BRIEFINGS IN BIOINFORMATICS. VOL 15. NO 6. 890–905
Advance Access published on 31 July 2013
doi:10.1093/bib/bbt052

Alignment-free genetic sequence comparisons: a review of recent approaches by word analysis

Oliver Bonham-Carter, Joe Steele and Dhundy Bastola

Submitted: 11th April 2013; Received (in revised form): 31st May 2013

Abstract

Modern sequencing and genome assembly technologies have provided a wealth of data, which will soon require an analysis by comparison for discovery. Sequence alignment, a fundamental task in bioinformatics research, may be used but with some caveats. Seminal techniques and methods from dynamic programming are proving ineffective for this work owing to their inherent computational expense when processing large amounts of sequence data. These methods are prone to giving misleading information because of genetic recombination, genetic shuffling and other inherent biological events. New approaches from information theory, frequency analysis and data compression are available and provide powerful alternatives to dynamic programming. These new methods are often preferred, as their algorithms are simpler and are not affected by synteny-related problems.

In this review, we provide a detailed discussion of computational tools, which stem from alignment-free methods based on statistical analysis from word frequencies. We provide several clear examples to demonstrate applications and the interpretations over several different areas of alignment-free analysis such as base–base correlations, feature frequency profiles, compositional vectors, an improved string composition and the D_2 statistic metric. Additionally, we provide detailed discussion and an example of analysis by Lempel–Ziv techniques from data compression.

Keywords: *alignment-free; sequence-alignment; information theory; word-analysis*

INTRODUCTION

Gene structure, function and phylogenetic relations are discovered by the basic comparison of known to unknown genetic material across organisms. Sequence comparison is pivotal to the success of basic phylogenetic and metagenomics research. For instance, large portions of common genetic material between organisms provide much evidence to suggest that they are somehow related. Furthermore, similar sequence data fuels conjecture that the associated functions are also similar.

Comparative research came from computer science that provided tools and algorithms to find specific substrings in larger sequences [1] for discovery. For instance, the Knuth–Morris–Pratt algorithm [2] and the Boyer–Moore [3] algorithm were used initially in the 1970's [4] to locate regions of common DNA by exact matching of larger sequences. Later, a modified version of the Boyer–Moore [5] was applied in the 1980's. As these algorithms assumed that the input strings contained exact matches, tiny mismatches found in DNA interrupted performance. This led to algorithms for approximate pattern matching [6] and others [7, 8].

Owing to the growth of inexpensive computing and improvements in sequence assembly

Corresponding author. Oliver Bonham-Carter, College of Information Science & Technology School of Interdisciplinary Informatics Peter Kiewit Institute University of Nebraska Omaha, NE USA. Tel.: (402) 554-2800; Fax: (402) 554-3284; E-mail: obonhamcarter@unomaha.edu

Oliver Bonham-Carter is a PhD student in the College of Information Science and Technology at the University of Nebraska at Omaha.

Joe Steele was a part of the application development team at the College of Information Science and Technology at the University of Nebraska at Omaha.

Dhundy Bastola is an assistant professor in Bioinformatics at the School of Interdisciplinary Informatics at the University of Nebraska at Omaha.

technologies, there is now more sequence data available to bioinformatics research than ever before. Comparative genomics has been an obstacle to discovery [9, 10] and still manages to be a major factor in more current applications. Some of these applications include sequence assembly [11], evolutionary history comparison involving complications from synteny [12], horizontal gene transfer (HGT) discovery [13, 14], analysis by gene-shuffling [15] and many other applications where proper sequence comparison must be used [16].

Dynamic programming [17] has often been applied to comparing sequences in the aforementioned applications. As global and local alignment algorithms [18, 19] work base-by-base, they stand to be confused by the inherent mismatches, gaps, alternating blocks of sequence material and inversions that are easily found in genetic material. These methods may erroneously conclude that the functionally related sequences are largely unrelated, as they do not demonstrate any statistically significant alignment. Sequence length is also important to address when running an alignment from dynamic programming. For example, local and global, implemented in softwares such as ClustalW [20], have complexities of $O(mn)$, and therefore it is clear that their resource requirements quickly escalate for larger sequences of lengths, m and n. It is often infeasible to perform comparisons of complete genomes by this approach owing to the large amount of time this would involve. For this reason, technologies requiring databases for speed such as BLAST [21], BLASTZ [22] and BLAT [23] have gained popularity. Other methods to help overcome some of the limitations of dynamic programming have come from diverse fields such as cloud computing [24], distributed computing [25] and parallel computing for multiple sequence comparison [26].

Frequency-based algorithms, which are driven by the statistics of word usage or similar, are becoming popular in research for discovery. This is because these approaches are not typically confused by the complexities caused by mismatches, gaps and sequence inversions that are often found between sequences for comparison [27]. For example, these methods function by evaluating the informational content between sequences, and therefore alternating blocks of DNA between two sequences will not be problematic. This form of alignment does not depend on where the features are found in the sequence, only that the sequence contains the features. Methods using frequency analysis also do not suffer from high algorithmic complexities as they are generally linear. They are, therefore, able to process larger sequences with fewer resources than dynamic programming algorithms and do not rely on having database support, as would BLAST, BLASTZ and BLAT. There is clearly a call for an alternative approach for sequence comparison done by methods that are not of dynamic programming, and therefore alignment-free methods are becoming attractive to bioinformatics research where the data are substantial and naturally dynamic.

In this article, we discuss some of the prominent methods stemming from vector or frequency-based analysis such as base–base correlations (BBC), feature frequency profiles (FFPs), compositional vectors (CVs), improved string composition and the D_2 statistic metric. These methods have been chosen for discussion because of their simplistic nature and ease of application to research. We provide clear examples for the implementation of these methods and discuss their interpretation. We also provide discussion and an example of a method inspired by the Lempel-Ziv (LZ) compression techniques. This review aims to show how these alignment-free methods are integral to the quantification and discovery of sequence function and structure.

BACKGROUND

Methods for differentiating sequence data by using statistical concepts (factor frequencies and approaches from data compression) have attracted much interest. In their often-cited 2003 publication, Vinga *et al.* [28] reviewed some related methods, metrics and algorithmic implementations. Mantaci *et al.* [29] continued by illustrating other methods recently introduced for the alignment-free comparison, which were also based on a statistical approach. The authors organize the comparison algorithms in the following basic groups: (i) count factor frequencies, (ii) data compression and (ii) edit distances or on block edit distance—a special case involving moving entire blocks of a sequence.

Recent developments and the release of new technologies from the scientific community have caused the aforementioned references to become out-dated. Here, we discuss some of the more recent statistical methods, which involve frequency data for comparison. The approaches that we cover were chosen based on their simplicity of application

and can be divided into the following categories: *factor frequencies* [30], *composition vectors* [31], *improved CVs* [32], *data compression* [33–35] and *common substrings* [6, 36].

