



## GRIMM: genome rearrangements web server

Glenn Tesler

Department of Computer Science and Engineering, University of California,  
San Diego, CA 92093-0114, USA

Received on September 14, 2001; revised on October 24, 2001; accepted on October 27, 2001

### ABSTRACT

**Summary:** Genome Rearrangements In Man and Mouse (GRIMM) is a tool for analyzing rearrangements of gene orders in pairs of unichromosomal and multichromosomal genomes, with either signed or unsigned gene data. Although there are several programs for analyzing rearrangements in unichromosomal genomes, this is the first to analyze rearrangements in multichromosomal genomes. GRIMM also provides a new algorithm for analyzing comparative maps for which gene directions are unknown.

**Availability:** A web server, with instructions and sample data, is available at <http://www-cse.ucsd.edu/groups/bioinformatics/GRIMM>.

**Contact:** [gptesler@ucsd.edu](mailto:gptesler@ucsd.edu)

When the Brothers Grimm (1812) transformed a man to a mouse in an instant, they could hardly have appreciated how long the normal course of evolution would take to do the same thing. Man and mouse share thousands of homologous genes, but they are arranged in a different order, and are located on different chromosomes. Such differences in gene orders are the results of rearrangement events that are common in molecular evolution of genomes. In unichromosomal genomes, the most common rearrangement events are *inversions*, usually referred to as *reversals* in bioinformatics. In a reversal, a contiguous interval of genes is put into the reverse order, also inverting the orientation of each gene. We represent genes by numbers  $1, \dots, n$ , and a unichromosomal genome as a signed permutation of these genes, where the signs indicate the two possible orientations. An example of a reversal is

$$6 \ 1 \ \boxed{3 \ -4 \ 5} \ -2 \ \longrightarrow \ 6 \ 1 \ \boxed{-5 \ 4 \ -3} \ -2.$$

In multichromosomal genomes, the most common rearrangement events are reversals, translocations, fissions, and fusions. We represent a multichromosomal genome on  $n$  genes and  $m$  chromosomes as a signed permutation of  $1, \dots, n$ , with delimiters '\$' inserted after chromosomes.

A sample *fusion* (or in reverse, *fission*) event is

$$\begin{array}{ccc} \boxed{5 \ 9 \ \$} & & 7 \ -2 \ 8 \ 3 \ \$ \\ 7 \ -2 \ 8 \ 3 \ \$ & \xrightarrow{\text{fusion}} & \boxed{5 \ 9 \ -6 \ -1 \ 12 \ \$} \\ \boxed{-6 \ -1 \ 12 \ \$} & \xleftarrow{\text{fission}} & 11 \ 4 \ 10 \ \$ \\ 11 \ 4 \ 10 \ \$ & & \end{array}$$

A *translocation* transforms two chromosomes  $A \ B$  and  $C \ D$  in a genome into  $A \ D$  and  $C \ B$ , where each letter represents a sequence of signed genes:

$$\begin{array}{ccc} 5 \ 9 \ \boxed{4 \ 10} \ \$ & & 5 \ 9 \ \boxed{11 \ 7 \ -2} \ \$ \\ 8 \ 3 \ \$ & \longrightarrow & 8 \ 3 \ \$ \\ -6 \ -1 \ \boxed{11 \ 7 \ -2} \ \$ & & -6 \ -1 \ \boxed{4 \ 10} \ \$ \end{array}$$

We restrict our focus to two genomes with  $n$  distinct genes in common. All other genes are ignored; insertions, deletions, duplications, and other such events, are not within our purview.

The *pairwise genome rearrangement problem* is to find an optimal scenario transforming one genome to another via these rearrangement events. There are in general many such scenarios, with no mathematical way to determine the actual one that occurred during evolution. A related problem with a well-defined answer is the *pairwise genomic distance problem*: determine a minimum number of rearrangement events effecting such a transformation. The genomic distance is usually quite close to the actual number of rearrangement steps that occurred, provided the number of steps is much smaller than the number of genes (Bourque and Pevzner, 2002).

Most comparative mapping techniques determine the physical locations and relative order of genes in each chromosome, but do not determine the orientation (sign) of each gene. This gives rise to analogous rearrangement and distance problems for unsigned permutations. In contrast to comparative maps, genome sequences do provide the gene orientations. The genome rearrangement problem (uni- and multichromosomal) for unsigned permutations is NP-hard (Caprara, 1999), but the same problems for signed data can be done in polynomial time (Hannenhalli and Pevzner, 1995, 1999). Fortunately, with many genomes currently being sequenced, it is likely that

**GRIMM - Genome rearrangement algorithms**

Source genome: -3 -2 \$  
-1 4 5 6 7 12 \$  
10 9 11 8 \$

Destination genome: 1 2 3 4 5 6 7 8 \$  
9 10 11 12 \$

Chromosomes:  circular  linear (oriented)  multichromosomal or unoriented

Signs:  signed  unsigned

run undo clear form Or, choose sample data

**Formatting options**

Report Style:  One line per genome (chromosomes concatenated)  One column (chromosomes separated)  Two column before & after (chromosomes separated)

Horizontal  Vertical  Show all chromosomes  Only affected

Show all possible initial steps of optimal scenarios

Highlighting style: Should operations (reversal, translocation, fission, fusion) be highlighted, and when?  
 before  after  between/both  no highlighting

