BIOINFORMATICS ORIGINAL PAPER Vol. 22 no. 12 2006, pages 1437–1439
doi:10.1093/bioinformatics/btl116

Genome analysis

PseudoPipe: an automated pseudogene identification pipeline

Zhaolei Zhang[1], Nicholas Carriero[2], Deyou Zheng[3], John Karro[4], Paul M. Harrison[5] and Mark Gerstein[3,*]

[1]Banting and Best Department of Medical Research, Donnelly CCBR, University of Toronto, 160 College Street, Toronto, ON M5S 3E1, Canada, [2]Department of Computer Science, Yale University, New Haven, CT 06520, USA, [3]Department of Molecular Biophysics and Biochemistry, New Haven, CT 06520, USA, [4]Department of Biology, 506 Wartik, Pennsylvania State University, University Park, PA 16802, USA and [5]Department of Biology, McGill University, Stewart Biology Building, 1205 Dr Penfield Avenue, Montreal, QC, H3A 1B1, Canada

Received on December 28, 2005; revised on March 1, 2006; accepted on March 22, 2006
Advance Access publication March 30, 2006
Associate Editor: Keith A Crandall

ABSTRACT

Motivation: Mammalian genomes contain many 'genomic fossils' i.e. pseudogenes. These are disabled copies of functional genes that have been retained in the genome by gene duplication or retrotransposition events. Pseudogenes are important resources in understanding the evolutionary history of genes and genomes.

Results: We have developed a homology-based computational pipeline ('PseudoPipe') that can search a mammalian genome and identify pseudogene sequences in a comprehensive and consistent manner. The key steps in the pipeline involve using BLAST to rapidly cross-reference potential "parent" proteins against the intergenic regions of the genome and then processing the resulting "raw hits" – i.e. eliminating redundant ones, clustering together neighbors, and associating and aligning clusters with a unique parent. Finally, pseudogenes are classified based on a combination of criteria including homology, intron-exon structure, and existence of stop codons and frameshifts.

Availability: The PseudoPipe program is implemented in Python and can be downloaded at http://pseudogene.org/

Contact: Mark.Gerstein@yale.edu or zhaolei.zhang@utoronto.ca

INTRODUCTION

Pseudogenes are those sequences in the genome that bear similarity to specific protein coding genes, but nevertheless are unable to produce functional proteins due to existence of frameshifts, premature stop codons or other deleterious mutations (Mighell *et al*., 2000). It is reported that human genome has 8000–12 000 pseudogenes and mouse genome has 5000 (Zhang *et al*., 2004, 2003). Most of these pseudogene sequences were the results of LINE1 mediated retrotransposition or genome duplication. Pseudogenes are important, as they are in essence genomic fossils that can be used to infer the ancestral sequence and evolutionary history of genes that are present today. They can often contribute to cross hybridization in high-throughput genomics experiments. Some pseudogenes reportedly have regulatory roles (Hirotsune *et al*., 2003).

In this paper, we describe our tested pseudogene identification algorithm (Zhang *et al*., 2004, 2003). The definition of pseudogene is somewhat ambiguous as it is more difficult to confirm non-functionality than confirm functionality. Our method is designed and best used to detect those pseudogenes that are unable to be translated into proteins. The algorithm has been implemented into a standalone software package, 'PseudoPipe'.

METHODS AND ALGORITHM

Program outline

The general data flow in PseudoPipe has been described previously (Zhang and Gerstein, 2004; Zhang *et al*., 2003). The inputs to the program are the genomic sequence after repeat-masking, the comprehensive and non-redundant set of protein sequences in the genome, and the chromosomal coordinates of the functional genes. The last piece of information is needed to efficiently distinguish pseudogene candidates from functional genes. The output of PseudoPipe is the complete annotation of the pseudogenes in the genome in question: their chromosomal location, nucleotide sequences, name and sequence of the parent gene, and alignment of the pseudogene with the functional gene.

Homology search

The first step in the annotation pipeline is to identify all the regions in the genome that share sequence similarity with any known protein, using BLAST (E-value $\leq 1 \times 10^{-4}$) (Altschul *et al*., 1997). We then partition the BLAST hits by chromosome and strand direction. Significant overlaps with functional gene annotations are then removed. In PseudoPipe, 'significant overlap' is defined as either complete overlap or overlap of $\geq$30 bps between a hit and a functional gene (Fig. 1A).

Eliminating redundancies

The next step is to eliminate redundant and overlapping BLAST hits, in places where a given chromosomal segment has multiple hits. We divide these overlapping hits into two categories depending on whether they match to the same or different query proteins (Fig. 1B). The first scenario arises because the BLAST program is prone to break continuous long sequence

[*]To whom correspondence should be addressed.

