Composition Vector Method Based on Maximum Entropy Principle for Sequence Comparison

Raymond H. Chan, Tony H. Chan, Hau Man Yeung, and Roger Wei Wang

Abstract—The composition vector (CV) method is an alignment-free method for sequence comparison. Because of its simplicity when compared with multiple sequence alignment methods, the method has been widely discussed lately; and some formulas based on probabilistic models, like Hao's and Yu's formulas, have been proposed. In this paper, we improve these formulas by using the entropy principle which can quantify the non-randomness occurrence of patterns in the sequences. More precisely, existing formulas are used to generate a set of possible formulas from which we choose the one that maximizes the entropy. We give the closed-form solution to the resulting optimization problem. Hence from any given CV formula, we can find the corresponding one that maximizes the entropy. In particular, we show that Hao's formula is itself maximizing the entropy and we derive a new entropy-maximizing formula from Yu's formula. We illustrate the accuracy of our new formula by using both simulated and experimental datasets. For the simulated datasets, our new formula gives the best consensus and significant values for three different kinds of evolution models. For the dataset of tetrapod 18S rRNA sequences, our new formula groups the clades of bird and reptile together correctly, where Hao's and Yu's formulas failed. Using real datasets with different sizes, we show that our formula is more accurate than Hao's and Yu's formulas even for small datasets.

Index Terms—Composition vector method, maximum entropy principle, optimization model, alignment-free sequence comparison, phylogenetics.

1 Introduction

Nowadays, powerful sequence comparison methods, together with comprehensive biological databases, have changed the practice of molecular biology and genomics. Sequence comparison methods can be divided into two categories: alignment-based [14], [22] and alignment-free [28]. The composition vector (CV) method [9], [17] is an alignment-free method, and has been extensively studied recently [12], [17], [30], [34]. Compared with the multiple sequence alignment methods which are widely employed, the CV method has several advantages. For instance, it can be used for phylogenetic analysis of complete genome sequences of bacteria, eukaryote, etc [5], [17], [34]. In contrast, as every species has its own gene content and gene order, it is difficult to align two complete genome sequences [17]. The CV method also requires no scoring matrix or gap penalty, and hence it has less parameters compared to the alignment method [17].

In CV method, the distance between two taxa can be computed in $\mathcal{O}(N \log N)$ operations and the memory requirement is $\mathcal{O}(N)$, where N is the length of the longer sequence. Hence the distance matrix of M taxa, each of length at most N, can be obtained within $\mathcal{O}(M^2 N \log N)$ operations. Their phylogenetic tree can be obtained in another $\mathcal{O}(M^3)$ operations by the neighbor-joining method [19], [26]. With the development of sequencing technologies, more and more complete genome sequences are available, and these advantages of CV methods are becoming more important or even necessary for sequence comparison methods.

Let us briefly introduce the CV method which consists of the following 4 steps:

1) Consider a taxon sequence of length N. Any consecutive k nucleotides within the sequence is called a *k-string*. For each k-string $\mathtt{u}$, we count the frequency of the pattern $\mathtt{u}$ occurring in the sequence and denote it by $f(\mathtt{u})$, the *frequency vector*. Since there are 4^k different possible k-strings, the vector $f(\cdot)$ is of length 4^k.
2) For every k-string $\mathtt{u}$, we estimate its expected frequency of appearance and denote it by $q(\mathtt{u})$. Then the *composition vector* of the taxon is just the 4^k-vector where each entry equals $[f(\mathtt{u}) - q(\mathtt{u})]/q(\mathtt{u})$.
3) The cosine angle between two composition vectors is used to compute the distance between the two taxa.
4) Once the distances amongst all taxa are obtained, the neighbor-joining method can be used to construct the phylogenetic trees.

As mentioned in [5], [6], [7], [12], [17], [30], [34], step 2 above is essential for the CV method, and there are quite a few models proposed and corresponding *estimation formulas* for estimating the expected frequencies $q(\mathtt{u})$ have been derived. Here we introduce two of them. For any k-string $\mathtt{u}$, let us write it as $\mathtt{LwR}$, where the characters

- *R. H. Chan, T. H. Chan and H. M. Yeung are with the Department of Mathematics, The Chinese University of Hong Kong, Shatin, N. T., Hong Kong. Research supported by HKRGC Grant CUHK 400508 and CUHK DAG 2060257. E-mail: rchan@math.cuhk.edu.hk.*
- *R. W. Wang is with CAS-MPG Partner Institute and Key Lab for Computational Biology, Shanghai Institutes for Biological Sciences, Chinese Academy of Sciences, Shanghai 200031, China.*
Email: wwang_00@yahoo.com.

"L" and "R" represent the first and the last nucleotides of u respectively, and "w" represents the $(k-2)$-string in the middle. The first formula was proposed by Hao *et al.* [17]:

$$q(\mathtt{LwR}) = \frac{f(\mathtt{Lw})f(\mathtt{wR})}{f(\mathtt{w})}, \quad k \geq 3. \tag{1}$$

It is derived under the Markov chain assumption. Yu *et al.* proposed [34]:

$$q(\mathtt{LwR}) = \frac{f(\mathtt{L})f(\mathtt{wR}) + f(\mathtt{Lw})f(\mathtt{R})}{2}, \quad k \geq 2. \tag{2}$$

It is derived under the assumption that L, w, and R occur independently. We remark that (1) was also used by Brendel *et al.* [4] for revealing functional and evolutionary relatedness of sequences, and (2) appeared in the area of complexity and dynamical systems [32].

Shannon's entropy measures the information content in random sequences [20]. Hu and Wang [11] used it to find statistically significant strings in biological sequences under the constraints:

$$\begin{cases} q(\mathtt{vA}) + q(\mathtt{vC}) + q(\mathtt{vG}) + q(\mathtt{vT}) = f(\mathtt{v}), \\ q(\mathtt{Av}) + q(\mathtt{Cv}) + q(\mathtt{Gv}) + q(\mathtt{Tv}) = f(\mathtt{v}), \end{cases} \tag{3}$$

where the entropy in $q(\cdot)$ is to be maximized given the frequency $f(\mathtt{v})$ for all $(k-1)$-strings v. An optimization problem was thereby proposed, and formula (1) was given without proof to be the solution to the optimization problem.

In this paper, we adopt their idea of using maximum entropy to estimate the expected frequency $q(\cdot)$. However, instead of assuming a relationship between $q(\cdot)$ and $f(\cdot)$ as in (3), we derive their relationship using existing formulas. More precisely, we put existing formulas such as (1) or (2) into the left hand side of (3) and get the corresponding right hand side in terms of $f(\cdot)$. In this way, the existing formula generates a set of possible estimation formulas $q(\cdot)$ to which the existing formula also belongs. We then choose the one formula in this set that maximizes the entropy.

