and circular chromosomes. *Journal of Computational Biology*, 16(10):1287–1309, 2009.
9. A. Bergeron, J. Mixtacki, and J. Stoye. A unifying view of genome rearrangements. In *International Workshop on Algorithms in Bioinformatics*, pages 163–173. Springer, 2006.
10. C. Berthelot, M. Muffato, J. Abecassis, and H.R. Crollius. The 3D organization of chromatin explains evolutionary fragile genomic regions. *Cell reports*, 10(11):1913–1924, 2015.
11. D. Bienstock and O. Günlük. A degree sequence problem related to network design. *Networks*, 24(4):195–205, 1994.
12. P. Biller, C. Knibbe, L. Guéguen, and E. Tannier. Breaking good: accounting for the diversity of fragile regions for estimating rearrangement distances. *Genome Biol Evol*, 8:1427–39, 2016.
13. M. Blanchette, T. Kunisawa, and D. Sankoff. Parametric genome rearrangement. *Gene*, 172(1):GC11–GC17, 1996.
14. L. Bulteau, G. Fertin, and E. Tannier. Genome rearrangements with indels in intergenes restrict the scenario space. *BMC bioinformatics*, 17(14):426, 2016.
15. A. Caprara. Sorting permutations by reversals and Eulerian cycle decompositions. *SIAM journal on discrete mathematics*, 12(1):91–110, 1999.
16. M.J.P. Chaisson, A.D. Sanders, X. Zhao, A. Malhotra, D. Porubsky, T. Rausch, E.J. Gardner, O.L. Rodriguez, L. Guo, R.L. Collins, et al. Multi-platform discovery of haplotype-resolved structural variation in human genomes. *Nature communications*, 10, 2019.
17. P.E.C. Compeau. A generalized cost model for DCJ-indel sorting. In *International Workshop on Algorithms in Bioinformatics*, pages 38–51. Springer, 2014.
18. F. Farnoud and O. Milenkovic. Sorting of permutations by cost-constrained transpositions. *IEEE Transactions on Information Theory*, 58(1):3–23, 2012.
19. T. Feder, A. Guetz, M. Mihail, and A. Saberi. A local switch Markov chain on given degree graphs with application in connectivity of peer-to-peer networks. In *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*, pages 69–76. IEEE, 2006.
20. G. Fertin, G. Jean, and E. Tannier. Algorithms for computing the double cut and join distance on both gene order and intergenic sizes. *Algorithms for Molecular Biology*, 12(1):16, 2017.
21. B.K. Fosdick, D.B. Larremore, J. Nishimura, and J. Ugander. Configuring random graph models with fixed degree sequences. *SIAM Review*, 60(2):315–355, 2018.
22. G. Fudenberg and K.S. Pollard. Chromatin features constrain structural variation across evolutionary timescales. *Proceedings of the National Academy of Sciences*, 116(6), 2019.

23 T. Hartmann, M. Bernt, and M. Middendorf. An exact algorithm for sorting by weighted preserving genome rearrangements. *IEEE/ACM transactions on computational biology and bioinformatics*, 16(1):52–62, 2019.

24 T. Hartmann, M. Middendorf, and M. Bernt. Genome Rearrangement Analysis: Cut and Join Genome Rearrangements and Gene Cluster Preserving Approaches. In *Comparative Genomics*, pages 261–289. Springer, 2018.

25 T. Hartmann, N. Wieseke, R. Sharan, M. Middendorf, and M. Bernt. Genome rearrangement with ILP. *IEEE/ACM transactions on computational biology and bioinformatics*, 15(5), 2018.

26 L. Huynh and F. Hormozdiari. TAD fusion score: discovery and ranking the contribution of deletions to genome structure. *Genome biology*, 20(1):60, 2019.

27 N.H. Lazar, K.A. Nevonen, B. O'Connell, C. McCann, R.J. O'Neill, R.E. Green, T.J. Meyer, M. Okhovat, and L. Carbone. Epigenetic maintenance of topological domains in the highly rearranged gibbon genome. *Genome research*, 28(7):983–997, 2018.

