# A Gene Family-Free model for Genome Rearrangements with Insertions and Deletions

Kevin Lamkiewicz and Pedro Feijão

Genome Informatics Group, Faculty of Technology, Bielefeld University, Germany.
klamkiew@cebitec.uni-bielefeld.de

**Abstract.** In comparative genomics methods for structural evolution, a common pre-processing step is to perform an orthology detection method to classify genes into gene families, in order to represent each chromosome as an ordering of genes of the detected families. This allows, for instance, the application of genome rearrangement distance methods, such as the Double Cut and Join model.
A recent approach called *family-free* aims to avoid this pre-processing step, receiving as input only the pairwise similarity between genes. Under this model, the family-free Double-Cut-and-Join distance was recently proposed. In this work, we extend this distance model to account for insertion and deletion events, also incorporating into the overall distance a way to balance the contributions from the rearrangement distance and sequence dissimilarity to achieve a combined measure of evolution. We generated several simulations to compare the original model with our proposed model, in terms of evolutionary distance estimation and recovery of number of rearrangements and indel events. The newly proposed distances reasonably estimates the number of indel events, and also gives a better measure of evolutionary distance than the original distance model.

## 1 Background

During the course of evolution, genomes are subject to mutations, such as substitutions, small insertions and deletions at the nucleotide level, but also to larger scale events that change the position and orientation of large blocks of DNA. Such events are called *genome rearrangements* and include inversions, translocations, fusions and fissions, block deletions, among others.

A classical problem in comparative genomics is to compute the rearrangement distance between genomes, that is, the minimum number of rearrangement events required to transform a given genome into another [10]. In order to study this problem, a pre-processing of the genome sequence data is required, so that we can compare the content of the genomes.

One common method is to identify homologous genes in all genomes and group them into *gene families*, so that genes in the same family are said to be equivalent. This setting is said to be *family-based*. Without gene duplications, that is, with the additional restriction that each family occurs exactly once in

each genome, many polynomial time models have been proposed to compute the genome rearrangement distance, such as the Double-Cut-and-Join (DCJ) operation [1, 12], when the genomes have the same number of genes. In the case that not all genes are present in all genomes, it is possible to include gene insertion and deletion events (here called *indels*), and the DCJ indel distance can also be calculated in polynomial time [2, 4]. However, when gene duplications are allowed, the DCJ distance problem is NP-hard [11].

The classification of genes into gene families is usually made automatically, and several methods have been proposed (see [7] for a recent approach, with a review of several methods). This automated process can be error-prone, therefore compromising the following comparative analysis. Also, even when genes are correctly classified into gene families, some information is lost, such as the similarity between each of the genes in the same family.

An alternative approach was recently proposed to study comparative genomics methods without the need for prior family assignment, where the pairwise similarity between genes is directly used [6, 3, 8]. This approach is called *family-free*. A recent example of the application of this method is the family-free Double-Cut-and-Join (FF-DCJ) distance [8]. This distance considers only a one-to-one matching between genes from both genomes in order to determine the FF-DCJ distance between two genomes, ignoring any gene that has no correspondence on the other genome. In this work we extend the FF-DCJ distance to treat *indel* events, by considering the unmatched genes, and show that this improves the evolutionary distance measure and can give an good estimate of the number of indel events that occurred between two genomes.

## 2 Methods

The input to a family-free problem is a *gene similarity graph*, where genes in the input genomes correspond to vertices, and weighted edges between genes indicate the similarity between them. For two genomes, this graph is bipartite, and the FF-DCJ distance problem is defined as finding a matching that minimizes the weighted rearrangement distance. This problem is NP-hard, and an integer linear program (ILP) was proposed to solve it [8]. In their model, genes not present in an optimal matching are ignored, which means that the original FF-DCJ distance does not give a good evolutionary distance estimate when there are many unique genes, that is, genes present in one genome but not on the other. Therefore, similarly with what was done in the family-based DCJ, we extend the FF-DCJ problem to include insertion and deletion events.

Given genomes with unique genes, in the *family-based* setting, there are DCJ indel models that find in polynomical time an optimal way of transforming one genome into the other by the minimal number of DCJ and indel events [2, 4]. However, applying these models in the *family-free* setting is not trivial, and could potentially explode the number of variables and constraints in the ILP, making it infeasible. Therefore, we tested two different approaches to extend the original ILP.

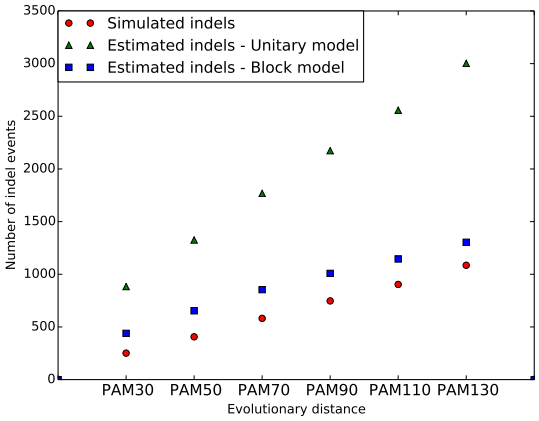


**Fig. 1.** Measured indel events with different models. The red squares indicate the number of simulated indels. Green squares show the measured number of indel events using the unitary indel model, and the block model is represented by blue squares.

