Genomics Proteomics Bioinformatics

www.elsevier.com/locate/gpb
www.sciencedirect.com

APPLICATION NOTE

CVTree: A Parallel Alignment-free Phylogeny and Taxonomy Tool Based on Composition Vectors of Genomes

Guanghong Zuo [*, §]

T-Life Research Center, Department of Physics, Fudan University, Shanghai 200433, China

Received 22 November 2020; revised 23 February 2021; accepted 6 March 2021
Available online 10 June 2021

Handled by Zhang Zhang

Abstract Composition Vector Tree (CVTree) is an alignment-free algorithm to infer phylogenetic relationships from genome sequences. It has been successfully applied to study phylogeny and taxonomy of viruses, prokaryotes, and fungi based on the whole genomes, as well as chloroplast genomes, mitochondrial genomes, and metagenomes. Here we presented the standalone software for the CVTree algorithm. In the software, an extensible parallel workflow for the CVTree algorithm was designed. Based on the workflow, new alignment-free methods were also implemented. And by examining the phylogeny and taxonomy of 13,903 prokaryotes based on 16S rRNA sequences, we showed that CVTree software is an efficient and effective tool for studying phylogeny and taxonomy based on genome sequences. The code of CVTree software can be available at https://github.com/ghzuo/cvtree.

KEYWORDS CVTree; Composition vector; Alignment-free; Dissimilarity matrix; Phylogeny; Taxonomy

Introduction

Comparative analysis of genome sequences is the fundamental approach for the phylogenetic study of biology. Traditionally, sequence comparison is based on pairwise, including global sequence alignment [1], local sequence alignment [2], and multiple sequence alignment (MSA). Software tools for sequence alignment, such as BLAST [3] and CLUSTAL [4], are the most widely used bioinformatics methods. The aligned sequences provide a very intuitive impression of the difference between sequences, especially for the sequences with high identity. However, the computation of an accurate MSA is an non-deterministic polynomial (NP)-hard problem. The MSA-based methods cannot be solved in a realistic time for the large datasets that are available today. Most MSA tools are based on heuristic algorithms. It has been found that alignment-based techniques are inaccurate in scenarios of low sequence identity [5,6]. Therefore, as an alternative solution to sequence alignment, many alignment-free approaches to sequence analysis have been developed in recent decades [6–10]. These methods are computationally less expensive than the alignment-based methods. Their scalability allows them to be applied to much larger datasets than conventional MSA-based methods.

Composition Vector Tree (CVTree) is a cluster of alignment-free methods based on subsequences of a defined length (named as k-string). They generated dissimilarity matrices (DMs) from a comparatively large collection of genome sequences for phylogenetic studies. The classical CVTree method was proposed by Prof. Bailin Hao and coworkers in 2004 [11]. In the classical CVTree algorithm, every genome sequence, including protein sequence, RNA, and DNA, was represented by a composition vector (CV), which was calculated by the difference between the

*Corresponding author.
E-mail: ghzuo@fudan.edu.cn (Zuo G).
[§]Current address: Wenzhou Institute, University of Chinese Academy of Sciences, Wenzhou 325001, China.

Peer review under responsibility of Beijing Institute of Genomics, Chinese Academy of Sciences / China National Center for Bioinformation and Genetics Society of China.
https://doi.org/10.1016/j.gpb.2021.03.006

frequencies of k-strings and the prediction frequencies by the Markov model. And the similarity between the two sequences was measured by the cosine of two CVs. The classical CVTree method was first developed to infer evolutionary relatedness of Bacteria and Archaea [11−14], and then successfully applied to fungi [15,16], viruses [17], chloroplasts [18], and mitochondria [19], as well as metagenomes [20,21]. After the proposal of the classical CVTree method, there are three versions of CVTree web server were released successively by our group [22−24]. The latest released CVTree web server, CVTree3 [24], is available from http://cvtree.online (Aliyun, Shenzhen, China).

In this study, we presented the standalone software for the CVTree algorithm. Due to the flexibility of the standalone software, the CVTree software is helpful for the researchers who are interested in the intermediate results (*e.g.*, the collection of CVs and DMs) or unwilling to upload their data to web server, as well as bioinformatics developers. In the CVTree software, the programs were redesigned in an object-oriented model. The OpenMP technique was employed to make the main programs parallel. An inbuilt automatic workflow helps users to obtain the phylogenetic tree from the Fasta files directly, and the intermediate result can be cached to avoid redundant calculation. Based on the scheme of the CVTree algorithm, other alignment-free phylogenetic methods based on the CVs were implemented [25]. Furthermore, by using CVTree software, we obtained the phylogeny of 13,903 prokaryotes based on their 16S rRNA sequences [26]. Interestingly, these CVTree methods are much faster than the alignment-based methods, and they are effective to obtain a taxonomy-compactible phylogenetic tree.

