A Lin-Kernighan Heuristic for the DCJ Median Problem of Genomes with Unequal Contents

Zhaoming Yin[2], Jijun Tang[1,3,⋆], Stephen W. Schaeffer[4], and David A. Bader[2,⋆]

[1] School of Computer Science and Technology, Tianjin University, China
[2] School of Compuational Science and Engineering,
Georgia Institute of Technology, USA
[3] Dept. of Computer Science and Engineering, University of South Carolina, USA
[4] The Huck Institutes of Life Sciences, Pennsylvania State University, USA

Abstract. In this paper, we designed a distance metric as *DCJ-Indel-Exemplar* distance to estimate the dissimilarity between two genomes with unequal contents (with gene insertions/deletions (*Indels*) and duplications). Based on the aforementioned distance metric, we proposed the *DCJ-Indel-Exemplar* median problem, to find a median genome that minimize the *DCJ-Indel-Exemplar* distance between this genome and the given three genomes. We adapted *Lin-Kernighan* (*LK*) heuristic to calculate the median quickly by utilizing the features of adequate subgraph decomposition and search space reduction technologies. Experimental results on simulated gene order data indicate that our distance estimator can closely estimate the real number of rearrangement events; while compared with the exact solver using equal content genomes, our median solver can get very accurate results as well. More importantly, our median solver can deal with *Indels* and duplications and generates results very close to the synthetic cumulative number of evolutionary events.

Keywords: Genome Rearrangement, Double-cut and Join (*DCJ*), Lin-Kernighan Heuristic.

1 Introduction

Inferring phylogenies (evolutionary history) of a set of given species is a fundamental problem in computational biology [23]. For decades, biologists and computer scientists have studied how to infer phylogenies by the measurement of genome rearrangement events using gene order data [13]. While evolution is not an inherently parsimonious process, maximum parsimony (*MP*) phylogenetic analysis has been widely applied to the phylogeny inference to study the evolutionary patterns of genome rearrangements. Given the input of gene order data with unequal contents (with gene insertions/deletions and duplications of genes), even the computation of distance between two genomes with only duplications is **NP**-hard [7,9,10] and **APX**-hard [1,11] by various distance measurement methods. There are attempts to perform phylogenetic reconstruction from

[⋆] Corresponding authors.

Z. Cai et al. (Eds.): COCOON 2014, LNCS 8591, pp. 227–238, 2014.

genome rearrangement data with unequal gene content, which can be roughly divided into distance-based methods [28], *MP* methods [29] and adjacency-based methods [17]. However, the first two approaches are generally quite limited by methods in distance and median computation.

Various distance metrics have been proposed to calculate the dissimilarity between two genomes, such as breakpoint distance [4], signed reversal distance [2], translocation distance [15], and Double-cut-and-join (*DCJ*) distance [34], which is currently the most extensively studied. However, there are still a lot of unclear subjects in distance computation between unequal content genomes, and computational biologists tried multiple ways to surpass this limit. Traditional approaches are based on breakpoint or reversal distances, such as efforts of employing exemplar distance [21, 25] to keep only one copy of duplicated gene families, or the methods by extending polynomial time reversal distance algorithm introduced by Hannenhalli Pevzner (*HP*), to handle *Indels* as well as duplications [18]. Contemporary research focusing on unequal contents are more concerned on *DCJ* model: For genomes with *Indels* only, there are exact algorithms to compute their *DCJ* distance [6, 12]; For genomes with duplications, there are several very useful methods to approximate or compute the exact *DCJ* distance [26, 27]. However, there are few efforts to combine these methods to measure distance of genomes with gene orders that contain both *Indels* and duplications.

The median problem is defined as to find a genome that minimizes sum of distances from itself to the three input genomes [5,19]; it's **NP**-hard under most distance metrics [3, 8, 22, 31]. Several exact algorithms have been implemented to solve the *DCJ* median problems on both circular [31, 33] and linear chromosomes [30, 32]. Some heuristics are introduced to improve the speed of median computation, such as linear programming (*LP*) [8], local search [16], evolutionary programming [14], or simply searching on one promising direction [24]. As all these algorithms are intended for solving the median problems with equal content genomes, their usage is limited in practice.

