June 27, 2017

Abstract

1 Introduction

Sequence clustering is a classic problem in biology. Clustering of complete biological sequences generates de-novo taxonomy and reduces the redundancy of biological database.

We developed a new sequence-clustering algorithm that can scale to hundred of millions protein sequences. Our algorithm is more sensitive, more specific, and faster than known published sequence clustering algorithms.

2 Related works

Clustering of biological sequences has a long history. Holm and Sander [1998] observed that two sufficiently long protein sequences that share no common decapeptide must have at most 90% sequence identity. Based on this observation, Holm and Sander [1998] developed a method called nrdb90 that clusters protein sequences at 90% sequence identity. However, nrdb90 is slow and works only for sequence identity of at least 90%. Li et al. [2001] observed that two protein sequences of lengths L_1 and L_2 must share at least x peptides of length y in order to share at least z% sequence identity, where L_1, L_2, x, y, and z are mathematically interdependent variables. Li et al. [2001] therefore generalized the decapeptide filter into short-word filter. Based on this generalization, Li et al. [2001] developed a method called CD-HI that clusters protein sequences at 70% or more sequence identity. Li et al. [2001] introduced the notion of greedy incremental update, where sequences are longest-first sorted and each sequence either belongs to an already formed cluster or forms a new cluster. Since then, nearly all clustering methods used greedy incremental update, so we will assume such unless explicitly stated otherwise. CD-HI is both significantly faster and can cluster at lower sequence identity than nrdb90 without compromising neither sensitivity nor specificity. However, the exponential increase in the size of a typical protein database such as nr even made CD-HI too slow to be practical. Li et al. [2002] extended the work of Li et al. [2001] by probabilistically filtering out sequence pairs that are likely to share less than a certain sequence-identity cutoff. Li et al. [2002] observed that two protein sequences of lengths L_1 and L_2 are not necessarily , but likely, to share at least x peptides of length y in order to share at least z% sequence identity. Based on this extension, Li et al. [2002] developed a method called CD-HIT that clusters protein sequences at 50% or more sequence identity. At the same sequence-identity cutoff, CD-HIT produces 0.4% more clusters than but is about 100 times faster than CD-HI. Li and Godzik [2006] observed that the algorithm in CD-HIT can be used for clustering nucleotide sequences and for comparing two biological databases. Fu et al. [2012] used OpenMP to parallelize CD-HIT.

Edgar [2010] developed a method called usearch which performs database search by sequence identity. In addition of using the short-word filter used by CD-HIT, usearch only searches a fixed maximum number of top hit target sequences per query sequence. Edgar [2010] also developed uclust which clusters biological sequences using the distance computed by usearch. Usearch and uclust are both closed-source commercial programs, so Rognes et al. [2016] developed vsearch which is open-source and similar to usearch and uclust in terms of functionality and performance.

Clustering of short reads produced by Next-Generation Sequencing (NGS) is also an ongoing research topic. Zorita et al. [2015] developed starcode, an algorithm for clustering sequences at high global similarity cutoff based edit distance. Starcode is especially useful for detection and correction of sequencing errors.

Table 1: The sequences and structures of all monomeric proteins in PDB are given as input to each program. Each program is run with each intra-cluster similarity threshold to generate each set of clusters. For each set of clusters, the number of clusters characterized by intra-cluster TM scores inclusively below each threshold is tabulated. The intra-cluster TM score of a cluster is the lowest TM score between the representative sequence in the cluster as template and each represented sequence in the cluster.