Algorithms and implementations

Scheme of CVTree

CVTree includes a cluster of alignment-free methods to obtain phylogenetic relationships based on genome sequences. **Figure 1** showed the scheme of the CVTree algorithm. There are three steps for the algorithm, *i.e.*, modeling the genomes to the CVs, calculating the DM from the CVs, and inferring a phylogenetic tree based on the DM. In the CVTree software, the classical CVTree CV generative model and algorithm were named as Hao model and Hao method, respectively, to honor Prof. Bailin Hao [27]. And in the Hao method, the genome sequences were cut into small k-strings. Then the CV of the genome was modeled by the frequencies of k-strings, including the lengths $k-2$, $k-1$, and k, based on a Markov model. The dissimilarity of the two genomes was measured by the cosine of the angle between two vectors. Finally, the phylogenetic tree was inferred by the neighbor-joint algorithm. Based on the scheme, other conventional dissimilarity methods, including Jaccard, Manhattan, and Euclidean, were integrated into the CVTree software [25]. Two CV models (*i.e.*, direct count model and Hao model) and an enhanced neighbor-joint tree method were also provided in the software. Users can compose the models and methods by the options of programs (see details in File S1).

Implementation

The CVTree software, written in standard C++, facilitates compilation compliant with CMake and execution on Linux/Unix, Macintosh, and Windows platforms. The CVTree software, including example data, documentation, and source codes, is freely available for academic use at https://github.com/ghzuo/cvtree.

The CVTree programs were designed in an object-oriented model. In the scheme of CVTree algorithm (Figure 1), there are four states for the information: genomes, CVs, DM, and phylogenetic tree. They were described by different classes in CVTree programs. In more detail, the k-strings were encoded into an unsigned long integer (64-bit

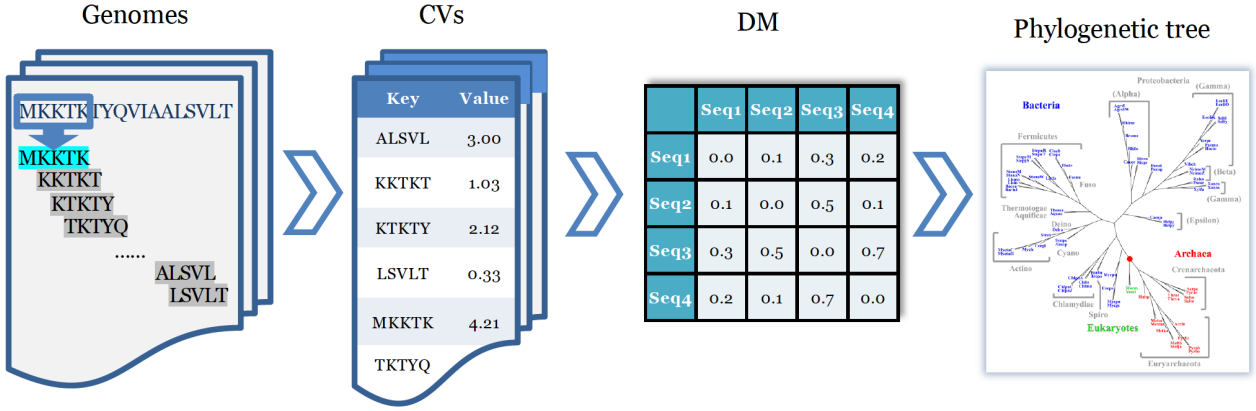

Figure 1 **Scheme of CVTree algorithm**
Four blocks indicate the four different states of the information, *i.e.*, genomes, CVs, DM, and phylogenetic tree, in the CVTree algorithm. Three chevrons indicate three steps to handle the information flow along with these four states, *i.e.*, from genomes to CVs, from CVs to a DM, and from a DM to a phylogenetic tree, in the CVTree algorithm. CV, composition vector; DM, dissimilarity matrix; Seq, sequence.

