Gene 529 (2013) 250–256

Contents lists available at ScienceDirect

Gene

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

Protein sequence comparison based on *K*-string dictionary

Chenglong Yu [a], Rong L. He [b], Stephen S.-T. Yau [c,*]

[a] Department of Mathematics, Statistics and Computer Science, University of Illinois at Chicago, IL 60607-7045,USA
[b] Department of Biological Sciences, Chicago State University, Chicago, IL,USA
[c] Department of Mathematical Sciences, Tsinghua University, Beijing, PR China

ARTICLE INFO

Article history:
Accepted 25 July 2013
Available online 9 August 2013

Keywords:
K-string
Sequence comparison
Frequency vector
Cardinality
Singular Value Decomposition

ABSTRACT

The current *K*-string-based protein sequence comparisons require large amounts of computer memory because the dimension of the protein vector representation grows exponentially with *K*. In this paper, we propose a novel concept, the "*K*-string dictionary", to solve this high-dimensional problem. It allows us to use a much lower dimensional *K*-string-based frequency or probability vector to represent a protein, and thus significantly reduce the computer memory requirements for their implementation. Furthermore, based on this new concept, we use Singular Value Decomposition to analyze real protein datasets, and the improved protein vector representation allows us to obtain accurate gene trees.

1. Introduction

With the development of biotechnology, more and more biological sequences have been acquired. The discovery of new protein sequences is accelerating, but many of these proteins show similarity to existing amino acid sequences. Sequence comparison problems arise when detecting the similarity of proteins, and explaining their phylogenetic relations as well as when handling the huge amount of data. Existing methods for sequence comparison can be classified into alignment-based methods and alignment-free methods. Alignment-based methods use dynamic programming, a regression technique that finds an optimal alignment by assigning scores to different possible alignments and picking the alignment with the highest score (Gotoh, 1982; Needleman and Wunsch, 1970; Smith and Waterman, 1981). However, the search for optimal solutions using sequence alignment turns out to be computationally difficult with large biological databases, especially when comparing three or more biological sequences at a time, i.e., multiple sequence alignment. Therefore, alignment-free approaches have been developed to overcome the critical limitations of alignment-based methods.

The recent reviews (Davies et al., 2008; Vinga and Almeida, 2003) on published methods of alignment-free sequence comparison report several concepts of distance measures, such as Markov chain models and Kullback–Leibler discrepancy (Wu et al., 2001), chaos theory (Almeida et al., 2001), Kolmogorov complexity (Li et al., 2001), decision tree induction algorithm (Huang et al., 2004), graphical representation (Liao and Wang, 2004; Randic et al., 2003; Yau et al., 2003), probabilistic measure (Pham and Zuegg, 2004; Yu et al., 2011a,b), and pseudo amino acid composition (Chou, 2011; Chou and Shen, 2009). Furthermore, sequence vector representation approaches without alignment are also prevalent, such as feature vector (Carr et al., 2010; Liu et al., 2006), moment vector (Yau et al., 2008; Yu et al., 2010, 2011a,b), and natural vector (Deng et al., 2011; Yu et al., 2013). Among all existing alignment-free methods, the *K*-string-based methods (Chu et al., 2004; Gao and Qi, 2007; Lu et al., 2008; Qi et al., 2004; Takahashi et al., 2009) have received substantial attention. Basically, the first step of these methods is, for a fixed integer *K*, to count the number of overlapping *K*-peptides in one protein sequence, and form a frequency or probability vector of dimension 20^K. Then using some probabilistic or optimization models these vectors are converted into more complicated composition vectors (Chan et al., 2012), but the dimension of the vectors remains unchanged in this process. Finally, the distance between two composition vectors is used to compute the distance between two taxa, and once the distances among all taxa are obtained, the phylogenetic trees can be reconstructed. These methods are able to provide good phylogenetic tree topologies for DNA or proteins; however, because large values needed to be chosen (see the discussion in Section 2), the resulting high memory usage becomes a disadvantage.

In this paper, we provide a novel concept, the "*K*-string dictionary", to solve this problem. It allows us to use a much lower dimensional frequency or probability vector to represent a protein, and thus significantly reduce the memory requirements for their implementation. Furthermore, after obtaining the lower dimensional frequency vectors, we use Singular Value Decomposition (SVD) to get an improved protein vector representation which allows us to obtain accurate gene trees. We have analyzed 290 proteins from 3 families and 50 beta-globin

Abbreviations: SVD, Singular Value Decomposition; MSA, multiple sequence alignment; ND1, NADH dehydrogenase 1.

* Corresponding author. Department of Mathematical Sciences, Tsinghua University, Beijing 100084, PR China. Tel.: +86 10 62787874; fax: +86 10 62798033.
E-mail address: yau@uic.edu (S.S.-T. Yau).