28 Y. Lin, V. Rajan, and B.M.E. Moret. TIBA: a tool for phylogeny inference from rearrangement data with bootstrap analysis. *Bioinformatics*, 28(24):3324–3325, 2012.

29 F.V. Martinez, P. Feijao, M.D.V. Braga, and J. Stoye. On the family-free DCJ distance. In *International Workshop on Algorithms in Bioinformatics*, pages 174–186. Springer, 2014.

30 I. Miklós and E. Tannier. Approximating the number of Double Cut-and-Join scenarios. *Theoretical Computer Science*, 439:30–40, 2012.

31 A. Ouangraoua and A. Bergeron. Combinatorial structure of genome rearrangements scenarios. *Journal of Computational Biology*, 17(9):1129–1144, 2010.

32 S. Pulicani, P. Simonaitis, E. Rivals, and K.M. Swenson. Rearrangement scenarios guided by chromatin structure. In *RECOMB International Workshop on Comparative Genomics*, pages 141–155. Springer, 2017.

33 M. Shao, Y. Lin, and B. Moret. An exact algorithm to compute the DCJ distance for genomes with duplicate genes. In *International Conference on Research in Computational Molecular Biology*, pages 280–292. Springer, 2014.

34 M. Shao, Y. Lin, and B.M.E. Moret. Sorting genomes with rearrangements and segmental duplications through trajectory graphs. In *BMC bioinformatics*. BioMed Central, 2013.

35 P. Simonaitis, A. Chateau, and K.M. Swenson. A General Framework for Genome Rearrangement with Biological Constraints. In *RECOMB International conference on Comparative Genomics*, pages 49–71. Springer, 2018.

36 P. Simonaitis and K.M. Swenson. Finding local genome rearrangements. *Algorithms for Molecular Biology*, 13(1):9, 2018.

37 P.H. Sudmant, T. Rausch, E.J. Gardner, R.E. Handsaker, A. Abyzov, J. Huddleston, Y. Zhang, K. Ye, G. Jun, M.H. Fritz, et al. An integrated map of structural variation in 2,504 human genomes. *Nature*, 526(7571):75, 2015.

38 K.M. Swenson and M. Blanchette. Large-scale mammalian genome rearrangements coincide with chromatin interactions. *Bioinformatics*, July 2019.

39 K.M. Swenson, P. Simonaitis, and M. Blanchette. Models and algorithms for genome rearrangement with positional constraints. *Algorithms for Molecular Biology*, 11(1):13, 2016.

40 A. Veron, C. Lemaitre, C. Gautier, V. Lacroix, and M.-F. Sagot. Close 3D proximity of evolutionary breakpoints argues for the notion of spatial synteny. *BMC Genomics*, 12, 2011.

41 R. Warren and D. Sankoff. Genome halving with double cut and join. *Journal of Bioinformatics and Computational Biology*, 7(02):357–371, 2009.

42 S. Yancopoulos, O. Attie, and R. Friedberg. Efficient sorting of genomic permutations by translocation, inversion and block interchange. *Bioinformatics*, 21(16):3340–3346, 2005.

43 R. Zeira and R. Shamir. Sorting cancer karyotypes using double-cut-and-joins, duplications and deletions. *Bioinformatics (Oxford, England)*, 2018.

44 R. Zeira and R. Shamir. Genome Rearrangement Problems with Single and Multiple Gene Copies: A Review. In *Bioinformatics and Phylogenetics*, pages 205–241. Springer, 2019.

45 X. Zeng, M.J. Nesbitt, J. Pei, K. Wang, I.A. Vergara, and N. Chen. OrthoCluster: a new tool for mining synteny blocks and applications in comparative genomics. In *Proceedings of the 11th international conference on Extending database technology*. ACM, 2008.