length) to improve efficiency. It is obvious that for a N length sequence consisting of m letters, when the length k of k-strings is large enough, the number of k-strings in the sequence, $N-k+1$, is much less than that of the types of k-strings, m^k. That is, the CV is sparse, *i.e.*, most of the dimensions are zero. Thus, only the non-zero dimensions were saved as key-value pairs in CVTree programs. All CVs were handled by associated arrays in the generation and by sorted sequential arrays in the calculation, respectively. The operations on these four states were also described by classes. And to organize different methods, we designed three virtual classes as the interface to describe the three operations in the CVTree scheme. In this way, a new method can be implemented by deriving from corresponding base classes (see details in File S1). To improve efficiency, the main programs of CVTree were implemented in parallel by OpenMP techniques. And these classes were carefully designed to keep threads safe.

The input data for CVTree software are the genomes in Fasta form, in which one file contains one genome. A file containing the list of the genomes is also required. The final output of CVTree is the phylogenetic tree in Newick form. In the scheme of CVTree algorithm, there are three steps from genomes to a phylogenetic tree. Thus, there are three programs, named as *g2cv*, *cv2dm*, and *dm2tree*, to perform these three tasks, respectively. Apart from the step-by-step way, an integrated program, named as *cvtree*, is also provided in the CVTree software. **Figure 2** showed the flowchart of the *cvtree* program. Instead of bundling those three programs into a command, the *cvtree* program automatically refers to the intermediate data to reduce computing resource consumption. Therefore, besides the final phylogenetic tree, the intermediate data, including CVs and DMs, are also saved for reuse in the next calculation. And to save storage, these intermediate data are compressed into binary format, which cannot be inspected directly. Thus, we also provided the tools to handle these compressed files in the CVTree software (see details in File S1).

Results and discussion

Performance of *cvtree* command

The CVTree was first developed to infer evolutionary relatedness based on whole genomes to obtain real species trees instead of gene trees. The phylogenetic tree for prokaryotes based on whole genomes can be accessed on the CVTree3 web server by an interactive interface. As an example, here we showed the CVTree software by performing *cvtree* command on 13,904 RNA sequences, in which 13,903 sequences were 16S rRNA sequences from the LTPs132 of the "All-Species Living" project [26] and one sequence was from the virus, *Ferret parechovirus*, as the outgroup to root the phylogenetic tree. It was found that the performance of *cvtree* is very remarkable. By the acceleration of multi-core CPUs, a typical phylogenetic tree for these 13,904 sequences can be obtained in 108.8 s on our Dell PowerEdge Server (4 × 20-Core Intel Xeon Gold 6248 @ 2.50GHz, Linux System) or in 493.4 s on our Apple MacBook Pro (8-Core Intel Core i9 @ 2.3GHz, MacOS System). **Figure 3** showed the elapsed time of *cvtree* command as a function of the number of threads in our Dell