2 Background

2.1 Genome Rearrangement Events and Their Graph Representations

Genome Rearrangement Events. The content of the *DNA* molecules are often similar, but their organizations often differ dramatically. The mutation that affect the organization of genes are called genome rearrangements. Fig 1 shows examples of different rearrangement events of a single chromosome. In the examples, we use signed numbers to represent different genes and their orientation in the genome strand. Genome rearrangements events involve with multiple combinatorial optimization problems, and graph representation is a very common way to abstract these problems. In this part, we will address the foundations of using breakpoint graph to model the genome rearrangement events.

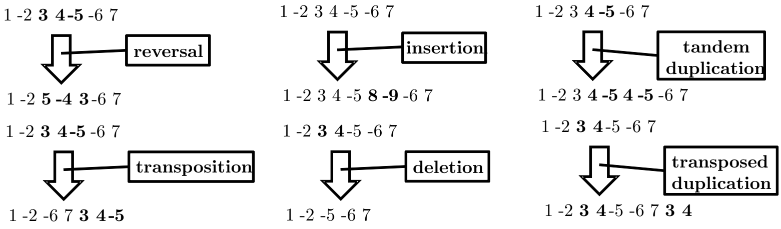
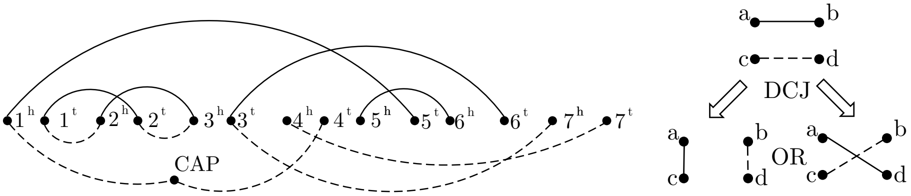


Fig. 1. Example of different rearrangement events



(a) Example of *BPG*

(b) Example of *DCJ*

Fig. 2. Examples of *BPG*; and *DCJ* operations

Breakpoint Graph. Given an alphabet $\mathcal{A}$, and two genomes Γ and Π are represented by two strings of signed (+ or −) symbols (representing genes) from $\mathcal{A}$. Each gene $a \in \mathcal{A}$ is represented by a pair of vertices head a_h and tail a_t, if a is positive a_h is putted in front of a_t, otherwise a_t is putted in front of a_h. For $a \in \mathcal{A}$ and $b \in \mathcal{A}$, if $a \in \Gamma$ (or Π) and $b \in \Gamma$ (or Π), and in Γ (or Π) a and b are adjacent to each other, their adjacent vertices will be connected by an edge. As for telomere genes, if they exist in a circular chromosome, two end vertices will be connected by an edge, and if they exist in a linear chromosome, two end vertices will be connected to a special vertex called *CAP* vertex. If we use one type of edges to represent adjacencies of Γ and another type of edges to represent adjacencies of Π, the resulting graph with two types of edges is called breakpoint graph (*BPG*). Fig 2(a) shows the *BPG* for gene order Γ (1,-2,3,-6,5) (solid edges) which is a genome with one circular chromosome and Π (1,2,3,7,4) (dashed edges) which is a genome with one linear chromosome.

***DCJ* Operation.** Double-cut and join (*DCJ*) operations are able to simulate all aforementioned rearrangement events applying *BPG*. The operations cut two edges (within one genome) and rejoin them using two possible combinations of end vertices (shown in Fig 2(b)). *DCJ* distance of genomes with the same content can be easily calculated by enumerating the number of cycles/paths in the *BPG*, which is linear [34]. Comparing with the complex model based on reversal operations, *DCJ* operations are simple and powerful.

2.2 Distance Computation

In the *BPG* with two genomes Γ and Π, the vertices and the edges of a closed walk form a cycle. In Fig 2(a), the walk $(1^t, (1^t; 2^h), 2^h, (2^h; 3^h), 3^h, (3^h; 2^t), 2^t, (2^t; 1^t), 1^t)$ is a cycle. A vertex v is *π-open* (*γ-open*) if $v \notin \Gamma$ ($v \notin \Pi$). An unclosed walk in *BPG* is a path. Based on different kinds of end points of the paths, we can classify paths into different types. If the two ends of a path are *CAP* vertices, we simply denote this path as p^0. If a path is ended by one open vertex and one *CAP*, we denote it as p^π (p^γ). If a path is ended by two open vertices, it is denoted by the type of its two open vertices, for example, $p^{\pi,\gamma}$ represent a path that ends with a *π-open* vertex and a *γ-open* vertex. In Fig 2(a), the walk $(5^t, (5^t; 1^h), 1^h, (1^h; CAP), CAP)$ is a p^γ path, and the walk $(6^t, (6^t; 3^t), 3^t, (3^t; 7^h), 7^h)$ is a $p^{\gamma,\pi}$ path. A path is even (odd), if it contains even (odd) number of edges. In [12], the *DCJ* distance between two genomes with *Indels* but without duplications is calculated by equation (1). We call this distance *DCJ-Indel* distance. From this equation, we can easily get the *DCJ-Indel* distance between Γ and Π in Fig 2(a) as 4.

