Computing evolutionary distance by genome rearrangements: a combinatorial approach

Steven Kelk[1], Leen Stougie[12], Judith Keijsper[2], Leo van Iersel[2], John Tromp[2], Cor Hurkens[2]
(1) Centrum voor Wiskunde en Informatica (CWI) and (2) Technische Universiteit Eindhoven (TU/e)

1. Introduction

Genes (regions of DNA that encode proteins) are arranged linearly along the genome. Research has shown that distinct species often have surprisingly many genes in common, albeit in a different order and with different orientations. It is widely believed that periodic large-scale genome rearrangement events, which alter the order and/or orientation of gene sequences, are responsible for this. One example of a rearrangement event is a *reversal*, which reverses the order and orientation of some successive sequence of genes e.g.

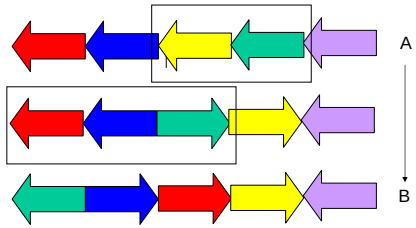


Figure 1. Here gene sequence A is transformed into B by two reversal events.

If one can predict the minimum number of such rearrangement events that can explain the transformation of genome A (from species A) into genome B (from species B), this (under the biologically reasonable assumption of *parsimony*) gives some measure for the "evolutionary distance" of the two species. Such evolutionary distance measures can in turn be useful for the construction of evolutionary trees, for example.

The question *"what is the smallest number of genome rearrangement events that can explain the transformation of genome A into genome B"* can be elegantly expressed within a mathematical/combinatorial model, and since its introduction in the early 1990s this model has generated a sustained level of interest from both biological and mathematical communities.

In this poster we describe the background of this field and then discuss our own work, in which we give efficient algorithms for a variant of this model: ***prefix reversals on binary and ternary alphabets***.

2. Background: reversals

A standard mathematical abstraction of the above problem regards the input problem (two sequences of *n* genes) as a (possibly signed) permutation on the numbers 1,2,3,…,*n*. Using the toy example above, we could represent A as +1 +2 +3 +4 +5 and B as +4 -2 -1 -3 +5:

+1	+2	+3	+4	+5	A
+1	+2	-4	-3	+5	
+4	-2	-1	-3	-5	B

Figure 2. This is Figure 1 expressed as substring reversals on signed permutations..

In this example, 2 is the minimum number of reversals required to transform A into B. But on more general inputs? In 1995 Hannenhali and Pevzner discovered (rather against expectations) an efficient (i.e. polynomial-time) algorithm for this problem [3]. This algorithm, which has been further refined and improved over the years (see a discussion of this in [1]), has been used to compute plausible evolutionary scenarios between species, for example between cabbage and turnip and between mice and man [3, 5].

3. Beyond reversals…

Reversals are only one type of genome rearrangement event. To more closely model biology, other events should be considered such as *transpositions, translocations, fissions,* and *fusions*. A transposition event, for example, does not reverse a sequence of genes but instead 'cuts' it out and 'pastes' it elsewhere e.g. here the genes 2 and 3 are removed and re-inserted between genes 4 and 5:

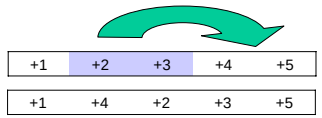


Figure 3. A transposition event moving the gene sequence 2,3 between genes 4 and 5

The variety of genome rearrangement events, and the need to adapt the described model to reflect more subtle biological features, leads to a wide array of intriguing mathematical problems. For example:

• combinations of events (e.g. reversals plus transpositions, often weighted corresponding to relative biological frequency) [1];
• circular and multi-chromosomal genomes;
• coping with the presence of duplicate genes;
• duplication/deletion events.

Unsurprisingly there remain many open problems to be solved, and the computational complexity of several fundamental questions (e.g. *"what is the smallest number of transpositions that can explain the transformation of gene sequence A into B"*) are to this day unknown. Most efficient algorithms, and approximation algorithms, have thus far been based on a construction known as the *breakpoint graph,* originally introduced by Bafna and Pevzner in 1993. This graph has many nice properties and captures, simultaneously, the difference between where a gene currently finds itself, and where it ultimately wants to be. For example, the following is the breakpoint graph for the permutation +3 -1 +2 -4:

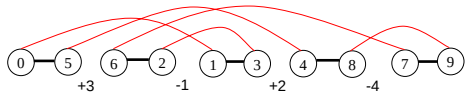
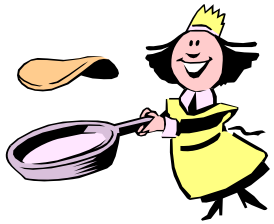


Figure 4. The Bafna/Pevzner breakpoint graph for +3 -1 +2 -4

The breakpoint graph, however, has its limitations. One problem that occurs is when there is more than one copy of each gene in the genome. In this case, the concept of a gene having a uniquely defined final destination does not necessarily apply. This motivated our research into the following problem, which took us on a detour from the world of genes to the world of pancakes…



4. Prefix reversals ("pancake flips") on binary and ternary alphabets

A *prefix reversal* (sometimes called a *"pancake flip"*) is similar to a reversal except that only a *prefix* of the permutation can be reversed. The complexity of the problem *"What is the smallest number of prefix reversals that can explain the transformation of permutation A into permutation B?"* has been open for a long time, and was made famous by Papadimitriou and (Bill) Gates [2].