A Proofs

A.1 Lemma 15

▶ **Lemma.** *For an embedding of a connected graph $G = (V, E)$ with $|V|$ edges incident to the outer face and all the inner faces being quadrilaterals, the number of internal faces $|F_{int}|$ is $\frac{|V|}{2} - 1$.*

Proof. Every internal face of G is bounded by four edges, G has $|V|$ edges incident to the outer face, and every edge of G is separating two faces. By counting edges we obtain the equality $4|F_{int}| + |V| = 2|E|$. Euler's formula for planar graphs is $|V| - |E| + |F| = 2$, thus $|E| = |V| + |F| - 2 = |V| + |F_{int}| - 1$. Combining these two equalities we obtain $4|F_{int}| + |V| = 2|V| + 2|F_{int}| - 2$ and finally $|F_{int}| = \frac{|V|}{2} - 1$. ◀

A.2 Corollary 17

▶ **Corollary.** *Consider a circle C, a parsimonious scenario ρ for C, and an internal face f of $\Sigma_{\mathcal{S}(C,\rho)}$ bounded by the edges $\{u, v\}, \{v, s\}, \{s, q\}$ and $\{q, u\}$, where $\{u, v\}$ is a black edge of C. Then ρ contains a 2-break $\{\{u, v\}, \{q, s\}\} \to \{\{u, q\}, \{v, s\}\}$.*

Proof. If $\{u, v\}$ is among the edges of at least two different 2-breaks in ρ, then $\{u, v\}$ is incident to at least two different internal faces of $\Sigma_{\mathcal{S}(C,\rho)}$ due to Proposition 16. However $\{u, v\}$ is a black edge of C, thus it is incident to a single internal face of $\Sigma_{\mathcal{S}(C,\rho)}$, and that face is f. This means that $\{u, v\}$ is among the edges of a single 2-break τ in ρ.

Due to Proposition 16, the edges of τ are exactly the edges of f. This means that τ is either $\{\{u, v\}, \{q, s\}\} \to \{\{u, q\}, \{v, s\}\}$, or $\{\{u, q\}, \{v, s\}\} \to \{\{u, v\}, \{q, s\}\}$. $\{u, v\}$ is not present in the terminal graph obtained from C after ρ is performed. This means that ρ includes a 2-break replacing $\{u, v\}$. Combining these two observations we conclude that τ is $\{\{u, v\}, \{q, s\}\} \to \{\{u, q\}, \{v, s\}\}$. ◀

A.3 Lemma 18

▶ **Lemma.** *For every circle C, a parsimonious scenario ρ for C, and an edge e of $\Sigma_{\mathcal{S}(C,\rho)}$, ρ can be partitioned into two (possibly empty) parsimonious scenarios, one for each sub-circle delimited by e.*

Proof. If C has two vertices, then the sub-circles in question are two copies of C, and empty scenarios satisfy the statement. Suppose that it is true for any circle having at most $2k - 2 \geq 2$ vertices and take a circle with $2k$ vertices together with its parsimonious scenario ρ and an edge e of $\Sigma_{\mathcal{S}(C,\rho)}$.

Denote the two sub-circles delimited by e by C_3 and C_4. Take the first 2-break τ of ρ and suppose that its vertices do not belong to the same sub-circle. In this case at least one of its edges is incident to the vertices that do not belong to the same sub-circle. This edge crosses e, which contradicts the planarity of $\Sigma_{\mathcal{S}(C,\rho)}$. This means that the vertices of τ belong to the same sub-circle. Without loss of generality we can suppose that it is C_3. τ transforms C into a union of two vertex-disjoint circles. Denote the one containing the endpoints of e by C_2 and the other one by C_1. Due to Corollary 7, the rest of ρ can be partitioned into two sub-sequences ρ_1 and ρ_2, that are respectively scenarios for C_1 and C_2.

The inductive hypothesis holds for the triplet C_2, ρ_2 and e, providing us with a partition of ρ_2 into two parsimonious scenarios for the sub-circles of C_2 delimited by e. One of these sub-circles is C_4. Denote the other by C_5 and denote the parsimonious scenarios obtained for them by ρ_4 and ρ_5 respectively.

By removing from ρ the 2-breaks in ρ_4 and a 2-break τ, we obtain a sequence of 2-breaks ρ', that is a shuffle of the parsimonious scenarios ρ_1 for C_1 and ρ_5 for C_5. Due to Corollary 8, ρ' is a parsimonious scenario for the union of C_1 and C_5, which is the graph obtained from C_3 after τ is performed. Finally, by adding τ at the beginning of ρ' we obtain a parsimonious scenario for C_3, which together with ρ_4 satisfies the statement. ◀

A.4 Theorem 20

We first prove the following lemma.