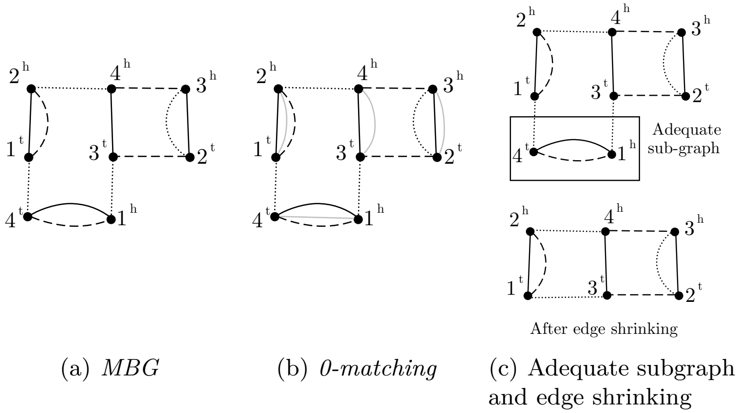
$$\begin{aligned} distance_{indel}(\Gamma, \Pi) = N - [c + p^{\pi,\pi} + p^{\gamma,\gamma} + \lfloor p^{\pi,\gamma} \rfloor] \\ + \frac{1}{2}(p^0_{even} + min(p^\pi_{odd}, p^\pi_{even}) + min(p^\gamma_{odd}, p^\gamma_{even}) + \delta) \end{aligned} \qquad (1)$$

Where $\delta = 1$ only if $p^{\pi,\gamma}$ is odd and either $p^\pi_{odd} > p^\gamma_{even}, p^\gamma_{odd} > p^\gamma_{even}$ or $p^\pi_{odd} < p^\gamma_{even}, p^\gamma_{odd} < p^\gamma_{even}$; Otherwise, $\delta = 0$.

There are in general two approaches to cope with duplicated genes. One is by removing all but keeping one copy of duplications in gene family to generate an exemplar pair [25] and another is by relabling duplicates such that all duplicated genes will have an unique label [26,27]. Lastly, mathematically optimized distance might not reflect the true number of biological events, distance estimation methods such as *EDE* or *IEBP* are used to rescale these computed distances [20].

2.3 Median Computation

If there are three given genomes, the graph constructed by borrowing the previous defined rule in *BPG* is called Multiple Breakpoint Graph (*MBG*). Figure 3(a) shows an example of *MBG*, With the input of three genomes: (1,2,3,4) (solid edges); (1,2,-3,4) (dashed edges) and (2,3,1,-4) (dotted edges). The *DCJ* median algorithm can be briefly described by a branch and bound (*BnB*) process [30,31,33] on *MBG*, which is to find a maximum matching (which is called 0-*matching*) in *MBG*. Figure 3(b) shows an example of 0-*matching* which is represented by gray edges. In [30,31,33], it's been proved that a type of subgraph called adequate sub-graph (*AS*) could be used to decompose the graph with edge shrinking operations. Figure 3(c) shows an example of *AS* and edge shrinking. The *BnB* algorithm is served to solve the *DCJ* median problem with equal content genomes. Unfortunately, there is no *BnB* based algorithm that deals with unequal content cases, and we will show that it's actually hard to design such algorithm in the following section.



(a) *MBG* (b) *0-matching* (c) Adequate subgraph and edge shrinking

Fig. 3. Examples of *MBG*; 0-matching and edge shrinking operations

3 Approaches

3.1 Applying *DCJ-Indel-Exemplar* Distance to Evaluate Dissimilarity

The *DCJ-Indel* distance can handle genomes which only have *Indels*, while the exemplar distance can only handle duplications. To process genomes with both *Indels* and duplications, a new distance metric named *DCJ-Indel-Exemplar* distance is designed by combining these two distances together. For gene families with duplicated genes, only one gene copy of a gene family in each genome is selected, and the rest of the gene copies in the gene family are deleted from both of the genomes. The resulting genomes are called 'exemplar' genomes. Of all possible selection of exemplar genomes, the one with the minimum *DCJ-Indel* distance is the *DCJ-Indel-Exemplar* distance for the original two genomes.

