

# Genome Rearrangement by the Double Cut and Join Operation

Richard Friedberg\*, Aaron E. Darling† and Sophia Yancopoulos<sup>‡</sup>

\* Department of Physics, Columbia University, NY, NY 10027 USA

†Institute for Molecular Bioscience, University of Queensland, St. Lucia, QLD 4072 Australia

<sup>‡</sup>The Feinstein Institute for Medical Research, 350 Community Dr, Manhasset, NY 11030, USA

## i. Abstract

The *Double Cut and Join* is an operation acting locally on four gene ends without regard to chromosomal context. We discuss its application and the resulting menu of operations for genomes consisting of arbitrary numbers of circular chromosomes, as well as for a general mix of linear and circular chromosomes. In the general case the menu includes: inversion, translocation, transposition, formation and absorption of circular intermediates, conversion between linear and circular chromosomes, block interchange, fission and fusion. We discuss the well known edge graph and its dual, the adjacency graph, recently introduced by Bergeron *et al.* We give step-by-step procedures for constructing these graphs and for manipulating them. We give simple algorithms in terms of the adjacency graph for computing the minimal DCJ distance between two genomes, for finding a minimal sorting, and we describe use of an online tool (Mauve) to generate synteny blocks and apply DCJ.

**ii. Key Words:** genome rearrangements, gene order, Mauve, synteny, inversion, reversal, translocation, transposition, block interchange, fission, fusion

## 1. Introduction

### 1.1 Background

Comparative analyses of genomes have identified regions of regions of similarity or “conserved segments” [1] among species which may be scrambled from one genome to

another. Such regions can be anywhere from a few nucleotides up to millions of base pairs depending on the criteria; at the small end they cover only a fraction of a gene or a small stretch of an intergenic element, while on the other end they span vast genomic tracts containing a multitude of genes. We will not focus on the identification of homologous regions or synteny blocks in this section (but see section 2.2.1), and for simplicity, we refer to these units as “genes”. The reader is cautioned that this nomenclature connotes neither the size nor functional characterization of these genomic elements.

The problem of transforming one genome to another, where the two genomes have the same gene content without paralogs, can be reduced to that of sorting a signed permutation via a defined set of operations. The set of operations possible not only depends on, but can affect, the kind of genomes considered. For single chromosomes, a significant amount of effort has been given to the study of inversions (reversals) of segments of arbitrary length [2]. When multiple linear chromosomes are involved, these can be generalized to include translocations, which exchange end-segments between two chromosomes, as well as fissions and fusions. The study of evolutionary “*distance*” (minimum number of steps) between two genomes containing the same genetic material in different arrangements depends on the choice of elementary operations. It can be seen that with a menu of “generalized reversals” the distance cannot be less than  $b-c$ , where  $b = \# \text{ breakpoints}$ , and  $c = \# \text{ cycles}$  in the *breakpoint graph* (see Section 1.2.3). But certain obstacles can prevent the lower bound from being achieved.

Recently, attention has been given to a single universal operation, the ***double cut and join*** (DCJ), which acts indiscriminately on gene ends without regard to the chromosomal disposition of the genes [3, 4]. Special cases of this operation are inversion, fission, fusion, translocation, and conversion between linear and circular chromosomes.

When the basic operation is taken to be unrestricted DCJ, the lower bound  $b - c$  is always achieved and the finding of optimal paths is vastly simplified. One also obtains this minimal distance in the restricted case of linear chromosomes, with the caveat that once a circular intermediate is created (by one DCJ) it must be annihilated (by absorption or linearization) by the DCJ which immediately follows. We shall see (section 1.5) that such a dual operation is equivalent to a single operation given a weight of 2, also known as “block interchange” or generalized transposition [3, 5, 6].

## ***1.2 DCJ on circular chromosomes***

### ***1.2.1 Genome graphs***

A genome consisting of  $N$  genes is completely described by specifying the connections between adjacent gene ends. There are  $2N$  gene ends and  $(2N-1)!! = 1*3*5*...*(2N-1)$  ways to pair them. Each such pairing corresponds to a genome configuration. This allows chromosomes consisting of a single gene eating its tail. It is possible to represent the configuration of a genome by a graph [Figure 1] in which the two ends of a single gene are connected by a white line, and the connection between two adjacent gene ends is represented by a black line. Such a graph may be called a genome graph; it must not be confused with the edge graph to be described later, which pertains to two genomes. We shall consistently speak of the path along a single gene as a “white line” although in the literature it has been variously represented as a solid line, a dashed line or a space.

### ***1.2.2 DCJ operation***

The DCJ operation can be defined entirely as a modification of the black lines without any reference to the white lines. One chooses two black lines and cuts them. This leaves four loose ends which can be rejoined in 3 different ways, one of which reestablishes

the previous configuration. One completes the DCJ by choosing either of the remaining two ways of rejoining the ends [Figure 2].

Sorting circular chromosomes by DCJ is equivalent to sorting by reversals, fusions and fissions. If the two black lines to be cut are initially on different chromosomes, either of the two possible DCJs will accomplish a fusion; the two possible outcomes differ by relative reversal of the two fragments. If the two black lines to be cut are initially on the same chromosome, one of the possible DCJs will accomplish a reversal within the chromosome, and the other a fission into two chromosomes [Figure 3].

The operations of block interchange [3-6] and of block interchange with reversal can each be accomplished by a fission followed by a fusion, and therefore can be reduced to two DCJs [3] [Figure 4]. Transposition and transversion are special cases of block interchange, where the two blocks to be exchanged are contiguous [7].

To distinguish these conventional operations from one another- for example to distinguish reversal from fusion- requires a knowledge of the placement of the gene ends on the chromosome, which in turn requires a knowledge of which gene ends lie at opposite ends of the same gene. In other words, it requires access to the white lines. By studying DCJ without making these distinctions, one addresses a simpler mathematical problem in which the white lines play no part. Not only does this lead to a relatively simple distance formula, but it may open the way to attack more complex rearrangement problems which would remain totally intractable in terms of the conventional operations that depend on the white lines.

### ***1.2.3 Edge graph***

The problem of sorting one genome (the initial genome) into another (the target genome) by DCJ operations is simply the problem of progressing by such operations from

the initial pairing of  $2N$  gene ends to the target pairing. At any stage in this sequence of operations the intermediate configuration reached will be called the “current genome”. It is conventional to represent the connections between adjacent gene ends by black lines in the initial as well as in the current pairing, and by gray lines in the target pairing.

The comparison between initial or current genome and target genome can be visualized by means of a graph (Figure 5c) consisting of the  $2N$  points joined by black and gray lines. The points represent gene ends; each point refers to that gene end in both genomes. The black lines tell how to connect the genes in the current genome, and the gray in the target genome. If nothing else is added, the graph gives no information about which gene ends belong to the same gene or which points belong to the same chromosome. It is often convenient, however, to supply this information by making the “white lines” (representing genes) visible or by labeling the gene ends as in the genome graph. This graph, first introduced in [8], has been variously called edge graph, breakpoint graph and comparison graph in the literature. (But see Section 1.2.5, paragraph 4.)

#### ***1.2.4 Adjacency graph***

It has been suggested recently [9] that there are advantages to replacing the edge graph by its dual. To construct the dual graph, replace every line of the edge graph by a point, and every point by a line. The resulting graph is called the adjacency graph [4]. Perhaps the most striking advantage of the adjacency graph is the ease with which it can be constructed, given the two genome graphs (initial and target) to be compared.

Let us begin with the initial (or current) genome graph, drawn with only white lines. That is, the black lines have been contracted to points. We label each point by the gene ends that meet there; thus if the head of gene 7 is adjacent to the tail of gene 3 one would label the point of adjacency 7h3t (Figure 6a.) In like fashion we draw the genome graph of the target

genome (Figure 6b.) This is placed underneath the first genome graph so that both appear in the same picture. Thus each gene is represented twice, once above and once below. To complete the adjacency graph, we simply join the endpoints of each gene in the current genome at the to the corresponding endpoints in the target genome below (Figure 6c.) For ease of reference we shall refer to the joining lines as green lines. Thus each adjacency is the terminus of two green lines. The adjacency graph proper is obtained by dropping the white lines and retaining only the green (Figure 6d.)