We show that Hao's formula is itself maximizing the entropy and we derive a new entropy-maximizing formula from Yu's formula. We illustrate the accuracy of our new formula by using both simulated and experimental datasets. For the simulated datasets, our new formula gives the highest consensus and significant values for three different kinds of evolution models. For the tetrapod sequences [31], our new formula can group the clades of bird and reptile together correctly, where Hao's and Yu's formulas failed. By applying the formulas to tetrapod datasets of different sizes, we also show that our formula is more accurate than Hao's and Yu's formulas for small datasets.

The rest of the paper is organized as follows. In Section 2, we introduce the optimization problem for maximizing the entropy in existing formulas. We then derive a closed-form solution to the optimization problem. From that we can generate new entropy-maximizing formulas from existing ones straightforwardly. In Section 3, we show the accuracy of our new estimation formula by applying it on both simulated and experimental datasets. Finally, we discuss on the computational complexity of CV method and its extension to protein sequence comparison. A way for choosing the optimal string length k is also provided.

2 OUR METHOD

2.1 The optimization problem

The CV method starts by computing the frequency of each of the 4^k's k-strings in the given DNA sequence. They are computed as follows. Given a k-string u, we count the number of times $n(\mathtt{u})$ that the pattern u appears in the sequence. The frequency of the k-string u is then defined to be $f(\mathtt{u}) = n(\mathtt{u})/(N-k+1)$. For instance, in the sequence GATCAGATTG, $f(\mathtt{G}) = 3/10$, $f(\mathtt{AT}) = 2/9$, and $f(\mathtt{ATC}) = 1/8$.

The second step of the CV method is to estimate the expected frequency $q(\mathtt{u})$ of each k-string u. Our idea is to use existing estimation formulas $q(\cdot)$ to determine the relationship between $q(\cdot)$ and $f(\cdot)$. Then we generate new estimation formulas by maximizing the entropy in $q(\cdot)$. As an example, if we substitute Hao's formula (1) for $q(\mathtt{u})$ into the left hand side of (3), we can easily verify that for any $(k-1)$-string $\mathtt{v} = \mathtt{Lw} = \mathtt{xR}$ where w and x are $(k-2)$-strings and L and R are 1-strings, we have

$$\begin{cases} q(\mathtt{vA}) + q(\mathtt{vC}) + q(\mathtt{vG}) + q(\mathtt{vT}) \\ = \dfrac{f(\mathtt{Lw})}{f(\mathtt{w})}[f(\mathtt{wA}) + f(\mathtt{wC}) + f(\mathtt{wG}) + f(\mathtt{wT})], \\ q(\mathtt{Av}) + q(\mathtt{Cv}) + q(\mathtt{Gv}) + q(\mathtt{Tv}) \\ = \dfrac{f(\mathtt{xR})}{f(\mathtt{x})}[f(\mathtt{Ax}) + f(\mathtt{Cx}) + f(\mathtt{Gx}) + f(\mathtt{Tx})]. \end{cases} \tag{4}$$

Note that the right hand sides of (4) are frequencies $f(\cdot)$ computable from the DNA sequence. Moreover, by construction, Hao's formula is just one of the many possible formulas that satisfy (4). Amongst all those formulas, we will choose the one that maximizes the entropy $-q(\mathtt{u})\log q(\mathtt{u})$.

In general, from any existing estimation formula $q(\cdot)$ given in terms of $f(\cdot)$, we can derive a set of constraints:

$$\begin{cases} q(\mathtt{vA}) + q(\mathtt{vC}) + q(\mathtt{vG}) + q(\mathtt{vT}) = l(\mathtt{v}), \\ q(\mathtt{Av}) + q(\mathtt{Cv}) + q(\mathtt{Gv}) + q(\mathtt{Tv}) = r(\mathtt{v}), \end{cases} \tag{5}$$

where v is a $(k-1)$-string, and $l(\mathtt{v})$ and $r(\mathtt{v})$ are numbers computable from the frequency $f(\mathtt{v})$. Note that in (5), there are $(2 \cdot 4^{k-1})$ constraints and 4^k unknowns $q(\mathtt{u})$, where u are all the possible k-strings. Thus the system is under-determined and the solution is not unique. By construction, the given estimation formula will only be one of the many $q(\cdot)$'s satisfying (5).

To obtain the unique $q(\mathtt{u})$, we maximize their entropy $-q(\mathtt{u})\log q(\mathtt{u})$. More precisely, we obtain $q(\mathtt{u})$ for all u by solving the optimization problem:

$$\begin{array}{ll} \text{maximize} & -\sum_{i=1}^{4^k} q(\mathtt{u}_i)\log q(\mathtt{u}_i) \\ \text{subject to} & \begin{cases} q(\mathtt{u}_i) \text{ satisfies (5)}, \\ q(\mathtt{u}_i) \geq 0 \text{ for } i=1,\ldots,4^k. \end{cases} \end{array} \tag{6}$$

2.2 Decoupling of the optimization problem (6)

In this subsection, we show that the optimization problem (6) can be decoupled into 4^{k-2} sub-problems of size 8-by-16 each, and hence can be readily solved.

The idea is to write the k-strings in the left hand sides of (5) out as LwR where w is a $(k-2)$-string and L and R are 1-strings. By exhausting different combinations of L and R, we see that the following constraints are all the constraints involving w:

$$\begin{cases} q(\mathtt{AwA})+q(\mathtt{AwC})+q(\mathtt{AwG})+q(\mathtt{AwT}) = l(\mathtt{Aw}), \\ q(\mathtt{AwA})+q(\mathtt{CwA})+q(\mathtt{GwA})+q(\mathtt{TwA}) = r(\mathtt{wA}), \\ q(\mathtt{CwA})+q(\mathtt{CwC})+q(\mathtt{CwG})+q(\mathtt{CwT}) = l(\mathtt{Cw}), \\ q(\mathtt{AwC})+q(\mathtt{CwC})+q(\mathtt{GwC})+q(\mathtt{TwC}) = r(\mathtt{wC}), \\ q(\mathtt{GwA})+q(\mathtt{GwC})+q(\mathtt{GwG})+q(\mathtt{GwT}) = l(\mathtt{Gw}), \\ q(\mathtt{AwG})+q(\mathtt{CwG})+q(\mathtt{GwG})+q(\mathtt{TwG}) = r(\mathtt{wG}), \\ q(\mathtt{TwA})+q(\mathtt{TwC})+q(\mathtt{TwG})+q(\mathtt{TwT}) = l(\mathtt{Tw}), \\ q(\mathtt{AwT})+q(\mathtt{CwT})+q(\mathtt{GwT})+q(\mathtt{TwT}) = r(\mathtt{wT}). \end{cases} \tag{7}$$