The *DCJ-Indel-Exemplar* distance does not reflect the true number of evolutionary events. For one thing, the number of duplications are not counted; furthermore, when there are large number of mutations, *DCJ* distance will underestimate the distance. Therefore, two steps are followed to adjust the *DCJ-Indel-Exemplar* distance. The first step is to use *EDE* [20] to rescale the distance. The second step is to add the count of duplicated genes by comparing the difference of the count of the same gene family in two genomes, if they are different, a duplication count is added. The *DCJ-Indel-Exemplar* distance after the adjustment of *EDE* distance and the addition of number of duplications is the final distance.

3.2 Adapting *Lin-Kernighan* Heuristic to Find Median

Problem Statement. Not surprisingly, finding the median genome that minimize the *DCJ-Indel-Exemplar* distance, is challenging. To begin with, given three input genomes, there are multiple choices of possible gene content selections for a median genome. Therefore, to make the problem easier, we can define a relaxed version of the median problem by providing known gene contents.

DCJ-Indel-Exemplar median
Instance. Given the gene content of a median genome, and gene orders of three modern genomes.
Question. Find an adjacency of the genes of the median genome that minimize the *DCJ-Indel-Exemplar* distance between the median genome and the three input genomes.

The *DCJ-Indel-Exemplar* median problem is not even in the class of **NP** because there is no polynomial time algorithm to verify the results. Furthermore, it's hard to design an exact *BnB* algorithm for *DCJ-Indel-Exemplar* median problem mainly because: To begin with, distance under DCJ does not hold when considering *Indels* [35]. when a *0-matching* edge is selected, edge shrinking is performed to generate the new *MBG*. The question is, when there are duplicated genes in a genome, it's possible that there are multiple edges of the same type connecting to the same vertex of a *0-matching*. This leads to ambiguity in the edge shrinking step, which makes the followed *BnB* search process very complicated and extremely hard to implement. Hence, we provided an adaption of Lin-Kernighan (*LK*) heuristic to help solving this challenging problem.

Design of *Lin-Kernighan* Heuristic. The *LK* heuristic can generally be divided into two steps: initialization of 0-*matching* for the median genome, and *LK* search to get the result.

The initialization problem can be described as: given gene contents of three genomes, find a median genome gene content that minimizes the sum of the number of *Indels* and duplications operations that transfer the median gene content to gene contents of other three genomes. In this paper, we designed a very simple rule to initialize the gene content of the median genome, which is, given the counts of one gene family of three genomes. If two or three counts are the same, we simply select this count as the number of occurence of the gene family in the median genome. If all three counts are different, we select the median count as the number of occurence of the gene family in the median genome.

After fixing the gene content for median genome, the next step is to set up the *0-matching* in the *MBG* and perform the *LK* heuristic. In this paper, we randomly set up the *0-matching*. As for the *LK* strategy, by selecting two *0-matching* edges on *MBG* of a given search node, and perform a *DCJ* operation, we can get the *MBG* of a neighbor search node. We expand the search frontier by keeping all neighboring search nodes to up until the search level $L1$. Then we only examine and add the most promising neighbors to the search list until level $L2$. The search is continued by the time when there is a neighbor solution yielding a better median score. This solution is then accepted and with it a new search is initiated from the scratch. The search will be terminated if there are no improvement on the result as the search level limit has been reached and all possible neighbors has been enumerated. If $L1 = L2 = K$, the algorithm is called *K-OPT* algorithm.

Adopting Adequate Subgraphs to Simplify Problem Space. There are two categories of vertices in the *MBG*. One connected with exactly one edge of each edge type, is called "regular" vertices; another connected with less or more than one edges of each edge type, is classified as "irregular" vertices. A subgraph in the *MBG* that only contains regular vertices, is defined as regular subgraph [30]. By using the adequate subgraphs [30,33], we can prove that they are still applicable for decomposing the graph in *DCJ-Indel-Exemplar* median problem.