▶ **Lemma.** *For every circle C, a parsimonious scenario ρ for C, and a 2-break in ρ replacing two black edges of C, the sequence of 2-breaks with this 2-break performed first and followed by the rest of ρ is also a parsimonious scenario for C.*

Proof. Take a circle C, its parsimonious scenario ρ and a 2-break τ in ρ replacing two black edges. Due to Proposition 16, there exists a face in $\Sigma_{\mathcal{S}(C,\rho)}$ bounded by the edges of τ. Using Theorem 19 we obtain a partition of ρ into four scenarios for the sub-circles of C delimited by the edges of τ and τ itself, however two of these scenarios are for the circles with 2 vertices, thus of length 0. Denote the other two by ρ_1 and ρ_2. This means that ρ without τ is a shuffle of two parsimonious scenarios for the vertex-disjoint circles. Using Corollary 8 we obtain that ρ without τ is a parsimonious scenario for the union of these circles, that is obtained from C after τ is performed. This means that moving τ to the beginning of ρ we obtain a parsimonious scenario for C. ◀

▶ **Theorem.** *For every circle C and its parsimonious scenarios ρ and μ, their scenario graphs are equal if and only if these scenarios are equivalent.*

Proof. If a circle C has two vertices, then its parsimonious scenarios are empty and the statement is true. Suppose that the statement is true for every circle with at most $2k-2 \geq 2$ vertices. Take a circle C with $2k$ vertices and two parsimonious scenarios ρ and μ for C.

First suppose that ρ and μ contain exactly the same 2-breaks but possibly in different order. By construction, the edges of $\Sigma_{\mathcal{S}(C,\rho)}$ are the edges of the 2-breaks in ρ, and these are exactly the same as the edges of the 2-breaks in μ, meaning that $\Sigma_{\mathcal{S}(C,\rho)} = \Sigma_{\mathcal{S}(C,\mu)}$.

Now suppose that $\Sigma_{\mathcal{S}(C,\rho)} = \Sigma_{\mathcal{S}(C,\mu)}$. Take τ, the first 2-break of ρ. By construction, it replaces two black edges of C and its edges bound a face in $\Sigma_{\mathcal{S}(C,\rho)}$ due to Proposition 16. $\Sigma_{\mathcal{S}(C,\rho)} = \Sigma_{\mathcal{S}(C,\mu)}$, thus the latter contains the same face that we denote by f. Due to Corollary 17, μ contains a 2-break replacing the black edges bounding f, this means that μ contains τ. Using the lemma proven above with a 2-break τ we get that the sequence of 2-breaks obtained by moving μ to the beginning of μ is a parsimonious scenario for C that we denote by μ'. Using the first part of this proof we obtain that $\Sigma_{\mathcal{S}(C,\mu')}$ is equal to $\Sigma_{\mathcal{S}(C,\mu)}$ and thus $\Sigma_{\mathcal{S}(C,\rho)}$.

Due to Corollary 7, τ transforms C into two vertex-disjoint circles C_1 and C_2. Denote the subgraph of $\Sigma_{\mathcal{S}(C,\rho)}$ induced by the vertices of C_1 (respectively C_2) by $\Sigma_{\mathcal{S}_1}$ (respectively $\Sigma_{\mathcal{S}_2}$). Due to Corollary 8, the rest of ρ (respectively μ') can be partitioned into two sub-sequences ρ_1, ρ_2 (respectively μ'_1, μ'_2) that are scenarios for C_1 and C_2. By construction, $\Sigma_{\mathcal{S}(C,\rho_1)} = \Sigma_{\mathcal{S}_1}$ and $\Sigma_{\mathcal{S}(C,\rho_2)} = \Sigma_{\mathcal{S}_2}$ (respectively $\Sigma_{\mathcal{S}(C,\mu'_1)} = \Sigma_{\mathcal{S}_1}$ and $\Sigma_{\mathcal{S}(C,\mu'_2)} = \Sigma_{\mathcal{S}_2}$). Using the inductive hypothesis we obtain that ρ_1 and μ'_1 contain exactly the same 2-breaks and so does ρ_2 and μ'_2, which means that ρ and μ' contain exactly the same 2-breaks. ◀