FACTOR FREQUENCIES

Producing seminary ideas in 1948, C.E. Shannon's *InformationTheory* is the branch of mathematics, which is concerned with quantifying information and signal processing [37]. As DNA contains observable structures and patterns [38–40], tools from information theory (e.g. mutual entropy *et al.*) are appropriate for frequency analysis. Many of these methods break each sequence for comparison into numeric parts such as frequencies from the occurrence of types of words or *k*-mers (substrings of length *k*) occurring in the sequences. If two sequences are similar, then the derived *k*-mer frequencies would have similar distributions to reflect this likeness. If the sequences are different, then so are the frequency distributions.

To perform a *k*-mer study, the size of the motif is an important factor to consider. When collecting word frequencies from motifs, the size of the motif does make a difference to the results. According to Wu *et al.* [41], where the length of motif or window size is extensively discussed, there is a general rule of play when collecting word frequencies. When the sequences are obviously different (e.g. they are not related), then size of *k*-mers or window-size should be short. However, when the sequences are similar (known to be related), then the *k*-mers or window sizes can be longer. The reader is invited to consult the aforementioned reference for the details behind their general rule.

BBC by analysis of mutual information

Mutual information is a tool from information theory, which measures the amount of common information (or interaction) between two entities. Liu *et al.* [30] described the development of BBC, an algorithmic approach for determining sequence similarity by mutual information to infer phylogenetic relationships from complete genomes. In their work, an interval is established containing *r*-bases, making up strings of DNA to be used for multiple sequence comparison. In this interval, a vector is created from all possible joint probabilities of DNA pairs, as the total possible pairs $= 4^*4 = 4^2 = 16$. In their article, they showed that the interval containing these joint probabilities in the sequence can often be expanded to get a better measurement of the difference between sequences.

For $(\alpha_1, \alpha_2, \alpha_3, \alpha_4) \equiv (\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T})$, the probability of finding base α_i is denoted p_i for $1 \leq i \leq 4$. For $T_{ij}(r)$, the average relevance of the two-base combination (the feature) with different gaps from 1 to r (a range of r), the authors define a BBC by the following:

$$T_{ij}(r) = \sum_{d=1} r_{p_{ij}}(d) \cdot \log_2\left(\frac{p_{ij}(d)}{p_i p_j}\right) \tag{1}$$

for $i, j \in \{1, 2, 3, 4\}$ where $p_{ij}(d)$ signifies the joint probabilities (e.g. the $4^2 = 16$ possible length-2 DNA words, which we refer to as *features*) of bases i and j at a distance of d. A BBC feature constitutes a 16D feature vector, V_{S_1} for a sequence S_1 having a length of n_1.

The statistical independence of two bases for a sequence of length- l is defined by $p_{ij}(l) = p_i p_j$ and its deviation is defined, $D_{ij} = p_{ij}(d) - p_i p_j$. Let $S_1 = \mathtt{ACGTGCTATG}$ and $S_2 = \mathtt{ACGCGCTA}$. We find the joint probabilities to populate the vector, $(\mathtt{AA}, \mathtt{AC}, \mathtt{AG}, \mathtt{AT}, \mathtt{CA}, \mathtt{CC}, \mathtt{CG}, \mathtt{CT}, \mathtt{GA}, \mathtt{GC}, \mathtt{GG}, \mathtt{GT}, \mathtt{TA}, \mathtt{TC}, \mathtt{TG}, \mathtt{TT})$, with the following equation for frequency, $f(W_k)$, from [42]:

$$f(W_k) = \frac{c(W_k)}{n - k + 1}, \tag{2}$$

where $c(W_k)$ signifies the number of occurrences of a length- k word in a sequence of length- n_1. The finalized vectors are the following.

$$V_{S_1} = (0.0, 0.2, 0.0, 0.1, 0.0, 0.0, 0.1, 0.1, 0.0, 0.1, 0.0, 0.0, 0.1, 0.0, 0.3, 0.0)$$

$$V_{S_2} = (0.0, 0.3, 0.0, 0.0, 0.0, 0.0, 0.5, 0.5, 0.0, 0.5, 0.0, 0.0, 0.5, 0.0, 0.0, 0.0)$$

For two sequences S_1 and S_2 having the same length n_1, the authors define the distance $H_{S_1 S_2}$ in the following equation.

$$H_{S_1 S_2} = \sqrt{\sum_{i=1}^{16} (V_{S_1 i} - V_{S_2 i})^2} \tag{3}$$

By this calculation, we find that $H_{S_1 S_2} = 0.8890$ for the example aforementioned. Higher values for this metric indicate a greater spread in the frequency distribution and increasing dissimilarity; however, lower values indicate levels of increasing similarity (e.g. 0 if and only if the distributions being compared

are equivalent). The authors note that $H_{S_1S_2}$ satisfies the definition of a sequence distance because (i) $H_{S_1S_2} > 0$ for different sequence lengths; $n_1 \neq n_2$; (ii) $H_{S_1S_2} = 0$; (iii) $H_{S_1S_2} = H_{S_2S_1}$ (symmetric); and (iv) $H_{S_1S_2} \leq H_{S_1} + H_{S_2}$ (triangle inequality).

Liu *et al.* used phylogenetic trees, using branch weights gained from their BBC mutual information calculations. From the sequence data of 48 different *Hepatitis E* viruses, they constructed a phylogenetic tree, which was consistent with previous studies by diverse approaches [30].

FFPs

In [43], a feature frequency approach (*UVWORD*) was presented, which compares the DNA words from two sequences. Known as *oligonucleotide profiling*, the sliding-window method compared the encountered word frequencies of one sequence (the *target*) with another (the *source*). The sequence similarity was determined by how many words were common to both sequences. Word-based statistical models were also presented in [42], which investigated the occurrence, type and frequency of overlapping and embedded DNA words for sequence comparison.

Sims *et al.* [44] were interested in comparing whole genomes, even in situations where there are no common genes with high homology. To do this, they developed a variation of text compression, where the distance between word frequency profiles of two texts would be taken as a measure of dissimilarity. They substituted relative *k*-mer frequencies (FFP) for word frequencies.

A sliding window of size k is run through the sequence from position 1 to $n - k + 1$ and counts the number of all $t = 4^k$ possible k-mers (the total number of features, for example) where four is the number of DNA bases. Although the k-mers extend themselves throughout the entire genome, the window is only allowed to span over the regions, which are completely free of sequencing gaps. The vector $C =< c_1, \ldots, c_t >$ holds the t number of raw frequency counts for all possible words of length- k and is conventionally found by the following equation:

$$\mathbf{F} = \mathbf{C} / \sum_i c_i. \tag{4}$$

The length of the genome must be considered carefully at this vector-forming stage. If the genomes are of approximately equal length, and a <4-fold difference exists between sequences (four is the number of bases), then the method is conveniently used. However, if the sequences for comparison have extremely different lengths, then it is necessary to implement the block-FFP method, which is similar to the method described by [41]. This pre-processing step works to ensure that diverse genome lengths do not yield misleading results.