Program	intra-cluster	Intra-cluster TM scores									
	similarity	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
mine	50	0	7	46	137	283	510	877	1403	2214	16575
linclust	50	0	29	118	253	402	656	1032	1565	2368	16964
CD-HIT	50	0	40	138	280	438	694	1107	1631	2445	15712
mine	70	0	7	42	131	253	444	761	1228	1960	18010
linclust	70	0	30	119	245	378	614	957	1446	2190	17828
CD-HIT	70	0	39	126	247	380	585	913	1390	2118	17668
mine	90	0	7	35	111	219	380	633	1022	1724	19400
linclust	90	0	23	89	188	316	502	796	1221	1922	19298
CD-HIT	90	0	29	114	223	342	514	772	1160	1852	19194

Table 2: All sequences in Uniref100_2017-03 are given as input to each program. Each program is run with each intra-cluster similarity threshold to generate each set of clusters.

Program	intra-cluster similarity	cluster count	amino-acid count	runtime

3 Novel key ideas

In greedy incremental update, a iterated sequence either forms a new cluster or is assigned to an existing cluster. However, such clustering decision can result in suboptimal solution. Instead, we delay the clustering decision until sufficient sequences are iterated over. Then, we make clustering decisions using an algorithm that is less greedy than the greedy incremental update. This reduction in greediness increases sensitivity.

In short-word filter, the similarity between two sequences is estimated by counting their common short words. However, a typical biological sequence has hundreds of short words. Instead, we transform short words into hash values and select a fixed number of lowest hash values. Then, we use the lowest hash values as short words. This reduction in short words reduces runtime.

Sequence identity is the number of matches divided by the length of the shorter sequence in an alignment. Usually, blosum62 matrix with linear gap model is used to compute such alignment. However, such computation of alignment is time-consuming. Moreover, deletions in the longer sequence are ignored. Therefore, normalization by the length of the shorter sequence can overestimate the true biological similarity. Conservation of biological function requires smaller number of matches of conserved residues than matches of non-conserved residues. However, smaller number of matches decreases sequence identity. Unfortunately, substitution matrices such as blosum62 tend to produce matches of conserved residues. Thus, sequence identity computed with a substitution matrix can underestimate the true biological similarity. Instead of using sequence identity to estimate true biological similarity, we used edit-similarity. The edit-similarity between a query and a target is defined as follows: the length of the target minus the minimum number of single-character edits required such that the target is a substring of the query, divided by the length of the target. Then, we use edit-similarity as sequence identity. Edit-similarity takes less time to compute and more correlated with true biological similarity. Therefore, the use of edit-similarity instead of sequence identity improves sensitivity, specificity, and runtime.

Program	intra-cluster similarity 90	70	50
mine	52581967	32658006	20735460
linclust	58268938	40473785	32077206
CD-HIT	53787568	33977706	timeout

4 Results and discussion

References

Liisa Holm and Chris Sander. Removing near-neighbour redundancy from large protein sequence collections. *Bioinformatics*, 14(5):423–429, 1998.

Weizhong Li, Lukasz Jaroszewski, and Adam Godzik. Clustering of highly homologous sequences to reduce the size of large protein databases. *Bioinformatics*, 17(3):282–283, 2001.

Weizhong Li, Lukasz Jaroszewski, and Adam Godzik. Tolerating some redundancy significantly speeds up clustering of large protein databases. *Bioinformatics*, 18(1):77–82, 2002.

Weizhong Li and Adam Godzik. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, 22(13):1658–1659, 2006.

Limin Fu, Beifang Niu, Zhengwei Zhu, Sitao Wu, and Weizhong Li. Cd-hit: accelerated for clustering the next-generation sequencing data. *Bioinformatics*, 28(23):3150–3152, 2012.

Robert C Edgar. Search and clustering orders of magnitude faster than blast. *Bioinformatics*, 26(19): 2460–2461, 2010.

Torbjørn Rognes, Tomáš Flouri, Ben Nichols, Christopher Quince, and Frédéric Mahé. Vsearch: a versatile open source tool for metagenomics. *PeerJ*, 4:e2584, 2016.

Eduard Valera Zorita, Pol Cuscó, and Guillaume Filion. Starcode: sequence clustering based on all-pairs search. *Bioinformatics*, page btv053, 2015.