Recall that the numbers $l(\cdot)$ and $r(\cdot)$ are computable from $f(\cdot)$ (cf. (4)). From (7), we also see that they are not arbitrary but must satisfy a consistence condition:

$$\begin{aligned} & l(\mathtt{Aw})+l(\mathtt{Cw})+l(\mathtt{Gw})+l(\mathtt{Tw}) \\ =& \sum_{\mathtt{L},\mathtt{R}\in\{\mathtt{A},\mathtt{C},\mathtt{G},\mathtt{T}\}} q(\mathtt{LwR}) \\ =& r(\mathtt{wA})+r(\mathtt{wC})+r(\mathtt{wG})+r(\mathtt{wT}). \end{aligned} \tag{8}$$

An important observation is that the system (7) for each w is decoupled from each other. In fact, for any $(k-2)$-string w, the unknowns $q(\mathtt{LwR})$ for different L and R can only occur in the constraints (7) for that particular w, and will never occur in the constraints for any other $(k-2)$-string $\tilde{\mathtt{w}}$. It is because if $\mathtt{w} \neq \tilde{\mathtt{w}}$, then $\mathtt{LwR} \neq \tilde{\mathtt{L}}\tilde{\mathtt{w}}\tilde{\mathtt{R}}$ no matter what L, R, $\tilde{\mathtt{L}}$, and $\tilde{\mathtt{R}}$ are. Obviously the objective function in (6) is already decoupled for each w as each term in the objective function involves only one $q(\mathtt{LwR})$. Since there are 4^{k-2} different $(k-2)$-strings, the optimization problem (6) can be decoupled into 4^{k-2} sub-problems of the form (7), each of which is of size 8-by-16. Specifically, we need to solve the following problem with 16 unknowns:

$$\begin{array}{ll} \text{maximize} & -\sum_{i,j=1}^{4} p_{ij}\log p_{ij} \\ \text{subject to} & \begin{cases} p_{i1}+p_{i2}+p_{i3}+p_{i4} = l_i, \\ \qquad i=1,2,3,4, \\ p_{1i}+p_{2i}+p_{3i}+p_{4i} = r_i, \\ \qquad i=1,2,3,4, \\ p_{ij} \geq 0, \\ \qquad i,j=1,2,3,4, \end{cases} \end{array} \tag{9}$$

where p_{ij} are the unknowns $q(\mathtt{LwR})$ to be sought in (7).

2.3 Solution of the optimization problem (9)

In this subsection, we use the Lagrange multiplier method [3] to show that the solution of (9) is:

$$p_{ij} = \begin{cases} \dfrac{l_i r_j}{\sigma}, & \text{if } \sigma \neq 0, \\ 0, & \text{if } \sigma = 0, \end{cases} \tag{10}$$

where σ is defined by the consistence condition (8):

$$\sigma \equiv l_1+l_2+l_3+l_4 = r_1+r_2+r_3+r_4. \tag{11}$$

Note that if any of the l_i or r_i is equal to 0, say for instance, $l_1 = 0$, then by (9) and the fact that $p_{ij} \geq 0$, we have $p_{11} = p_{12} = p_{13} = p_{14} = 0$. Hence (10) is true for those variables. Thus in the following, we can assume that all the l_i and r_i are positive.

We first consider solving (9) without the nonnegative constraints $p_{ij} \geq 0$. The Lagrange function is:

$$\begin{aligned} F \;=\; & -\sum_{i,j=1}^{4} p_{ij}\log p_{ij} + \sum_{i=1}^{4} \lambda_i(l_i - p_{i1} - p_{i2} - p_{i3} - p_{i4}) \\ & + \sum_{j=1}^{4} \mu_j(r_j - p_{1j} - p_{2j} - p_{3j} - p_{4j}) \end{aligned}$$

where λ_i and μ_j are the Lagrange multipliers for the equations involving l_i and r_j respectively. By setting $\partial F/\partial p_{ij} = 0$ for all $i,j = 1,2,3,4$, we have $\log p_{ij} + 1 + \lambda_i + \mu_j = 0$. This gives

$$p_{ij} = e^{-(\lambda_i+\mu_j+1)}. \tag{12}$$

By inserting (12) into (9), we obtain

$$\begin{cases} e^{-(\lambda_i+1)}(e^{-\mu_1}+e^{-\mu_2}+e^{-\mu_3}+e^{-\mu_4}) = l_i, \\ e^{-(\mu_j+1)}(e^{-\lambda_1}+e^{-\lambda_2}+e^{-\lambda_3}+e^{-\lambda_4}) = r_j. \end{cases} \tag{13}$$

Hence

$$\begin{aligned} & (e^{-\lambda_1}+e^{-\lambda_2}+e^{-\lambda_3}+e^{-\lambda_4})(e^{-\mu_1}+e^{-\mu_2}+e^{-\mu_3}+e^{-\mu_4}) \\ =& e\cdot(l_1+l_2+l_3+l_4) = e\sigma. \end{aligned} \tag{14}$$

Combining (13) and (14), we obtain $e^{-(\lambda_i+\mu_j)} = e l_i r_j/\sigma$. Then from (12), $p_{ij} = l_i r_j/\sigma$, which is the expression in (10). Clearly all such $p_{ij} \geq 0$. Therefore, they are the solution to (9).

2.4 New estimation formulas

In this subsection, we derive new estimation formulas that maximize the entropy by using existing formulas. As the first example, we try Hao's formula (1). If (1) is used for the $q(\mathtt{LwR})$ in the left hand side of (7), then we have

$$l(\mathtt{Lw}) = \frac{f(\mathtt{Lw})}{f(\mathtt{w})}\left[\sum_{\mathtt{I}} f(\mathtt{wI})\right], \tag{15}$$

$$r(\mathtt{wR}) = \frac{f(\mathtt{wR})}{f(\mathtt{w})}\left[\sum_{\mathtt{I}} f(\mathtt{Iw})\right], \tag{16}$$

cf. (4). For simplicity, here and in the following, all summations are over the four possible nucleotides A, C, G, and T. Substituting (15) and (16) into (11), we have

$$\sigma = \frac{1}{f(\mathtt{w})}\left[\sum_{\mathtt{I}} f(\mathtt{Iw})\right]\left[\sum_{\mathtt{I}} f(\mathtt{wI})\right]. \tag{17}$$

By (10), our estimation formula is:

$$\begin{aligned} q(\mathtt{LwR}) &= \frac{f(\mathtt{Lw})f(\mathtt{wR})}{\sigma f^2(\mathtt{w})}\left[\sum_{\mathtt{I}} f(\mathtt{Iw})\right]\left[\sum_{\mathtt{I}} f(\mathtt{wI})\right] \\ &= \frac{f(\mathtt{Lw})f(\mathtt{wR})}{f(\mathtt{w})}, \end{aligned} \tag{18}$$

where the last equality follows from (17). We note that this formula is exactly the same as (1). Thus we have formally proved the result in [11] that Hao's formula (1) satisfies the maximum entropy principle.

