BIOINFORMATICS APPLICATIONS NOTE Vol. 22 no. 13 2006, pages 1658–1659
doi:10.1093/bioinformatics/btl158

Sequence analysis

Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences

Weizhong Li* and Adam Godzik

Burnham Institute for Medical Research, La Jolla, CA 92037, USA

Received on March 23, 2006; revised and accepted on April 20, 2006
Advance Access publication May 26, 2006
Associate Editor: Golan Yona

ABSTRACT

Motivation: In 2001 and 2002, we published two papers (*Bioinformatics*, **17**, 282–283, *Bioinformatics*, **18**, 77–82) describing an ultrafast protein sequence clustering program called cd-hit. This program can efficiently cluster a huge protein database with millions of sequences. However, the applications of the underlying algorithm are not limited to only protein sequences clustering, here we present several new programs using the same algorithm including cd-hit-2d, cd-hit-est and cd-hit-est-2d. Cd-hit-2d compares two protein datasets and reports similar matches between them; cd-hit-est clusters a DNA/RNA sequence database and cd-hit-est-2d compares two nucleotide datasets. All these programs can handle huge datasets with millions of sequences and can be hundreds of times faster than methods based on the popular sequence comparison and database search tools, such as BLAST.

Availability: http://cd-hit.org

Contact: liwz@sdsc.edu

1 INTRODUCTION

In recent years, the amount of biological sequence data has been growing explosively, which has imposed growing difficulties on analyzing them. The complexity of many sequence analyses is of the order of n^2, where n is the number of sequences to be considered. One such example is protein sequence clustering, which groups similar proteins into clusters based on their sequence similarities. To address this computational challenging problem, we developed a novel method and published a program, cd-hit (Li *et al*., 2001, 2002a), which can efficiently handle huge databases. For example, it takes only a few hours to cluster the NCBI-nr with ∼3 million proteins on a single high-end workstation.

Since its release, cd-hit has been used by many groups, including Uniprot (Apweiler *et al*., 2004) and PDB (Bourne *et al*., 2004), in various research fields. In our group, we applied it to generate non-redundant protein datasets to reduce the database search efforts and to improve the homology detection sensitivity (Li *et al*., 2002a).

*To whom correspondence should be addressed.

The algorithm behind cd-hit is short word filtering, which can determine that the similarity between two sequences is below a certain value without performing an actual sequence alignment. This algorithm is not limited to protein sequence clustering; it can also be applied to many other analyses that involve a large amount of sequence comparisons.

Here, we present several new programs based on cd-hit algorithm including cd-hit-2d, cd-hit-est and cd-hit-est-2d. Cd-hit-2d compares two protein datasets and reports similar matches between them; cd-hit-est clusters a DNA/RNA database and cd-hit-est-2d compares two nucleotide datasets. The common advantages of these programs are ultrahigh speed and the ability to handle huge databases.

2 ALGORITHMS

2.1 Short word filtering

The details of the algorithm for short word filtering were described in our earlier papers (Li *et al*., 2001, 2002a). In short, the minimum number of identical short substrings, called 'words', such as dipeptides, tripeptides and so on, shared by two proteins is a function of their sequence similarity. We calculated this function by analytical and large-scale statistical analyses. Therefore, we can effectively estimate that the similarity of two sequences is below a certain threshold by simple word counting and without an actual sequence alignment. For nucleotide sequences, we can also obtain such a short word requirement by a similar combination of analytical and statistical analyses.

We implemented this idea using an index table. For instance, the total number of possible pentapeptides is only 21^5 (each position has 21 possibilities, 20 amino acids plus 'X'), and such an index table requires only 4 million entries, which just matches the RAM size of current computers. Index tables maximize the speed of short word counting. Details regarding how to choose an appropriate short word are documented in the cd-hit user's guide.

2.2 Cd-hit and cd-hit-est clustering

The original cd-hit program clusters a protein database, and its variant, cd-hit-est, clusters a DNA/RNA database. For eukaryotic genes, long introns can cause long gaps in sequence alignments, which significantly reduces the efficiency of short word filtering. So,