A.5 Proposition 22

▶ **Proposition.** *For every circle C and its quadrangulation Σ_S there exists a parsimonious scenario ρ such that $\Sigma_{\mathcal{S}(C,\rho)} = \Sigma_S$.*

Proof. If a circle has 2 vertices, then an empty scenario satisfies the statement. Suppose that the statement is true for every circle having at most $2k-2 \geq 2$ vertices and take a circle C with $2k$ vertices together with its quadrangulation Σ_S (see Figure 3 for an example).

Due to Lemma 15, Σ_S has $k-1$ internal face and k outer edges that are black in C. An outer edge of C belongs to at most one internal face of Σ_S, thus we obtain that Σ_S has an internal face f bounded by the edges among which there are at least two black edges of C.

Perform a 2-break on these black edges of C transforming it into a vertex-disjoint union of two sub-circles C_1 and C_2. The subgraphs of Σ_S induced by the vertices of C_1 and C_2 are quadrangulations of C_1 and C_2 that we denote by Σ_{S_1} and Σ_{S_2}. They satisfy the inductive hypothesis and provide us with parsimonious scenarios ρ_1 and ρ_2 for circles C_1 and C_2. Combining them with the initial 2-break we obtain a parsimonious scenario ρ for C with $\Sigma_{\mathcal{S}(C,\rho)} = \Sigma_S$. ◀

A.6 Lemma 23

▶ **Lemma.** *For a sub-circle $C[i,j]$ of a circle C with $i<j$, its cost $\omega[i,j]$ is 0 if it has two vertices, otherwise*

$$\omega[i,j] = min_{r,s}\big(\omega[i,r] + \omega[r,s] + \omega[s,j] + \omega(\tau)\big)$$

for vertices $i<r<s<j$ of a valid quadrilateral for $C[i,j]$, where $\tau = \big\{\{i,r\},\{s,j\}\big\} \to \big\{\{i,j\},\{r,s\}\big\}$ if i is odd, and $\tau = \big\{\{i,j\},\{r,s\}\big\} \to \big\{\{i,r\},\{j,s\}\big\}$ if i is even.

Proof. If $j = i+1$, then $C[i,j]$ has two vertices, its parsimonious scenario is of length 0, and thus of cost 0. We start by showing a lower bound on $\omega[i,j]$. Take a scenario ρ_o of minimum cost among the parsimonious scenarios for $C[i,j]$, and a face f in $\Sigma_{S(C[i,j],\rho_o)}$ incident to the edge $\{i,j\}$. Denote its other two vertices by r and s with $r<s$. By construction $i+r$, $r+s$ and $s+j$ are all odd. Due to Theorem 19, ρ_o can be partitioned into four parsimonious scenarios ρ_1,ρ_2,ρ_3 and ρ_4 for the sub-circles of $C[i,j]$ delimited by the edges bounding f plus a 2-break τ whose edges bound f. The sub-circles of $C[i,j]$ in question are respectively $C[i,r]$, $C[r,s]$, $C[s,j]$ and a circle of size 2 containing the edge $\{i,j\}$. If i is even, then $\{i,i+1\}$ is gray and $\{i,j\}$ is black by construction. Using Corollary 17 we obtain that τ is $\big\{\{i,j\},\{r,s\}\big\} \to \big\{\{i,r\},\{j,s\}\big\}$. Similarly τ is $\big\{\{i,r\},\{s,j\}\big\} \to \big\{\{i,j\},\{r,s\}\big\}$ if i is odd. This establishes that $\omega(C) = \omega(\rho_o) = \omega(\rho_1)+\omega(\rho_2)+\omega(\rho_3)+\omega(\rho_4)+\omega(\tau) \geq \omega[i,r]+\omega[r,s]+\omega[s,j]+0+\omega(\tau)$.