Next let us use Yu's formula (2) to see if we can derive a new estimation formula. Putting (2) into (7) for $q(\mathtt{LwR})$, and using the fact that $\sum_{\mathtt{I}} f(\mathtt{I}) = 1$, we have

$$l(\mathtt{Lw}) = \frac{1}{2}\left[f(\mathtt{Lw}) + f(\mathtt{L})\sum_{\mathtt{I}} f(\mathtt{wI})\right], \tag{19}$$

$$r(\mathtt{wR}) = \frac{1}{2}\left[f(\mathtt{wR}) + f(\mathtt{R})\sum_{\mathtt{I}} f(\mathtt{Iw})\right]. \tag{20}$$

Hence by (11),

$$\sigma = \frac{1}{2}\left[\sum_{\mathtt{I}} f(\mathtt{Iw}) + \sum_{\mathtt{I}} f(\mathtt{wI})\right]. \tag{21}$$

By (10), the estimation formula is:

$$\begin{aligned} q(\mathtt{LwR}) &= \frac{1}{4\sigma}\left[f(\mathtt{Lw}) + f(\mathtt{L})\sum_{\mathtt{I}} f(\mathtt{wI})\right] \\ &\quad \times \left[f(\mathtt{wR}) + f(\mathtt{R})\sum_{\mathtt{I}} f(\mathtt{Iw})\right], \end{aligned} \tag{22}$$

where σ is defined in (21). This formula, which satisfies the maximum entropy principle, is different from (2). We will call this formula Yu1.

In order to show that our approach is generic, let us derive another entropy-maximizing formula from the following Yu-like formula (cf. (2)):

$$q(\mathtt{LYxZR}) = \frac{1}{2}\left[f(\mathtt{LY})f(\mathtt{xZR}) + f(\mathtt{LYx})f(\mathtt{ZR})\right], k \geq 5. \tag{23}$$

Here L, Y, Z and R are single nucleotide and x is a $(k-4)$-string in the middle, i.e. $\mathtt{w} \equiv \mathtt{YxZ}$ is a $(k-2)$-string. Putting this into (7) for $q(\mathtt{LYxZR}) = q(\mathtt{LwR})$, we have

$$l(\mathtt{LYxZ}) = \frac{1}{2}\left[f(\mathtt{LY})\sum_{\mathtt{I}} f(\mathtt{xZI}) + f(\mathtt{LYx})\sum_{\mathtt{I}} f(\mathtt{ZI})\right], \tag{24}$$

$$r(\mathtt{YxZR}) = \frac{1}{2}\left[f(\mathtt{xZR})\sum_{\mathtt{I}} f(\mathtt{IY}) + f(\mathtt{ZR})\sum_{\mathtt{I}} f(\mathtt{IYx})\right]. \tag{25}$$

Hence by (11),

$$\sigma = \frac{1}{2}\left[\sum_{\mathtt{I}} f(\mathtt{IY})\sum_{\mathtt{I}} f(\mathtt{xZI}) + \sum_{\mathtt{I}} f(\mathtt{IYx})\sum_{\mathtt{I}} f(\mathtt{ZI})\right]. \tag{26}$$

By (10), the estimation formula is:

$$\begin{aligned} q(\mathtt{LYxZR}) =& \frac{1}{4\sigma}\left[f(\mathtt{LY})\sum_{\mathtt{I}} f(\mathtt{xZI}) + f(\mathtt{LYx})\sum_{\mathtt{I}} f(\mathtt{ZI})\right] \\ &\times \left[f(\mathtt{xZR})\sum_{\mathtt{I}} f(\mathtt{IY}) + f(\mathtt{ZR})\sum_{\mathtt{I}} f(\mathtt{IYx})\right], \end{aligned} \tag{27}$$

where σ is defined in (26). Thus we obtain another new formula. We will call this formula Yu2.

2.5 Construction of phylogenetic trees

Once we have the $q(\mathtt{u})$'s, the corresponding entry of the composition vector $\mathbf{c}$ is constructed as below:

$$c(\mathtt{u}) = \begin{cases} \dfrac{f(\mathtt{u}) - q(\mathtt{u})}{q(\mathtt{u})}, & \text{if } q(\mathtt{u}) \neq 0, \\ 0, & \text{otherwise}, \end{cases} \tag{28}$$

see [9]. Note that entropy measures the randomness in stochastic sequences, see [20]. By maximizing the entropy in $q(\mathtt{u})$ and then subtracting $q(\mathtt{u})$ from the frequency vector $f(\mathtt{u})$, we are quantifying the non-randomness occurrence of the pattern u, see Hu and Wang [11].

Let $\mathbf{c}_1$ and $\mathbf{c}_2$ be the composition vectors of two DNA sequences from two taxa $\mathcal{C}_1$ and $\mathcal{C}_2$. Then the distance between $\mathcal{C}_1$ and $\mathcal{C}_2$ can be computed as follows [2], [17], [24], [25]:

$$d(\mathcal{C}_1, \mathcal{C}_2) = \frac{1 - \cos\langle \mathbf{c}_1, \mathbf{c}_2 \rangle}{2},$$

where $\cos\langle \mathbf{c}_1, \mathbf{c}_2 \rangle$ is the cosine of the angle between the two composition vectors $\mathbf{c}_1$ and $\mathbf{c}_2$. Once the distances amongst all taxa are obtained, the neighbor-joining method [19], [26], such as the one in the software MEGA4 [27], can be used to construct the phylogenetic trees.

We remark that here we follow the papers [2], [17], [24], [25] and use the cosine angle to measure the distance between vectors. However, there are some other ways to measure the distances, see [28].

3 EXPERIMENTS

In this section, we compare the effectiveness of the estimation formulas by Hao (1), Yu (2), and by us, i.e. Yu1 (22) and Yu2 (27).

3.1 Simulated datasets

In order to compare the effectiveness, we use three different evolution models to generate sets of 10 sequences that have a tree topology as shown in Figure 1. Then for each set, we apply the estimation formulas to see how many branches they can identify correctly. The 18S rRNA sequence of Human ribosomal DNA complete repeating unit (GenBank: U13369.1), which is 1871bp long, is employed as the root sequence. We set $k = 8$ (see Section 4 for the choice of k) and we repeat the experiments 100 times for each model. Consensus values [13] and binomial significant tests [21] are used to gauge the accuracy.

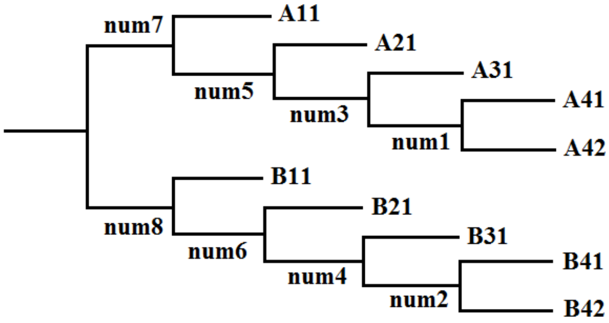

Fig. 1. The tree topology used in the simulation test.

