BIOINFORMATICS APPLICATIONS NOTE Vol. 27 no. 12 2011, pages 1731–1733
doi:10.1093/bioinformatics/btr259

Data and text mining

Advance Access publication May 5, 2011

ReMark: an automatic program for clustering orthologs flexibly combining a Recursive and a Markov clustering algorithms

Kangseok Kim[1], Wonil Kim[2,*] and Sunshin Kim[3,*]

[1]Department of Knowledge Information Security, Ajou University, Suwon, [2]College of Electronics and Information, Sejong University, Seoul and [3]Department of Biomedical Science, College of Life Science, CHA University, Seongnam, Korea

Associate Editor: John Quackenbush

ABSTRACT

Summary: ReMark is a fully automatic tool for clustering orthologs by combining a Recursive and a Markov clustering (MCL) algorithms. The ReMark detects and recursively clusters ortholog pairs through reciprocal BLAST best hits between multiple genomes running software program (RecursiveClustering.java) in the first step. Then, it employs MCL algorithm to compute the clusters (score matrices generated from the previous step) and refines the clusters by adjusting an inflation factor running software program (MarkovClustering.java). This method has two key features. One utilizes, to get more reliable results, the diagonal scores in the matrix of the initial ortholog clusters. Another clusters orthologs flexibly through being controlled naturally by MCL with a selected inflation factor. Users can therefore select the fitting state of orthologous protein clusters by regulating the inflation factor according to their research interests.

Availability and Implementation: Source code for the orthologous protein clustering software is freely available for non-commercial use at http://dasan.sejong.ac.kr/∼wikim/notice.html, implemented in Java 1.6 and supported on Windows and Linux.

Contact: wikim@sejong.ac.kr; sskim04@hotmail.com

Received on December 22, 2010; revised on March 29, 2011; accepted on April 6, 2011

1 INTRODUCTION

Identifying orthologs automatically is very useful for functional annotation, and studies on comparative and evolutionary genomics.

An orthology group identified by a semi-automated method is the Clusters of Orthologous Group (COG) database (Tatusov *et al*., 1997, 2003) in National Center for Biotechnology Information (NCBI). It first identifies groups of three proteins with best reciprocal BLAST (Altschul *et al*., 1990) hits, and then performs case-by-case manual analysis to eliminate false-positives. Remm *et al*. (2001) introduces InParanoid, a fully automatic program to detect orthologs and inparalogs between two species. Li *et al*. (2003) developed OrthoMCL, which generates orthologs from multiple species using the MCL algorithm (Van, 2000). The MCL algorithm is known to be effective for detecting protein families with especially complicated domain structure (Enright *et al*., 2002). It is a graph partitioning algorithm, based on network flow, which is to simulate the flow within a graph such that the flow is encouraged where the current is strong, but it is discouraged where the current is weak.

We here introduce the ReMark tool that clusters orthologs flexibly through adjusting a parameter according to the user's interest. The fundamental algorithms and its evaluation were introduced in the previous work (Kim *et al*., 2007, 2008). Based on reciprocal BLAST best hits of gene pairs between two genomes, the method utilizes, to get more reliable results, the diagonal scores in the matrix of the initial ortholog clusters and clusters orthologs flexibly through being controlled naturally by MCL with a selected inflation factor. Users can therefore select the fitting state of orthologous protein clusters by regulating the inflation factor according to their research interests.

2 IMPLEMENTATION

The ReMark provides multiple, flexible approaches to cluster orthologous proteins. The ReMark software program works in two steps (shown in Fig. 1). In the first step, *n* genomes (protein sequence sets in the FASTA format) of our interest are selected. Next is to cluster orthologous proteins recursively using our recursive clustering algorithm from the initial table with reciprocal best hits derived by our software program (RecursiveClustering.java) written in Java. The clustering recursively detects and merges gene pairs with identical genes. In this step, score matrices are constructed from the initial clusters produced in the previous step. The second step is to split the initial clusters into more refined, tighter and consistent clusters using the MCL algorithm implemented by our software program (MarkovClustering.java) written in Java. In this step, the score matrices produced in the previous step are transformed into Markov matrices to simulate random walks on a graph. Finally, the terminal ortholog clusters are produced through adjusting its inflation factor.