The power of the adjacency graph is seen when it is recognized as the perfect dual (lines and points interchanged) of the edge graph described in the preceding section. Therefore anything that can be found from one can be found from the other. However, the adjacency graph has distinct advantages as visual display. First, it is considerably easier to construct by hand than the edge graph – the reader is encouraged to try both constructions starting from the same genome pair, see Sections 3.3 and 3.4 – and the correctness of the construction is much easier to verify for the adjacency graph. Second, in the edge graph the two genomes are represented tangled together and at least one of them is necessarily difficult to visualize from an examination of the graph. In the adjacency graph the two genomes are visually distinct and each one can be represented in a way that closely resembles the set of chromosomes.

The duality between the adjacency and the edge graphs can be visualized with the aid of a more complex graph which we call the master graph (Figure 7). The master graph contains all the black and gray lines of the edge graph as well as the green lines of the adjacency graph. Starting from the master graph, one obtains the edge graph by contracting the green lines to single points, or correspondingly the adjacency graph by contracting the black and gray lines to single points.

### 1.2.5 Distance

To solve the problem of sorting sorting circular chromosomes by DCJ, we observe that in the edge graph each point has two connections and therefore the graph resolves itself into closed cycles consisting of alternating black and gray lines. When linear chromosomes are allowed the situation is more complicated (see section 1.3.) If every DCJ is begun by cutting two black lines in the same cycle, the ends can be rejoined in a way that causes the cycle to be split in two (Figure 8.) At the end of the sorting, the current genome will be identical to the target genome, and the edge graph will consist of  $N$  cycles each composed of one black and one gray line (Figure 9.) We shall call these 1-cycles. Since each DCJ increased the number of cycles by 1, the number of DCJ steps performed was  $N-C$ , where  $C$  is the number of cycles in the beginning edge graph.

From the preceding argument it is also clear that the number of cycles cannot be increased by more than one at a step, so that no sorting is possible in fewer than  $N-C$  steps [10]. One thus arrives at the distance formula:  $d=N-C$ .

A frequent convention is to eliminate each 1-cycle as soon as it is formed by deleting the black line and the gray line of which it is comprised (Figure 10.) The lines of the edge graph with all 1-cycles deleted are called breakpoints. The number of either black or gray breakpoints is called  $b$ . We shall denote the number of cycles excluding the number of 1-cycles by  $c$  as is usually done in this convention. Since the number of 1-cycles can be written either as  $N-b$  or as  $C-c$ , the distance formula can be written as  $d=N-C=b-c$ . The proof presented above for the distance can be presented equally well with  $b-c$ .

In this chapter we generally allow 1-cycles to be retained in the edge graph and write the distance as  $N-C$ . Strictly, the term “breakpoint graph” is not applicable to our edge

graph, since in most literature the term “breakpoint” refers only to a connection between two gene ends which is present in one genome but is broken in the other.

The formula  $d = N - C$  can equally be evaluated from the adjacency graph, since the green lines there form cycles that correspond one for one to the black-gray cycles of the edge graph. A 1-cycle then consists of an A point and a B point joined by two green lines.

In the adjacency graph the DCJ is defined in terms of two of the points belonging to the current genome. These points are deleted leaving 4 green lines lacking a termination. These 4 lines are paired in either of the two remaining ways and for each pair a new point is provided to serve as the common terminus. (For linear chromosomes see section 1.4.) This procedure is completely parallel to the one described above in terms of the edge graph, and one obtains the distance formula by essentially the same argument.

### ***1.3 DCJ on circular and linear chromosomes (applications of edge graph)***

#### ***1.3.1 Caps and null chromosomes***

In this section we shall allow both circular and linear chromosomes to be present in each genome. Now if we pair adjacent gene ends the two end points of each chromosome are not paired, so that there are only  $2N - 2L$  points to be paired where  $L$  is the number of linear chromosomes. In this way we obtain a genome graph consisting of  $N - L$  black lines and  $N$  white lines (Figure 11b.) It would be possible to use this as a basis for constructing the edge graph between two genomes, but the resulting DCJ operation would not be capable of handling end points of a linear chromosome. Thus for example, the operation of reversing an end segment of a linear chromosome could not be subsumed under the DCJ operation. In order to include operations on endpoints including those of fission and fusion, it is convenient to consider each chromosomal endpoint in the current genome as the terminus of

a black line whose other end is not attached to any gene. This unattached end is called a “cap” [14]. The resulting genome graph has  $N + L$  black and  $N$  white lines (Figure 11c).

### ***1.3.2 DCJ on capped lines***

The DCJ operation is now defined just as in section I. The capped black lines are treated completely on a par with those that connect two genes. When one of the black lines to be cut is capped, the half attached to the cap provides one of the four loose ends to be rejoined. The rejoining then connects the cap to one of the two gene ends originally connected by the other black line; this gene end now becomes a chromosomal end point in place of the original one (Figure 12a).

When two capped black lines are cut it is possible to rejoin the loose ends so that the two caps are connected, and the two chromosomal endpoints are joined. This operation decreases  $L$  by 1, either by converting a linear to a circular chromosome, or by fusing two linear chromosomes into one. The structure consisting of two caps joined by a black line is an artifact of the construction which we call a null chromosome (Figure 12b). Null chromosomes do not contain genes and can be added or deleted without affecting the genome.

By including any number of null chromosomes in the initial genome, one may achieve the inverse of the above process. A null chromosome is destroyed by cutting the black line it contains and rejoining the ends to those of another black line that was internal to a chromosome. Thus  $L$  is increased by 1, either by fission of a linear chromosome or by conversion of a circular chromosome to a linear chromosome.

### ***1.3.3 Menu of operations***

As noted in section 1.2.2, when only circular chromosomes are present (and by tacit agreement no null chromosomes) the conventional operations mimicked by DCJ are

reversals, fusions and fissions. When linear chromosomes are allowed, we must add to this list translocations between linear chromosomes and conversions from linear to circular or circular to linear chromosomes.

A complete list of possible DCJ operations with their effect on genomic structure is presented in Table 1. The genomic effects are also summarized in Figure 13.

#### ***1.3.4 Paths and cycles***

We now consider the DCJ distance between an initial genome A and a target genome B, when linear chromosomes are permitted. The edge graph may not consist entirely of cycles. Besides any cycles present there may be a number of paths, beginning at one cap and ending at another. In defining paths we regard the caps in genome A, which terminate black lines, as distinct from the caps in genome B which terminate gray lines. A path will be called “odd” if it starts with a cap belonging to one genome and ends with a cap in the other. It will be called “even” if both its caps are in the same genome (Figure 14.) Odd paths may also be called AB paths; even paths may be called AA or BB paths, according to the location of the end caps.

#### ***1.3.5 Closure and distance***

The distance between two genomes can be found by closing all paths into cycles and applying the formula  $d = N' - C'$  where  $N'$  and  $C'$  are found from the closed graph. To close an AB path, identify the two end caps. To close an AA path, introduce a null chromosome into the B genome and identify its caps with those of the AA path. To close a BB path, introduce a null chromosome into the A genome and identify its caps with those of the BB path. After closure the graph will contain an equal number  $N'$  of black lines and gray lines, and a number  $C'$  of cycles including those formed from paths.

### ***1.4 DCJ on circular and linear chromosomes (applications of the adjacency graph)***

### ***1.4.1 Telomeres***

If the adjacency graph is constructed in accordance with Section 1.2.4, each genome will in general contain not only adjacency points corresponding to the connections between adjacent genes in a chromosome, but also telomere points, which correspond to the endpoints of linear chromosomes. An adjacency has two labels and two green lines attached; a telomere has only one of each (Figure 15d.) To perform a DCJ on two adjacencies, the four green lines incident on them are detached and reshuffled as described in Section 1.2.5 (last paragraph).

To achieve the full menu of operations given in Section III, it must be possible to perform DCJ also on telomeres. Then one has only three green lines to reshuffle, or only two if both points are telomeres; in the last case the result is a single adjacency point. Bergeron et al [4] treat these possibilities as separate cases, along with another case in which a single adjacency is attacked, yielding two telomeres. Thus the correct dualism is achieved between the edge and adjacency graph, but at the sacrifice of uniformity in the definition of DCJ.