0378-1119/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.gene.2013.07.092

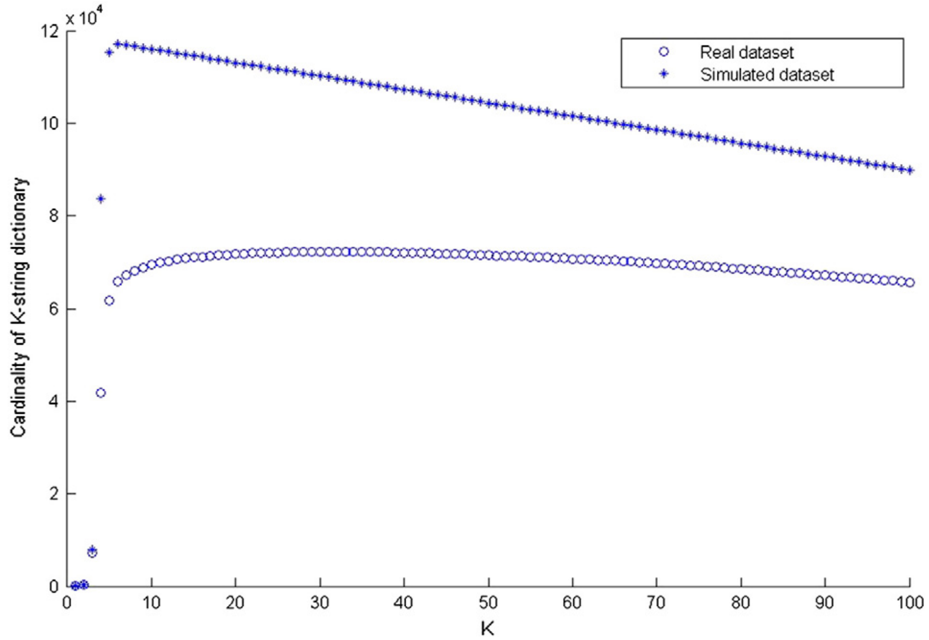

Fig. 1. The cardinalities of *K*-string dictionary of real and simulated datasets including 290 proteins.

proteins from different animal species using this method, and found it to be a powerful classification tool for proteins.

2. Materials and methods

2.1. Background on K-string frequency or probability vector

Given a protein sequence of length L, the frequency of appearances of a K-string $\alpha = a_1a_2, \ldots a_K$ in this sequence is defined as $f(\alpha)$, where α_i is an amino acid single-letter symbol. This frequency divided by the total number $(L - K + 1)$ of K-strings in the given protein sequence is defined as the probability $p(\alpha)$ of appearance of the K-string α in the sequence: $p(\alpha) = \frac{f(\alpha)}{L-K+1}$. For example, given a protein sequence (AMFAMCAMFS), $f(\alpha) = 2$ for 3-string $\alpha =$ (AMF), and $p(\alpha) = \frac{2}{10-3+1} = 0.25$.

Table 1
The cardinalities of *K*-string dictionary of real and simulated dataset.

	Cardinality	
K value	Real dataset	Simulated dataset
1	20	20
2	400	400
3	7186	8000
4	41703	83601
5	61792	115394
6	65733	117083
7	67214	116892
8	68182	116604
9	68898	116314
10	69450	116024
11	69895	115734
12	70255	115444
13	70551	115154
14	70804	114864
15	71012	114574
16	71188	114284
17	71343	113994
18	71482	113704
19	71607	113414
20	71720	113124

There are a total of $N = 20^K$ possible types of such K-strings for protein sequences. Thus the K-string frequency vector of one protein sequence is defined as $(f(\alpha_1), f(\alpha_2), \ldots, f(\alpha_N))$, and the corresponding K-string probability vector of one protein sequence is defined as $(p(\alpha_1), p(\alpha_2), \ldots, p(\alpha_N))$.

Many current alignment-free works are based on the K-string frequency or probability vectors as we mentioned in Section 1. However, the choice of suitable K has always been an important concern. The main problem is that the dimension of these vectors can quickly become large. For example, the dimension of the protein K-string frequency or probability vector for $K = 6$ is $20^6 = 64{,}000{,}000$. Trying to work with vectors of such a large dimension will exceed the memory limits of ordinary personal computers. Thus, when using these vectors, we cannot evaluate the results for larger K. To overcome this disadvantage, we propose a novel concept "*K*-string dictionary" to solve this problem.

2.2. K-string dictionary

The K-string dictionary of a group of protein sequences is the set of all K-strings existing in these sequences. Note that a set is a collection of distinct objects, so we only record repeated K-strings once in the dictionary. For example, given a group of two protein sequences (AMTHGS) and (MTHAKW), the 3-string dictionary for this group is the set {AMT, MTH, THG, HGS, THA, HAK, AKW}. The key point is that the cardinality of a K-string dictionary is far less than 20^K. This will significantly reduce the memory requirements for computer calculations.

For example, titin is currently the largest known protein; its human variant (GenBank No.: NP_001243779) consists of 34,350 amino acids (Minajeva et al., 2001). For example, we take $K = 10$, then titin has $34{,}350 - 10 + 1 = 34{,}341$ K-strings. Assume that we are dealing with 1000 big proteins like titin's size, and all 10-strings of them are totally different, then the cardinality of the 10-string dictionary of this group is $34{,}314 \times 1000 = 3.4341 \times 10^7$. However, this number is still far less than $20^{10} = 1.024 \times 10^{13}$.

2.3. The cardinality of K-string dictionary

Given a group of protein sequences, for different K, we have different K-string dictionaries. We will use the real and simulated protein