Computational Biology and Chemistry 48 (2014) 64–70

Contents lists available at ScienceDirect

Computational Biology and Chemistry

journal homepage: www.elsevier.com/locate/compbiolchem

Research article

Subgrouping Automata: Automatic sequence subgrouping using phylogenetic tree-based optimum subgrouping algorithm

Joo-Hyun Seo[a,c,1], Jihyang Park[a,2], Eun-Mi Kim[a,3], Juhan Kim[a,4], Keehyoung Joo[b], Jooyoung Lee[c], Byung-Gee Kim[a,*]

[a] School of Chemical and Biological Engineering, Seoul National University, Seoul 151-742, Republic of Korea
[b] Center for Advanced Computation, Korea Institute for Advanced Study, Seoul 130-722, Republic of Korea
[c] School of Computational Sciences, Korea Institute of Advanced Study, Seoul 130-722, Republic of Korea

ARTICLE INFO

Article history:
Received 10 June 2013
Received in revised form 12 October 2013
Accepted 23 November 2013

Keywords:
Subgrouping
Protein family discrimination
Optimum subgrouping node
Phylogenetic tree
Statistical analysis

ABSTRACT

Sequence subgrouping for a given sequence set can enable various informative tasks such as the functional discrimination of sequence subsets and the functional inference of unknown sequences. Because an identity threshold for sequence subgrouping may vary according to the given sequence set, it is highly desirable to construct a robust subgrouping algorithm which automatically identifies an optimal identity threshold and generates subgroups for a given sequence set. To meet this end, an automatic sequence subgrouping method, named 'Subgrouping Automata' was constructed. Firstly, tree analysis module analyzes the structure of tree and calculates the all possible subgroups in each node. Sequence similarity analysis module calculates average sequence similarity for all subgroups in each node. Representative sequence generation module finds a representative sequence using profile analysis and self-scoring for each subgroup. For all nodes, average sequence similarities are calculated and 'Subgrouping Automata' searches a node showing statistically maximum sequence similarity increase using Student's *t*-value. A node showing the maximum *t*-value, which gives the most significant differences in average sequence similarity between two adjacent nodes, is determined as an optimum subgrouping node in the phylogenetic tree. Further analysis showed that the optimum subgrouping node from SA prevents under-subgrouping and over-subgrouping.

© 2013 Published by Elsevier Ltd.

1. Introduction

The generation of subgroups containing functionally relevant sequences based on sequence similarity or identity can give good information for functional discrimination of sequences in a given set of sequences (Heger and Holm, 2000). Through clustering or subgrouping, functions of anonymous protein sequences can be easily inferred from the functions of other sequences in the same cluster (or subgroup) or the sequences in neighbor subgroups. A homologous sequence set for clustering can be prepared by text mining or several search methods such as BLAST (Altschul et al., 1990) or profile analysis (Altschul et al., 1997; Eddy, 1998). When we assume that the functions of homologous proteins are diversified along time owing to point mutation, deletion, insertion, multimerization and duplication, the function of the collected anonymous sequences in a given sequence set can be predicted more in detail by the functional inference based on phylogenomic analysis method (Eisen, 1998). When the functions of the several members in all subgroups in the phylogenetic tree are revealed by experiments, we can roughly identify the function of anonymous sequences in a given sequence set. Then the next remaining question is how we can make functionally meaningful subgroups from large set of sequences (Eisen, 1998; Eisen and Fraser, 2003; Krause et al., 2002). As summarized by Lee et al. (2010), developed algorithms can be categorized into three main types, i.e. phylogenomics, pattern recognition and clustering. According to Lee et al., SCI-PHY, which utilize pattern recognition and clustering, gives better results in protein function prediction than sequence-only methods such as Secator (Wicker et al., 2001), Ncut (Abascal and Valencia, 2002) and CD-HIT (Li and Godzik, 2006). However, these methods need conserved domains (for pattern recognition) or pre-determined sequence identity cut-off (for clustering). Some methods use mathematical models. Brown introduced model-based sequence clustering method utilizing Dirichlet process (Brown, 2008). This method is reported to show