### ***1.4.2 Uniform definition and 0 labels***

We can make DCJ uniform in the adjacency graph by adding structures dual to the caps and null chromosomes. The dual of a cap would be a dangling green line attached to a telomere point, like a road with a beginning and no end. Although this structure can lead to a correct uniform definition of DCJ, it would mar the visual elegance of the adjacency graph. Therefore we suggest the logically equivalent addition of a second label "0" to each telomere, representing the absence of a green line. (Figure 15e.) The "0" labels play the role of caps. We also introduce null points, to play the role of null chromosomes. A null point belongs to one genome and has two "0" labels and no green line attached. Every point now has two labels. For further illustration we present in Figure 16 the adjacency graph corresponding to

Figure 14. The corresponding master graph is shown in Figure 17.

The DCJ can now be performed by choosing any two points in the current genome, reshuffling the four labels attached to them, and reconnecting the green lines (however many there are) in accordance with the new labeling of the points. This is a uniform definition, and it yields all the special cases described in Table 1.

#### ***1.4.3 Paths and cycles, odd and even paths***

Like the edge graph, the adjacency graph consists of paths and cycles. The paths begin and end on telomeres. An AB path (odd) begins in genome A and ends in genome B. An AA path (even) begins and ends in genome A. A BB path (even) begins and ends in genome B.

It is possible to close the adjacency graph and use the distance formula  $d = N' - C'$  given in Section 1.3.5. To close an AB path, join the two endpoints (telomeres) by a green line. To close an AA path, introduce a null point into genome B and connect it by green lines to both ends of the AA path. To close a BB path, introduce a null point into genome A and connect it by green lines to both ends of the BB path. Then  $C'$  is the number of cycles including those formed from paths, and  $N'$  is the number of points in each genome, including null points.

#### ***1.4.4 Distance formula (Bergeron et al)***

Bergeron *et al* [4] have also given a distance formula that can be applied directly to the adjacency graph without introducing "0" labels, null points, or closure. They arrive at  $d = N - C - I/2$ , where  $N$  is the number of genes,  $C$  the number of cycles, and  $I$  the number of odd paths. This is a lower bound on the distance because no DCJ can do better than increase  $C$  by  $I$  or increase  $I$  by 2, but not both. To see that this lower bound can be achieved, see Section 3.10 where we give a sorting procedure that achieves it.

This formula can be proved equivalent to the one based on closure as follows. One has  $C' = C + P$  where  $P$  is the number of paths. One also has  $2N' = 2N + L_A + L_B + Z_A + Z_B$  where  $L_A, L_B$  are the number of linear chromosomes in each genome, and  $Z_A, Z_B$  are the number of null points introduced in each genome for closure. By counting telomeres one has  $P = L_A + L_B$  and also  $P = I + Z_B + Z_A$  since closure introduces one null point for each even path. Putting these equations together, one obtains  $N' - C' = N - C - I/2$ . The formula  $d = N - C - I/2$  can also be applied to the edge graph since  $N, C$ , and  $I$  are the same in both graphs.

### ***1.5 Linear chromosomes with restriction of circular intermediates***

Of biological interest is the case where only linear chromosomes are allowed in the initial and target genomes, and never more than one circular chromosome in the current genome. It has been shown [3] that in this case the DCJ distance is the same as though circular chromosomes were unrestricted. This case is equivalent to forbidding circulars altogether and allowing inversions, translocations, and block interchanges (with weight 2).

### ***1.6 Outline of procedures (to be found in section 3)***

**The interested reader may follow the procedures detailed in section 3 for the construction of the graphs that have been described or to develop algorithms for the distance or sorting by DCJ:**

- 3.1 Construction of black-white genome graph
- 3.2 Construction of white genome graph
- 3.3 Construction of edge graph
- 3.4 Construction of adjacency graph
- 3.5 From edge graph to adjacency graph
- 3.6 From adjacency graph to edge graph
- 3.7 Distance without linear chromosomes
- 3.8 Distance with or without linear chromosomes

3.9 Sorting without linear chromosomes

3.10 Sorting with or without linear chromosomes

## 2. Materials and Online Resources

The procedures in this chapter are of two kinds: constructing graphs that have been described in Section 1, and carrying out sorting algorithms to transform one genome into another by the minimum number of DCJ's.

### *2.1 Constructing graphs by hand or by computer*

The constructions can be done in two ways: by hand or by computer. Hand construction is easier unless one has facility with computer software for creating diagrams. Some of the constructions involve alterations of a graph, and this requires an excellent eraser or else the willingness to draw the graph from scratch after each alteration.

### *2.2 Online software for performing algorithms:*

#### *2.2.1 Applying DCJ to genome sequence data using Mauve*

The DCJ algorithm provides a measure of genomic rearrangement distance between genomes that have been coded as syntenic blocks. Identification of syntenic blocks among genomes remains a non-trivial task and is the subject of ongoing research and software development. As discussed in the chapter on GRIMM/MGR, GRIMM-Syntenic provides one mechanism for identifying syntenic blocks. Here we describe the Mauve genome alignment software [11], which can be used to both identify syntenic blocks and to compute pairwise DCJ distances among multiple genomes. Mauve is free, open-source software for Linux, Windows, and Mac OS X, available from <http://gel.ahabs.wisc.edu/mauve>.

Mauve creates syntenic blocks using an algorithm that identifies homologous tracts of sequence that are unique in each genome. First, Mauve identifies putatively homologous local multiple alignments [12]. Next, the local multiple alignments are clustered into groups that are free from rearrangement, called Locally Collinear Blocks (LCBs). Each LCB is assigned an *LCB weight* equal to the sum of lengths of the ungapped local alignments which comprise the LCB. Some LCBs may represent spurious homology predictions or paralogous

sequence, and it is necessary to filter out such LCBs. Such LCBs typically have low LCB weight relative to LCBs representing non-paralogous homology. Mauve discards all LCBs that have an LCB weight less than a user-specified threshold value in a process called greedy breakpoint elimination. For more details, refer to the algorithm descriptions in [11, 13].

### 2.2.2 Performing a DCJ analysis of *E. coli* and *Shigella* genomes with Mauve 1.3.0

Using the Mauve genome alignment software, one can generate synteny blocks and perform a DCJ analysis in a five step process:

1. Download and run Mauve from <http://gel.ahabs.wisc.edu/mauve>
2. Download whole genome sequence data in either Multi-FastA or GenBank format from NCBI. For this example we will use the genomes of five *Shigella* and *E. coli* strains. The genomes are available from:

[ftp://ftp.ncbi.nih.gov/genomes/Bacteria/Shigella\\_boydii\\_Sb227/NC\\_007613.gbk](ftp://ftp.ncbi.nih.gov/genomes/Bacteria/Shigella_boydii_Sb227/NC_007613.gbk)

[ftp://ftp.ncbi.nih.gov/genomes/Bacteria/Shigella\\_flexneri\\_2a\\_2457T/NC\\_004741.gbk](ftp://ftp.ncbi.nih.gov/genomes/Bacteria/Shigella_flexneri_2a_2457T/NC_004741.gbk)

[ftp://ftp.ncbi.nih.gov/genomes/Bacteria/Shigella\\_flexneri\\_5\\_8401/NC\\_008258.gbk](ftp://ftp.ncbi.nih.gov/genomes/Bacteria/Shigella_flexneri_5_8401/NC_008258.gbk)

[ftp://ftp.ncbi.nih.gov/genomes/Bacteria/Shigella\\_dysenteriae/NC\\_007606.gbk](ftp://ftp.ncbi.nih.gov/genomes/Bacteria/Shigella_dysenteriae/NC_007606.gbk)

[ftp://ftp.ncbi.nih.gov/genomes/Bacteria/Escherichia\\_coli\\_K12/NC\\_000913.gbk](ftp://ftp.ncbi.nih.gov/genomes/Bacteria/Escherichia_coli_K12/NC_000913.gbk)

3. Identify synteny blocks (LCBs) in Mauve
  - a. Select the “Align ...” option from the “File” menu. The “Align sequences...” dialog will appear, as shown in Figure 18.
  - b. In the dialog, add each genome sequence file that was downloaded from NCBI. Files can be dragged and dropped, or selected using the “Add sequence...” button, as shown in Figure 18, panel A. Also set the output file location. In this example, output will be written to the file C:\shigella\5way\_comparison

