Identifying Orthologs: Cycle Splitting on the Breakpoint Graph

Krister M. Swenson, Nicholas D. Pattengale, Bernard M.E. Moret
Department of Computer Science
University of New Mexico
Albuquerque, NM 87131, USA
{kswenson,nickp,moret}@cs.unm.edu

1. Introduction

Gene rearrangements have successfully been used in phylogenetic reconstruction and comparative genomics (see the survey of [4] and the monograph of [6]), but usually under the assumption that all genomes have the same gene content and that no gene is duplicated. While these assumptions allow one to work with organellar genomes, they are too restrictive when comparing nuclear genomes [1], where the main challenge is how to deal with gene families, specifically, how to identify orthologs. While searching for orthologies is a common task in computational biology, it is usually done using sequence data. We approach that problem using gene rearrangement data. Sankoff [5] first addressed this problem with his introduction of *exemplars*, in which he suggested identifying a single gene within each family (the exemplar) on the basis of a parsimonious criterion (using the fewest rearrangements) and discarding all others. Our group provided an alternate approach in which a correspondence is established between gene families on the basis of conserved segments [3, 8]; our results suggested that considering all members of a gene family yields better results than keeping only exemplars, but were limited in that the assignment of orthologs did not take into account any rearrangement structure beyond conserved segments.

Here we take steps to remedy this problem by providing an optimization framework derived from the breakpoint graph (the basic structure behind the last decade of work in gene rearrangements [2]) in which to phrase the problem; we give preliminary theoretical results in support of our framework.

2. The Breakpoint Graph

The basic structure describing a pair of genomes with no duplicates and equal gene content is the *breakpoint graph* (really a multigraph)—for a careful and readable description of its construction, see [7]. In our case, however, gene families in G_1 need not be singletons, so we need to extend the construction. Let $B(G_1) = (V, E)$ denote the breakpoint graph for G_1 and G_2 (because of our conventions, G_2 is known once G_1 is). As in the regular breakpoint graph, each singleton gene g in G_1 becomes a pair of vertices, g^- and g^+ (the "negative" and "positive" terminals), joined by an edge; we leave out the gene families with multiple members, since only the singletons have a well-defined structure, but we now need to accommodate gaps left in the sequence where duplicate genes exist in G_1. We add a *desire* edge (in the charming terminology of [7]—also known elsewhere as a gray edge) (a_i^-, b_j^+), for each member i and j of gene families a and b, respectively, whenever a and b differ by one in the indexing (i.e., are neighbors in G_2). We add a *reality* edge (a^p, b^q) if a is the element to the left of b in G_1 and either $p = q$ if a and b have different parities (in G_1 naturally) or $p \neq q$ if a and b have the same parity. Figure 1 illustrates the construction.

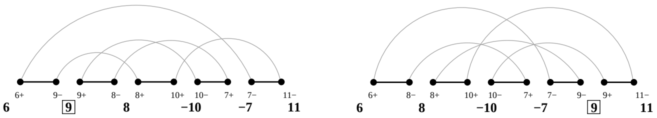

Figure 1. The breakpoint graphs for the two candidates for gene 9.

3. The Cycle Splitting Problem

We can formulate the orthology assignment problem as an optimization problem within the context of the breakpoint graph $B(G_1)$: choose an assignment of orthologs (one from each multigene family in G_1) such that the number of cycles in the augmented breakpoint graph ($B(G_1)$ to which the chosen candidates have been added) has the largest possible number of cycles.

Consider the signed genome $(6, 9, 8, -10, -7, 9, 11)$. There are two occurrences of gene 9 and we must choose which one to call orthologous with gene 9 in G_2. Figure 1 shows the two breakpoint graphs. Note that the graph on left, where the candidate lies between 6 and 8, has one more cycle than the graph on the right, where the candidate lies between 7 and 11; thus the first candidate is a better choice. The choice of candidate is advantageously viewed on breakpoint graph inscribed in a circle as shown in Figure 2. Now overlay the two choices into a single graph, as shown in Figure 2(b). Two curved lines meet on the perimeter between 10^- and 8^+, denoting the two choices. The solid line indicates that choosing the candidate between 6 and 8 gives rise to desire edges that do not cross in the inscribed representation. The dashed line indicates that the other candidate gives rise to crossing desire edges. Each line meets the perimeter at one end between the two terminals of the candidate and at the other end between its bookends. Figure 3(a) illustrates a more general instance with three multigene families.