In our first experiment, we generate the 10 sequences by MySSP [15], [18] where the HKY model [15], [18] with the mutation rate 10% was employed as the substitution model. The other parameters were set by default. The consensus trees were then generated using the neighbor joining program in the PHYLIP package. In Table 1, we show the consensus values for the correct tree topology in each branch for the four different formulas. The table clearly shows that the new entropy-maximizing formula Yu1 (22) derived from Yu's formula (2) gives the highest consensus values.

In the second and third experiments, we generate the 10 sequences by using the Markov model (Hao's model) and the totally independent model (Yu's model) respectively with mutation rate of 10%. For the Markov model, they are generated as follows. Given the parent sequence, we first compute the frequencies of all its $(k-1)$-strings and k-strings. Then we randomly choose 10% of the necleotides in the sequence. For each necleotide chosen, we compute the probability of change conditioned on the $(k-1)$-string on its left, and change the necleotide according to the probability. For the independent model, we just randomly choose 10% of the necleotides and change them randomly.

When all ten sequences are generated, we apply the formulas to get their resulting tree topologies. We count the result as a "success" if all the branches are the same as in Figure 1, or as a "failure" if at least one of the branches is wrong. We then apply a one tail binomial significant test [21] to compare the formulas pairwise, see Table 2. In the table, the p-value for "Formula X vs Formula Y" is the probability that Formula X failed to outperform Formula Y. Thus the smaller the p-value, the more confident we can say that Formula X outperforms Formula Y. The binomial significant test clearly shows that Yu1 (22) is the best amongst all four formulas for all three different evolution models. This indicates that Yu1 can provide more consistent phylogenies.

Since the new formula Yu2 (27) is not as good as Yu1 (22), in the following, we only compare Yu1 with Hao's and Yu's formulas.

3.2 Experimental datasets

We test the three formulas on two real datasets: the archaeal 16S rRNA sequences ($\sim$ 1400 bp) in Arahal *et al.* [1], and the tetrapod 18S rRNA sequences ($\sim$ 1800bp) in Xia *et al.* [31]. We note that the use of 16S rRNA for the prokaryotic organisms and 18S rRNA for the eukaryotic organisms is well-documented, see [23] and [29].

Arahal et al. [1] analyzed 22 halophilic archaeal strains in the family Halobacteriaceae collected from Dead Sea. The strains were first separated into five groups according to their phenotypic features, and one representative strain (E1, E2, E8, E11 and E12) was then selected from each group. A 16S rRNA neighbor-joining tree was constructed on these 5 strains as well as the published sequences of 27 halophilic archaea and two non-halophilic archaea. The five unknown strains were assigned into three genera, Haloferax, Haloarula and Halobacterium in the neighbor-joining tree. All three formulas support this assignment of the five strains, see Figure 2.

The phylogenetic relationship amongst tetrapods has been widely discussed in the area of phylogeny and evolution. One early topic is whether birds are more closely related to crocodilians or to mammals. Several studies based on 18S rRNA sequences supported the grouping of birds and mammals [31]. But according to the traditional classification and the results derived from a large amount of molecular, morphological and paleontological data, birds are thought to be grouped with crocodilians. This opinion is more acceptable in the biological area [10], [31].

Here we apply the CV method on the tetrapod dataset in Figure 3 of [31], except that we deleted the sequence Latimeria since it is a fish and is irrelevant to the clades we are considering. Using any one of the three formulas, every taxon is grouped to their corresponding amphibian, reptile, bird or mammal clade correctly. However, for Hao's (1) and Yu's (2) formulas, birds are grouped with mammals whereas for Yu1 formula (22), the bird and reptile clades are grouped together, see Figure 3. Thus our groupings conform with the traditional classification [10], [31].

Finally in order to show that our formula is better even for small datasets, we tried the following experiment. We started with Xia's dataset which consists of

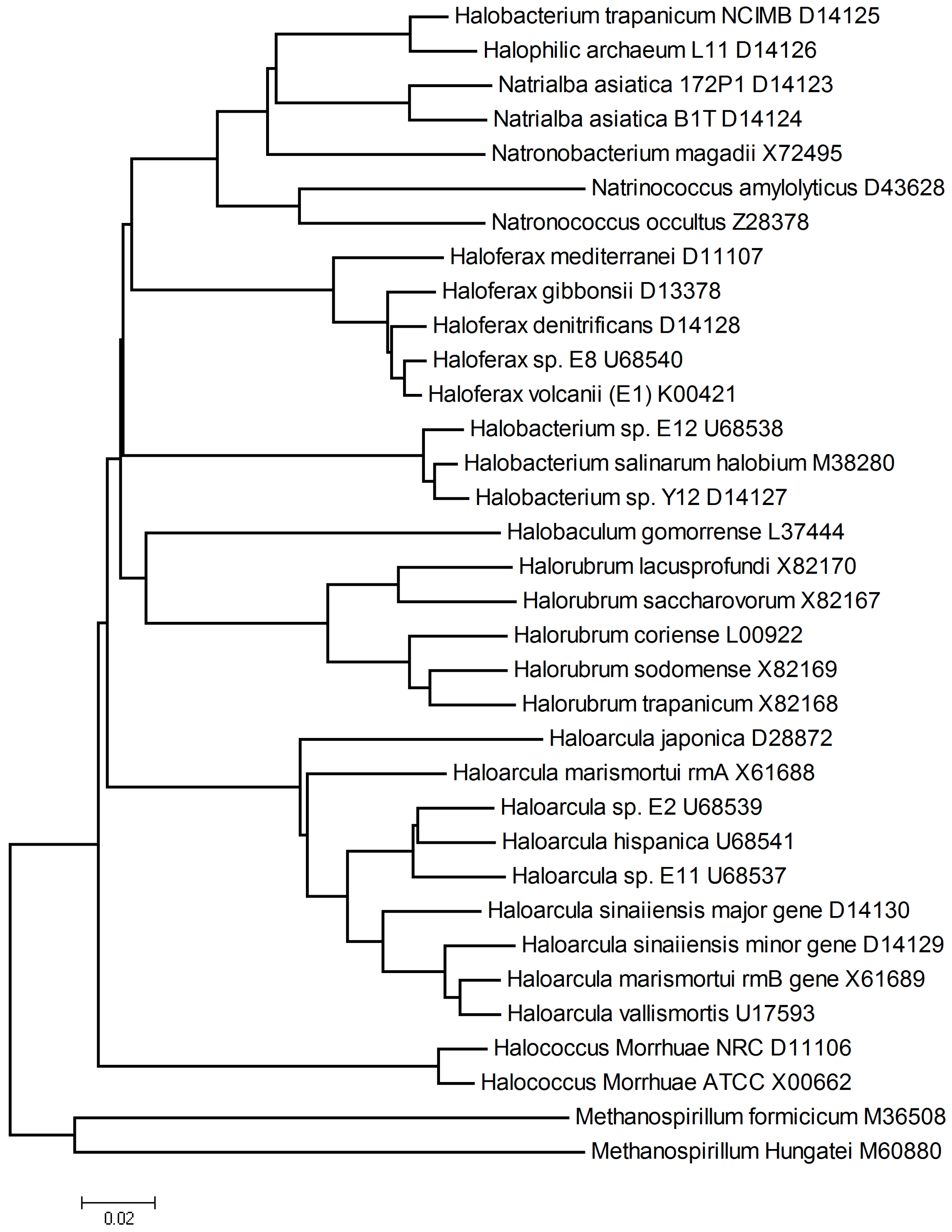
Halobacterium trapanicum NCIMB D14125
Halophilic archaeum L11 D14126
Natrialba asiatica 172P1 D14123
Natrialba asiatica B1T D14124
Natronobacterium magadii X72495
Natrinococcus amylolyticus D43628
Natronococcus occultus Z28378
Haloferax mediterranei D11107
Haloferax gibbonsii D13378
Haloferax denitrificans D14128
Haloferax sp. E8 U68540
Haloferax volcanii (E1) K00421
Halobacterium sp. E12 U68538
Halobacterium salinarum halobium M38280
Halobacterium sp. Y12 D14127
Halobaculum gomorrense L37444
Halorubrum lacusprofundi X82170
Halorubrum saccharovorum X82167
Halorubrum coriense L00922
Halorubrum sodomense X82169
Halorubrum trapanicum X82168
Haloarcula japonica D28872
Haloarcula marismortui rrnA X61688
Haloarcula sp. E2 U68539
Haloarcula hispanica U68541
Haloarcula sp. E11 U68537
Haloarcula sinaiiensis major gene D14130
Haloarcula sinaiiensis minor gene D14129
Haloarcula marismortui rrnB gene X61689
Haloarcula vallismortis U17593
Halococcus Morrhuae NRC D11106
Halococcus Morrhuae ATCC X00662
Methanospirillum formicicum M36508
Methanospirillum Hungatei M60880
0.02