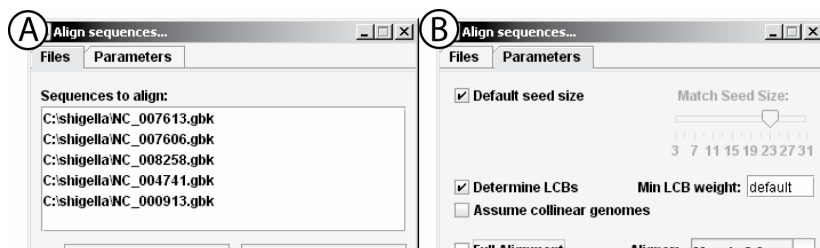
- c. Select the “Parameters” tab (Figure 18, panel B) and disable the “Extend LCBs” option. Then disable the “Full Alignment” option. The synteny block generation process will run much more quickly without performing a full genome alignment.
  - d. Click the “Align” button
4. Choose the appropriate minimum LCB weight. A slider control at the top-right of the Mauve window allows one to change the minimum LCB weight, shown in Figure 19. The default value is often too low and thus many spurious LCBs appear. By sliding the control to the right, a higher LCB weight can be chosen that better represents the true set of LCBs. For the present data we select a minimum LCB weight of 1147, which roughly corresponds to the average gene size in bacteria.
  5. Perform a DCJ analysis by clicking the DCJ button in the Mauve window (shown at top in Figure 19). The results appear as a textual table in a pop-up window. The table is tab-delimited text that can be copied-and-pasted into a program such as Microsoft Excel. The pairwise DCJ distances for the five *Shigella* and *E. coli* genomes are shown in Table 2:

	Sb227	Sd197	5 str. 8401	2a 2457T	K12
<i>S. boydii</i> Sb227	-				
<i>S. dysenteriae</i> Sd197	86	-			
<i>S. flexneri</i> 5 str. 8401	40	79	-		
<i>S. flexneri</i> 2a 2457T	44	79	14	-	
<i>E. coli</i> K12	28	65	16	18	-

**Table 2. Pairwise DCJ distances between genomes of *Shigella* and *E. coli*. as computed on the Mauve alignment with minimum LCB weight 1147.**

We see that in general, pairwise distances between *Shigella* spp. and *E. coli* K12 are lower than distances among pairs of *Shigella* genomes—a counterintuitive result for organisms from different genera. *Shigella* and *E. coli* were originally given different genus names because the diarrhoeal disease caused by *Shigella* had different symptoms than that caused by *E. coli*. *Shigella* and *E. coli* are now commonly considered as being members of the same bacterial “species”. Some strains of *Shigella* appear to have acquired an elevated rate of genomic rearrangement, resulting in the high DCJ distances between strains

**Figure 1**



Panel (A) shows the selection of five GenBank sequence files that correspond to four genomes of *Shigella* spp. and one *E. coli* (NC\_000913.gb).

Panel (B) shows selection of the alignment options. Note that the “Extend LCBs” option and the “Full



### 3. Procedures

#### 3.1 Construction of black-white genome graph

Construct a black-white genome graph with a given gene content.

Let the gene content be given in signed permutation form, thus  $[4, 1, -2] (-3, -5)$  signifies that there are two chromosomes, that the first chromosome is linear and reads from left to right as gene 4 pointing forward, gene 1 pointing forward, gene 2 pointing backward, and that the second chromosome is circular and reads counterclockwise as gene 3 pointing backward, gene 5 pointing backward.

- 1) For each circular chromosome of  $n$  genes, place  $2n$  points in a horizontal line.
- 2) Join these  $2n$  points by alternating white and black lines, starting with white. (The white lines may be left invisible.)
- 3) Join the first and last point by a black arc passing above.
- 4) For each linear chromosome of  $n$  genes, place  $2n+2$  points in a horizontal line. (The first and last point may be distinguished as caps.)
- 5) Join these  $2n+2$  points by alternating black and white lines, starting with black.
- 6) As a result of steps (2) and (5), every chromosome now has as many white lines as genes. The white lines represent the genes, reading from left to right as in the input description. They may be labelled as follows:
- 7) If the gene is positive (pointing forward), label the right hand end of the white line with the gene number followed by "h", and the left hand end with the gene number followed by "t". If it is negative (pointing backward) label the left hand end with the gene number followed by "h", and the right hand end with the gene number followed by "t". Caps are not labelled.

With the input example given above, the first chromosome is given  $2 \times 3 + 2 = 8$

points, the second is given  $2 \times 2 = 4$  points, and the labels of the 12 points from left to right (caps indicated by x) are x, 4t, 4h, 1t, 1h, 2h, 2t, x, 3h, 3t, 5h, 5t.

### ***3.2 Construction of white genome graph***

Construct a white genome graph with a given gene content.

Let the input be given as in 3.1.

- 1) For each circular chromosome of  $n$  genes, place  $n$  points in a horizontal line.
- 2) Join these  $n$  points by  $n-1$  white lines, and join the first and last point by a white arc passing above. (This white arc must be visible, at least until its endpoints have been labelled in step (6) below.)
- 3) For each linear chromosome of  $n$  genes, place  $n+1$  points in a horizontal line.
- 4) Join these  $n+1$  points by  $n$  white lines.
- 5) Each chromosome of  $n$  genes now has  $n$  white lines. Each white line will be labeled at each of its endpoints, so that the point of connection between two consecutive white lines will receive two labels. Label in accordance with the input description as follows:
- 6) For the white arc passing above a circular chromosome, label "in reverse". If the gene is positive, the left hand end receives the gene number followed by "h", and the right hand end receives the gene number followed by "t". If the gene is negative, the right hand end receives the gene number followed by "h", and the left hand end receives the gene number followed by "t".
- 7) For all other white lines, label "directly". If the gene is positive, the right hand end receives the gene number followed by "h", and the left hand end receives the gene number followed by "t". If the gene is negative, the left hand end receives the gene number followed by "h", and the right hand end receives the gene number followed by "t".
- 8) All points now have two labels, except the endpoints of linear chromosomes.

With the input example given in 3.1, the first chromosome will have  $3+1=4$  points, and the second will have 2 points. The labels of the six points will be (4t); (4h,1t); (1h,2h); (2t); (5t,3h);(3t,5h). (See Figure 15a.)

### ***3.3 Construction of edge graph***

Construct an edge graph, given the initial and target genomes.

Let the two genomes be given in signed permutation form, for example:

genome A (initial) = [4, 1, -2] (-3, -5)

genome B (target) = [1, 2] (3, 4, 5).

- 1) Construct the black-white genome graph of genome A, following 3.1.
- 2) Construct the black-white genome graph of genome B, but use gray lines instead of black.
- 3) For each gray line in the B graph, connecting two labeled points, find the corresponding labeled points in the A graph and connect them by a gray arc passing beneath the horizontal line.
- 4) For each gray line in the B graph connecting a labeled point to a cap, find the corresponding labeled point in the A graph and run a gray line vertically downward to a cap below.
- 5) Discard the B graph.
- 6) The white lines may now be deleted if desired.

The resulting elaborated A graph is the edge graph between genomes A and B. It should have  $N$  labeled points,  $2L_A$  caps terminating horizontal black lines, and  $2L_B$  caps terminating vertical gray lines, where  $N$  is the number of genes in each genome,  $L_A$  is the number of linear chromosomes in genome A, and  $L_B$  is the number of linear chromosomes in genome B.

Thus, for the input example given above, the edge graph has 10 labeled points,

$2x1 = 2$  caps terminating horizontal black lines, and  $2x1=2$  caps terminating vertical gray lines.

### ***3.4 Construction of adjacency graph***

Construct an adjacency graph, given the initial and target genomes.

Let the input genomes be given as for Procedure 3.

- 1) Construct the white genome graph for genome A, using 3.2. (See Figure 15a.)
- 2) Construct the white genome graph for genome B, directly below the A graph. (See Figure 15b.)
- 3) Join each label in the A graph to the corresponding label in the B graph by a green line. (See Figure 15c.)
- 4) Delete white lines. (See Figure 15d.)
- 5) If desired, add a "0" label to each telomere (point with only one label). (See Figure 15e.)

The resulting adjacency graph will have  $2N$  green lines, where  $N$  is the number of genes in each genome. For the input given before 3.3, there should be  $2 \times 5 = 10$  green lines.

### ***3.5 From edge graph to adjacency graph***

Given an edge graph, construct the corresponding adjacency graph.

We shall assume that the white lines are invisible in both graphs.

- 1) The edge graph is built on a horizontal level called the upper level. Also visualize a lower level, some distance below the upper level.
- 2) For each gray arc in the edge graph, mark a point on the lower level. Join this point by two green lines to each endpoint of the gray arc. Delete the gray arc.
- 3) Replace each vertical gray line running from a point on the upper level down to a cap by a vertical green line running from that point on the upper level down to a point on the lower level. Delete the cap.