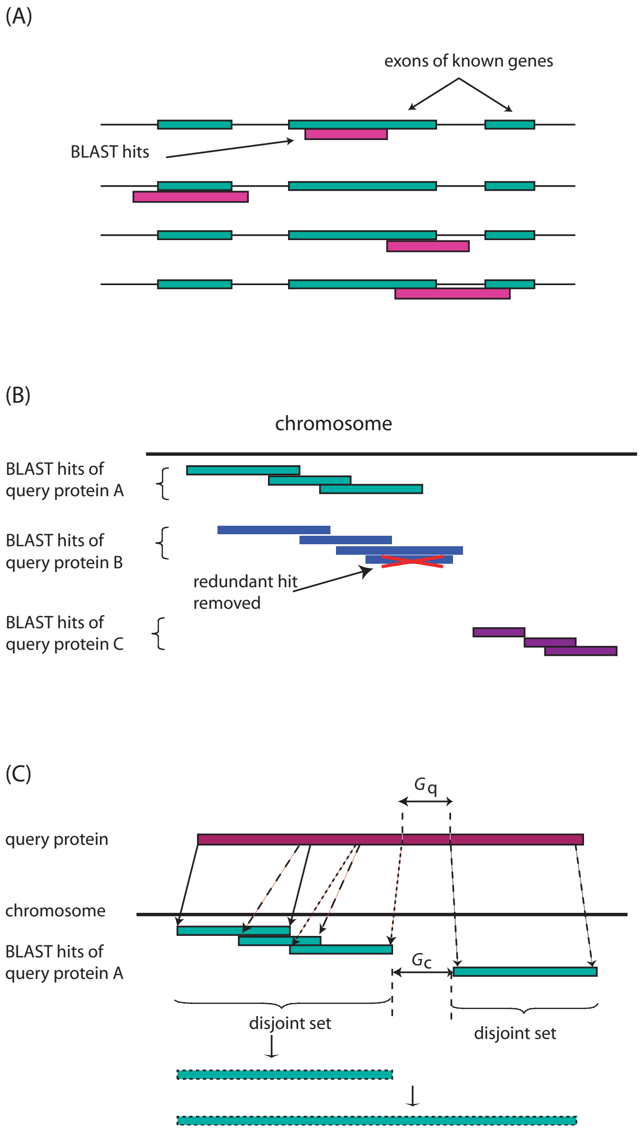

Fig. 1. **(A)** Schematic illustrations of overlap between BLAST hits and functional genes. Green rectangles represent exons of annotated genes and red rectangles represent BLAST hits. We expect some of the BLAST hits to identify the proteins' parent genes or their homologs. Given the locations of these genes' exons, we eliminate any BLAST hits that overlap as pictured here. **(B)** Separating BLAST hits into disjoint sets based on query and position on the chromosome. Three disjoint sets are shown, which match to distinct query protein A, B and C. The first two disjoint sets map to the same region on the chromosome, they were both kept at this step. Overlapping BLAST hits within each disjoint set are filtered and removed. **(C)** Merging neighboring hits. BLAST hits within each disjoint set are first merged into 'superhits'. The distance between the neighboring superhits G_c is compared with the distance on the query protein, G_q; the neighboring superhits are merged together if they are determined to be part of the same ancestral pseudogene structure and the distance G_c is not too great.

homologies into short overlapping fragments. The second scenario occurs because the BLAST query sequences include homologous proteins or protein domains. For example, in Figure 1B, both query protein A and B have BLAST hits in the same genomic region. At this step, we do not try to determine which query protein (A or B) is the 'real parent' of the pseudogene candidate. Instead, we first separate the BLAST hits that match to distinct queries, and then partition them into disjoint sets that do not have any overlap at all (Fig. 1C). The partitioning is transitive, i.e. if hit X overlaps Y and Y overlaps Z, then X, Y and Z are all placed into the same set even if X and Z do not overlap. Within each set, we sort the hits by their BLAST *E*-values and remove any hits that either completely or partially overlaps with another hit but with much worse BLAST *E*-values.

Merging the hits

Here, we merge the overlapping and successive BLAST hits into a continuous pseudogene structure (Fig. 1C). We first analyze the overlapping BLAST hits inside a single disjoint set, and merge them into a single 'super hit' or 'pseudo-exon'. We then select those neighboring disjoint sets of hits that match the same query protein. Based on the distance between the hits on the chromosome (G_c in Fig. 1C) and the distance on the query protein (G_q), we determine whether these merged hits belong to the same pseudogene structure. These gaps G_c can arise from (1) low complexity or very decayed regions of the pseudogene that are discarded by BLAST, (2) short DNA sequences inserted into the pseudogene, (3) ancestral intron sequence in the duplicated pseudogenes and (4) repetitive elements. These four scenarios can be distinguished by comparing the length of G_c and G_q, and by calculating the repeat content of the gaps between the neighboring hits.