Fig. 2. The CV tree ($k = 7$) with estimation formula (22) based on the 16S rRNA sequences analyzed by Arahal *et al.* [1].

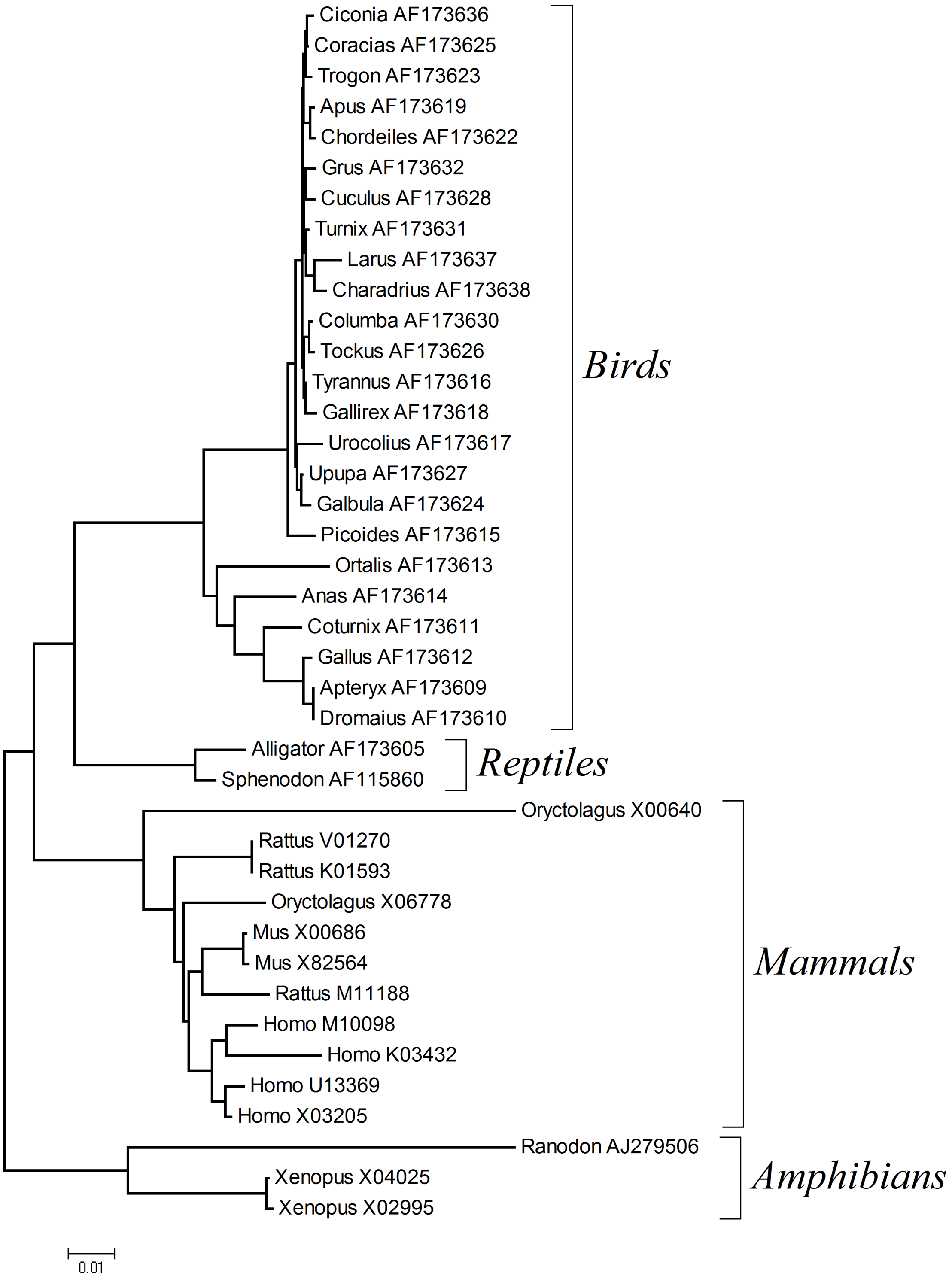

Fig. 3. The CV tree ($k = 8$) with estimation formula (22) based on the 18S rRNA sequences analyzed by Xia *et al.* [31].

Estimation formula	num1	num2	num3	num4	num5	num6	num7	num8
Hao (1)	99	99	67	62	53	45	88	88
Yu (2)	100	100	95	91	50	51	100	100
Yu1 (22)	100	100	97	93	62	63	100	100
Yu2 (27)	100	100	95	90	50	50	100	100

Table 1: Consensus values of CV trees (k=8) for simulated datasets from MySSP.

