BIOINFORMATICS APPLICATIONS NOTE Vol. 21 no. 11 2005, pages 2780–2782
doi:10.1093/bioinformatics/bti412

Genome analysis

ROBIN: a tool for genome rearrangement of block-interchanges

Chin Lung Lu[1,*], Tsui Ching Wang[1], Ying Chih Lin[2] and Chuan Yi Tang[2]

[1]Department of Biological Science and Technology, National Chiao Tung University, Hsinchu 300, Taiwan, ROC and [2]Department of Computer Science, National Tsing Hua University, Hsinchu 300, Taiwan, ROC

Received on January 2, 2005; revised and accepted on March 24, 2005
Advance Access publication April 6, 2005

ABSTRACT

Summary: ROBIN is a web server for analyzing genome rearrangement of block-interchanges between two chromosomal genomes. It takes two or more linear/circular chromosomes as its input, and computes the number of minimum block-interchange rearrangements between any two input chromosomes for transforming one chromosome into another and also determines an optimal scenario taking this number of rearrangements. The input can be either bacterial-size sequence data or landmark-order data. If the input is sequence data, ROBIN will automatically search for the identical landmarks that are the homologous/conserved regions shared by all the input sequences.

Availability: ROBIN is freely accessed at http://genome.life.nctu.edu.tw/ROBIN

Contact: cllu@mail.nctu.edu.tw

INTRODUCTION

With the increasing number of sequenced genomes, the study of genome rearrangement, which measures the evolutionary difference between two organisms by conducting a large-scale comparisons of their genomic data, has received a lot of attention in computational biology and bioinformatics. One of the most promising ways to do this research is to compare the orders of the identical landmarks in two different genomes, where the identical landmarks can be the homologous/conserved regions (including genes) shared by the sequences. The genomes considered are usually denoted by a set of ordered (signed or unsigned) integers with each integer representing an identical landmark in the genomes and its sign (+ or −) indicating the transcriptional orientation. Given a set of ordered landmarks from each genome, many existing tools (Tesler, 2002; Pevzner and Tesler, 2003; Darling *et al*., 2004b) have focused on inferring an optimal series of reversal events that transform one genome organization into another, where the reversal events act on the genome by inverting a contiguous interval of landmarks into the reverse order and also inverting the orientation of each landmark. Other rearrangements like transpositions (Bafna and Pevzner, 1998; Walter *et al*., 1998), translocations (Hannenhalli, 1996; Kececioglu and Ravi, 1995), fissions, fusions (Hannenhalli and Pevzner, 1995; Meidanis and Dias, 2001) and block-interchanges (Christie, 1996; Lin *et al*., 2005) have been proposed to determine the evolutionary distance between two related genomes. Christie (1996) first introduced the block-interchange events, a new kind of global rearrangements affecting on a genome by swapping two non-intersecting intervals of landmarks of any length, and proposed an $\mathcal{O}(n^2)$ time algorithm for solving the so-called block-interchange distance problem that is to find a minimum series of block-interchanges for transforming one linear genome into another, where n is the number of landmarks. In fact, the block-interchanges can be considered as a generalization of the transpositions because the intervals of landmarks swapped by a block-interchange event are not necessarily adjacent. Recently, Lin *et al*. (2005) have designed a simpler algorithm for solving the block-interchange problem on linear or circular genomes with time complexity of $\mathcal{O}(\delta n)$, where δ is the the minimum number of block-interchanges required for the transformation and can be calculated in $\mathcal{O}(n)$ time in advance. They also demonstrated that the block-interchange events seem to play a significant role in the evolution of bacterial (*Vibrio*) species. Actually, the proof in the paper of Lin *et al*. (2005) for showing their circular algorithm being able to apply to linear chromosomes can be easily extended to prove that the block-interchange problem on circular genomes is equivalent to that on linear genomes. Here, we adopt their algorithms to implement the kernel of ROBIN (Rearrangement Of Block-INterchanges) program for analyzing rearrangements of landmark orders between two linear/circular chromosomal genomes via the block-interchange events. In addition, by integrating Mauve (Darling *et al*., 2004a) into our ROBIN system, not only landmark-order data but also sequence data are allowed to be the input of ROBIN system. If the input is sequence data, ROBIN can automatically search for the identical landmarks that are the homologous/conserved regions shared by all the input sequences.