This step breaks up each sequence into smaller, manageable fragments of length- n_1 (called FFP-blocks). In the case where the length of the shorter sequence is evenly divisible by the length of the longer sequence, the intervals (e.g. blocks) are made so that they have the same length as the shorter sequence. If a sequence (length n_2) is not evenly divisible by the shorter sequence (length n_1), then the total number of possible blocks for comparative analysis that can be made is n_2 modulus n_1.

A comparison by frequencies and the Jensen–Shannon Divergence test

Comparing genomes is actually comparing the sets of frequencies, which have been taken over an interval of sequence data. To make this comparison, we will follow Sims *et al.* [44] approach to use the Jensen–Shannon Divergence (JSD) test. The JSD test is a close relation to the Kullback–Leibler Divergence test, an information theoretic non-symmetric divergence measure of two probability distributions, that is extensively discussed in [45].

Once the vectors have been properly created, we are ready to apply the calculations that determine their distance apart. For two arbitrary vectors, V_{S_1} and V_{S_2}, prepared from sequences S_1 and S_2 for t, the number of features collected, the JSD is given below:

$$JS(V_{S_1}, V_{S_2}) = \frac{1}{2} KL(V_{S_1}, V_M) + \frac{1}{2} KL(V_{S_2}, V_M), \tag{5}$$

where,

$$V_{M_i} = \frac{V_{S_1 i} + V_{S_2 i}}{2} \tag{6}$$

for $i = \{1, \ldots, t\}$ and KL is the Kullback–Leibler Divergence, below.

$$KL(V_{S_1}, V_M) = \sum_{i=1}^{t} V_{S_1 i} \log_2 \frac{V_{S_1 i}}{V_{M_i}}, \tag{7}$$

where t is the number of features.

We now return to our earlier example of the two sequences $S_1 =$ `ACGTGCTATG` and $S_2 =$ `ACGCGCTA`, which we compared by this JSD

Table 1: Positions 1 through 16 of the table of vectors for V_{S_1} from $S_1 = \text{ACGTGCTATG}$ and V_{S_2} from $S_2 = \text{ACGCGCTA}$, aligned with position

2mers position *i*	AA1	AC2	AG3	AT4	CA5	CC6	CG7	CT8	GA9	GC10	GG11	GT12	TA13	TC14	TG15	TT16
V_{S_1}	0	$\frac{1}{9}$	0	$\frac{1}{9}$	0	0	$\frac{1}{9}$	$\frac{1}{9}$	0	$\frac{1}{9}$	0	$\frac{1}{9}$	$\frac{1}{9}$	0	$\frac{2}{9}$	0
V_{S_2}	0	$\frac{1}{7}$	0	0	0	0	$\frac{2}{7}$	$\frac{1}{7}$	0	$\frac{2}{7}$	0	0	$\frac{1}{7}$	0	0	0
V_M	0	$\frac{8}{63}$	0	$\frac{1}{18}$	0	0	$\frac{25}{126}$	$\frac{8}{63}$	0	$\frac{25}{126}$	0	$\frac{1}{18}$	$\frac{8}{63}$	0	$\frac{1}{9}$	0

The elements of combined vector **M** by index are also shown. Frequencies of each 2mer are made by normalizing the occurrences of each 2mer in S_1 and S_2, respectively, by the total number of 2mer occurrences in each sequence.

analysis. In this example, we populate vectors for these sequences using all length-2 words (2mers) in the sequences. The possible 2mers are ordered in the following order:

AA, AC, AG, AT, CA, CC, CG, CT, GA, GC, GG, GT,
TA, TC, TG, TT.

The FFP vectors V_{S_1} and V_{S_2} are created and populated by all available 2mers from sequences S_1 and S_2.

$$V_{S_1} =< 0,1,0,1,0,0,1,1,0,1,0,1,1,0,2,0 > *\frac{1}{9}$$
$$V_{S_2} =< 0,1,0,0,0,0,2,1,0,2,0,0,1,0,0,0 > *\frac{1}{7}$$

At each position i of both vectors, we apply $V_{M_i} = \frac{V_{S_1 i} + V_{S_2 i}}{2}$ to get an average vector V_M. The calculated values for all three vectors are shown in Table 1.

To help the reader to keep track of the vectors and their frequencies at each position, we offer Table 1. We apply vectors V_{S_1} and V_M (and then vectors V_{S_2} and V_M) to Equation (7), which we illustrate below.

$$\begin{aligned} KL(V_{S_1}, V_M) &= \sum_{i=1}^{t} V_{S_1 i} \log_2 \frac{V_{S_1 i}}{V_{M_i}} \\ &= \frac{1}{9} * \log_2\left(\frac{\frac{1}{9}}{\frac{8}{63}}\right) + \frac{1}{9} * \log_2\left(\frac{\frac{1}{9}}{\frac{1}{18}}\right) \\ &= \frac{1}{9} * \log_2\left(\frac{\frac{1}{9}}{\frac{8}{63}}\right) + \frac{1}{9} * \log_2\left(\frac{\frac{1}{9}}{\frac{1}{18}}\right) \\ &+ \frac{1}{9} * \log_2\left(\frac{\frac{1}{9}}{\frac{25}{126}}\right) + \frac{1}{9} * \log_2\left(\frac{\frac{1}{9}}{\frac{1}{18}}\right) \\ &+ \frac{1}{9} * \log_2\left(\frac{\frac{1}{9}}{\frac{8}{63}}\right) + \frac{2}{9} * \log_2\left(\frac{\frac{2}{9}}{\frac{1}{9}}\right) \\ &= 0.1943 \end{aligned}$$

Following this theme for sequence S_2 (vector V_{S_2}), we find that $KL(V_{S_2}, V_M) = 0.3734$.

$$\begin{aligned} JS(V_{S_1}, V_{S_2}) &= \frac{1}{2} KL(V_{S_1}, V_M) + \frac{1}{2} KL(V_{S_2}, V_M) \\ &= \frac{1}{2} * (0.1943) + \frac{1}{2} * (0.3734) \\ &= 0.2839 \end{aligned}$$

Provided the base 2 logarithm is used, the JSD is bounded below by 0 and 1 [45]. Higher values indicate increasing dissimilarity, but lower values indicate increasing similarity (e.g. 0 if and only if the distributions are identical). As $JS(V_{S_1}, V_{S_2}) = 0.2839$ is close to zero, we conclude that the sequences S_1 and S_2 are similar by this test.