P-value	MySSP	Markov	Independent
Formula Yu vs Formula Hao	9.60E-03	7.62E-06	2.46E-05
Formula Yu1 vs Formula Hao	7.68E-09	1.23E-07	2.31E-07
Formula Yu1 vs Formula Yu	1.05E-05	1.13E-01	1.5E-02
Formula Yu1 vs Formula Yu2	3.6E-03	1.21E-02	3.83E-07

Table 2: Binomial significant test for CV trees (k=8) from 3 different evolution models.

40 sequences. We randomly delete a certain number of sequences from it while keeping at least one sequence in each clade: birds, reptiles and mammals. The deletion of the sequences from the dataset increases the distances amongst the remaining sequences. We then apply the three formulas to the remaining dataset and see if a correct phylogeny [[birds, reptiles], mammals] can be obtained. The result is shown in Figure 4. In the figure, the x-axis is the number of sequences in the dataset and the y-axis is the number of correct phylogeny out of ten trials (we repeated each experiment ten times). In this experiment, all sequences are grouped correctly to their corresponding clades by all 3 formulas. However, for the correct phylogeny [[birds, reptiles], mammals], our Yu1 formula (22) outperforms the other two formulas for all datasets. Moreover, the accuracy of (22) is monotonic increasing with respect to the number of sequences in the dataset, indicating that our formula is more stable than Hao's and Yu's. Notice that though Hao's and Yu's formulas give the same performance in the figure, we observed from the results that Hao may perform better on some samples, while Yu may perform better on others. There is no obvious pattern which one is better.

4 Discussion

Compared with sequence alignment methods, the CV method has several advantages which were mentioned in Section 1. In this paper, we proposed a general way of constructing new estimation formulas for the CV method based on the maximum entropy principle. Existing estimation formulas can be used to give new estimation formulas that maximize the entropy. In this paper, we used Hao's (1) and Yu's (2) formulas to derive new formulas. The new Hao's formula happens to be Hao's formula itself (hence is maximizing the entropy). The Yu's formula leads to a new and accurate formula (22), which is shown to be better than the Hao (1) and Yu (2) formulas in simulated as well as experimental datasets. Of course one may also use other existing formulas to derive new estimation formulas via our approach as we have done in §2.4.

We note that only nucleotide sequences were considered here. If amino acid sequences were considered, the system will be of size $(2 \cdot 20^{k-1})$-by-20^k. But it can still be decoupled into 20^{k-2} small systems of size 40-by-400 each, and new estimation formulas can be derived similarly.

One problem in CV method is how to choose the optimal string length k. Here we propose to use the simulated dataset to determine k. Specifically, using the simulated dataset, the consensus trees for different k are generated, and we choose the best k where the consensus values are maximal. As an example, consider the same setting as in Figure 1 and Table 1. The consensus trees with estimation formula (22) for k=6, 7, 8, 9, 10, are given in Table 2. We find that $k = 8$ is the best.

In implementing the CV method, we note that the maximum number of possible k-strings in a sequence of length N is $(N-k+1)$. Thus the number of non-zero entries in the frequency vector is at most $(N-k+1)$. We can save the indices and the values of these non-zero entries in two vectors of length at most N. When constructing the index vector, we can sort the indices in ascending order for easy searching later. Hence the total cost of constructing the index vector will be of $\mathcal{O}(N \log N)$ for any given k. Using our estimation formulas, e.g. (22), the cost of computing the expected frequency $q(\cdot)$ is $\mathcal{O}(1)$ for each entry. Hence the composition vector $\mathbf{c}$ for each taxon can be constructed in $\mathcal{O}(N \log N)$ operations. To compute the distance between two taxa, the total cost will then be $\mathcal{O}(N \log N)$ and the memory requirement is $\mathcal{O}(N)$. Moreover, we note that the construction of the composition vectors can be done in parallel for each taxon, and so is the computation of the entries in the distance matrix after the composition vectors are formed. These are important advantages of CV method especially when large datasets are considered.

References

[1] D.R. Arahal, F.E. Dewhirst, B.J. Paster, B.E. Volcani, and A. Ventosa, "Phylogenetic analyses of some extremely Halophilic archaea isolated from Dead sea water, determined on the basis of their 16S rRNA sequences," *Applied and Environmental Microbiology*, vol. 62, pp. 3779–3786, 1996.

[2] M.W. Berry, Z. Drmac, and E.R. Jessup, "Matrices, vector spaces, and information retrieval," *SIAM Review*, vol. 41, pp. 335–362, 1999.

[3] D.P. Bertsekas, "Nonlinear programming," Athena Scientific, 1999.

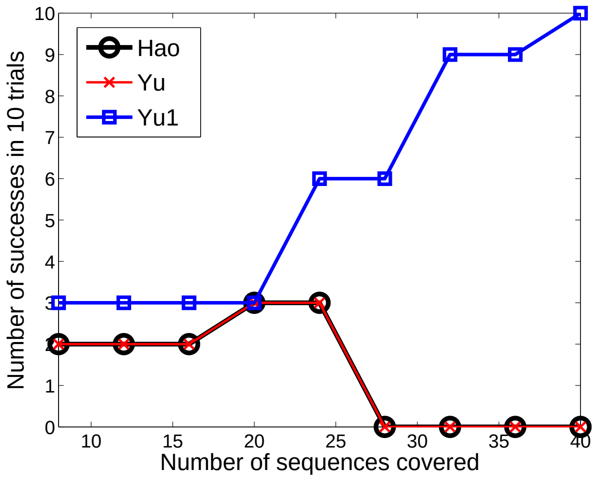

Fig. 4. Number of sequences versus successful rate.

String length	num1	num2	num3	num4	num5	num6	num7	num8
$k = 6$	100	100	94	94	60	61	100	100
$k = 7$	100	100	95	95	59	59	100	100
$k = 8$	100	100	97	93	62	63	100	100
$k = 9$	100	100	96	95	52	62	100	100
$k = 10$	100	100	94	93	51	52	100	100

Table 3: Consensus values of CV trees with estimation formula (22) for different k.

[4] V. Brendel, J.S. Beckmann, and E.N. Trifonov, ”Linguistics of nucleotide sequences: morphology and comparison of vocabularies,” *Journal of Biomolecular Structure and Dynamics*, vol. 4, pp. 11–20, 1986.

[5] K.H. Chu, J. Qi, Z.G. Yu, and V. Anh, ”Origin and phylogeny of chloroplasts: A simple correlation analysis of complete genomes,” *Molecular Biology and Evolution*, vol. 21, pp. 200–206, 2004.

[6] L. Gao and J. Qi, ”Whole genome molecular phylogeny of large dsDNA viruses using composition vector method,” *BMC Evolutionary Biology*, vol 7, pp. 1–7, 2007.

[7] L. Gao, J. Qi, H. Wei, Y. Sun, and B.L. Hao, ”Molecular phylogeny of coronaviruses including human SARS-CoV,” *Chinese Science Bulletin*, vol. 48, pp. 1170–1174, 2003.