To demonstrate how to use the ReMark tool, we have downloaded 12 genomes from NCBI at ftp://ftp.ncbi.nih.gov/genomes/. The dataset was used by the Recursive program in ReMark. After executing the program, initial clusters (score matrices) were generated. We computed the initial clusters with the MCL program in ReMark adjusting the inflation factor from 1.3 to 2.0. We obtain a group of protein clusters after running MCL with the score matrices.

3 DISCUSSION

Chen *et al*. (2006) showed that both methods of InParanoid and OrthoMCL have the best overall performance of 10 others using a

*To whom correspondence should be addressed.

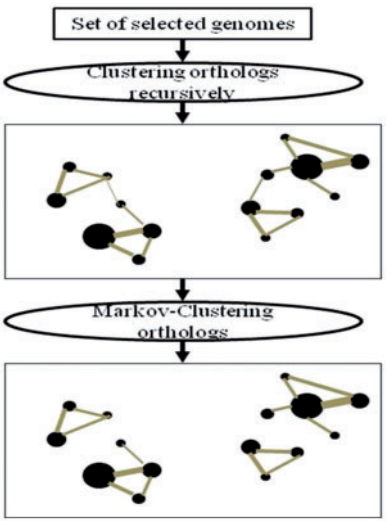

Fig. 1. Both main algorithms in the ReMark tool. Circles represent genes (proteins) with self-BLAST best scores, and the circle sizes mean the self-similarity, represented by scores, of amino acid sequences of genes themselves. Links represent the reciprocal BLAST best scores of gene pairs between two genomes, and the link thickness means the similarity of amino acid sequences of gene pairs (orthologs).

statistical technique called latent class analysis (LCA) (Hui *et al*., 1998). However, there is no gold standard datasets to evaluate ortholog clusters as they mentioned. According to each strategy, the results of the 10 methods are different each other. Resampled Inference of Orthology (RIO) (Zmasek *et al.*, 2002) has the least FP (1-Specificity) rate, and euKaryotic Orthologous Groups (KOG) (Tatusov *et al*., 2003) the least FN (1-Sensitivity) rate. Each strategy has its positive and negative points. Nevertheless, in order to identify which method can generate more accurate results, Chen *et al.* (2006) notably investigated the consistency of OrthoMCL and KOG with enzyme commission assignments, concerning protein function and domain architectures. By an intuitive measure similar to this (Kim *et al*., 2007, 2008), based on the consistency of Kegg Orthology (KO) database (Kanehisa *et al*., 2004, 2006) with the ortholog group tables containing curated orthologous genes extracted from metabolic and regulatory pathways, we tested how much our results are consistent with KO using InParanoid as the baseline approach. We then recognized that our method performed significantly better than InParanoid as investigated, changing an inflation factor in the condition of not including inparalogs.

In this study, we selected 12 genomes in different domains (Table 1) to test our method more extensively, and made three different datasets consisting of six genomes and three different datasets comprising nine genomes (Table 2). In this experiment, we compared ours, produced by ReMark, with results produced by InParanoid version 4.1 with default conditions. The inparalogs are in this time included in InParanoid's results unlike the previous test (Kim *et al*., 2007, 2008). As shown in Figures 2 and 3, users can get better ortholog clusters with proper inflation factors. According to the size of selected genomes, the factors can be modulated to optimize the quality of ortholog clusters. We can generally get almost the same as or better results than InParanoid's at the inflation factor 1.4.

Table 1. Different domains including 12 genomes from 12 species to test the ReMark method more extensively

Eukaryota	
ECU	*Encephalitozoon_cuniculi*_uid155 (NC_003229)
SPO	*Schizosaccharomyces_pombe*_uid127 (NC_001326)
SCE	*Saccharomyces_cerevisiae*_uid128 (NC_001133)
Bacteria	
AAE	*Aquifex_aeolicus*_VF5_uid57765 (NC_000918)
SYN	*Synechocystis*_PCC_6803_uid57659 (NC_000911)
TMA	*Thermotoga_maritima*_MSB8_uid57723 (NC_000853)
Gamma	
ECO	*Escherichia_coli*_K_12_substr__DH10B_uid58979 (NC_010473)
HIN	*Haemophilus_influenzae*_86_028NP_uid58093 (NC_007146)
YPE	*Yersinia_pestis*_Angola_uid58485 (NC_010157)
Gramplus	
CAC	*Clostridium_acetobutylicum*_ATCC_824_uid57677 (NC_001988)
LLA	*Lactococcus_lactis*_Il1403_uid57671 (NC_002662)
SPY	*Streptococcus_pyogenes*_M1_GAS_uid57845 (NC_002737)