Lemma 1. *As long as the irregular vertices do not involve, regular subgraphs are applicable to decompose* MBG*.*

Proof. If there are d number of vertices that contain duplicated edges in *MBG*, then we can disambiguate the *MBG* by generating different subgraphs that contain only one of the duplicate edges (we call these subgraphs disambiguate *MBG*, *d-MBG*). And there are $O(\prod_{i<d} deg(i))$ number of *d-MBG*s. Suppose a regular adequate subgraph exists in the *MBG*, then it must also exist in every *d-MBG*. Based on the *0-matching* solution, we can transform every *d-MBG* into completed *d-MBG* (*cd-MBG*) by constructing the optimal completion [12] between *0-matching* and all the other 3 types of edges. After this step, the adequate subgraphs exist in every *d-MBG* still exist in every *cd-MBG*. Which means, we can use these adequate subgraphs to decompose *cd-MBG* for each median problem without losing accuracy. □

Search Space Reduction Methods. The performance bottleneck with the median computation is in the exhaustive search step, because for each search level we need to consider $O(2g)^2$ possible number of edge pairs, which is $O((2g)^{2L1})$ in total. In traveling salesman problem (*TSP*), it's cheap to find the best neighbor, but for *DCJ* operations, to evaluate a neighbor, we need to compute *NP*-hard *DCJ-Indel-Exemplar* distance, which makes this step extremely expensive to conclude. Noticing that if we search neighbors on edges that are on the same *0-i* color altered connected component (*0-i-comp*), the *DCJ-Indel-Exemplar* distance for genome 0 and genome i is more likely to reduce [36]. We can sort each edge pair by how many *0-i-comp* they share. Suppose the number of *0-i-comp* that an edge pair x share is $num\_pair(x)$. When the algorithm is in the exhaustive search step ($currentLevel < L1$), we set a threshold δ and select the edge pairs that satisfy: $num\_pair(x) > \delta$ to be added into the search list. When it comes to the recursive deepening step; we select the edge pair that satisfy $\underset{x}{argmax}\ num\_pair(x)$ to be added into the search list. This strategy has two merits, 1) some of the non-promising neighbor solution is eliminated to reduce the search space. 2) the expensive evaluation step which make a function call to *DCJ-Indel-Exemplar* distance is postponed to the time when a solution is retrieved from the search list.

4 Experimental Results

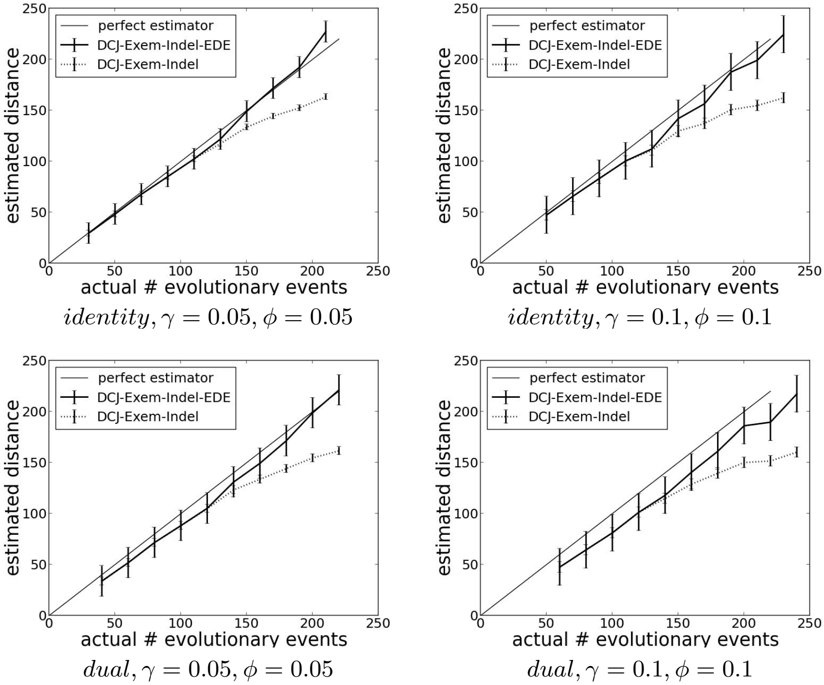
Distance Estimation. We simulated the data sets using genomes with 200 genes. To show how *Indels* and duplications affect the estimation of the distance, we divide the data set into multiple groups with varied *Indels* rate (γ, which varies from 5% to 10%), and duplication rate (ϕ, which varies from 5% to 10% as well). For each *Indels* or duplication event, only one gene is inserted/deleted or duplicated. We compare the change of distance estimation with the change of mutation rate (θ, which varies from 10% to 100%, we used reversal operation to simulate the mutation mainly because *DCJ* distance and reversal distance are quite similar when using genome data of same contents), with each specific setting of γ and ϕ. With two genomes (one is called target and the other is called subject) we conduct experiments on two sets of data. One set of data that set target genome as identity genome (for example $(1, 2, 3, ..., i, j, ..., n)$), and the subject genome is evolved from the identity genome with full ratio of θ, γ, ϕ, we call this set '*identity*'. Another set of data assigns half ratio of θ, γ, ϕ to both of target and subject genomes to let them evolve from identity genome, we call this set '*dual*'.