Sims *et al.* [44] reconstructed phylogenies from concatenated mammalian 'intronic genomes' by this method and found that their method closely reflected the accepted evolutionary history and agreed to results from a codon-sequence-based alignment technique [46].

Suffix trees by *k*-mer frequencies

The abundance of sequence noise (e.g. insertions, mismatches and similar) often necessitates frequency-based analysis. Similar to the work of [44] earlier in the text, another method of applying frequency data have been extensively explored by Soares *et al.* [47] to measure Euclidean distance between sequence data. When collecting frequency data, typically a window is opened at the beginning of the sequence, and the frequencies are found for all encountered words. The authors depart from this method by presenting a new approach that determines a single optimal word length (*k*-mers) from which to generate a frequency distribution for application to suffix trees.

To collect these optimal *k*-mer frequencies, Soares *et al.* [47] began by determining all words in the DNA alphabet (e.g. {A,C,G,T}) of length- *k*. An optimal resolution range of *k*-mers for the given

set of genomes was described in [44] and later applied to the work of Soares *et al.* [47] for instance, $k_{H\max} = \log_4(n_1)$ for a sequence of length- n_1. To find a value applicable to all sequences under analysis, we choose n_1 as the length of the greater sequence and K as the smaller integer greater than $\log_4(m)$. For m sequences of different lengths, the peak value of word length (K) that is applicable to all sequences of the study is described by the following two equations:

$$n_1 = \max\{\text{length}(S_i), 1 \leq i \leq m\},$$

$$K = \lceil \log_4(n_1) \rceil$$

In the logarithmic equation, L is given the smallest integer not less than the calculated value.

The exhaustive lists of DNA L-words for n sequences were created by combinatorial means. For words of length- L, the size of the list can be described mathematically: $t = 4^L$. The frequencies of these words are similarly found as in [44]. Once amassed, they are added to an $n \times t$ matrix to create a global profile of all L-word frequencies of all input sequences. Next is the development of the genetic distance for the suffix trees. This pairwise standard Euclidean distance between pairs of sequences is calculated by the following:

$$SED(S_1, S_2) = \sqrt{\sum_{w \in t} (f_{S_1 w} - f_{S_2 w})^2} \tag{8}$$

for w, representing the k-mer and t representing the exhaustive list of words, respectively, and $f_{S_i w}$ represents the relative frequency of w in the sequence S_i. These values may be applied to suffix trees for convenient sequence analysis across large sets of sequences.

Composition vectors based on *k*-mer frequencies

There are two main string composition vectors that we will discuss; the CV and the complete composition vector (CCV). Discussed in [48–50], a k-mer frequency CV for a genomic sequence is a distribution of frequencies of length- k motifs, which are used for comparison across sequences. The CV method contains motif frequencies of the same length, whereas the CCVs contain motif frequencies of unequal length.

The basic steps of creating CVs are the following: (i) find the frequencies of the motifs in a sequence, (ii) create a vector by organizing the frequencies in some order, (iii) compute the distance between every two composition vectors to form a distance matrix, and optionally (iv) construct the phylogenic tree based on the differences. This last step is not essential but may be helpful when evaluating the degree of closeness between a set of sequences.

Creation of Composition Vectors (CVs, CCVs)

The creation of a CCV is similar to that of a CV except that the input frequencies are made from strings of differing lengths. Despite its extra computational expense, the CCV method was found to provide finer evolutionary information than the CV method [32].

Following the discussion from [31, 32], we define S_1 to be a sequence consisting of n_1 nucleotides and let $f(\alpha_1 \ldots \alpha_k)$ to be the observed frequency of the length- k motif in S_1. We define α_i for $1 \leq i \leq k$ to be a nucleotide such that $\alpha_i \in \{\text{A, C, G, T}\}$ for $1 \leq k < n_1$. Next, for some constant K, the largest string length we consider, we define $V_{S_1} = (f_1, f_2, \ldots, f_{4^K})$ as the combined vector. Finally, we define $V_S = (S_1, S_2, \ldots, S_K)$ as the combined vector. The vectors, V_{S_1} and V_S, reflect both random mutation and selection. Lu *et al.* noted that there is an underestimation of selective evolution for both these vectors when the data are normalized according to Equation (9), which is also discussed in [51] and [52]. For the observed frequency of $\alpha_1, \alpha_2, \cdots, \alpha_k$, the normalizing equation is described by the following:

$$a(\alpha_1 \ldots \alpha_k) = \frac{f(\alpha_1 \ldots \alpha_k) - f_e(\alpha_1 \ldots \alpha_k)}{f_e(\alpha_1 \ldots \alpha_k)} \tag{9}$$

where f_e represents the expected frequency and is defined by:

$$f_e(\alpha_1 \ldots \alpha_k) = \frac{f(\alpha_1 \ldots \alpha_{k-1}) f(\alpha_2 \ldots \alpha_k)}{f(\alpha_2 \ldots \alpha_{k-1})} \times \frac{(n_1 - k + 1)(n_1 - k + 3)}{(n_1 - k + 2)^2}.$$

for $k \geq 3$ and $f_e(\alpha_1 \ldots \alpha_k)$. Lu *et al.* [32] note that this normalization method underestimates the actual reality of the data. Next, we describe their modified method, which overcomes this problem.

Creation of improved CCVs

To overcome this setback, Lu *et al.* [32] propose an improvement to CVs and CCVs. The improved

CCV (ICCV) is made assuming that the sequence bases all occur with equal probability, according to the expected frequency of a k-mer string. The variance of frequency of a given sequence S_1 of length- n_1 is also based on this assumption. We first define the expected motif frequencies and their variance in the vectors. For any given k-mer, a position in the sequence is given as:

$$x_i = \begin{cases} 1, & \text{if the } k\text{−mer begins at position } i \\ 0, & \text{otherwise} \end{cases}$$

for integer i such that $1 \leq i \leq (n_1 - k + 1)$. This upper bound is the maximum observed frequency for a string $\alpha_1 \cdots \alpha_k$ in S_1. Therefore, it can be shown that

$$f(\alpha_1 \cdots \alpha_k) = \sum_{i=1}^{n_1-k+1} x_i. \tag{10}$$

The expectation and variance of $f(\alpha_1 \cdots \alpha_k)$ are described in the following equations.

$$E[f(\alpha_1 \cdots \alpha_k)] = \sum_{i=1}^{n_1-k+1} E[x_i] = \frac{n_1 - k + 1}{4^k} \tag{11}$$

and the variance;

$$\begin{aligned} \mathrm{Var}[f(\alpha_1 \cdots \alpha_k)] = {} & \frac{n_1 - k + 1}{4^k}\left(1 - \frac{1}{4^k}\right) \\ & - \frac{2}{4^{2k}}(k-1)\left(n_1 - \frac{3}{2}k + 1\right) \\ & + \frac{2}{4^k}\sum_{i=1}^{k-1}(n_1 - k + 1 - t)\frac{J_r}{4^t}, \end{aligned}$$

where J_r, is defined by the following.

$$J_r = \begin{cases} 1, & \text{if } (\alpha_1 \cdots \alpha_{k-r}) = (\alpha_{r+1} \cdots \alpha_k) \\ 0, & \text{otherwise} \end{cases}$$

for integer r such that $1 \leq r \leq k - 1$. See [53] for a full derivation. One of the problems with the original CV and CCV concerns the denominator, which requires a square root operation without which Lu *et al.* warn of a problem of over-estimation. To mitigate the over-estimation problem, the authors apply the data's expectation and variance to the normalizing equation given later in the text and complete the construction of the ICCV. The normalization of each observed frequency of a k-mer string, k_{norm} is given by the following equation:

$$k_{norm} = \frac{f(a_1 \cdots a_k) - E[(a_1 \cdots a_k)]}{\sqrt{\mathrm{Var}[f(a_1 \cdots a_k)]}} \tag{12}$$

for $k \geq 1$.