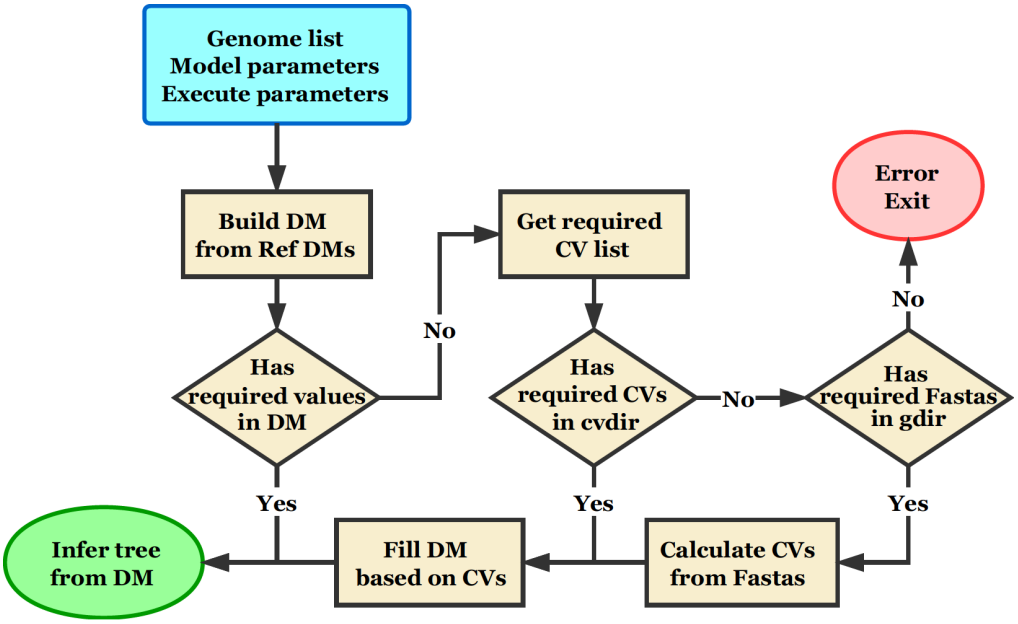

Figure 2 Flowchart of *cvtree* command

The workflow of the *cvtree* command starts from the blue square, follows the arrow lines, and normally ends at the green ellipse. In the workflow, the *cvtree* command automatically checks the cache data of every step to avoid redundant calculation. Ref, reference.

PowerEdge Server. As shown in Figure 3, the speedup of parallel was very significant. The calculation was accelerated about 1.7 times when the number of threads doubled. Detailed studies showed that most of the time was spent in the last two steps, calculating the DM and inferring the phylogenetic tree, and the speedup by parallel in calculating the DM was more significant. It was about 1.9 times with double of threads. We noted that with the increase of the length of genome sequences, the complexity of calculating a DM is lower than linear complexity, while the complexity of inferring a phylogenetic tree is constant. That is, the amount of computing resource of CVTree methods is scaled with the length of the sequence below linear. And CVTree method may obtain a rich benefit by parallel. Therefore, the CVTree programs are efficient enough to obtain the all-species living tree based on whole genomes.

Compatibility with taxonomy

To examine the accuracy of CVTree, we compared the phylogenetic trees with the taxonomy system of these prokaryotes. It was a frequently asked question that what is the best length of the k-string, *i.e.*, how to set the parameter k. According to our studies, a reasonable length was in the range $\log_m N < k < (\log_m N) + 2$ for the Hao method, where m is the number of the genome tyeps (*i.e.*, the types of amino acids or nucleotides) and N is the average length of the genome sequences. And the reasonable k for new methods in the CVTree software should be a little bigger than that of the Hao method and has a larger value range. A detailed discussion of this problem can refer to our previous studies [28,29]. In this study, we set $k = 6$ for the Hao method and $k = 7$ for the InterList method and the InterSet method. **Figure 4** showed the relative entropy difference between the taxonomy and the phylogenetic tree at every taxon level. The relative entropy difference between the taxonomy and the phylogenetic tree from the LTPs132, which was obtained by the alignment method, was also plotted in the figure as the benchmark. It was found that the results of CVTree methods and LTPs132 had similar performance at the high taxon levels of phylum, class, and order. At the low taxon levels of family, genus, and species, however, the results of CVTree methods were more consistent with the taxonomy than that of LTPs132. That is, the taxa were more monophyly in the CVTree methods. Moreover, our previous study showed that the CVTree methods may have much better performance with whole genomes for prokaryotes [30]. This indicated that the CVTree was an effective tool for deducing the taxonomy system.

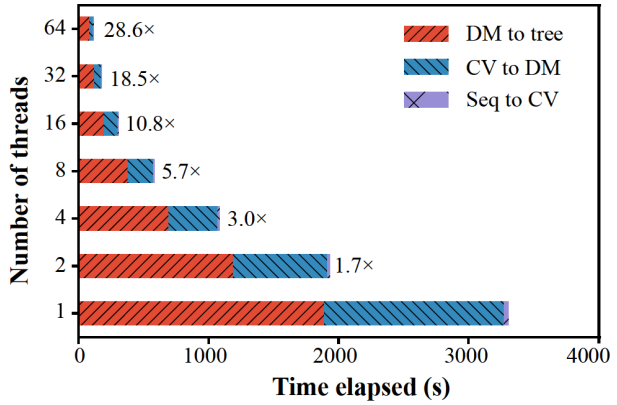

Figure 3 **Speedup of *cvtree* command by multi-thread**
The Hao model with $k = 6$ was performed by *cvtree* command on the Dell PowerEdge Server (4 × 20-Core Intel Xeon Gold 6248 @ 2.50GHz, Linux System). The elapsed time of the three steps displayed in different patterns and colors in the bar plot. The number at the right of the bars showed the speedup of the *cvtree* command by multi-thread.