We decided to investigate the complexity of the problem not on permutations but (inspired by [6]) on strings with fixed-size alphabets, allowing us (to some extent) to accommodate duplicate gene models. Specifically, we decided to investigate the problem on unsigned (i.e. without orientation) *binary alphabets* (i.e. {0,1}) and unsigned *ternary alphabets* (i.e. {0,1,2}.)

For example, what is the smallest number of prefix reversals required to transform 02201 into 01202? The answer is 3, and here is such a sequence:

0	2	2	0	1
2	0	2	0	1
1	0	2	0	2
0	1	2	0	2

Figure 5. Three prefix reversals (length-2, length-5, length-2) suffice to transform 02201 into 01202.

We identified a number of variants of this problem:

• **Distance.** Given two length-*n* strings A and B, find the smallest number of prefix reversals required to transform A into B;
• **Sorting.** Given a length-*n* string A, find the smallest number of prefix reversals required to put A in non-descending order;
• **Grouping.** Given a length-*n* string A, find the smallest number of prefix reversals required to group all like symbols in A together.

5. Our results

The following results can be found in [4]:

• A total characterization of binary and ternary strings into classes corresponding to the difficulty of the sorting and grouping problems on those strings;

• Based on the above, efficient polynomial-time algorithms for the *sorting* and *grouping* problems on both binary and ternary strings;

• The distance problem on even binary strings is NP-hard, which is very strong evidence that no polynomial-time algorithm exists for that problem;

• Binary strings are sometimes *n*-1 prefix reversals away from each other, but never more;

• Ternary strings are never more than (4/3)*n* prefix reversals away from each other;

• For fixed *k*, polynomial-time approximation schemes for the grouping and sorting problems on *k*-ary alphabets (i.e. alphabets with *k* symbols.)

6. Conclusions / open problems

Extending [6] and inspired by earlier works such as [3] we have produced a number of efficient polynomial-time algorithms for several prefix reversal problems on binary and ternary alphabets. We have also produced a number of hardness, diameter and approximation results.

The surprising complexity of the polynomial-time algorithms, and the (at this time) infeasibility of extending them to alphabets with four or more symbols, indicates that further work should primarily focus on developing a deeper, more abstract understanding of how prefix reversals behave when acting on strings with repeated symbols.

Other open problems/areas include:

• Approximation algorithms for the NP-hard distance problem;
• Tightening incomplete diameter results (we conjecture that for ternary alphabets (4/3)*n* can be improved to *n*-1*);*
• Exploring more deeply the link between sorting and grouping problems;
• Extending the problem to signed alphabets (i.e. where symbols have + and - orientations);
• Improving biological relevance by examining the behaviour of prefix reversals over alphabets that are relatively large compared to the overall length of the string (i.e. low levels of gene duplication.)
• In the spirit of papers such as [1] investigating combinations of rearrangement events e.g. prefix reversals and reversals (over binary and ternary alphabets.)

Questions?

If you have questions, comments etc. then do not hesitate to contact the corresponding author Steven Kelk:

Email: S.M.Kelk@cwi.nl
Webpage: http://homepages.cwi.nl/~kelk

References

[1] M. Bader, E. Ohlebusch, Sorting by Weighted Reversals, Transpositions and Inverted Transpositions, *Proceedings of the 10th annual conference on research in computational molecular biology* (RECOMB), (2006).

[2] W. H. Gates, C. H. Papadimitriou, Bounds for sorting by prefix reversal, *Discrete Mathematics*, 27 (1979), pp. 47-57.

[3] S. Hannenhali and P. Pevzner, Transforming cabbage into turnip (polynomial time algorithm for sorting by reversals), *Proceedings 27th ACM STOC* (1995), pp. 178-187.

[4] C. Hurkens, L. van Iersel, J. Keijsper, S. Kelk, L. Stougie, J. Tromp, Prefix reversals on binary and ternary strings, submitted July 2006 to SIAM Journal on Discrete Mathematics. Pre-print available from S.Kelk's webpage.

[5] P. Pevzner, Computational molecular biology: an algorithmic approach, MIT Press (2000).

[6] A. J. Radcliffe, A. D. Scott, E. L. Wilmer, Reversals and transpositions over finite alphabets, *SIAM Journal on Discrete Mathematics*, 19(1) (2005), pp. 224-244.

Part of this research has been funded by the Dutch BSIK/BRICKS project BRICKS