Table 2. Datasets of InParanoid (IP) and ReMark (RM) comprising different domains

Set #1 (IP1, RM1)	Eukaryota, Bacteria	Six genomes
Set #2 (IP2, RM2)	Bacteria, Gamma	
Set #3 (IP3, RM3)	Bacteria, Gramplus	
Set #4 (IP4, RM4)	Eukaryota, Bacteria, Gamma	Nine genomes
Set #5 (IP5, RM5)	Eukaryota, Bacteria, Gramplus	
Set #6 (IP6, RM6)	Bacteria, Gamma, Gramplus	

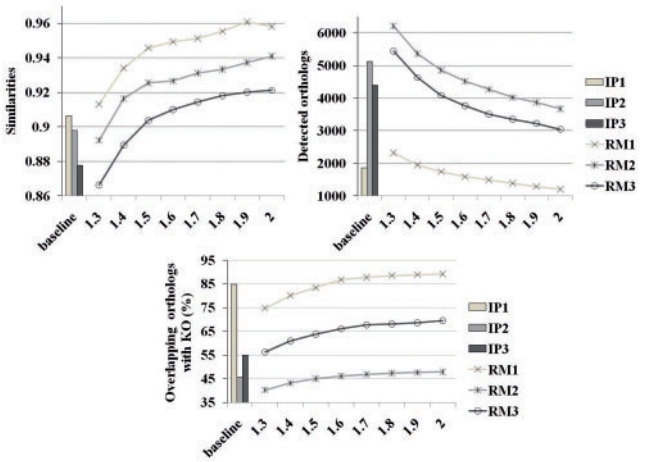

Fig. 2. Bar graphs representing the results (IP1, IP2 and IP3) of InParanoid and line graphs representing the results (RM1, RM2 and RM3) of ReMark. The accuracy of RM1 in Set #1 is better than IP1 from the inflation factor 1.3 to the factor 2.0 (about 1–6% better) and has more orthologs (24.6% at 1.3 and 4.6% at 1.4). The accuracy of RM2 in Set #2 is almost the same as IP2 at 1.3 and better than IP2 from the factor 1.4 to 2.0 (about 1–4%), and has more orthologs (21.1% at 1.3 and 4.6% at 1.4). The accuracy of RM3 in Set #3 is better than IP3 from 1.4 to 2.0 (about 1–4%) and has more orthologs (24% at 1.3 and 5.5% at 1.4). The overlapping orthologs with KO have distribution of about 40–90%.

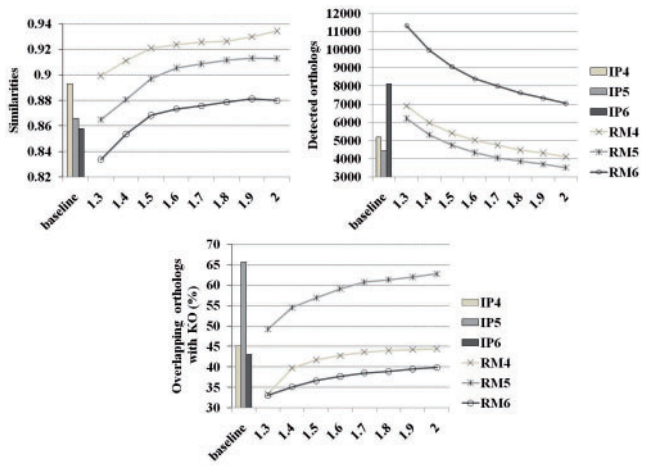

Fig. 3. Bar graphs representing the results (IP4, IP5 and IP6) of InParanoid and line graphs representing the results (RM4, RM5 and RM6) of ReMark. The accuracy of RM4 in Set #4 is better than IP4 from 1.3 to 2.0 (about 1–3% better) and has more orthologs (32.9% at 1.3, 14.9% at 1.4 and 4% at 1.5). The accuracy of RM5 in Set #5 is almost the same as the IP5 at 1.3, and better than IP5 from 1.4 to 2.0 (about 1–5% better), and has more orthologs (40.1% at 1.3, 19.8% at 1.4 and 6.9% at 1.5). The accuracy of RM6 in Set #6 is almost the same as the IP6 at 1.4, and better than IP6 from 1.5 to 2.0 (about 1–3% better), and has more orthologs (39.4% at 1.3, 22.6% at 1.4, 11.5% at 1.5 and 3.4% at 1.6). The overlapping orthologs with KO have distribution of about 30–60%.