Distance measurement

We next discuss how the distance between vectors is measured. For two sequences, S_1 and S_2, let their vectors be defined, $V_{S_1} = (\alpha_1, \alpha_2, \cdots, \alpha_k)$ of length- k and $V_{S_2} = (\beta_1, \beta_2, \cdots, \beta_k)$, also of length- k. We define the normalized distance between the vectors by the following:

$$D(V_{S_1}, V_{S_2}) = \frac{1 - C(V_{S_1}, V_{S_2})}{2}, \tag{13}$$

where $C(V_{S_1}, V_{S_2})$ is the cosine distance of the angle between V_{S_1} and V_{S_2} and is described by the following.

$$C(V_{S_1}, V_{S_2}) = \frac{\sum_{i=1}^{k} \alpha_i \cdot \beta_i}{\sqrt{\sum_{i=1}^{k} \alpha_i^2 \cdot \sum_{i=1}^{k} \beta_i^2}} \tag{14}$$

Lu *et al.* show that the ICCV method fixes the observed overestimation problems with the previous method and generates more accurate and robust results. They also show that its results are consistent with methods based on alignment by dynamic programming in phylogeny.

A revised string composition method

Chan *et al.* [31] revisit the composition vector method and apply an analysis of entropy from information theory and operations research. Their method begins by finding the frequencies of each base of a k-string sequence. For example, from ACTGCTATGC, the base frequencies are the following: $f(\mathtt{A}) = \frac{1}{5}$, $f(\mathtt{C}) = \frac{3}{10}$, $f(\mathtt{G}) = \frac{1}{5}$, $f(\mathtt{T}) = \frac{3}{10}$.

The second step is to estimate the expected frequency $q(u)$ for each k-string. For this step, the authors suggested determining the relationship between $q(\cdot)$ and $f(\cdot)$ by maximizing the following system of equations from Hua *et al.* [54]. Here, the entropy in $q(\cdot)$ is maximized given the frequency $f(v)$ for all $(k-1)$-strings v.

$$\begin{cases} q(vA) + q(vC) + q(vG) + q(vT) = f(v), \\ q(Av) + q(Cv) + q(Gv) + q(Tv) = f(v) \end{cases} \tag{15}$$

Chan *et al.* depart from the work of Hua *et al.* by making no assumptions between $q(\cdot)$ and $f(\cdot)$. Instead, they maximized the following equations, which estimate the expected frequencies $q(u)$.

$$q(\mathtt{LwR}) = \frac{f(\mathtt{Lw})f(\mathtt{wR})}{f(\mathtt{w})} \tag{16}$$

for $k \geq 3$, which was introduced by, Qi *et al.* [49] and,

$$q(\mathtt{LwR}) = \frac{f(\mathtt{L})f(\mathtt{wR}) + f(\mathtt{Lw})(\mathtt{R})}{2} \tag{17}$$

for $k \geq 2$, from Yu *et al.* [55]. For any k-string u, L and R represent the left and right nucleotides of the word and w represents the middle $(k-2)$-string located between them. In the second equation, all these elements are assumed to occur independently. From these equations, the authors created a new system of equations (later in the text) to solve where the right-hand side concerns sequence frequencies and the left-hand side concerns the estimations:

$$\begin{cases} q(\mathtt{vA}) + q(\mathtt{vC}) + q(\mathtt{vG}) + q(\mathtt{vT}) \\ \quad = \frac{f(\mathtt{Lw})}{f(\mathtt{w})}\left[f(\mathtt{wA}) + f(\mathtt{wC}) + f(\mathtt{wG}) + f(\mathtt{wT})\right] \\ q(\mathtt{Av}) + q(\mathtt{Cv}) + q(\mathtt{Gv}) + q(\mathtt{Tv}) \\ \quad = \frac{f(\mathtt{xR})}{f(\mathtt{x})}\left[f(\mathtt{Ax}) + f(\mathtt{Cx}) + f(\mathtt{Gx}) + f(\mathtt{Tx})\right]. \end{cases} \tag{18}$$

When this system is maximized, Chan *et al.* [31] note that the system generates a set of all possible estimation formulas $q(\cdot)$ from which one can be selected to maximize the entropy. In general, from any existing estimation formula $q(\cdot)$ given in terms of $f(\cdot)$, the authors note that the set of constraints such as the following can be derived:

$$\begin{cases} q(vA) + q(vC) + q(vG) + q(vT) = l(v), \\ q(Av) + q(Cv) + q(Gv) + q(Tv) = r(v) \end{cases} \tag{19}$$

where the left- and right-side frequency values, $l(v)$ and $r(v)$ are derived from frequency information $(f(v))$ for each length-$(m-1)$ motif v. To obtain the unique $q(u)$ for all u, the following optimization problem is solved:

$$\text{maximize}: -\sum_{i=1}^{4^k} q_i \log q_i$$

$$\text{subject to}: \begin{cases} q_i \text{ satisfies the system of equations} \\ q_i \geq 0 \text{ for all } i \end{cases}$$

where $-q_i \log q_i$ is Shannon's entropy calculation. The authors apply this information to phylogenetic tree analysis in a similar fashion as we saw in Lu *et al.* [32].

Maximum Entropy Principle. After solving the problem aforementioned, a system of noise estimation formulas is provided. Note: a motif appears as the following: $(\alpha_1 \cdots \alpha_m \alpha_n \cdots \alpha_k)$ and can be split into to sub words.

$$q^{MEP}(\alpha_1 \cdots \alpha_m \alpha_n \cdots \alpha_k) = \frac{l(\alpha_1 \cdots \alpha_m) r(\alpha_n \cdots \alpha_k)}{\sigma}, \tag{20}$$

where, q^{MEP} is the maximized entropy principle score for the sequence data and,

$$\sigma = \sum_{L \in \{A, C, G, T\}} l(\alpha_1 \cdots \alpha_m) = \sum_{R \in \{A, C, G, T\}} r(\alpha_n \cdots \alpha_k). \tag{21}$$

We note that $q^{MEP} = 0$ if $\sigma = 0$ and that $l(\cdot)$ and $r(\cdot)$ are parametric functions. Different $l(\cdot)$ and $r(\cdot)$ will give different estimation formulas and will have varying levels of success. The authors applied this test to create phylogenetic trees from simulated data sets. Their results showed differentiation of 'closely related' sequences.