A first naive approach to simply add a cost for all unmatched genes in the objective function of the ILP. This leads to the *unitary indel model*, where each unmatched gene represents one indel operation, with a fixed cost, usually the same as a DCJ operations. This implementation most likely overestimates the number of real indel events, as a block of consecutive genes is usually deleted in a single evolutionary operation, instead of several unitary deletions.

To improve the measurement of indel operations, we also tested a *block indel model*, where new constraints where added to the ILP such a block of consecutive unmatched genes counts as an indel operation, rather than each individual gene.

## 3 Results

To test the new ILP formulations, we created several simulated dataset with the ALF simulator [5], that simulates both sequence and structural evolution. For each dataset, starting with a *Escherichia coli* genome, we simulated two new genomes by using ALF default parameters, just scaling the PAM distance parameter from 30 to 110 in increments of 20, with 10 repetitions, to test different rates of evolution. By running Blast all vs. all between the genes of both genomes and using the relative reciprocal BLAST score (RRBS) [9] as the similarity between genes, we obtain the gene similarity graph, which is then given as input to generate the ILPs for the unitary indel and block indel models. The ILPs were solved with `CPLEX`[1].

[1] http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/

Figure 1 shows the number of estimated indel events using our approaches compared to the exact number of operations performed by ALF. The unitary indel model largely overestimates the number of indels, while the block model has much better results, even without involving any complex methods. There is still some overestimation, most likely due to rearrangement events occurring after indels, in the same regions. The results could be improved by implementing more complex FF-DCJ indel models with ideas from the family-based DCJ models, in a way that does not exponentially increases the ILP size, and this is our current goal.

## References


1. Anne Bergeron, Julia Mixtacki, and Jens Stoye. A unifying view of genome rearrangements. In *Proc. of WABI 2006*, volume 4175 of *LNBI*, pages 163–173, 2006.
2. Marília D V Braga, Eyla Willing, and Jens Stoye. Double cut and join with insertions and deletions. *Journal of Computational Biology*, 18(9):1167–84, September 2011.
3. Marília Dias Vieira Braga, Cedric Chauve, Daniel Dörr, Katharina Jahn, Jens Stoye, Annelyse Thévenin, and Roland Wittler. The potential of family-free genome comparison. In C. Chauve, N. El-Mabrouk, and E. Tannier, editors, *Models and Algorithms for Genome Evolution*, chapter 13, pages 287–307. Springer, London, 2013.
4. Phillip Ec Compeau. DCJ-Indel sorting revisited. *Algorithms for molecular biology : AMB*, 8(1):6, March 2013.
5. Daniel A Dalquen, Maria Anisimova, Gaston H Gonnet, and Christophe Dessimoz. ALF–a simulation framework for genome evolution. *Mol. Biol. Evol.*, 29(4):1115–1123, 2012.
6. Daniel Dörr, Annelyse Thévenin, and Jens Stoye. Gene family assignment-free comparative genomics. *BMC Bioinformatics*, 13(Suppl 19):S3, 2012.
7. Marcus Lechner, Maribel Hernandez-Rosales, Daniel Doerr, Nicolas Wieseke, Annelyse Thévenin, Jens Stoye, Roland K. Hartmann, Sonja J. Prohaska, and Peter F. Stadler. Orthology Detection Combining Clustering and Synteny for Very Large Datasets. *PLoS ONE*, 9(8):e105015, August 2014.
8. Fábio V Martinez, Pedro Feijão, Marília Dv Braga, and Jens Stoye. On the family-free DCJ distance and similarity. *Algorithms for Molecular Biology*, 10(1):1–10, 2015.
9. Catia Pesquita, Daniel Faria, Hugo Bastos, António E N Ferreira, André O Falcão, and Francisco M Couto. Metrics for GO based protein semantic similarity: a systematic evaluation. *BMC bioinformatics*, 9 Suppl 5:S4, 2008.
10. David Sankoff. Edit distance for genome comparison based on non-local operations. In *Proc. of CPM 1992*, volume 644 of *LNCS*, pages 121–135, 1992.
11. Mingfu Shao, Yu Lin, and Bernard Moret. An exact algorithm to compute the DCJ distance for genomes with duplicate genes. In *Proc. of RECOMB 2014*, volume 8394 of *LNBI*, pages 280–292, 2014.
12. Sophia Yancopoulos, Oliver Attie, and Richard Friedberg. Efficient sorting of genomic permutations by translocation, inversion and block interchanges. *Bioinformatics*, 21(16):3340–3346, 2005.