Determining paternity of the pseudogenes

In this step, we resolve the paternity ambiguity of the pseudogenes, i.e. determine among the paralogous query proteins which one most likely gave rise to the pseudogene. To do this, we consider (1) the sequence identity between the pseudogene and the query proteins at either DNA or translated amino acid sequence levels, (2) the best BLAST *E*-value associated with any of the original hits (before merging), (3) the length of the protein subsequence that yields the pseudogene. In effect, we are assuming that after millions of years of evolution under neutral selection, the pseudogene remains more similar to the modern form of the original parent gene, than to paralogous genes.

Refining alignment

BLAST is a tool that is intended for fast, heuristic homology search instead of accurate sequence alignment. It does not accommodate frame shifts, thus it may break a potential pseudogene into smaller fragments. Therefore, in PseudoPipe, each remaining hit is re-processed using a more accurate dynamic programming alignment program, specifically tfasty from the FASTA suite (Pearson, 1997). A pseudo-exon is extended in both directions with a small buffer region (30 nt), which is then aligned to the query protein to achieve an optimal alignment, to calculate accurate sequence similarity and to annotate positions of disablements (frame shifts and stop codons), as well as insertions and deletions.

Classification of pseudogenes

The PseudoPipe program applies a set of sequence identity and completeness cutoffs to report a final set of good-quality pseudogene sequences. The parameters and cutoffs in the PseudoPipe can be easily altered to produce more or fewer pseudogenes. Previously (Zhang *et al*., 2002, 2003), we used the following cutoffs: (1) amino acid sequence identity >40%, (2) BLAST *E*-value lower than 1E-10 and (3) the pseudogene covers 70% of the parent gene. These high-confidence pseudogenes are then classified as (1) retrotransposed pseudogenes, (2) duplicated pseudogenes and (3) pseudogeneic fragments. Retrotransposed pseudogenes lack introns, have small flanking direct repeats and a 3′ polyadenine tail. PseudoPipe distinguishes retrotransposed pseudogenes from duplicated pseudogenes by a combination of these features, with the emphasis on the evidence of ancient introns. *Pseudogenic fragments* are protein/chromosome homologies that have high-sequence similarity, but are too decayed to be reliably assessed as processed or duplicated.

Implementation and run time

The program was originally written in PERL but we have re-implemented it in Python. Except for the step of whole-genome BLAST search, the annotation pipeline can be run on an entire genome in a few hours, on a reasonably robust Linux workstation (3.0 GHz, 1 GB RAM). Multiple concurrent independent pipeline runs could be started on multiple computers, e.g. several chromosomes can be grouped together and processed on a single computer.

ACKNOWLEDGEMENTS

Z.Z. acknowledges start-up fund from University of Toronto Faculty of Medicine. M.G. acknowledges funding support from the NIH/NHGRI (IU01HG003156-01). Funding to pay the Open Access publication charges was provided by the NIH (grant no. NIH/NHGRI HG02357).

Conflict of Interest: none declared.

REFERENCES

Altschul,S.F. *et al*. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res*., **25**, 3389–3402.

Hirotsune,S. *et al*. (2003) An expressed pseudogene regulates the messenger-RNA stability of its homologous coding gene. *Nature*, **423**, 91–96.

Mighell,A.J. *et al*. (2000) Vertebrate pseudogenes,. *FEBS Lett*., **468**, 109–114.

Pearson,W.R. (1997) Comparison of DNA sequences with protein sequences. *Genomics*, **46**, 24–36.

Zhang,Z. and Gerstein,M. (2004) Large-scale analysis of pseudogenes in the human genome. *Curr. Opin. Genet. Dev*., **14**, 328–335.

Zhang,Z. *et al*. (2002) Identification and analysis of over 2000 ribosomal protein pseudogenes in the human genome. *Genome Res*., **12**, 1466–1482.

Zhang,Z. *et al*. (2003) Millions of years of evolution preserved: a comprehensive catalog of the processed pseudogenes in the human genome. *Genome Res*., **13**, 2541–2558.

Zhang,Z. *et al*. (2004) Comparative analysis of processed pseudogenes in the mouse and human genomes. *Trends Genet*., **20**, 62–67.