All Possible Optimal Solutions for Multichromosomal Genome Rearrangements using DCJ

Oshi Taneja

Department of Computer Science,
Birla Institute of Technology, Mesra, Jharkhand, India,
tanejaoshi@gmail.com

Abstract

The double cut and join (DCJ) operation, introduced by Yancopoulos et al., allows the representation of rearrangement operations that can occur in multichromosomal genomes namely as reversals/inversions, fissions, fusions, translocations, excisions, integrations, circularizations and linearizations with no restriction on the genome structure i.e. considering the both linear and circular chromosomes. Several works concerning DCJ operations have been published and in particular, an algorithm was proposed to find one optimal DCJ rearrangement sequence for sorting one genome into the another one. Here we study the whole space of solutions for this problem and give some propositions, on the basis of which a theoretical algorithm has been proposed.

1. Introduction

Genome Rearrangements proves be a greatest tool for tracking evolutionary scenarios of the living species at the genome level. The approach is the determination of the minimum number of rearrangements events/operations**[2]** required to transform one genome into another genome. This value is also known as the genomic distance. The algorithms available will give only one rearrangement sequence out of a possibly large number of rearrangement scenarios. It becomes very tedious when in addition to genomic distance, more than one scenarios of rearrangements are to be predicted and founded.

I. Genome representation

A multichromosomal genome $\prod$ over a set of markers π is a collection of linear or/and circular chromosomes in which each marker in π is a gene, a DNA (deoxyribonucleic acid) fragment & has an orientation and it occurs exactly once in $\prod$.

$\forall a \in \pi$, is represented as a non-zero integer whose extremities **[2]** can be defined as “-a” (tail) and as“+a” or “a” (head).

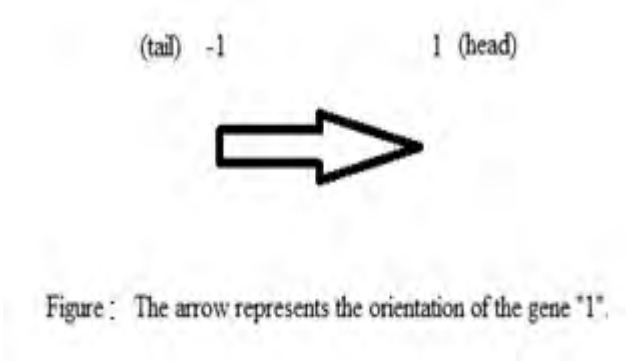

Figure 1

The adjacency **[2]** between two markers (e.g. x and y where x,y are two genes & both are positive integers) in $\prod$ is represented by an unordered pair containing the extremities of both x & y that are adjacent.

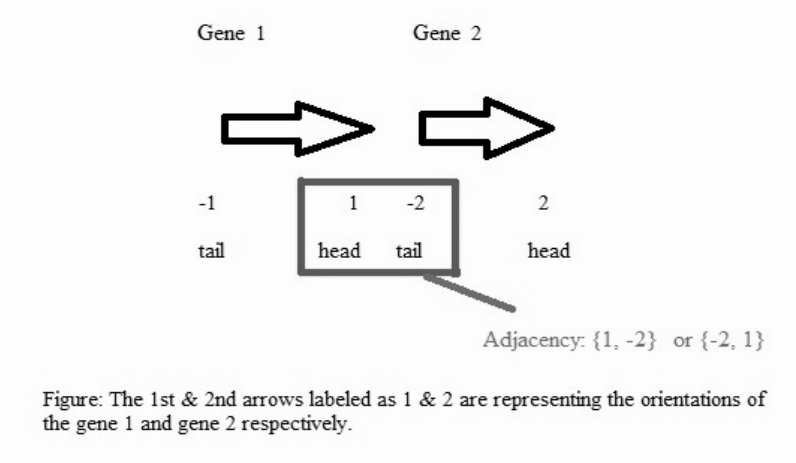

Figure: The 1st & 2nd arrows labeled as 1 & 2 are representing the orientations of the gene 1 and gene 2 respectively.