Conclusion

CVTree is a cluster of alignment-free methods to infer phylogenetic relationships based on genome sequences. It has been applied to viruses, prokaryotes, and fungi with remarkable success, as well as chloroplasts, mitochondria, and metagenomes. Here we released the standalone CVTree software. The main programs of the software are parallel by OpenMP techniques. It is efficient to obtain a taxonomy-compactible phylogenetic tree based on the 16S rRNA sequences. Since the complexity of the CVTree algorithm is

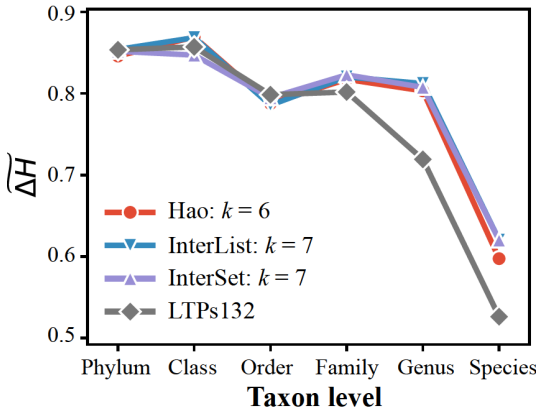

Figure 4 **Relative entropy difference between phylogeny and taxonomy**
The relative entropy difference $\widetilde{\Delta H} = \frac{H_{max} - H_{phy}}{H_{max} - H_{tax}}$, where $H = -\sum_i p_i \cdot \log_2 p_i$, is the Shannon entropy of a distribution. H_{phy} and H_{tax} are the Shannon entropy of the distribution in phylogeny and taxonomy at a taxon level, respectively. H_{max} indicates that every class includes only one strain, *i.e.*, $p_i = 1/N$. It is obvious that all taxa of a level make a partition of all strains, and $p_i = n_i / N$. To obtain H_{phy} at a taxon level, we obtained the branches of all taxa of this level in the phylogenetic tree; it is also a partition of all strains. Thus, $0 \le \widetilde{\Delta H} \le 1$, in which "1" indicates that all taxa of the taxon level are monophyly, and "0" indicates that every stain of all taxa is polyphyly in the phylogenetic tree.

lower than linear complexity with the length of genome sequences, CVTree is efficient to handle huge whole genomes to obtain the phylogenetic relationship, especially for the prokaryotes. We believe that CVTree software is an efficient and effective tool for establishing a phylogeny-based prokaryotic taxonomy.

Code availability

CVTree software code can be downloaded at https://github.com/ghzuo/cvtree and https://ngdc.cncb.ac.cn/biocode/tools/BT007094.

CRediT author statement

Guanghong Zuo: Conceptualization, Methodology, Software, Data curation, Visualization, Investigation, Supervision, Validation, Writing - original draft, Writing - review & editing. The author has read and approved the final manuscript.

Competing interests

The author has declared that no competing interests exist.

Acknowledgments

We thank Dr. Qiang Li for his helpful discussion.

Supplementary material

Supplementary data to this article can be found online at https://doi.org/10.1016/j.gpb.2021.03.006.

ORCID

0000-0002-7822-5969 (Guanghong Zuo)