For the landmarks used in the analyses of rearrangement among genomes, we considered the exact matches such as the maximal unique matches (MUMs) as in MUMmer (Delcher *et al*., 1999, 2002), the approximate matches without gaps such as the yielding fragments as in DIALIGN (Morgenstern *et al*., 1998; Morgenstern, 1999) and LAGAN (Brudno *et al*., 2003), the approximate matches with gaps, such as the hit fragments as in BLASTZ (Schwartz *et al*., 2003) or the regions of local collinearity, such as the locally collinear blocks (LCBs) as in Mauve (Darling *et al*., 2004a). Conceptually, an LCB can be considered as a collinear (consistent) set of the multi-MUMs, where multi-MUMs are exactly matching subsequences shared by all the considered genomes that occur only once in each of genomes and that are bounded on either side by mismatched nucleotides. Here, we adopt the LCBs for representing the landmarks in genomes. The main reason is that each LCB may correspond to a homologous region of sequence shared by all genomes and does not contain any genome rearrangements. In addition, Darling *et al*. (2004a) have implemented

*To whom correspondence should be addressed.

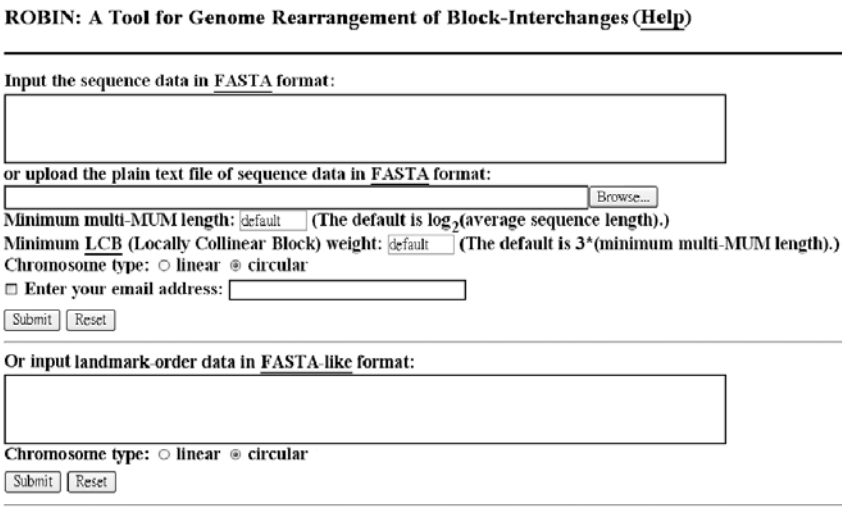

Fig. 1. The web interface of ROBIN.

a package called Mauve that contains a program which is able to efficiently identify all the LCBs shared by all the large-scale genomes being studied. Usually, each identified LCB is associated with a weight that can serve as a measure of confidence that it is a true homologous region rather than a random match, where the weight of an LCB is defined as the sum of the lengths of multi-MUMs in this LCB. The user can identify the larger LCBs that are truly involved in genome rearrangement by selecting a high minimum weight, whereas by selecting a low minimum weight, the user can trade some specificity for sensitivity to identify the smaller LCBs that are possibly involved in genome rearrangement. For the detailed algorithm of computing LCBs, we refer the reader to the paper by Darling *et al*. (2004a,b).

The kernel algorithms of ROBIN are written in C++ and the web interface is written in PHP. It can be easily accessed via a simple web interface (Fig. 1). The input of ROBIN can be two or more linear/circular chromosomes with bacterial size that can be either genomic sequences or unsigned integer sequences with each integer representing an identical landmark on all input chromosomes. If the input is genomic sequences, our ROBIN will automatically identify all the LCBs (i.e. homologous/conserved regions) that meet the user-specified minimum weight, where the minimum LCB weight is a user-definable parameter and our ROBIN chooses its default to be three times the minimum multi-MUM length. The output of ROBIN is the block-interchange distance between any two input chromosomes and its optimal scenario of block-interchange rearrangements for transforming one chromosome into another. Our ROBIN also provides an online help with some testing examples to show the user how to use this system. To test our ROBIN system, we rerun the experiments conducted by Lin *et al*. (2005) for detecting the evolutionary relationships among three human *Vibrio* pathogens, *V.vulnificus*, *V.parahaemolyticus* and *V.cholerae*. Note that Lin *et al*. used common MUMs, which were computed in advance with another tool of finding consensuses or signatures among these three *Vibrio* genomes, to represent the identical landmarks in their experiments. In our experiments, however, we used as the landmarks the LCBs that were automatically computed by our ROBIN system with default parameters from three input *Vibrio* genomic sequences. Our experimental results (whose details are shown in the ROBIN web site), showing that the block-interchange distance between *V.vulnificus* and *V.parahaemolyticus* is smaller than that between *V.vulnificus* and *V.cholerae* and that between *V.parahaemolyticus* and *V.cholerae*, indeed coincide with those obtained by Lin *et al*. (2005) and by Chen *et al*. (2003).