Figure 2

The telomere is referred to that extremity of a marker which is not the part of any adjacency of the chromosome and can also be termed as singleton (i.e. contains only one extremity instead of a pair).

Example: ∏= 1 -3 2 ; 4 -6 5

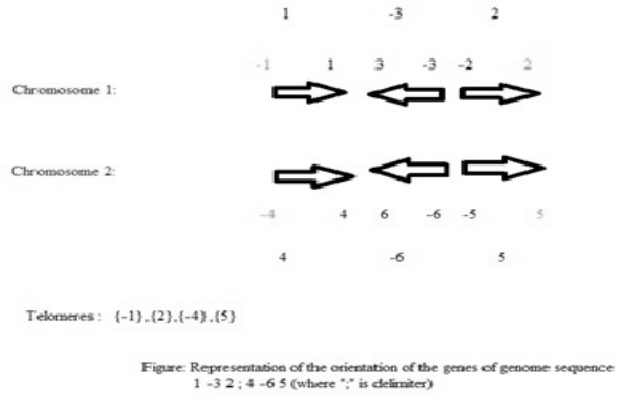

Figure: Representation of the orientation of the genes of genome sequence 1 -3 2 ; 4 -6 5 (where ";" is delimiter)

Figure 3

Thus, a genome ∏ can be represented by the set **[1]** V (∏) containing its adjacencies & telomeres. Using this set V (∏), an adjacency graph **[2]** can be constructed for both genomes (one is source genome ∏1 and another is the sorted sequence of genes termed as the target genome ∏2).

Definition:

Given two genomes ∏1 and ∏2 over the same set of markers ╥ (i.e. both contains same gene content) and without duplications. The adjacency graph AG (∏1, ∏2) is a graph in which

1) Set of vertices, V= V (∏1) U V (∏2)
2) ∀a ∈ ╥, there exists an edge connecting the vertices in V (∏1) to the corresponding V (∏2).

2. Whole solution space for Sorting

In accordance with the rules **[2]** and based on propositions for optimal DCJ sorting (**[1], page 4)**, sorting a genome ∏1 (source) into a genome ∏2 (target), the newobservation deduced in order to find all possible optimal rearrangement sequences is as follows:

Proposition 1. *Sorting the source genome with respect to each target genome's unsorted adjacencies & telomeres by using double cut and join can work successfully to give the one or more optimal sequences only when the exchange operations between adjacency-adjacency or adjacency-telomere is valid.*

Note: Unsorted adjacencies & telomeres in the source genome are those which are not in the proper places with respect to the target genome.

Proof.

The term valid means that no resulting adjacency should get the same magnitude (i.e. here both numeric values in resultant are same irrespective of their sign)after the exchange operation(**[1], page 3).** For e.g. the resultant adjacency {-2, 2} after exchange operation is invalid. If this type of invalidity occurs, we can never get the optimal DCJ rearrangement sequence (because the adjacency in DCJ optimal sorting can never contain the same magnitude values as there is non-duplication of genes) or it may lead to an undesired target. This leads to the conclusion that there exists a validity constraint to perform double cut and join operation (It means that we cannot blindly apply double cut and join anywhere in the genome sequence).

This proposition will work in the direction of finding all possible DCJ rearrangement sequences because only unsorted adjacencies & telomeres of target are taken into account while performing sorting & whenever one unsorted adjacency or telomeres are sorted, it always increases either the number of cycles or number of odd paths, thus always reducing the dDCJ distance (double cut and join distance) **[2]**. In other words, exchanges are made or predicted (like in proposed algorithm discussed later in this paper) only on those positions where adjacencies & telomeres are unsorted, i.e. a telomere or an adjacency in source genome which is not equivalent to any of adjacencies or telomeres in the target genome and its validity constraint is verified.

3 .Proposed Algorithm for whole space of optimal solutions using DCJ sorting

Let the sequence of source genome be 'S' and the set of all intermediate genome sequences at hierarchy/phase level denoted as 'x' be 'I ' and the set of all independent components at hierarchy/phase level denoted as 'x' be 'IC '.

(Note: "components" in the independent components are unsorted adjacencies and/or telomeres.)