References

[1] Needleman SB, Wunsch CD. A general method applicable to the search for similarities in the amino acid sequence of two proteins. J Mol Biol 1970;48:443–53.
[2] Smith TF, Waterman MS. Identification of common molecular subsequences. J Mol Biol 1981;147:195–7.
[3] Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. J Mol Biol 1990;215:403–10.
[4] Thompson JD, Higgins DG, Gibson TJ. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. Nucl Acids Res 1994;22:4673–80.
[5] Earl D, Nguyen N, Hickey G, Harris RS, Fitzgerald S, Beal K, et al. Alignathon: a competitive assessment of whole-genome alignment methods. Genome Res 2014;24:2077–89.
[6] Zielezinski A, Vinga S, Almeida J, Karlowski WM. Alignment-free sequence comparison: benefits, applications, and tools. Genome Biol 2017;18:186.
[7] Ren J, Bai X, Lu YY, Tang K, Wang Y, Reinert G, et al. Alignment-free sequence analysis and applications. Annu Rev Biomed Data Sci 2018;1:93–114.
[8] Zielezinski A, Girgis HZ, Bernard G, Leimeister CA, Tang K, Dencker T, et al. Benchmarking of alignment-free sequence comparison methods. Genome Biol 2019;20:144.
[9] Vinga S. Information theory applications for biological sequence analysis. Brief Bioinform 2014;15:376–89.
[10] Vinga S, Almeida J. Alignment-free sequence comparison—a review. Bioinformatics 2003;19:513–23.
[11] Qi J, Wang B, Hao B. Whole proteome prokaryote phylogeny without sequence alignment: a K-string composition approach. J Mol Evol 2004;58:1–11.
[12] Zuo G, Xu Z, Hao B. Phylogeny and taxonomy of *archaea*: a comparison of the whole-genome-based CVTree approach with 16S rRNA sequence analysis. Life (Basel) 2015;5:949–68.
[13] Zuo G, Xu Z, Hao B. *Shigella* strains are not clones of *Escherichia coli* but sister species in the genus *Escherichia*. Genomics Proteomics Bioinformatics 2013;11:61–5.
[14] Zuo G, Hao B, Staley JT. Geographic divergence of "Sulfolobus islandicus" strains assessed by genomic analyses including electronic DNA hybridization confirms they are geovars. Antonie Van Leeuwenhoek 2014;105:431–5.
[15] Wang H, Xu Z, Gao L, Hao B. A fungal phylogeny based on 82 complete genomes using the composition vector method. BMC Evol Biol 2009;9:195.
[16] Kjærbølling I, Vesth TC, Frisvad JC, Nybo JL, Theobald S, Kuo A, et al. Linking secondary metabolites to gene clusters through genome sequencing of six diverse *Aspergillus* species. Proc Natl Acad Sci U S A 2018;115:E753–61.
[17] Gao L, Qi J. Whole genome molecular phylogeny of large dsDNA viruses using composition vector method. BMC Evol Biol 2007;7:41.
[18] Chu KH, Qi J, Yu ZG, Anh V. Origin and phylogeny of chloroplasts revealed by a simple correlation analysis of complete genomes. Mol Biol Evol 2004;21:200–6.
[19] Yuan J, Zhu Q, Liu B. Phylogenetic and biological significance of evolutionary elements from metazoan mitochondrial genomes. PLoS One 2014;9:e84330.
[20] Zhang Q, Wu Y, Wang J, Wu G, Long W, Xue Z, et al. Accelerated dysbiosis of gut microbiota during aggravation of DSS-induced colitis by a butyrate-producing bacterium. Sci Rep 2016;6:27572.
[21] Liu J, Wang H, Yang H, Zhang Y, Wang J, Zhao F, et al. Composition-based classification of short metagenomic sequences elucidates the landscapes of taxonomic and functional enrichment of microorganisms. Nucleic Acids Res 2013;41:e3.
[22] Qi J, Luo H, Hao B. CVTree: a phylogenetic tree reconstruction tool based on whole genomes. Nucleic Acids Res 2004;32:W45–7.
[23] Xu Z, Hao B. CVTree update: a newly designed phylogenetic study platform using composition vectors and whole genomes. Nucleic Acids Res 2009;37:W174–8.
[24] Zuo G, Hao B. CVTree3 web server for whole-genome-based and alignment-free prokaryotic phylogeny and taxonomy. Genomics Proteomics Bioinformatics 2015;13:321–31.
[25] Li Q. A heuristic probabilistic model for the evolution of K-string of biological sequences and the problem of unique reconstruction of a sequence from its constituent K-string. A Ph.D. thesis. Department of Physics, Fudan University 2009.
[26] Yarza P, Richter M, Peplies J, Euzeby J, Amann R, Schleifer KH, et al. The All-Species Living Tree project: a 16S rRNA-based phylogenetic tree of all sequenced type strains. Syst Appl Microbiol 2008;31:241–50.
[27] Yu J. A *scientist guerilla fighter* in the frontiers of bioinformatics—

in memory of Bailin Hao. Genomics Proteomics Bioinformatics 2018;16:307–9.

[28] Zuo G, Xu Z, Yu H, Hao B. Jackknife and bootstrap tests of the composition vector trees. Genomics Proteomics Bioinformatics 2010;8:262–7.

[29] Zuo G, Li Q, Hao B. On K-peptide length in composition vector phylogeny of prokaryotes. Comput Biol Chem 2014;53:166–73.

[30] Zuo G, Qi J, Hao B. Polyphyly in 16S rRNA-based LVTree *versus* monophyly in whole-genome-based CVTree. Genomics Proteomics Bioinformatics 2018;16:310–9.