The result for *DCJ-Indel-Exemplar* distance and *DCJ-Indel-Exemplar* distance corrected by *EDE* are shown in Fig 4. As for the impact of different evolution operation rates, the main factor that affects the accuracy of distance estimation is the change of rate γ and ϕ. This is mainly because an *Indel* after a duplication can cancel the count of both *Indel* and duplication and makes the distance underestimated. As for the effect of two different data sets, it seems that the '*dual*' set underestimates the result more than '*identity*' set, which is mainly because both of two genomes will delete a common set of genes, which makes the actual size of alphabet $\mathcal{A}$ shrunk.

Median Computation. We simulate the median data of three genomes using the same simulation strategy as in the distance simulation. In our experiments, each genome is "evolved" from a seed genome, which is identity, and they all have the same evolution rate (θ, γ and ϕ). We compare the result of using *LK* algorithm with $L1 = 2$ and $L2 = 3$, and the *K-OPT* algorithm of $K = 2$. We use the search space reduction methods and set $\delta = 2$ and $\delta = 3$ respectively.

To test the accuracy of our *LK* and *K-OPT* methods, we first set both γ and ϕ to 0 and increased the mutation rate θ from 10% to 100%, so that each of the three genomes has the same gene content. We run the exact *DCJ* median solver (we use the one in [36]) to compare the exact result with our heuristic. In Fig 5(a), it shows the accuracy of our heuristic compared with the exact result. It is shown that when $\theta \leq 60\%$, all results of the *LK* and *K-OPT* methods are quite close to the exact solver. For parameter of $\delta = 2$, both *LK* and *K-OPT* methods can generate exact results for most of the cases.

As for the median results for unequal contents, we set both γ and ϕ to 5% and increase the mutation (inversion) rate θ from 10% to 60%. We compare our result with the accumulated distance of three genomes to their simulation seed. Although it can not show the accuracy of our method (since we do not have an



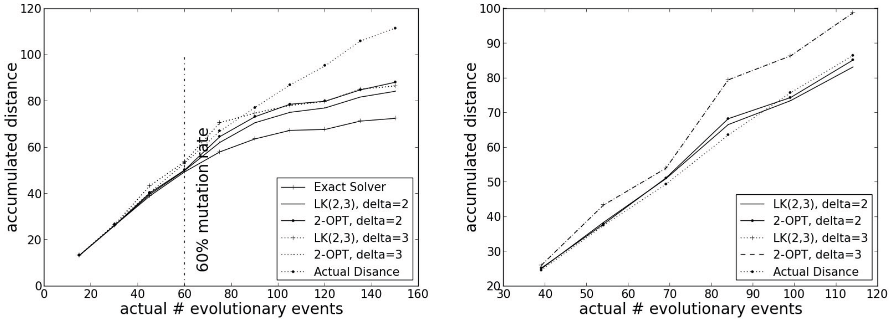
$identity, \gamma = 0.05, \phi = 0.05$

$identity, \gamma = 0.1, \phi = 0.1$

$dual, \gamma = 0.05, \phi = 0.05$

$dual, \gamma = 0.1, \phi = 0.1$

Fig. 4. Distance computation results, the x-axis represents the actual number of *DCJ* operations and the y-axis represent the computed distance for the methods using *DCJ-Indel-Exemplar* distance, *DCJ-Indel-Exemplar* distance rectified by *EDE*, and the true estimator. γ is the rate of *Indels* and ϕ is the rate of duplications. The results are grouped by two sets of data, which are *identity* and *dual*.



(a) $\gamma = \phi = 0\%$ and θ varies from 10% to 100%.

(b) $\gamma = \phi = 5\%$ and θ varies from 10% to 60%.

Fig. 5. Experimental results for median computation

exact solver), it can be used as an indicator of how close of our method was to the real evolution. Fig 5(b) shows the median results for unequal gene contents. It indicates that when $\delta = 3$, both *LK* and *K-OPT* algorithms get results quite close to the real evolutionary distance.