- 4) Label each point on the lower level by the same labels as appear at the upper ends of the green lines connected to it. If there is only one green line, add a label "0" if desired.
- 5) On the upper level, contract each horizontal black line to a point. Let this point inherit the labels of the previous endpoints of the black line. If one endpoint was a cap, add a label "0" if desired, otherwise give the point only one label.
- 6) Treat the black arcs according to the same principle. This may be done by deleting the arc and its right hand endpoint, and transferring the green line and label of the deleted endpoint to the left hand endpoint, which is now the point of contraction.

The resulting structure is the adjacency graph. Every point should have two labels., if "0" labels were added.

### ***3.6 From adjacency graph to edge graph***

Given an adjacency graph, construct the corresponding edge graph.

- 1) If the adjacency graph has "0" labels, delete them.
- 2) Label each green line in the adjacency graph with the single label that is found on both its endpoints. If the two endpoints have both labels in common, there will be two green lines joining them; one green line should receive one label and one the other.
- 2) Draw out each point on the upper level (in genome A) into two points connected by a horizontal black line. Give each point one of the green lines connected to the original point, with the corresponding label. If the original point has only one green line, let one of the new points inherit the green line with its label, and the other be a cap.
- 3) Replace each pair of green lines having a common lower endpoint by a gray arc connecting the respective upper endpoints.
- 4) Replace each green line having only one lower endpoint by a vertical gray line running from the upper endpoint downward to a cap. The resulting structure is the edge graph.

### ***3.7 Distance without linear chromosomes***

Find the DCJ distance between two genomes, given the adjacency graph;

Restricted case: no linear chromosomes in either genome (no telomeres in the graph).

- 1)  $N$  is the number of genes in each genome, found by counting the points on either level.
- 2)  $C$  is the number of cycles in the graph, found by the following steps.
- 3) Start from any point and follow the green lines continuously, marking each point you reach, until you return to the starting point. You have traversed a cycle.
- 4) Start again from an unmarked point (if there is one) and traverse a second cycle by repeating step (3).
- 5) Repeat until all points are marked. The number of cycles traversed is  $C$ .
- 6) The distance is  $N - C$ .

### ***3.8 Distance with or without linear chromosomes***

Find the DCJ distance between two genomes, given the adjacency graph;

General case: linear chromosomes may be present.

- 1) If the number  $N$  of genes in each genome is not known, determine it by counting the green lines and dividing by 2.
- 2) Explore all the paths by the following steps.
- 3) Start at a telomere; it has only one green line attached. Move along the green lines continuously, marking each point you reach, until you reach another telomere. You have traversed a path.
- 4) If you began on one level and ended on the other, the path is odd. Keep count of the odd paths.
- 5) If there remains an unmarked telomere, start again at that point and traverse a second path. Repeat until all paths are traversed (no unmarked telomere).

- 6)  $I$  is the number of odd paths you have found. This number must be even.
- 7) The part of the graph remaining unmarked consists completely of cycles. Find  $C$ , the number of cycles, by applying steps (3-5) of 3.7.
- h) The distance is  $N - C - (I/2)$ .

### ***3.9 Sorting without linear chromosomes***

Perform a series of DCJs to transform a given initial genome consisting of circular chromosomes into a given target genome consisting of circular chromosomes.

- 1) Construct the adjacency graph between initial and target genome (Section 3.4). All points will be adjacencies (two green lines).
- 2) Proceed from left to right on the lower level. Choose the first point that is connected by green lines to two different points on the upper level. If there is none, sorting has been completed.
- 3) Cut the two points on the upper level that are connected to the chosen point.
- 4) Reconnect the four loose ends so that the two green lines from the chosen point are connected to one point on the upper level, and the other two green lines to the other point.
- 5) Repeat steps (2), (3), (4) until sorting is complete.

Each DCJ has increased the number of cycles by one. The final configuration consists completely of 1-cycles.

### **3.10 Sorting with or without linear chromosomes**

Perform a series of DCJs that transform a given initial genome of arbitrary type (circular and linear chromosomes both permitted) to a given target genome of arbitrary type.

- 1) Construct the adjacency graph between the initial and target genome (Section 3.4). Include “0” labels for telomere points (only one green line).
- 2) Proceed from left to right on the lower level. Choose the first point that is connected

by green lines to two different points on the upper level. If there is none, proceed to step (7).

- 3) Cut the two points on the upper level that are connected to the chosen point.
- 4) Shuffle the four labels on the two cut points so that the two labels corresponding to those on the chosen point are placed on one point on the upper level. Place the other two labels on the other point. If the latter are both "0", the resulting null point may be deleted.
- 5) Reconnect all loose ends of green lines according to the labels on the points.
- 6) Repeat steps (2-5) until every point on the lower level is joined to no more than one point on the upper level. Every DCJ so far has increased the number of cycles by 1, and has not changed the number of odd paths, although it may have shortened some. The graph now consists of (i) 1-cycles; (ii) AB paths (odd) of length 1; (iii) BB paths (even) of length 2.
- 7) Proceed from left to right on the upper level. Choose the first point that is connected by green lines to two different points on the lower level. (This configuration will be a BB path.) If there is none, sorting is complete.
- 8) Cut the chosen point and a null point introduced for this purpose on the upper level.
- 9) Shuffle the four labels so that the two "0" labels are on different points.
- 10) Reconnect the two loose ends of green lines. This has converted an adjacency (the chosen point) and a null point into two telomeres. The BB path has been split into two AB paths.
- 11) Repeat steps (7-10) until sorting is complete. Each DCJ in this second loop has left the number of cycles unchanged and has increased the number of odd (AB) paths by 2. Each DCJ in the whole procedure has increased the number  $C+I/2$  by 1, where

$C$ =number of cycles,  $I$ =number of odd paths. The final configuration consists of 1-cycles and AB paths of length 1. The labeled points on the upper level correspond exactly to those on the lower level.

An example of this procedure is shown in Figs 16 and 20.

## 4. Notes

**1.1** Homology is a binary characteristic. There is no such thing as 75% homologous, "more" homologous or "homologous depending on criteria". Given sequence data, we use computational methods determine whether a given genomic segment is homologous to another genomic segment. These computational methods use parameters and "criteria" to make predictions about homology. The true relationship of homology does not depend on those criteria, but our predictions do.

3.1, 3; 3.2, 2) In all these constructions we represent the genome graph as far as possible as laid out in a straight horizontal row. This can be done completely for linear chromosomes, but a circular chromosome must contain one arc that returns from the last point to the first. In 3.1 we have the option to choose either a black or a white line to be this arc. We choose a black arc because a white arc, if invisible, would be difficult to infer from the appearance of the graph, and so circular chromosomes would not be easily distinguished from linear. (In principle the distinction could be made by studying the labels.)

In 3.2 we have no choice but to use a white arc. Therefore the arc must not be deleted until its endpoints have been labeled.

3.1, 4) Caps seem to be necessary at least in the current genome if one wants a uniform definition of DCJ in terms of the edge graph. We have thought of making the linear chromosome "pseudocircular" by adding a black line to connect the endpoints, but marking this line to show that it is not really there. However, a DCJ made by cutting this line would

have ambiguous effect, because it would remain unspecified which end of the chromosome was to receive new material. Another possibility [14] is to omit the caps in the target genome; we have preferred to use them in both genomes for the sake of symmetry between the two.

3.2, 6) The rules for labeling can be understood by imagining a toy train proceeding counterclockwise along a closed track, left to right in the foreground and right to left on the return segment in the background. Some of the cars (positive genes) have been put on correctly, heads forward and tails behind; some (negative genes) have been put on backwards. In the foreground the positive cars have their heads to the right and tails to the left, and the negative (backwards) cars have heads left and tails right. In the background the train is headed right to left, therefore the positive cars have heads left and the negative cars have heads right. The white arc corresponds to the background portion of the track.

3.4, 5; 3.5, 4&5; 3.6, 1; 3.10, 1) The adjacency graph was presented Bergeron *et al* [3] without “0” labels, and for some purposes they are superfluous. Therefore they are made optional in 3.4 and 3.5. They only get in the way in 3.6, and so we delete them at the outset. They have no effect on the computation of distance, and so are not mentioned in 3.8. Their value is in sorting by DCJ, which becomes simpler if the operation is defined with only one universal case. Therefore we explicitly include them in 3.10.