ACKNOWLEDGEMENTS

The authors would like to thank the anonymous referees for many constructive comments for the presentation of this paper.

REFERENCES

Bafna,V. and Pevzner,P.A. (1998) Sorting by transpositions. *SIAM J. Discrete Math.*, **11**, 221–240.

Brudno,M. *et al*. (2003) LAGAN and Multi-LAGAN: efficient tools for large-scale multiple alignment of genomic DNA. *Genome Res.*, **13**, 721–731.

Chen,C.Y. *et al*. (2003) Comparative genome analysis of *Vibrio vulnificus*, a marine pathogen. *Genome Res.*, **13**, 2577–2587.

Christie,D.A. (1996) Sorting by block-interchanges. *Information Process. Lett.*, **60**, 165–169.

Darling,A.C.E. *et al*. (2004a) Mauve: multiple alignment of conserved genomic sequence with rearrangements. *Genome Res.*, **14**, 1394–1403.

Darling,A.E. *et al*. (2004b) GRIL: genome rearrangement and inversion locator. *Bioinformatics*, **20**, 122–124.

Delcher,A.L. *et al*. (1999) Alignment of whole genomes. *Nucleic Acids Res.*, **27**, 2369–2376.

Delcher,A.L. *et al*. (2002) Fast algorithms for large-scale genome alignment and comparison. *Nucleic Acids Res.*, **30**, 2478–2483.

Hannenhalli,S. (1996) Polynomial algorithm for computing translocation distance between genomes. *Discrete Appl. Math.*, **71**, 137–151.

Hannenhalli,S. and Pevzner,P.A. (1995) Transforming men into mice (polynomial algorithm for genomic distance problem). In *Proceedings of the 36th IEEE Symposium on Foundations of Computer Science (FOCS1995)*, IEEE Computer Society, San Francisco, CA, pp. 581–592.

Kececioglu,J.D. and Ravi,R. (1995) Of mice and men: algorithms for evolutionary distances between genomes with translocation. In *Proceedings of the 6th ACM–SIAM Symposium on Discrete Algorithms (SODA1995)*, ACM/SIAM, IEEE Computer Society, San Francisco, CA, pp. 604–613.

Lin,Y.C. *et al*. (2005) An efficient algorithm for sorting by block-interchanges and its application to the evolution of *Vibrio* species. *J. Comput. Biol.*, **12**, 102–112.

Meidanis,J. and Dias,Z. (2001) Genome rearrangements distance by fusion, fission, and transposition is easy. In Navarro,G., (ed.), *Proceedings of the 8th International Symposium on String Processing and Information Retrieval (SPIRE2001)*, IEEE Computer Society, San Francisco, CA, pp. 250–253.

Morgenstern,B. (1999) DIALIGN 2: improvement of the segment-to-segment approach to multiple sequence alignment. *Bioinformatics*, **15**, 211–218.

Morgenstern,B. *et al*. (1998) DIALIGN: finding local similarities by multiple sequence alignment. *Bioinformatics*, **14**, 290–294.

Pevzner,P. and Tesler,G. (2003) Genome rearrangements in mammalian evolution: lessons from human and mouse genomes. *Genome Res.*, **13**, 37–45.

Schwartz,S. *et al*. (2003) Human–mouse alignments with BLASTZ. *Genome Res.*, **13**, 103–107.

Tesler,G. (2002) GRIMM: genome rearrangements web server. *Bioinformatics*, **18**, 492–493.

Walter,M.E.M.T., Dias,Z. and Meidanis,J. (1998) Reversal and transposition distance of linear chromosomes. In *Proceedings of String Processing and Information Retrieval (SPIRE1998)*, IEEE Computer Society, San Francisco, CA, pp. 96–102.