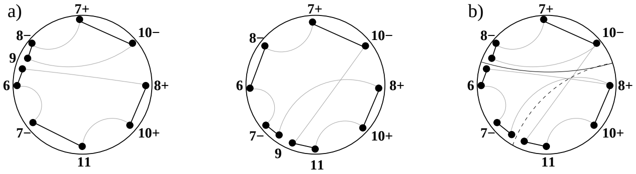

Figure 2. (a) The graphs of Figure 1 inscribed in a circle. (b) The result of overlaying the two graphs from part (a).

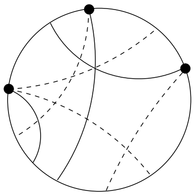

Figure 3. (a) An instance of the many-to-one cycle splitting problem.

The collection of all lines (which we refer to as *operations*) that share an endpoint represents all members of the gene family in G_1, so we also call it a family and call its common endpoint the *family home*. We can now state the constraints for the optimization problem: (i) each family home is a distinct point on the circle; (ii) the family home is not the endpoint of any operation not in that family; and (iii) the other endpoint (on the circle) of each operation is unique to that operation. The problem thus becomes picking as many operations as there are homes per family such that the cycle count is maximized.

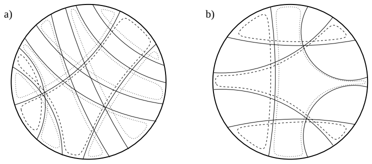

Figure 4. (a) A 3-star and two 4-chains. (b) Four 3-stars.

4 Theoretical Results

We characterize certain operations and groups of operations that don't positively contribute to the cycle count. Two such structures we call k-stars and k-chains. Figure 4 gives two examples of how these structures can interact.

We also identify a graph-theoretical framework that can guarantee optimality to a sub-form of our problem.

References

[1] J. Earnest-DeYoung, E. Lerat, and B. Moret. Reversing gene erosion: reconstructing ancestral bacterial genomes from gene-content and gene-order data. In *Proc. 4th Int'l Workshop Algs. in Bioinformatics (WABI'04)*, volume 3240 of *Lecture Notes in Computer Science*, pages 1–13. Springer Verlag, 2004.

[2] S. Hannenhalli and P. Pevzner. Transforming cabbage into turnip (polynomial algorithm for sorting signed permutations by reversals). In *Proc. 27th Ann. ACM Symp. Theory of Comput. (STOC'95)*, pages 178–189. ACM Press, New York, 1995.

[3] M. Marron, K. Swenson, and B. Moret. Genomic distances under deletions and insertions. *Theor. Computer Science*, 325(3):347–360, 2004.

[4] B. Moret, J. Tang, and T. Warnow. Reconstructing phylogenies from gene-content and gene-order data. In O. Gascuel, editor, *Mathematics of Evolution and Phylogeny*, pages 321–352. Oxford University Press, 2005.

[5] D. Sankoff. Genome rearrangement with gene families. *Bioinformatics*, 15(11):990–917, 1999.

[6] D. Sankoff and J. Nadeau, editors. *Comparative Genomics: Empirical and Analytical Approaches to Gene Order Dynamics, Map Alignment, and the Evolution of Gene Families.* Kluwer Academic Pubs., Dordrecht, Netherlands, 2000.

[7] J. Setubal and J. Meidanis. *Introduction to Computational Molecular Biology*. PWS Publishers, Boston, MA, 1997.

[8] K. Swenson, M. Marron, J. Earnest-DeYoung, and B. Moret. Approximating the true evolutionary distance between two genomes. In *Proc. 7th SIAM Workshop on Algorithm Engineering & Experiments (ALENEX'05)*. SIAM Press, Philadelphia, 2005.