[8] Z. Gu, X. Zhao, N. Li, and C. Wu, ”Complete sequence of the yak (Bos grunniens) mitochondrial genome and its evolutionary relationship with other ruminants,” *Molecular Phylogenetics and Evolution*, vol. 42, pp. 248–255, 2007.

[9] B.L. Hao, J. Qi, and B. Wang, ”Prokaryotic phylogeny based on complete genomes without sequence alignment,” *Modern Physics Letters B*, vol. 2, pp. 1–4, 2003.

[10] S.B. Hedges, K.D. Moberg, and L.R. Maxson, ”Tetrapod phylogeny inferred from 18S and 28S ribosomal RNA sequences and a review of the evidence for amniote relationships,” *Molecular Biology and Evolution*, vol. 7, pp. 607–633, 1990.

[11] R. Hu and B. Wang, ”Statistically significant strings are related to regulatory elements in the promoter regions of Saccharomyces cerevisiae,” *Physica A*, vol. 290, pp. 464–474, 2001.

[12] G. Lu, S. Zhang, and X. Fang, ”An improved string composition method for sequence comparison,” *BMC Bioinformatics*, **9 (Suppl 6)**, S15, 2008.

[13] T. Margush and F.R. McMorris, ”Consensus n-trees,” *Bulletin of Mathematical Biology*, vol. 43, pp. 239–244, 1981.

[14] S.B. Needleman and C.D. Wunsch, ”A general method applicable to the search for similarities in the amino acid sequence of two proteins,” *Journal of Molecular Biology*, vol. 48, pp. 443–453, 1970.

[15] P.A.S. Nuin, Z. Wang, and E.R.M. Tillier, ”The accuracy of several multiple sequence alignment programs for proteins,” *BMC Bioinformatics*, vol 7, pp. 1–18, 2006.

[16] G.J. Phillips, J. Arnold, and R. Ivarie, ”Mono-through hexanucleotide composition of the Escherichia coli genome: a Markov chain analysis,” *Nucleic Acids Research*, vol. 15, pp. 2611–2626, 1987.

[17] J. Qi, B. Wang, and B.L. Hao, ”Whole proteome prokaryote phylogeny without sequence alignment: A k-string composition approach,” *Journal of Molecular Evolution*, vol. 58, pp. 1–11, 2004.

[18] M.S. Rosenberg, ”MySSP: Non-stationary evolutionary sequence simulation, including indels,” *Evolutionary Bioinformatics Online*, vol. 1, pp. 51–53, 2005.

[19] N. Saitou and M. Nei, ”The neighbor-joining method: a new method for reconstructing phylogenetic trees,” *Molecular Biology and Evolution*, vol. 4, pp. 406–425, 1987.

[20] A.E. Shannon, ”A Mathematical Theory of Communication,” *Bell System Technical Journal*, vol. 27, pp. 379–423, 1948.

[21] D. Sheskin, ”Handbook of Parametric and Nonparametric Statistical Procedures. 3rd ed.,” *CRC Press*, 2004.

[22] T.T. Smith and M.S. Waterman, ”Identification of common molecular subsequences,” *Journal of Molecular Biology*, vol. 147, pp. 195–197, 1981.

[23] S.J. Sogin, M.L. Sogin, and C.R. Woese, ”Phylogenetic measurement in procaryotes by primary structural characterization”, *Journal of Molecular Evolution*, vol. 1, pp. 173–184, 1972.

[24] G.W. Stuart and M.W. Berry, ”An SVD-based comparison of nine whole eukaryotic genomes supports a coelomate rather than ecdysozoan lineage,” *BMC Bioinformatics*, vol 5, 204, 2004.

[25] G.W. Stuart, K. Moffett, and J.J. Leader, ”A comprehensive vertebrate phylogeny using vector representations of protein sequences from whole genomes,” *Molecular Biology and Evolution*, vol. 19, pp. 554–562, 2002.

[26] J.A. Studier and K.J. Keppler, ”A note of the neighbor-joining algorithm of Saitou and Nei,” *Molecular Biology and Evolution*, vol. 5, pp. 729–731, 1988.

[27] K. Tamura, J. Dudley, M. Nei, and S. Kumar, ”MEGA4: molecular evolutionary genetics analysis (MEGA) software version 4.0,” *Molecular Biology and Evolution*, vol. 24, pp. 1596–1599, 2007.

[28] S. Vinga and J. Almeida, ”Alignment free sequence comparison-a review,” *Bioinformatics*, vol. 19, pp. 513–523, 2003.

[29] C.R. Woese, ”Bacterial evolution”, *Microbiology Review*, pp.221–271, 1987.

[30] X. Wu, X.F. Wan, G. Wu, D. Xu, and G. Lin, ”Phylogenetic analysis using complete signature information of whole genomes and clustered Neighbor-Joining method,” *International Journal of Bioinformatics Research and Applications*, vol. 2, pp. 219–248, 2006.

[31] X. Xia, Z. Xie, and K.M. Kjer, ”18S ribosomal RNA and tetrapod phylogeny,” *Systematic Biology*, vol. 52, pp. 283–295, 2003.

[32] H.M. Xie, *Grammatical Complexity and One-dimensional Dynamical Systems.* World Scientific. Singapore, 1996.

[33] Z.G. Yu, X.W. Zhan, G.S. Han, R.W. Wang, V. Anh, and K.H. Chu, ”Proper distance metrics for phylogenetic analysis using complete genomes without sequence alignment”, *International Journal of Molecular Sciences*, vol. 11, pp. 1141–1154, 2010.

[34] Z.G. Yu, L.Q. Zhou, V. Anh, K.H. Chu, S.C. Long, and J.Q. Deng, ”Phylogeny of prokaryotes and chloroplasts revealed by a simple composition approach on all protein sequences from whole genome without sequence alignment,” *Journal of Molecular Evolution*, vol. 60, pp. 538–545, 2005.

Raymond H. Chan is a Chair Professor of Mathematics in the Department of Mathematics at The Chinese University of Hong Kong. He obtained his Ph.D. degree from the Courant institute of Mathematical Sciences at New York University. His research interests include Bioinformatics, Scientific Computation, Numerical Linear Algebra and Image Processing.

Tony H. Chan is with the Department of Mathematics at The Chinese University of Hong Kong. His research interests include Bioinformatics and Computational Biology.

Hau Man Yeung is with the Department of Mathematics at The Chinese University of Hong Kong. His research interests include Bioinformatics and Computational Biology.

Roger Wei Wang received the Ph.D. degree in mathematics from The Chinese University of Hong Kong. He is currently a postdoctoral research fellow at CAS-MPG Partner Institute and Key Lab for Computational Biology, Chinese Academy of Sciences. His research interests include Bioinformatics and Computational Biology.