D_2 statistic

The statistic D_2, is the number of approximate word matches of length k between sequences $S_1 = (\alpha_1, \ldots, \alpha_k)$ and $S_2 = (\beta_1, \ldots, \beta_k)$, with α_i and β_j belonging to an alphabet $\mathcal{A}$ (in this case, the DNA bases), which is distributed according to a letter distribution parameterized by η [56]. This statistic is applied to two populations of differing means, but identical dispersion matrices [57], to determine distance. Recently, the statistic has evolved to provide more exact approximations by asymptotic regimes for uniform and non-uniform distributions [58, 59]. Mathematically, the D_2 statistic is defined by the following. From [60], given sequences $S_1 = (\alpha_1, \ldots, \alpha_{n_1})$ of length- n_1 and $S_2 = (\beta_1, \ldots, \beta_{n_2})$ of length- n_2 and $W = \{w_1, \ldots, w_k\} \in \mathcal{A}^k$, then D_2 is defined by the following:

$$D_2 = \sum_{W \in \mathcal{A}^k} C_{s_1}(W) C_{s_2}(W) \tag{22}$$

where $C_{s_i}(W)$ is the number of occurrences of W in sequence S_i.

The $D2Z$ statistic [61] was developed to compare gene regulatory sequences and offered an improvement in performance to D_2, but could still fail due to noise complications [62, 60]. To combat this problem of noise, Reinert *et al.* [60] propose a new statistic D_2^S, which is a self-standardized D_2.

$$D_2^S = \sum_{W \in \mathcal{A}^k} \frac{\tilde{C}_{s_1}(W) \tilde{C}_{s_2}(W)}{\sqrt{\tilde{C}_{s_1}(W)^2 + \tilde{C}_{s_2}(W)^2}} \tag{23}$$

For $p_W = \prod_{i=1}^{t} p_{w_i}$, the probability of occurrence of w_i for $1 \leq i \leq k$ and $\tilde{n}_i = n_i - k + 1$ for i sequences, the centralized count variables, $\tilde{C}_{s_1}(W)$ and $\tilde{C}_{s_2}(W)$, are therefore denoted by the following.

$$\tilde{C}_{s_1}(W) = C_{s_1}(W) - \tilde{n}_1 p_W \text{ and } \tilde{C}_{s_2}(W) = C_{s_2}(W) - \tilde{n}_2 p_W$$

Reinert *et al.* also proposed a second statistic, D_2^*, which we shall presently define. To introduce this statistic, we replace $p(a)$, the unobserved feature probabilities, by $\tilde{p}(a)$ (the observed) for the relative count of letter a in the concatenation of the two sequences that are based on the assumption that the two sequences are independent. We note that these sequences are both independent and contain identically distributed (i.i.d.) bases. The estimated probability of occurrence of $W = \{w_1, \ldots, w_k\}$ is obtained by $\tilde{p}_W = \prod_{i=1}^{k} \tilde{p}_{w_i}$. We now define D_2^* by the following.

$$D_2^* = \sum_{W \in \mathcal{A}^k} \frac{\tilde{C}_{s_1}(W)\tilde{C}_{s_2}(W)}{\sqrt{\tilde{n}_1 \tilde{n}_2 \tilde{p}_W}} \tag{24}$$

The authors found that the D_2^* statistic outperformed both the D_2 and D_2^S statistics in terms of accurate detection of relatedness between two sequences. The statistical power of both D_2^* and D_2^S increases with sequence length and tends to 1 as the sequence length tends to infinity under a common motif model. When applied to organizing sequence reads of next-generation sequence assembly tasks, and to phylogeny tasks, the D_2^S statistic provided a powerful alignment-free comparison tool [63]. However, when studying phenomena in the patten transfer model such as HGT, the power of these statistics declines and converges to a limit that is generally <1 as the sequence length tends to infinity. The primary reason for this limitation is that the means of the word counts in these statistics eventually become increasingly similar to each other. This resemblance works to desensitize the detection of patterns between the sequences.

To improve the detection of relationships across sequences using alignment-free methods in the pattern transfer model, Liu *et al.* [64] developed new statistics (T^*, T^S and T^*_{sum}, described later in the text), which they claim have a better statistical power. The authors present them with simulations to demonstrate their power and to show that they are more appropriate for applications where long sequence-lengths are a concern.

Based on approximating the mean by a sample mean, the approach of the new statistic is to partition a long sequence of length- n_1 into consecutive non-overlapping (discrete) subintervals of length- r, $d_{sub} = \left\lfloor \frac{n_1}{r} \right\rfloor$. Then, the D_2^* and D_2^S values are calculated over each i^{th} subinterval for word counts w and are denoted $D(i)_2^*$ and $D(i)_2^S$, respectively. For, two sequences of length n_1 where, $S_1 = \{\alpha_1, \ldots, \alpha_k\}$ and $S_2 = \{\beta_1, \ldots, \beta_k\}$, these statistics are defined by the following equations.

$$T^* = \sum_{i=1}^{d_{sub}} D_2^*(i) \tag{25}$$

and

$$T^S = \sum_{i=1}^{d_{sub}} D_2^S(i) \tag{26}$$

The final statistic from [64] is drawn over two sequences S_1 and S_2 of lengths n_1 and n_2, respectively, to conclude the degree of relatedness.

$$T^*_{sum} = \sum_{i=1}^{n_1-k+1} S^*_{1i} + \sum_{i=1}^{n_2-k+1} S^*_{2i} \tag{27}$$

for,

$$S^*_{1i} = \max_{\{1 \leq j \leq n_1-k+1\}} M^*[i,j,k] \tag{28}$$

and

$$S^*_{2i} = \max_{\{1 \leq i \leq n_1-k+1\}} M^*[i,j,k] \tag{29}$$

where,

$$M^*[i,j,k] = D_2^*(S_1[i, i+k-1], S_2[j, j+k-1]) \tag{30}$$

Although D_2^* and D_2^S are generally more powerful statistics than T^*_{sum} and T^s_{sum} for the common motif model, this is not the case for studies concerning the pattern transfer model. For this reason, the statistics presented by Liu *et al.* are desirable in pattern transfer model applications when the sequence data are long.