3.5, 2) This step is ambiguous as to the placement of points on the lower level, and may therefore lead to an appearance of the adjacency graph different from that resulting from 3.2. But the connections will be the same.

3.8, 1) The number of genes can of course be determined by counting the white lines, but if they are not visible the easiest way may be by counting the green lines. Counting the points requires a distinction between adjacencies and telomeres.

3.8, 6) The number of AB paths,  $I$ , is always an even number, and therefore the distance formula  $N - C - I/2$  always gives an integer.  $I$  is even because the number of A-caps is equal to  $I$  plus twice the number of AA paths. Therefore if  $I$  were odd, the number of A-caps would be odd, which is impossible because this number is double the number of linear chromosomes in genome A.

3.11, 12) In sorting with linear chromosomes present, the elements to be expected when sorting is complete are 1-cycles and AB paths of length 1. Each of these structures involves one point in each genome, with identical labeling. A 1-cycle represents an adjacency common to both genomes. An AB path of length 1 represents a chromosomal endpoint common to both genomes.

## Acknowledgements

We are grateful to David Sankoff for his advice and encouragement; to Anne Bergeron for communicating to us the idea of the adjacency graph in advance of publication; to Mike Tsai for his online implementation of the DCJ; and to Betty Harris for invaluable logistic support and encouragement. S.Y. thanks Dr. Nicholas Chiorazzi for his enthusiasm, encouragement and support and A.E.D is supported by NSF grant DBI-0630765.

## REFERENCES

1. Nadeau, J H & Taylor, BA. 1984. Lengths of chromosomal segments conserved since divergence of man and mouse. *Proc. Natl Acad. Sci. USA* 81, 814–818.
2. Pevzner, P. A. (2000) *Computational Molecular Biology: an Algorithmic Approach*, MIT Press, Chapter 10.
3. Yancopoulos, S., Attie, O. and Friedberg, R. 2005. Efficient sorting of genomic permutations by translocation, inversion and block interchange. *Bioinformatics* 21, 3340 – 3346
4. Bergeron, A., Mixtacki, J. and Stoye, J. (2006) □A Unifying View of Genome Rearrangements. *WABI 2006*. pp. 163-173.
5. Christie, D.A. (1996) Sorting permutations by block interchanges. *Inform. Processing Lett.*, 60,165–169.
6. Lin, Y.C. et al. (2005) An efficient algorithm for sorting by block-interchanges and its application to the evolution of *Vibrio* species. *J. Comput. Biol.*, 12, 102–112.
7. Meidanis, J. and Dias, Z. (2001) Genome rearrangements distance by fusion, fission, and transposition is easy. In *Proceedings of SPIRE' 2001—String Processing and Information Retrieval*, Laguna de San Rafael, Chile, pp. 13–15.
8. Kececioğlu, J. and Sankoff, D. (1995) Exact and approximation algorithms for sorting by reversals, with application to genome rearrangement. *Algorithmica* 13, p. 180-210
9. Bergeron, A. (private communication.)
10. Bafna, V. and Pevzner, P.A. (1993) Genome rearrangements and sorting by reversals. In *Proceedings of the 34th Annual IEEE Symposium on Foundations of Computer Science*, IEEE

Press, pp. 148–157.

11. Darling ACE, Mau B, Blattner FR, Perna NT.(2004) Mauve: multiple alignment of conserved genomic sequence with rearrangements. *Genome Research* 14(7):1394-403.
12. Darling AE, Treangen TJ, Zhang L, Kuiken C, Messeguer X, Perna NT. (2006) “Procrastination leads to efficient filtration for local multiple alignment.” *Lecture Notes in Bioinformatics* 4175:126-137 Springer-Verlag.
13. Darling AE, Treangen TJ, Messeguer X, Perna NT. (2007) Analyzing patterns of microbial evolution using the Mauve genome alignment system. In *Comparative Genomics* (Bergman Eds.), Humana Press, In Press.
14. Hannenhalli, S. and Pevzner, P.A. (1995) Transforming men into mice (polynomial algorithm for genomic distance problem). In *Proceedings of the 36th Annual IEEE Symposium on Foundations of Computer Science*, Milwaukee, WI, pp. 581–592.

**Table 1. Outcomes of DCJ on linear & circular chromosomes**

	lines cut	1 or 2 chr	initial chromosome configuration	2 outcomes			
				operation	result	operation	result
1	int +int	1	C	fission	C C	reversal	C
2	int +int	1	L	(int) fission	C L	(int) reversal	L
3	int +int	2	C C	fusion	C	fusion	C
4	int +int	2	C L	(int) fusion	L	(int) fusion	L
5	int +int	2	L L	(reciprocal) translocation	L L	(reciprocal) translocation	L L
6	int + tel	1	L	(ext) fission	L C	(ext) reversal	L
7	int + tel	2	C L	(ext) fusion	L	(ext) fusion	L
8	int + tel	2	L L	(1-way) translocation	L L	(1-way) translocation	L L
9	tel + tel	1	L	conversion	C N	no change	L
10	tel + tel	2	L L	fusion	L N	no change	L L
11	int + null	2	C N	conversion	L	no change	C N
12	int + null	2	L N	fission	L L	same fission	L L
13	tel + null	2	L N	no change	L N	no change	L N
14	null + null	2	N N	no change	N N	no change	N N

**Legend:**

L = linear chromosome    C = circular chromosome    N = null chromosome  
chr = chromosome        tel = telomere                    int = internal line

**Table 1.** A DCJ on a given edge graph is defined by choosing two black lines to be cut and selecting one of two ways to rejoin the cut ends. The black lines are of three kinds: internal to a chromosome; telomere at the end of a chromosome, with one cap; and null, with two caps. Accordingly there are six possible cases for two black lines. For each case there are various subcases according to the number and type (circular, linear, null) of chromosomes initially containing the cut lines. For each subcase there are two outcomes, depending on the rejoining.

For each outcome we give the conventional name of the operation, and the subcase to which the final configuration belongs. For fission of L into CL, we distinguish between internal and external depending on whether the ejected fragment was an interior segment or an end segment of the original chromosome. Likewise for fusion of CL to L, whether the circular material is inserted into the linear chromosome or appended to the end; and for reversal within L, whether an interior or an end portion is reversed. For translocation we distinguish between reciprocal (exchange of end portions) and 1-way (transfer of end portion from one to another).

Operating on the last two subcases, tel + null and null + null, leads only to the initial configuration because all caps are indistinguishable. For the same reason, operating on tel + tel or int + null yields only one outcome different from the initial state.

## FIGURES

### Figure 1. Genome graph for a genome consisting of one circular chromosome.

- (a) signed labeling of each gene: a positive gene points counterclockwise around the circle.  
 (b) labeling of gene ends: the gene points from its tail (“t”) to its head (“h”).

QuickTime™ and a  
TIFF (Uncompressed) decompressor  
are needed to see this picture.

---

### Figure 2. Localized view of DCJ [1].

The letters a, b, c, d represent gene ends.  
 The black lines represent attachments between genes. The rest of the genome is not shown.

QuickTime™ and a  
TIFF (Uncompressed) decompressor  
are needed to see this picture.

---

### Figure 3. DCJ on the genome shown in Figure 1.

Signed labeling of genes. The black lines cut are between 3h and 2t, above; and between 5t and 4t, below. The four gene ends 3h, 2t, 5t, 4t are labeled respectively a, b, c, d to facilitate comparison with Figure 2. In the upper outcome, the chromosome is split into two. In the lower outcome, the segment -3,-5 is reversed as can be seen by untwisting the chromosome so that the order of genes becomes 4, 1, -2, 5, 3. The reader may wish to follow the steps by relabeling the figure as in Fig 1b.

QuickTime™ and a  
TIFF (Uncompressed) decompressor  
are needed to see this picture.