Initially I and IC are both null and x=0(root).

1) Consider all the possible exchanges of adjacencies/telomeres in S or i (where iϵI) with respect to target and store their respective positions.
2) In accordance with their positions calculate the independent components & add them to IC where 'x' value is of current level.
3) With respect to each of independent components obtained in step 2, calculate the next set of intermediate genome sequences for next hierarchy/phase level and add them to 'Ix+1'.
4) Repeat steps 1-3 for all other z (where z ϵI) until all are exhausted. Skip this step if x=0.
5) x←x+1
6) Repeat all of the above steps until no possible exchanges are possible. (Which means all elements of both I and IC for all x have been exhausted).
7) Using this hierarchy, traversal from phase level 0(root) to the last level (gives one optimal solution),subsequently traversing structure from left to right one can get all possible solutions using DCJ operations.(Note: while traversing consider only those paths i.e. root to last level/leaf node as optimal solutions whose level number is equal to DCJ distance **[2]**.)

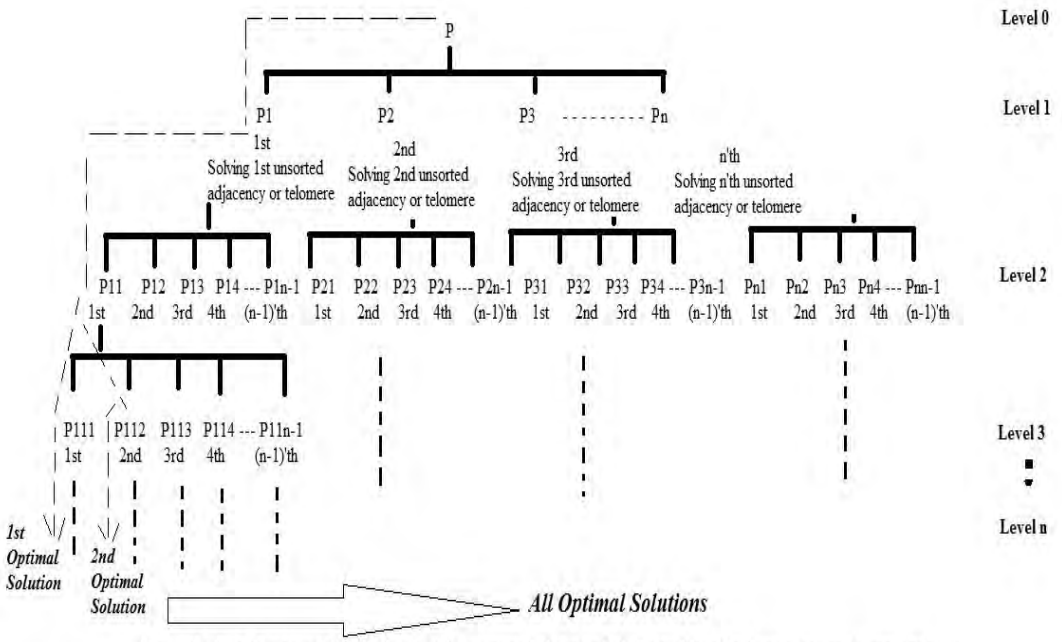

Figure: A pictorial representation for the algorithm presented above to find all possible optimal rearrangement sequences using DCJ. Here P represents a set of unsorted adjacencies & telomeres (of the source genome with respect to the target genome) and Pi or Pii or Piii or Piii.... represents a set of unsorted adjacencies & telomeres (of the intermediate genome with respect to the target genome) where i=1,2,3,4.....
n= No. of unsorted adjacencies & telomeres in P