DATA COMPRESSION AND DICTIONARIES

Alignment-free methods, involving data compression and dictionaries, are based on the idea that the more similar two sequences are to each other, then the better one sequence can be created from the parts of another. Inspired by LZ compression

technologies [65], we offer an example of sequence comparison, from Otu *et al.* [34].

For the sequences S_1, S_2 and S_Q, we define $H_E(S_1)$, $H_E(S_2)$ and $H_E(S_Q)$ to be the exhaustive sets of all words found using an approach from LZ-compression. We then analyze the sets of *sequence histories* to determine how much of one sequence can be built out of the sequence histories of another. We define $c_H(\cdot)$ to be the number of components in a history of a sequence S and $c_{min}(\{c_H(S)\}$ over all histories of S.

For S_1 and S_2, we have $c_{min}(S_1 S_Q) \leq c_{min}(S_1) + c_{min}(S_Q)$, by the sub-additivity of the LZ-complexity. To compute the closest similarity of S_1 and S_Q, $d(S_1, S_Q)$, and S_2 to S_Q, $d(S_2, S_Q)$, we take the smallest value of $\max\{c_{min}(S_1 S_Q) - c_{min}(S_1), c_{min}(S_Q S_1) - c_{min}(S_Q)\}$ and $\max\{c_{min}(S_2 S_Q) - c_{min}(S_2), c_{min}(S_Q S_2) - c_{min}(S_Q)\}$, respectively.

Compare the sequence similarity of S_1 to S_Q and S_2 to S_Q. We first find the sequence histories to compare distances. We introduce an example to demonstrate how this is performed.

$S_1 =$ ATGGC
$S_2 =$ ACGGT
$S_Q =$ ATGGC

- $S_1 =$ ATGGC
  - $H_E(S_1) =$ A, T, G, GC
  - $c_{min}(S_1) = 4$
- $S_2 =$ ACGGT
  - $H_E(S_2) =$ A, C, G, GT
  - $c_{min}(S_2) = 4$
- $S_Q =$ ATGGC
  - $H_E(S_Q) =$ A, T, G, GC
  - $c_{min}(S_Q) = 4$
- $S_1 S_Q =$ ATGGCATGGC
  - A, T, G, GC, ATGGC
  - $c_{min}(X S_Q) = c_{min}(S_Q S_1) = 5$
- $S_2 S_Q =$ ACGGTATGGC
  - A, C, G, GT, AT, GGC
  - $c_{min}(S_2 S_Q) = c_{min}(S_Q S_2) = 6$

- $d(S_1, S_Q) = \max\{c_{min}(S_1 S_Q) - c_{min}(S_1), c_{min}(Q S_1) - c_{min}(S_Q)\} = 1$
- $d(S_2, S_Q) = \max\{c_{min}(S_2 S_Q) - c_{min}(S_2), c_{min}(S_Q S_2) - c_{min}(S_Q)\} = 2$

By the author's method, we conclude that S_1 and S_Q are more similar, as $1 = d(S_1, S_Q) < d(S_2, S_Q) = 2$. The authors used this method to populate phylogenetic trees from simulated sequences to show clusterings of 'related' sequences.

Text compression algorithms

Data compression is nearly out of the scope of this article; however, they are worth mentioning because they also provide an alignment-free approach to comparing sequence data. These general purpose compression algorithms may be based on the Ziv and Lempel [65] methods (as seen earlier in the text). Recent advances have been developed in [66, 67] and [68]. Cao *et al.* [33] proposed a memory-based algorithm called *expert model* to compress DNA by applying statistical information, gained from previous encounters of a particular symbol.

Average common substring

Ulitsky *et al.* [35] built on information theoretic tools, such as Kullback–Leibler relative entropy, to find a distance between entire genomes, even if their lengths vary. The average common substring measure that they proposed is based on computing the average lengths of maximum common substrings. They used these average lengths between the sequences to construct phylogenetic trees from an efficient algorithm.

Let S_1 and S_2 be sequences, of lengths n_1 and n_2 where, $S_1 = (\alpha_1, \ldots, \alpha_{n_1})$ and $S_2 = (\beta_1, \ldots, \beta_{n_2})$. For any position i, let $r(i)$ be the length of longest substring in S_1 that *exactly matches* a substring in S_2 starting at some position j. These lengths $r(i)$ are averaged to get a measure, $L(S_1, S_2) = \sum_{i=1}^{n} r(i)/n_1$. As $L(S_1, S_2)$ represents a common sequence found in both sequences, then the longer it is, the more similar the sequences are to each other. This value is only a *similarity* measure and must still be converted to a distance value. The inverse is taken to get the distance, and then a 'correction term' is subtracted to ensure that the distance $d(S_1, S_1) = 0$ (will always be zero). This allows for, $d(S_1, S_2) = \frac{\log n_2}{L(S_1, S_2)} - \frac{\log n_1}{L(S_1, S_1)}$ where $L(S_1, S_1) = \frac{n_1}{2}$ to provide the correctional term, $2 \cdot \frac{\log(n_1)}{n_1}$, which converges to 0 as $n_1 \to \infty$. As the measure, $d(S_1, S_2)$ is not symmetric, the authors compute the final average common substring measurement between the two strings, $d_s(S_1, S_2)$ by the following.

$$d_s(S_1, S_2) = d_s(S_2, S_1) = \frac{d(S_1, S_2) + d(S_2, S_1)}{2} \tag{31}$$

We now show how to apply this method to determine the distance between two sequences. Let $S_1 =$ ACGTGCTATG and $S_2 =$ ACGCGCTA, of lengths $n_1 = 10$ and $n_2 = 8$, the method finds all common substrings as shown in Table 2.

$$L(S_1, S_2) = \frac{(1+2+3)+(1)+(1+2+3+4)+(1)+(1)}{10} = \frac{19}{10} = 1.9$$

and

$$L(S_1, S_1) = \frac{1+2+3+4+5+6+7+8+9+10}{10} = \frac{55}{10} = 5.5$$

Then the distance between the two sequences is

$$d(S_1, S_2) = \frac{\log 8}{1.9} - \frac{\log 10}{5.5} = 0.293$$

Similarly, we can calculate $D(B, A)$ as follows:

$$L(S_2, S_1) = \frac{(1+2+3)+(1)+(1+2+3+4)+(1)+(1)}{8} = \frac{19}{8} = 2.375,$$

$$L(S_2, S_2) = \frac{1+2+3+4+5+6+7+8}{8} = \frac{36}{8} = 4.5$$

and,

$$d(S_2, S_1) = \frac{\log 10}{2.375} - \frac{\log 8}{4.5} = 0.220.$$

For our example aforementioned, where $d_s(S_1, S_2)$ is not symmetric, the symmetric distance is $d_s(S_1, S_2) = d_s(S_2, S_1) = \frac{d(S_1, S_2)+d(S_2, S_1)}{2} = \frac{0.293+0.220}{2} = 0.257$. This value can be used as a weight for a sequence in a phylogenetic tree to show relations between sequences of a set.