Now we show the upper bound on $\omega(C)$. Take $i<r<s<j$ with odd $i+r, r+s, s+j$, and minimum cost parsimonious scenarios ρ_1,ρ_2 and ρ_3 for the sub-circles $C[i,r], C[r,s]$ and $C[s,j]$. If i is even, then $\{i,i+1\}$ is gray, thus $\{i,j\}$ is black in $C[i,j]$, and $\{r,s\}$ is gray in $C[r,s]$. This means that we can perform ρ_2 on $C[i,j]$, and then follow it by the 2-break τ equal to $\big\{\{i,j\},\{r,s\}\big\} \to \big\{\{i,r\},\{j,s\}\big\}$, followed by ρ_1 and ρ_3. This is a parsimonious scenario for $C[i,j]$ that we denote ρ, and $\omega[i,j] \leq \omega(\rho) = \omega(\rho_2)+\omega(\tau)+\omega(\rho_1)+\omega(\rho_3) = \omega[r,s]+\omega(\tau)+\omega[i,r]+\omega[s,j]$. If i is odd, then similarly we obtain that $\{i,r\}$ and $\{s,j\}$ are both black in $C[i,r]$ and $C[s,j]$ respectively, and that ρ_1 followed by ρ_3, a 2-break τ equal to $\big\{\{i,r\},\{s,j\}\big\} \to \big\{\{i,j\},\{r,s\}\big\}$ and ρ_2 is a parsimonious scenario for $C[i,j]$ that we denote by ρ. This leads to the inequality $\omega[i,j] \leq \omega(\rho) = \omega(\rho_1)+\omega(\rho_3)+\omega(\tau)+\omega(\rho_2) = \omega[i,r]+\omega[s,j]+\omega(\tau)+\omega[r,s]$. ◀

B Methods

B.1 Downloading and Preprocessing Orthologs

Protein coding Human (GRCh38.p12), Chimpanzee (Pan_tro_3.0), Gibbon (Nleu_3.0), Mouse (GRCm38.p6) and Chicken (GRCg6a) genes were downloaded from `Ensemble Release 96 using Biomart`. Orthologous groups of protein coding genes between Human and the rest of the species were downloaded from the same database. Ensemble provides orthologous pairs with a `confidence score` that is either low or high. Low confidence pairs were filtered out together with the orthologous pairs containing genes in unlocalized or unplaced scaffolds. At this point we were left with 17319 orthologous pairs for Chimpanzee, 14923 for Gibbon, 16848 for Mouse, and 13024 for Chicken.

B.2 OrthoCluster: Identifying syntenic blocks

For the construction of syntenic blocks we chose to use OrthoCluster [45] due to its ease of use, speed, and allowance for different parametric constraints. OrthoCluster takes the orthologous groups as input and identifies the syntenic blocks. It handles many-to-many orthologs and overlapping genes, thus no further filtering of the previously prepared orthologs was required. OrthoCluster deals with various types of mismatches, however in this study we required the genes in the syntenic blocks to share exactly the same order and strandedness, and contain no mismatches. We processed these blocks by deleting the ones that were included in some other blocks, and by merging those blocks that were consecutive in both genomes and either 1) had the same order and strandedness, or 2) opposite order and strandedness in both genomes. Table 1 contains the coverage of the blocks before merging, and the number of the blocks after merging. The final blocks might have some overlaps due to the overlapping genes.

B.3 OrthoCluster Identifies Blocks of High Coverage Without Performing a Whole Genome Alignment

Golden path length of the Human assembly `GRCh38.p12` is around 3.1Gbp. For a set of syntenic blocks, we divide the number of the nucleotides from Human included in at least one of the blocks by this number to obtain the *coverage of the blocks*. The number of the blocks and their coverage computed using OrthoCluster are comparable to those of the blocks available on `Ensemble Release 96 Compara` database. These were computed using whole genome alignments and their number of blocks and coverage are respectively 192 and 92% for Chimpanzee, 277 and 90% for Gibbon, 363 and 88% for Mouse, 398 and 68% for Chicken.