Figure 4

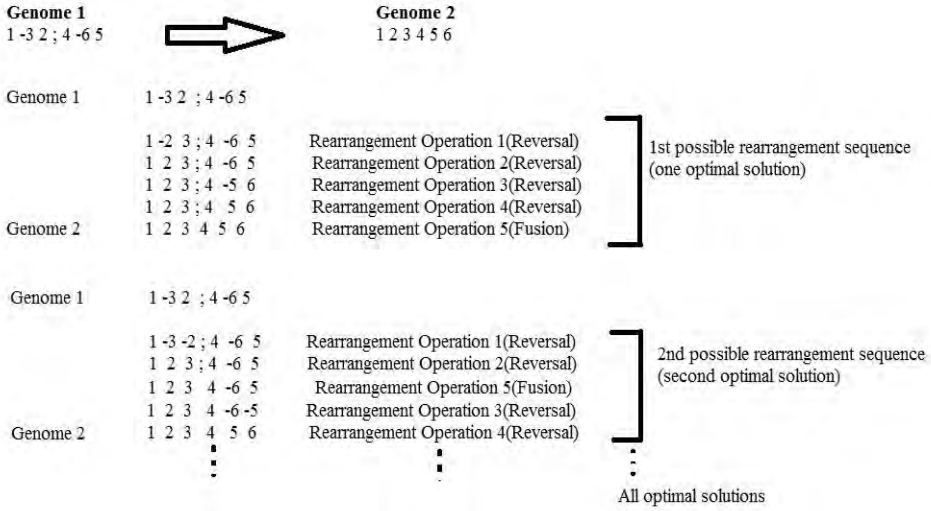

Figure : An example of transforming Genome 1(source genome) to Genome 2(target Genome) whose dDCJ (i.e. double cut and join distance [***refer to reference no [2] in this paper***]) is 5.

Figure 5

Final Remarks

An approachto work towards solving an open problem of finding the all possible optimal rearrangement sequences for a given genome sequence using double cut and join discovered. A theoretical algorithm has been proposed to achieve this purpose. This technique can work for the most of the cases to find out the whole space of optimal solutions using double cut and join. In addition to this, anothernew thing came to the surface is that it is not necessary that the double cut and join operation can be applied anywhere on the genome sequence i.e. one may not apply double cut and join at some places in the genome sequence in some cases due to invalid exchanges in the genome sequence.

References

[1] Mar´ılia D. V. Braga and Jens Stoye, Counting All DCJ Sorting Scenarios,TechnischeFakult¨at, Universit¨at Bielefeld, Germany,2009

[2] The Double Cut And Join Operation And Its Applications To Genome Rearrangements,JuliaZakotnikGeb. Mixtacki,Gedruckt auf alterungsbest¨andigemPapier – ISO 9706,April 2008

[3] Sridhar hannenhalli and pavel a. Pevzner, Transforming Cabbage into Turnip: Polynomial Algorithm for Sorting Signed Permutations by Reversals, Journal of the ACM, Vol. 46, No. 1, January 1999, pp. 1–27.

[4] S. Hannenhalli and P. A. Pevzner. Transforming men into mice (polynomial algorithm for genomic distance problem). In Proceedings of the 36th IEEE Sym- posium on Foundations of Computer Science (FOCS 1995), pages 581–592. IEEE Comuter Society Press, 1995.

[5] G. Tesler. Efficient algorithms for multichromosomal genome rearrangements. Journal of Computer and System Sciences, 65(3):587–609, 2002.

[6] G. Tesler. GRIMM: Genome rearrangements web server. Bioinformatics, 18(3):492–493, 2002.

[7] Glenn Tesler GRIMM: Genome Rearrangements Web Server Department of Computer Science and Engineering, University of California, San Diego, CA 92093-0114

[8] G. Bourque, P. A. Pevzner, and G. Tesler. Reconstructing the genomic architecture of ancestral mammals: lessons from human, mouse and rat genomes. Genome Research, 14(4):507–516, 2004.

[9] G. Bourque and P. A. Pevzner. Genome-scale evolution: Reconstructing gene orders in the ancestral species. Genome Research, 12(1):26–36, 2002.

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

[11] Braga M. D. V., Sagot M.-F., Scornavacca C. and Tannier E.: Exploring the solution space of sorting by reversals with experiments and an application to evolution. IEEE/ACM Trans. Comput. Biol. Bioinf. 5(3), 348–356, 2008. (Preliminary version in Proceedings of ISBRA 2007, LNBI 4463, 293–304, 2007).