---

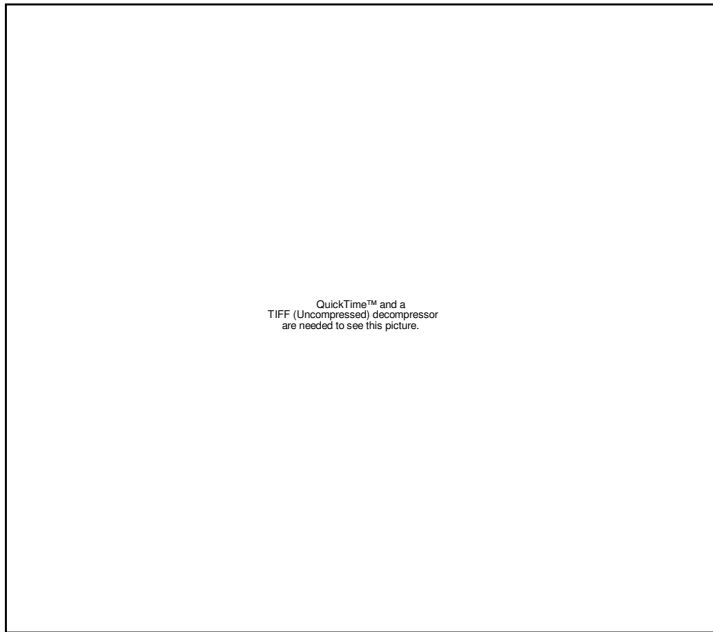
### Figure 4. Block interchange achieved by two successive DCJs.

(a) Fission of one chromosome into two, taken from the upper outcome of Figure 3.

(b) The two chromosomes resulting from (a) are cut in a different way and fused into one circular chromosome. The result differs from the starting configuration of (a) by interchanging -3 and 4.

QuickTime™ and a  
TIFF (Uncompressed) decompressor  
are needed to see this picture.

**Figure 5. Making an edge graph (only circular chromosomes).**

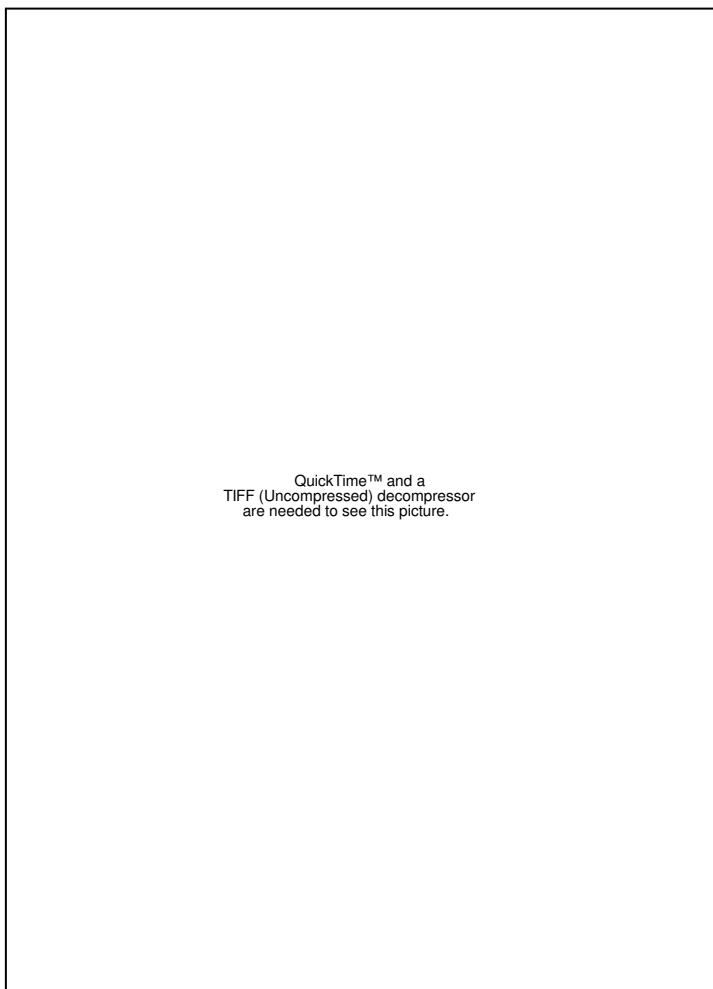


(a) Genome graph of the current genome, with gene ends labeled. This genome is identical with the upper outcome of Figure 3, namely  $(4,1,-2); (-3,-5)$ , but displayed in such a way that all the black lines are horizontal except one in each chromosome.

(b) Genome graph of the target genome,  $(1,2); (3,4,5)$ , similarly displayed.

(c) The edge graph formed from (a) and (b). All of (a) is retained, and the labeled points are connected below by gray arcs in accordance with the connections shown in (b). Note that the large black-gray cycle visits both chromosomes of each genome.

**Figure 6. Construction of an adjacency graph (only circular chromosomes).**



The current genome is  $(4, 1, -2); (-3,-5)$ .  
The target genome is  $(1, 2); (3, 4, 5)$ .

(a) Current genome graph with black lines shrunk to points, white lines shown as dotted, and gene end labeling.

(b) Target genome graph below (a).

(c) Same as (a, b) with corresponding gene ends joined by green lines.

(d) Same as (c) with white lines removed. This is the adjacency graph.

### Figure 7. Duality via the master graph.

The current genome is (3, 2, -4, 1).

The target genome is (1, 2, 3, 4).

At left is shown the usual edge graph, above; the same edge graph with points reordered to render the gray lines horizontal rather than the black lines, below; and the master graph, middle.

At right is shown the adjacency graph. “Green lines” in the master and adjacency graph are those that connect the upper half with the lower. To go from the master to either version of the edge graph, contract the green lines. To go from the master to the adjacency graph, contract the black and gray lines.

QuickTime™ and a  
TIFF (Uncompressed) decompressor  
are needed to see this picture.

QuickTime™ and a  
TIFF (Uncompressed) decompressor  
are needed to see this picture.

QuickTime™ and a  
TIFF (Uncompressed) decompressor  
are needed to see this picture.

### Figure 8. DCJ acting on a cycle [1].

The cuts are shown as in Figure 2, but the dashed arcs here represent the rest of the cycle, composed of black and gray lines, not the rest of the chromosome as is depicted in Figure 3. This figure gives no indication of chromosomal structure. The upper outcome represents fission of a cycle, and the lower outcome reversal of part of a cycle, but there is no way to tell whether either outcome represents fission of a chromosome, fusion of two chromosomes, or reversal within a chromosome.

### Figure 9.

(a) Sorting of an edge graph. At left, the edge graph of Figure 5c with labels deleted. At right, the edge graph obtained by modifying the current genome until it matches the target genome. The gray arcs are the same as at left, but the black lines have been redrawn so as to mirror the gray arcs. All black lines are now shown as arcs above the points.

(b) Cycle decomposition. The points of each graph in (a) have been shuffled so as to exhibit the cycles separately. There are 2 cycles in the graph at left, and 5 in the sorted graph at right; therefore the sorting requires a minimum of  $5-2=3$  DCJ steps.



**Figure 10. Removal of a 1-cycle in an edge graph of circular chromosomes.**

(a) The edge graph of Fig 5c, with labels removed. There are  $N=5$  genes (hence 5 black lines) and  $C=2$  cycles.

(b) Same as (a) except that the 1-cycle has been removed. Now there are  $b=4$  black lines (breakpoints) and  $c=1$  cycle. The DCJ distance between the two genomes is  $5-2 = 4-1 = 3$ .

**Figure 11. Genome graphs for circular and linear chromosome.**



(a) The circular chromosome (4,1,-2) displayed as in the left hand part of Fig 5a, but with signed labeling of genes. There are 3 genes and 3 black lines.

(b) The linear chromosome [4,1,-2] (we use [brackets] for linear, (parentheses) for circular chromosomes) displayed in the same way. The difference from (a) is the absence of the arc above. There are 3 genes and 2 black lines.

(c) The same chromosome as in (b), displayed with caps. There are 3 genes and 4 black lines.

**Figure 12. DCJ on a linear chromosome (same as Fig 11c) with cutting of capped line(s).**



(a) One capped line and one uncapped line are cut. Outcomes are fission into a linear and a circular, and reversal of the 4,1 segment.

(b) Two capped lines are cut. Outcomes are conversion to circular chromosome (same as Fig 11a) and reversal of the entire chromosome; the latter is no change. There is a null chromosome (black line bound by to “x”-caps) in the upper outcome.

**Figure 13.** The five possible “DCJ triangles” analogous to Figure 3, sorted by number and type of chromosome participating.

QuickTime™ and a  
TIFF (Uncompressed) decompressor  
are needed to see this picture.

In (c) and (d) the third member of the triangle is not shown because it differs from the left hand member shown only by the exchange of two indistinguishable caps.