Table 1. Example runs of cd-hit programs

Program	db1 (No. of seqs and letters/bps)	db2 (No. of seqs and letters/bps)	Threshold (%)	Results	CPU
cd-hit	NCBI-nr, February 2006 3 287 897 seqs and 1 128 123 096 letters		90	1 982 695 non-redundant seqs	7h 50m
cd-hit	NCBI-nr, February 2006		60	1 233 210 non-redundant seqs	61h 30m
cd-hit-2d	Swissprot February 2006 190 405 seqs and 71 123 307 letters	NCBI-pdbaa February 2006 23 626 seqs 5 358 851 letters	60	19 408 seqs in pdbaa match seqs in swissprot	7.5 m
cd-hit-2d	Swissprot February 2006	NCBI-nr February 2006	90	692 727 seqs in nr match seqs in swissprot	1h 53m
cd-hit-est	Human EST of the month, February 2006 60 460 seqs and 16 109 549 bp		95	34 029 non-redundant seqs	3m 50s
cd-hit-est	Human EST Apr. 2005 6 056 958 seqs and 3 219 654 196 bps		95	2 782 847 non-redundant seqs	139h
cd-hit-est-2d	Human mRNA Refseq, Apr. 2005 29 168 seqs and 76 489 646 bps	Human EST of the month, Feb. 2006	95	2 244 seqs in db2 match seqs in db1 in either strand	6m 30s
cd-hit-est-2d	Human mRNA Refseq, Apr. 2005	Human EST Apr. 2005	95	2 799 899 seqs in db2 match seqs in db1 in either strand	7h 44m

practically, cd-hit-est can be applied only for non-intron-containing sequences, such as ESTs.

The clustering algorithm in both cd-hit and cd-hit-est is a greedy incremental clustering algorithm. Briefly, sequences are first sorted in order of decreasing length. The longest sequence becomes the representative of the first cluster. Then, each remaining sequence is compared with the representatives of existing clusters. If the similarity with any representative is above a given threshold, it is grouped into that cluster. Otherwise, a new cluster is defined with that sequence as the representative. For each sequence comparison, short word filtering is applied to the sequences to confirm whether the similarity is below the clustering threshold. If this cannot be confirmed, an actual sequence alignment is performed.

2.3 Cd-hit-2d and cd-hit-est-2d

Program cd-hit-2d compares two protein databases and identifies similar sequences between them above a certain threshold. Cd-hit-est-2d works for two DNA/RNA databases. For the same reason that we mentioned earlier, cd-hit-est-2d is a practical choice only for non-intron-containing sequences.

Given two databases, db1 and db2, cd-hit-2d or cd-hit-est-2d works in a straightforward way. Sequences in db1 are first sorted in order of decreasing length. Then, each sequence in db2 is compared with db1 from the top (the longest one), and if the similarity to any one in db1 is above the threshold, this sequence is attached to the matched one in db1. At the end of comparing, the program reports matches between db1 and db2 and also outputs a list of proteins in db2 that is not similar to any sequence in db1.

3 RESULTS

The cd-hit package was written in C++ and was tested on Linux systems. It is distributed as an open source package and can be run on almost all systems that support C++ with little or no modification.

Some example runs are listed in Table 1. All these examples were performed on a Linux workstation with dual 3.0 GHz Xeon processors and 4 GB RAM. The programs used only one processor. For example, cd-hit took <8 h to cluster the NCBI-nr with more than 3.2 million proteins at 90% sequence identity level. Cd-hit-est-2d took a similar amount of time to identify the matches above 95% identity in both strands between human ESTs with ∼6 million sequences and ∼30 thousand human mRNAs.

Many options and functions were implemented for the users to control the clustering or comparing process. For example, a useful function is the incremental clustering that offers not only a higher speed but also a stable clustering structure for regularly updated databases. Full set of options are described in the documentation for the program.

In addition to the programs described above, the package contains several utility tools. Some tools help analyze, sort and format the clustering results. One script runs clustering in parallel mode by distributing jobs on a computer cluster (details can be found in the user's guide). The cd-hit package will be under regular maintenance and further development, which will focus on the efficiency at low sequence similarity thresholds. We are also open to adding new functionalities as suggested by users.

ACKNOWLEDGEMENTS

Funding to pay the Open Access publication charges was provided by the institutional funds of Burnham Institute for Medical Research.

Conflict of Interest: none declared.

REFERENCES

Apweiler,R. *et al*. (2004) UniProt: the Universal Protein knowledgebase. *Nucleic Acids Res*., **32** (Database issue), D115–D119.

Bourne,P.E. *et al*. (2004) The distribution and query systems of the RCSB Protein Data Bank. *Nucleic Acids Res*., **32** (Database issue), D223–D225.

Li,W. *et al*. (2001) Clustering of highly homologous sequences to reduce the size of large protein databases. *Bioinformatics*, **17**, 282–283.

Li,W. *et al*. (2002a) Sequence clustering strategies improve remote homology recognitions while reducing search times. *Bioinformatics*, **15**, 643–649.

Li,W. *et al*. (2002b) Tolerating some redundancy significantly speeds up clustering of large protein databases. *Protein Eng*., **18**, 77–82.