Chromosome end format:  numeric (10)  subscripts (C<sub>10</sub>)  omit

Color coding: Genes should be colored according to their chromosome in which genome:  
 source  destination

run undo clear form

[Click here or scroll up to enter new data or change options.](#)

3 chromosomes, 12 genes, 6 caps Multichromosomal Distance: 6

**One optimal rearrangement scenario**

Step	Description	
0	(Source)	-8 -11 -9 -10 -3 -2 -12 -7 -6 -5 -4 1
1	Reversal	-8 -11 9 -10 -3 -2 -12 -7 -6 -5 -4 1
2	Reversal	-8 -11 10 -9 -3 -2 -12 -7 -6 -5 -4 1
3	Reversal	-8 -11 -10 -9 -3 -2 -12 -7 -6 -5 -4 1
4	Fusion	-8 -11 -10 -9 -3 -2 -1 4 5 6 7 12
5	Translocation	-8 -7 -6 -5 -4 1 2 3 9 10 11 12
6	Reversal (Destination)	-8 -7 -6 -5 -4 -3 -2 -1 9 10 11 12

GRIMM 1.01 by Glenn Tesler, University of California, San Diego.  
Copyright © 2001, The University of California.  
Contains code from GRAPPA © 2000-2001, The University of New Mexico and The University of Texas at Austin.

**Fig. 1.** Output for a scenario involving 12 genes. The source genome has three chromosomes and the destination genome has two.

many comparative maps (giving unsigned permutations) will soon be replaced by sequencing data (giving signed permutations).

Genome Rearrangements In Man and Mouse (GRIMM) is a web server combining pairwise distance and rearrangement algorithms for unichromosomal and multichromosomal genomes, with either signed or unsigned gene data. In each case, it computes the minimum possible number of rearrangement steps, and determines a possible scenario taking this number of steps. The scenarios may be formatted in a variety of different ways, depending on the size of the genomes and whether the user is interested in the biological data or in the mathematical details of the rearrangement algorithm.

GRIMM implements the Hannenhalli–Pevzner algorithms for computing unichromosomal and multichromosomal genomic distances (Hannenhalli and Pevzner, 1995, 1996, 1999; Pevzner, 2000). This code makes extensive use of code that was adapted from GRAPPA for computing unichromosomal distance in linear time (Bader *et al.*, 2001); how the multichromosomal algorithm was integrated with this, as well as corrections to errors in Hannenhalli and Pevzner (1995), will be described in Tesler (in preparation). GRIMM

also implements the Hannenhalli–Pevzner algorithm for computing the reversal distance between two unsigned unichromosomal genomes (Hannenhalli and Pevzner, 1996), and a new algorithm due to this author (Tesler, in preparation) for computing the distance between two unsigned multichromosomal genomes. While versions of all the unichromosomal algorithms have previously been implemented, this is the first full implementation of these multichromosomal algorithms. See the GRIMM website for links; especially noteworthy are GRAPPA and a Java applet (Mantin and Shamir, 1999) for reversal distance.

The various distance algorithms and rearrangement scenario generators are written in C, and the web front-end is written in Perl. The C code has been integrated into Guillaume Bourque’s program MGR (links available at the GRIMM web site), which studies the related problem of constructing optimal phylogenetic trees for multiple genomes, both uni- and multichromosomal.

## ACKNOWLEDGEMENTS

I would like to thank Derren Barken, Guillaume Bourque, Pavel Pevzner, and Yang Yu for valuable discussions and assistance.

## REFERENCES

- Bader, D., Moret, B. and Yan, M. (2001) A linear-time algorithm for computing inversion distances between signed permutations with an experimental study. In *Proceedings of the 7th International Workshop on Algorithms and Data Structures (WADS 2001)*. Springer, Berlin, pp. 365–376.
- Bourque, G. and Pevzner, P. (2002) Genome-scale evolution: reconstructing gene orders in the ancestral species. *Genome Res.*, **12**, 26–36.
- Caprara, A. (1999) Sorting permutations by reversals and Eulerian cycle decompositions. *SIAM J. Discrete Math.*, **12**, 91–110 (electronic).
- Grimm, J. and Grimm, W. (1812) Puss in boots. In *Grimm’s Fairy Tales*.
- Hannenhalli, S. and Pevzner, P. A. (1995) Transforming men into mice (polynomial algorithm for genomic distance problem). In *36th Annual Symposium on Foundations of Computer Science (Milwaukee, WI, 1995)*. IEEE Computer Society Press, Los Alamitos, CA, pp. 581–592.
- Hannenhalli, S. and Pevzner, P. (1996) To cut ... or not to cut (applications of comparative physical maps in molecular evolution). In *Proceedings of the 7th Annual ACM-SIAM Symposium on Discrete Algorithms (Atlanta, GA, 1996)*. ACM, New York, pp. 304–313.
- Hannenhalli, S. and Pevzner, P. A. (1999) Transforming cabbage into turnip: polynomial algorithm for sorting signed permutations by reversals. *J. ACM*, **46**, 1–27.
- Mantin, I. and Shamir, R. (1999) An algorithm for sorting signed permutations by reversals, <http://www.math.tau.ac.il/~rshamir/GR/>.
- Pevzner, P. A. (2000) *Computational Molecular Biology, an Algorithmic Approach*, Chapter 10, MIT Press, Cambridge, MA.