For (a) see Table 1, rows 1 and 3; for (b), rows 2, 4, 6 and 7; for (c), rows 10 and 12; for (d), rows 9 and 11; for (e), rows 5 and 8.

**C = circular    L = linear    N = null.**  
**fi = fission    fu = fusion**  
**rv = reversal    tr = translocation,**  
**cv = conversion**

QuickTime™ and a  
TIFF (Uncompressed) decompressor  
are needed to see this picture.

**Figure 14.** A complex edge graph.

The current genome is [2,-3]; (1,4); (5, -6).  
 The target genome is [1, 2, 3]; [4]; (5); (6).  
 There are two AB paths (odd),  
 one BB path (even) and one cycle.

**Figure 15.** Construction of an adjacency graph involving circular and linear chromosomes.

Obtained from Fig. 6 by making two of the chromosomes linear.

Current genome is [4, 1, -2]; (-3, -5).

Target genome is (1, 2); [3, 4, 5].

(a) Current genome above

(b) target genome below.

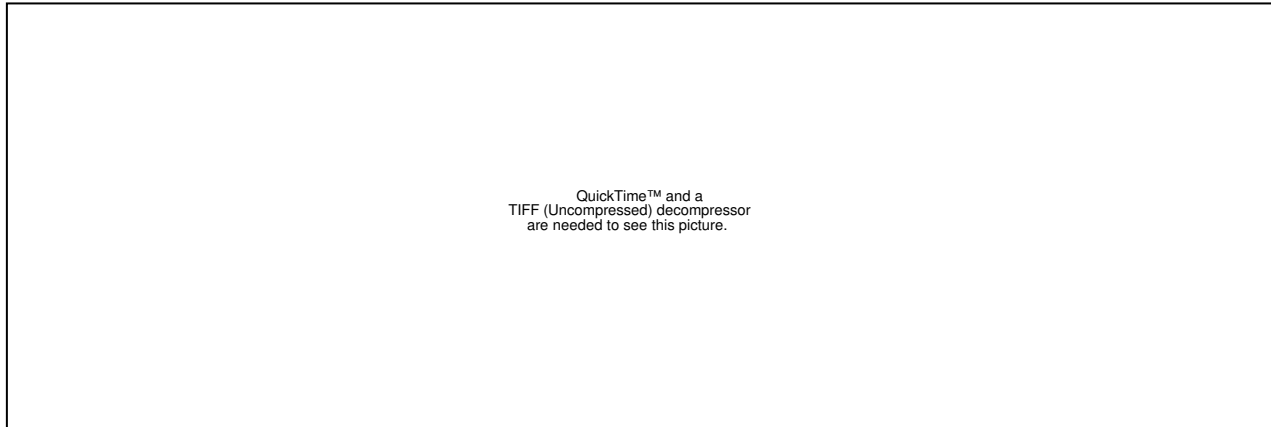
(c) Same as (a,b) with green lines added.

(d) Same as (c) with white lines deleted.

(e) Same as (d) with “0” labels added to telomeres.

The two cycles in Fig. 6 are replaced here by even paths.

QuickTime™ and a  
TIFF (Uncompressed) decompressor  
are needed to see this picture.



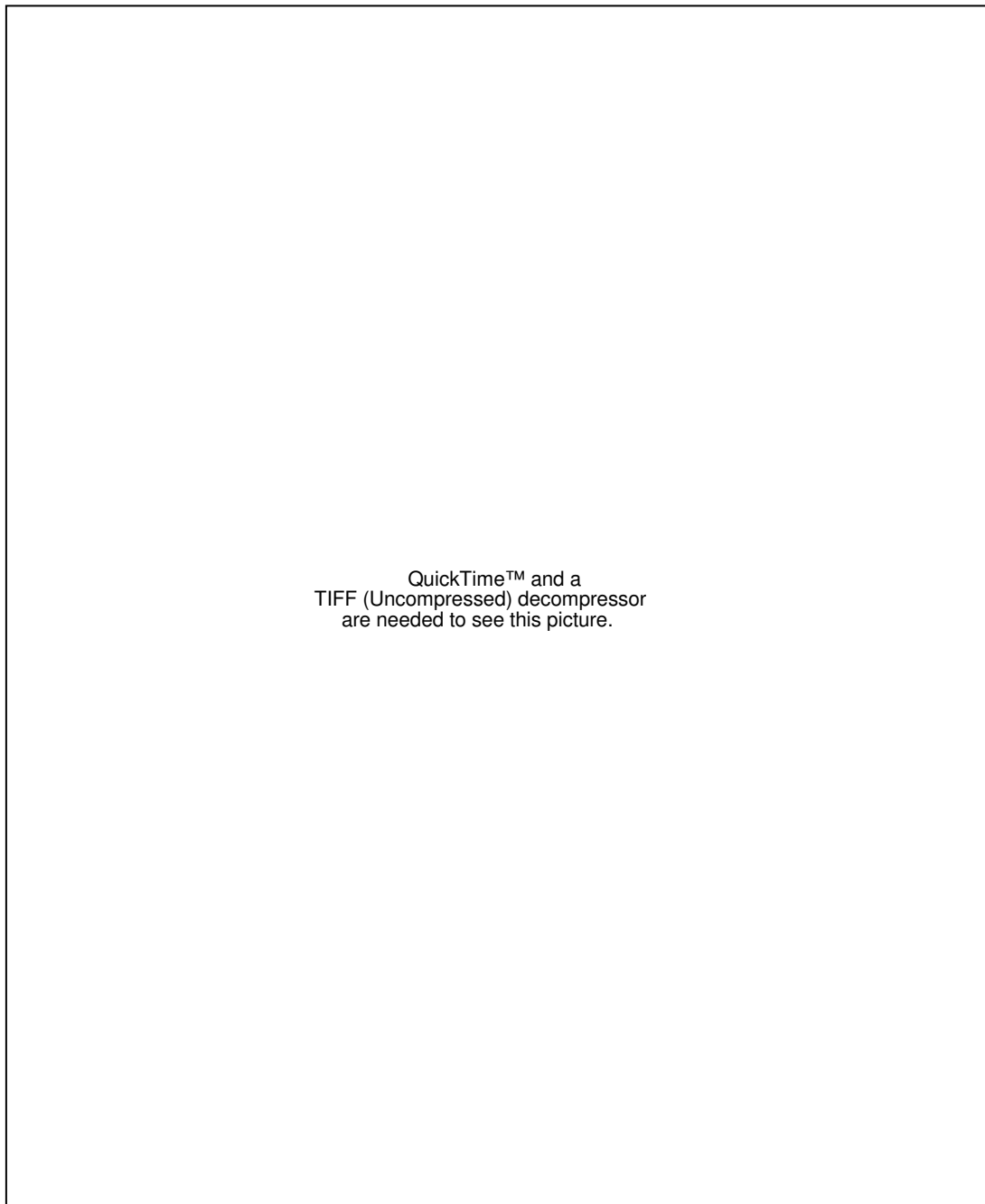
**Figure 16.** The adjacency graph corresponding to Figure 14.

Current genome [2,-3]; (1, 4); (5, -6). Target genome [1, 2 ,3]; [4]; (5); (6). “0” labels have been added.



**Figure 17.** The master graph for Figs. 14 and 16.

Contract green lines to get Figure 14. Contract black and gray lines to get Figure 16.



**Figure 20. Sorting of Fig. 16 by DCJ.** Procedure as given in 3.10.

(a) Start in Fig. 16 from 1h2t below, cut 02t and 1h4t above. Operation is fusion of circular and linear.

New genome is [4, 1, 2, -3]; (5, -6).  $\Delta C = 1$ ,  $\Delta I = 0$ .

(b) Start in (a) with 2h3t below, cut 2h3h and 3t0 above. Operation is reversal.

New genome is [4, 1, 2, 3]; (5, -6).  $\Delta C = 1$ ,  $\Delta I = 0$ .

(c) Start in (b) from 5h5t below, cut 6h5t and 5h6t above. Operation is fission of circular.

New genome is [4, 1, 2, 3]; (5); (6).  $\Delta C = 1$ ,  $\Delta I = 0$ .

(d) Start in (c) from 4h1t above, cut 4h1t and 00 (null point has been introduced for this purpose). Operation is fission of linear to two linears. New genome is [1, 2, 3]; [4]; (5); (6), identical to target genome.  $\Delta C = 0$ ,  $\Delta I = 2$ .