* Corresponding author. Tel.: +82 2 880 6774; fax: +82 2 874 1206.
E-mail address: byungkim@snu.ac.kr (B.-G. Kim).
[1] Current address: Samsung Advanced Institute of Technology, Nongseo-dong, Giheung-gu, Yongin-si, Gyoenggi-do 446-712, Republic of Korea.
[2] Current address: Samsung Corning Precision Materials, Yongdu-ri, Tangjeong-myeon, Asan-si, Chungcheongnam-do 336-841, Republic of Korea.
[3] Current address: Joint BioEnergy Institute, Emeryville, CA 94608, USA.
[4] Current address: CIRES, University of Colorado, Boulder, CO 80309, USA.

1476-9271/$ – see front matter © 2013 Published by Elsevier Ltd.
http://dx.doi.org/10.1016/j.compbiolchem.2013.11.004

good group purity and VI score, which divides the given sequence set into functionally different subgroups. However, according to Andreopoulos et al. (2009), most models used in model-based clustering methods are often oversimplified, so that leading to inaccurate result. In addition, another disadvantage of model-based method is slow processing time for large sequence sets.

From the standpoint of enzymologists, bioinformatical methods should be feasible regardless of the level of input sequence set. Input sequence set could consist of sequences in superfamily, family or subfamily level. If the sequence set consists of superfamily level, sequences may be very diverse to draw conserved domains. In addition, sequence identity cut-off for sequence subgrouping could be never known for sequence set collected from database by utilizing any homology search method. Therefore, sequence clustering algorithm should produce family or subfamily sequence sets from superfamily or family-level sequence set without prior knowledge of sequence identity cut-off.

In this work, we designed a versatile algorithm for subclassification of input sequence set, which uses sequence comparison and clustering method. Because there are many outperforming sequence alignment methods, we focused on the development of algorithm to generate subgroups utilizing the sequence alignment result produced from other multiple sequence alignment methods.

2. Materials and methods

2.1. Data set

Sequences of β-alanine:pyruvate aminotransferase, γ-aminobutyrate aminotransferase, L-ornithine aminotransferase and lysine decarboxylase were searched with EC number and retrieved from BRENDA. Redundant sequences were removed by sequence clustering with 100% threshold using CD-HIT. 97 sequences of β-alanine:pyruvate aminotransferase, 272 sequences of γ-aminobutyrate aminotransferase, 72 sequences of L-ornithine aminotransferase and 111 sequences of lysine decarboxylase were used for subgrouping.

In the case of branched-chain aminotransferase (bcAT), sequences of bcATs were searched using the EC number of bcAT (2.6.1.42) at RefSeq database in NCBI. Putative sequences and fragment sequences were removed from the search result. To remove the redundant sequences, sequence clustering with the sequence identity threshold of 100% was performed using CD-HIT. A total of 691 sequences were gathered. The sequence subgrouping using the developed subgrouping algorithm was performed for the final 691 sequences. For aspartate ammonia-lyase, sequence sets were searched with EC number, and retrieved from BRENDA. Redundant sequences were removed by sequence clustering with 100% threshold using CD-HIT. HMM profile was built using 'hmmbuild' and 'hmmcalibrate' program in HMMER package. 843 microbial genomes (as of March 2009) were searched using the generated profile. From the result of the search for each microbial genome, hit sequences scoring over default *E*-value of 10 were collected. Sequence clustering was performed to remove redundant sequences using CD-HIT with 100% threshold. Final 379 sequences were used for subgrouping.