5 Conclusion

In this paper, we proposed a new way to compute the distance and median between genomes with unequal contents (with *Indels* and duplications). Nevertheless, there are still a lot of aspects to be improved. For example, we need to design a scheme to better estimate the gene contents. A way to deal with ambiguation when shrinking an edge is needed; therefore, a branch and bound algorithm could be designed to infer the exact median genome. Last but not least, since the *LK* algorithm can only process hundreds of genes, algorithm engineering and high performance computing methods are required to provide a way helping us to design faster algorithms to deal with high resolution data.

Acknowledgements. This Research was sponsored in part by the NSF OCI-0904461 (Bader), OCI-0904179, IIS-1161586 (Tang) and OCI- 0904166 (Schaeffer).

References

1. Angibaud, S., Fertin, G., Rusu, I., Thévenin, A., Vialette, S.: On the approximability of comparing genomes with duplicates. J. Graph Algorithms Appl. 13(1), 19–53 (2009)
2. Bader, D.A., Moret, B.M.E., Yan, M.: A linear-time algorithm for computing inversion distance between signed permutations with an experimental study. Journal of Computational Biology 8, 483–491 (2001)
3. Bergeron, A., Mixtacki, J., Stoye, J.: On sorting by translocations. Journal of Computational Biology, 615–629 (2005)
4. Blin, G., Chauve, C., Fertin, G.: The breakpoint distance for signed sequences. In: Proc. CompBioNets 2004. Text in Algorithms, vol. 3, pp. 3–16. King's College, London (2004)
5. Bourque, G., Pevzner, P.A.: Genome-Scale Evolution: Reconstructing Gene Orders in the Ancestral Species. Genome Res. 12(1), 26–36 (2002)
6. Braga, M.D.V., Willing, E., Stoye, J.: Genomic distance with DCJ and indels. In: Moulton, V., Singh, M. (eds.) WABI 2010. LNCS, vol. 6293, pp. 90–101. Springer, Heidelberg (2010)
7. Bryant, D.: The complexity of calculating exemplar distances. In: Sankoff, D., Nadeau, J. (eds.) Comparative Genomics. Kluwer (2001)
8. Caprara, A.: The Reversal Median Problem. INFORMS Journal on Computing 15(1), 93–113 (2003)
9. Chauve, C., Fertin, G., Rizzi, R., Vialette, S.: Genomes containing duplicates are hard to compare. In: Alexandrov, V.N., van Albada, G.D., Sloot, P.M.A., Dongarra, J. (eds.) ICCS 2006. Part II. LNCS, vol. 3992, pp. 783–790. Springer, Heidelberg (2006)