The COGs are groups to cluster orthologs among multiple genomes without any premature threshold. However, the quality of ortholog clusters depends on the BLAST algorithm, which may not give accurate results between evolutionary distant species. The COG method dependant on the reciprocal BLAST best hits may therefore need the manual analysis of biologists. This takes time-consuming processes. To overcome this difficulty, both InParanoid and OrthMCL made automatic ortholog clusters with taking a premature threshold. However, applying the threshold could raise sensitivity, but reduce specificity. That is, this may lose real true positive orthologs prematurely. ReMark, therefore, uses the natural threshold regulated by the MCL with adjusting inflation factors, gaining 3–40% more orthologs than InParanoid. In addition, the ReMark uses diagonal components that have self-BLAST best scores in contrast with OrthoMCL. The reason we use the diagonal terms here is that the node with a high return weight can give an effect for changing cluster granularity (Van, 2000). It would be natural to apply the diagonal terms since the self-score of each gene is different according to the size of the amino acid sequences of a gene considered. In our experiment, it was observed that the ortholog clusters, generated with diagonal scores, were more consistent with KO than without, showing about 7% more accuracy (Kim *et al*., 2007).

In conclusion, this method clusters orthologs flexibly and more accurately through being controlled naturally by employing MCL with an inflation factor and diagonal components that are self-BLAST best scores.

ACKNOWLEDGEMENT

Many thanks to Mr Joohwan Lee at Intelligent Systems Lab in Sejong University for testing and feedback on the software program described here.

Funding: The Priority Centers Program of the National Research Foundation of Korea (NRF), which is funded by the Ministry of Education, Science and Technology (No. 2009-0093821).

Conflict of Interest: none declared.

REFERENCES

Altschul,S.F. *et al*. (1990) Basic local alignment search tool. *J. Mol. Biol*., **215**, 403–410.
Chen,F. *et al*. (2006) OrthoMCL-DB querying a comprehensive multi-species collection of ortholog groups. *Nucleic Acids Res*., **34**, D363–D368.
Enright,A.J. *et al*. (2002) An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Res*., **30**, 1575–1584.
Hui,S.L. and Zhou,X.H. (1998) Evaluation of diagnostic tests without gold standards. *Stat. Methods Med. Res.*, **7**, 354–370.
Kanehisa,M. *et al*. (2004) The KEGG resource for deciphering the genome. *Nucleic Acids Res*., **32**, D277–D280.
Kanehisa,M. *et al*. (2006) From genomics to chemical genomics: new developments in KEGG. *Nucleic Acids Res*., **34**, D354–D357.
Kim,S. (2007) Clustering methods for finding orthologs among multiple species. Chungbuk National University. Ph.D. Thesis, August 2007.
Kim,S. *et al*. (2008) Clustering orthologous proteins across phylogenetically distant species. *Proteins: Struc. Funct. Bioinformatics*, **71**, 1113-1122.
Li,L. *et al*. (2003) OrthoMCL: identification of ortholog groups for eukaryotic genomes. *Genome Res.*, **13**, 2178–2189.
Remm,M. *et al*. (2001) Automatic clustering of orthologs and in-paralogs from pairwise species comparisons. *J. Mol. Biol.*, **314**, 1041–1052.
Tatusov,R.L. *et al*. (1997) A genomic perspective on protein families. *Science*, **278**, 631–637.
Tatusov,R.L. *et al*. (2003) The COG database: an updated version includes eukaryotes. *BMC Bioinformatics*, **4**, 41.
Tatusov,R.L. *et al*. (2003) The COG database: an updated version includes eukaryotes. *BMC Bioinformatics*, **4**, 41.
Van Dongen,S. (2000) Graph clustering by flow simulation. Ph.D. Thesis, University of Utrecht, The Netherlands.
Zmasek,C.M. and Eddy,S.R. (2002) RIO: analyzing proteomes by automated phylogenomics using resampled inference of orthologs. *BMC Bioinformatics*, **3**, 14.