In the case of (*S*)-2-aminoadipate semialdehyde dehydrogenase (aasDH) sequence sets, there are two kinds of sequences in BRENDA database. The sequences of approximately 400 amino acids were chosen, but the number of the sequences was too small. Therefore, simple BLASTP search was performed with aasDH from *Stenotrophomonas* sp. SKA14 in NCBI and top 100 sequences were retrieved. Redundant sequences were removed by sequence clustering with 100% threshold using CD-HIT. Final 99 sequences were remained and used for subgrouping.

The sequences for the subgrouping analysis of aminotransferase group I, II and all the other aminotransferase groups were manually collected from the NR database. Putative sequences and fragment sequences were manually removed. BcAT sequences were re-collected manually with the same method for sequences of other aminotransferase subgroups. Sequence set of all aminotransferases consists of 11 sequences of alanine aminotransferase, 152 sequences of aspartate aminotransferase, 39 sequences of aromatic aminotransferase, 22 sequences of histidinol-phosphate aminotransferase, 16 sequences of ω-aminotransferase, 10 sequences of L-ornithine aminotransferase, 55 sequences of *N*-acetyl-L-ornithine aminotransferase, 129 sequences of 7,8-diaminopelargonate aminotransferase, 48 sequences of γ-aminobutyrate aminotransferase, 10 sequences of D-alanine aminotransferase, 53 sequences of branched-chain aminotransferase, 22 sequences of phosphoserine aminotransferase and 50 sequences of serine aminotransferase. Aminotransferase group I consists of alanine aminotransferase, aspartate aminotransferase, aromatic aminotransferase and histidinol-phosphate aminotransferase. Aminotransferase group II consists of ω-aminotransferase, L-ornithine aminotransferase, *N*-acetyl-L-ornithine aminotransferase, 7,8-diaminopelargonate aminotransferase and γ-aminobutyrate aminotransferase. Classification of aminotransferase in the work of Mehta et al. (Mehta et al., 1993) was adopted.

2.2. Algorithm and used programs

ClustalW 1.83 (Thompson et al., 1994) was used for the multiple alignment of input sequence sets. In this work, we used ClustalW because some input sequence sets have large number of sequences. Although we used ClustalW for multiple sequence alignment, any alignment programs can be used to generate multiple sequence alignment. "hmmbuild", "hmmcalibrate" and "hmmsearch" in HMMER package (Eddy, 1998) were used for profile building and profile search. Subgrouping algorithm and parsing algorithm for the result of ClustalW and programs in HMMER package were coded using Python programming language.

Two assumptions were made to construct the algorithm. (1) If sequence sets consist of a certain level (e.g. family level), subgrouping should be performed to discriminate sub-level (e.g. subfamily level); (2) at the optimum subgrouping node in phylogenetic tree, an average sequence similarity shows a maximum increase. Program starts with the 'PHYLIP' format of the tree and 'clustal' format of the alignment result. Therefore, if a user can make these two files, any alignment program can be used. In phylogenetic tree recognition and analysis, node number is calculated from 'PHYLIP' format of the phylogenetic tree. Starting from node index of zero, +1 is added when an opening parenthesis appears and −1 is added when a closing parenthesis appears. Therefore, the origin of the rooted tree is designated as node 1. After the node number assignment, the subgroup at node i is picked by selecting sequences between an opening parenthesis with node number $(i+1)$ and a closing parenthesis with node number i. The node number and the subgroup tag for each sequence can be assigned as in Fig. 1(a). For example, because node number of tree origin is 1, subgrouping at node 1 results in one subgroup and it contains all the sequences.

Next, subgroups at each node were identified. For instance, three subgroups can be identified at node 2 and four subgroups at node 3 (Fig. 1(a)). Therefore, sequence a in Fig. 1(a) can have the multiple subgroup tags of 3-1, 4-2 and so on. For sequences in each subgroup, to calculate the average sequence similarity, pairwise sequence similarities were calculated for all the sequence

Download English Version:

https://daneshyari.com/en/article/15143

Download Persian Version:

https://daneshyari.com/article/15143

Daneshyari.com