10. Chen, X., Zheng, J., Fu, Z., Nan, P., Zhong, Y., Lonardi, S., Jiang, T.: Assignment of orthologous genes via genome rearrangement. IEEE/ACM Trans. Comput. Biology Bioinform. 2(4), 302–315 (2005)
11. Chen, Z., Fu, B., Zhu, B.: Erratum: The approximability of the exemplar breakpoint distance problem. In: Snoeyink, J., Lu, P., Su, K., Wang, L. (eds.) FAW-AAIM 2012. LNCS, vol. 7285, p. 368. Springer, Heidelberg (2012)
12. Compeau, P.E.C.: A simplified view of dcj-indel distance. In: Raphael, B., Tang, J. (eds.) WABI 2012. LNCS (LNBI), vol. 7534, pp. 365–377. Springer, Heidelberg (2012)
13. Fertin, G., Labarre, A., Rusu, I., Tannier, E., Vialette, S.: Combinatorics of Genome Rearrangements, 1st edn. The MIT Press (2009)
14. Gao, N., Yang, N., Tang, J.: Ancestral genome inference using a genetic algorithm approach. PLoS One 8(5) (2013)
15. Hannenhalli, S.: Polynomial-time algorithm for computing translocation distance between genomes. Discrete Applied Mathematics 71(1-3), 137–151 (1996)
16. Lenne, R., Solnon, C., Stützle, T., Tannier, E., Birattari, M.: Reactive Stochastic Local Search Algorithms for the Genomic Median Problem. In: van Hemert, J., Cotta, C. (eds.) EvoCOP 2008. LNCS, vol. 4972, pp. 266–276. Springer, Heidelberg (2008)
17. Lin, Y., Hu, F., Tang, J., Moret, B.M.: Maximum likelihood phylogenetic reconstruction from high-resolution whole-genome data and a tree of 68 eukaryotes. In: Proc. 18th Pacific Symp. on Biocomputing, PSB 2013, pp. 285–296. IEEE Computer Society, Washington, DC (2013)
18. Marron, M., Swenson, K.M., Moret, B.M.E.: Genomic distances under deletions and insertions. In: Warnow, T., Zhu, B. (eds.) COCOON 2003. LNCS, vol. 2697, pp. 537–547. Springer, Heidelberg (2003)
19. Moret, B.M.E., Tang, J., San Wang, L., Warnow, Y.: Steps toward accurate reconstructions of phylogenies from gene-order data. J. Comput. Syst. Sci 65, 508–525 (2002)
20. Moret, B.M.E., Wang, L.S., Warnow, T., Wyman, S.K.: New approaches for reconstructing phylogenies from gene order data. In: ISMB (Supplement of Bioinformatics), pp. 165–173 (2001)
21. Nguyen, C.T., Tay, Y.C., Zhang, L.: Divide-and-conquer approach for the exemplar breakpoint distance. Bioinformatics 21(10), 2171–2176 (2005)
22. Pe'er, I., Shamir, R.: The median problems for breakpoints are np-complete. Elec. Colloq. on Comput. Complexity 71 (1998)
23. Pevzner, P.A.: Computational Molecular Biology: An Algorithmic Approach, 1st edn. Computational Molecular Biology. A Bradford Book (August 2000)
24. Rajan, V., Xu, A.W., Lin, Y., Swenson, K.M., Moret, B.M.E.: Heuristics for the inversion median problem. BMC Bioinformatics 11(S-1), 30 (2010)
25. Sankoff, D.: Genome rearrangement with gene families. Bioinformatics 15(11), 909–917 (1999)
26. Shao, M., Lin, Y.: Approximating the edit distance for genomes with duplicate genes under dcj, insertion and deletion. BMC Bioinformatics 13(S-19), S13 (2012)
27. Shao, M., Lin, Y., Moret, B.: An exact algorithm to compute the DCJ distance for genomes with duplicate genes. In: Sharan, R. (ed.) RECOMB 2014. LNCS (LNBI), vol. 8394, pp. 280–292. Springer, Heidelberg (2014)
28. Swenson, K.M., Marron, M., Earnest-DeYoung, J.V., Moret, B.M.E.: Approximating the true evolutionary distance between two genomes. In: Demetrescu, C., Sedgewick, R., Tamassia, R. (eds.) ALENEX/ANALCO, pp. 121–129. SIAM (2005)

29. Tang, J., Moret, B.M.E.: Phylogenetic reconstruction from gene-rearrangement data with unequal gene content. In: Dehne, F., Sack, J.-R., Smid, M. (eds.) WADS 2003. LNCS, vol. 2748, pp. 37–46. Springer, Heidelberg (2003)
30. Xu, A.W.: DCJ median problems on linear multichromosomal genomes: Graph representation and fast exact solutions. In: Ciccarelli, F.D., Miklós, I. (eds.) RECOMB-CG 2009. LNCS (LNBI), vol. 5817, pp. 70–83. Springer, Heidelberg (2009)
31. Xu, A.W.: A fast and exact algorithm for the median of three problem: A graph decomposition approach. Journal of Computational Biology 16(10), 1369–1381 (2009)
32. Xu, A.W., Moret, B.M.E.: Gasts: Parsimony scoring under rearrangements. In: Przytycka, T.M., Sagot, M.-F. (eds.) WABI 2011. LNCS (LNBI), vol. 6833, pp. 351–363. Springer, Heidelberg (2011)
33. Xu, A.W., Sankoff, D.: Decompositions of multiple breakpoint graphs and rapid exact solutions to the median problem. In: Crandall, K.A., Lagergren, J. (eds.) WABI 2008. LNCS (LNBI), vol. 5251, pp. 25–37. Springer, Heidelberg (2008)
34. Yancopoulos, S., Attie, O., Friedberg, R.: Efficient sorting of genomic permutations by translocation, inversion and block interchange. Bioinformatics 21(16), 3340–3346 (2005)
35. Yancopoulos, S., Friedberg, R.: Sorting genomes with insertions, deletions and duplications by DCJ. In: Nelson, C.E., Vialette, S. (eds.) RECOMB-CG 2008. LNCS (LNBI), vol. 5267, pp. 170–183. Springer, Heidelberg (2008)
36. Yin, Z., Tang, J., Schaeffer, S.W., Bader, D.A.: Streaming breakpoint graph analytics for accelerating and parallelizing the computation of dcj median of three genomes. In: ICCS, pp. 561–570 (2013)