APPLICATIONS OF ALIGNMENT-FREE METHODS

Biological data and sequence assembly

In genetic sequence assembly work, alignment technologies are important for determining the adjacency of reads (or contigs which are partially combined reads) to reconstruct the original sequence. During a typical *de novo* assembly task, a sequencing machine may split the genome into many millions (trillions) of reads that must be reassembled like from a jigsaw puzzle. This reconstruction task is computationally intensive, as each piece must be compared with every other piece in the pool to determine adjacency. This task is frustrated when there are foreign reads of other sequences to be assembled in the same data pool. The extra sequence data serve to massively broaden the search space when determining the adjacency of a read, as there are many more comparison operations to perform. To reduce the workload of the assembly project, it is therefore desirable to place all related reads into a unique groups (*bins*) and apply the main assembly algorithms to each organism separately.

Table 2: The similar and different chunks, taken in order from each sequence

Sequence	Same	Different	Same	Different
S_1	ACG	T	GCTA	TG
S_2	ACG	C	GCTA	

A novel approach, requiring no database support, was introduced by [69, 70] to order the organisms in the pool into separate bins. The authors' method creates CVs from restriction sites [71] to determine inter-sequence relatedness and place the sequences from the mixed pool into separate groups. This type of proposed alignment is for a global analysis, as it is able to process and compare sequences in a pool of arbitrarily size. They applied their work to the sequence assembly reads and contigs of *Bifidobacterium longum*, *Mycobacterium bovis*, *Clostridium tetani*, *Staphylococcus aureus*, *Burkholderia pseudomallei* and *Campylobacter jejuni*. Based on the similarity of proportional values contained in the CVs, the authors were able to differentiate the sequence material by organism.

The method uses *spectrum sets* that are lists of motifs made up of permutations of restriction enzymes, which are specific and unique sites in DNA where enzymes are able to cleave. To create a spectrum set from the bacterial restriction site, `GAATTC`, we observe that the motif contains, two `A`′*s*, two `T`′*s*, one `C` and one `G`. A spectrum set contains all motifs, which have exactly the same number of each base. For example, for the bacterial restriction site, `GAATTC`, there are 156 motifs in the spectrum set that have the same base composition. A vector of length-156 is constructed from the proportions of each of these motifs, which are contained in the sequence data. For example, to populate the vector V_{S_1} of the motif proportions of w_i for $i = \{1, \ldots, 156\}$ for sequence S_1 of length- n_1, the following equation is used,

$$V_{S_1} = \frac{c(w_i) * |w_i|}{n_1}, \quad (32)$$

where $c(\cdot)$ represents the number of occurrences of the motif in the sequence. This equation serves to normalize the proportions so that the values can be compared across diverse data sets. The authors noted that similar sequence data gave rise to similar vectors that they used to organize the sequence data.

Chromosomal data and phylogeny

In addition, in [70], it was shown that the method could also be applied to create phylogenetic trees, which were extremely similar to trees created by NCBI's taxonomy tree making software. In this work, they used chromosomal sequences of arbitrarily chosen organisms (*Caenorhabditis elegans*, *Canis lupus familiaris*, *Drosophila melanogaster*, *Mus musculus*, *Mycoplasma hyorhinis*, *Oryctolagus cuniculus* and *Rattus norvegicus*) and built a tree that replicated that of NCBI's taxonomy analysis software (available at http://www.ncbi.nlm.nih.gov/guide/taxonomy/).

HGT

HGT is the phenomenon where genetic material is shared between unrelated organisms. Evolutionary [72] and Phylogenetic studies [73] have observed common material between unrelated bacterial organisms, which suggests a parallel evolutionary history. The discovery of similar regions of DNA between two enormous genomes is not a trivial task, and therefore alignment-free methods have proven to be helpful in this field. In [36], the authors present *Alignment-Free Local Homology* (*alfy*), a method to determine HGT by an alignment-free approach. As determining evolutionary distances from word frequency data is a non-trivial task, the authors report that their method is conveniently able to make this determination.

We cite and discuss the method and example presented in [36] where the query sequence, denoted as S_Q of Table 3, is compared with the subject sequence, S_1. For each position in the query S_Q, the *alfy* algorithm determines the shortest substring that starts in query, which is absent from the subject sequence.

In Tables 4 and 5, this comparison task is shown by a string of numbers (match scores), which show the length of the substring starting in (S_Q) that are absent in (S_1). If the consecutive intervals created by these matching scores are wide (e.g. long strings of uninterrupted consecutive integers), then the sequences are closely related (similar); however, if the intervals are generally short, then the sequences are not closely related (dissimilar).

Table 3: The sequences to compare by the *alfy* method

Query	(S_Q) =	CGCGATTACT$
Subject	(S_1) =	CGCCCGGACT$
Subject	(S_2) =	TGAGATTCAG$

To compare sequences, we find the shortest sequence in the query (S_Q), which is absent from a subject.

Table 4: S_1 is compared with S_Q to determine the shortest substring in S_Q, which is absent from S_1

Subject	(S_1)	CGCCCGGACT$
Query	(S_Q)	CGCCCTGACT$
Matching score		6543325432

The matching numbers indicate the shortest unique substring starting at this position that is absent from the subject.

Table 5: Sequences S_1 and S_2 are compared with S_Q

Subject	(S_1)	CGCCCGGACT$
Subject	(S_2)	TGAGATTCAG$
Query	(S_Q)	CGCCCTGACT$
Matching score		4325432432
Implied HGT	(S_1) and (S_2)	*Bbbccccbbb*

The matching numbers indicate the shortest unique substring starting at this position that is absent from the subject. The HGT is described by a string of S_1 and S_2 characters to indicate where the subsequences likely originated.

We cite an example from another study concerning HGT by the same authors [13]. This method is applied to locating regions of common genetic material in *Escherichia coli* and recombinant HIV-1 strains. This method is similar because it locates local regions in subject sequences that are closely related to the query sequence. In Table 6, sequences S_1 to S_3 are the subjects and S_Q is the query. We find which parts of S_Q most closely resemble the subject sequences. The sections of sequence material are written in an interval notation: $S_Q[1,2] = \texttt{TA}$ matches $S_3[1,2]$, $S_Q[3,4] = \texttt{GC}$ matches $S_2[1,2]$. By this system, we claim that $S_Q[1,2]$ is most closely related to S_3, and $S_Q[3,4]$ is most closely related to S_2.

During the sequence comparison task of query-to-subject, in [13], the authors denote the length of the

Table 6: We wish to determine the sequence relations based on common sequence material

	1	2	3	4	5
S_Q	T	A	G	C	$
S_1	G	A	$		
S_2	G	C	C	$	
S_3	T	A	$		

The query sequence is S_Q and subjects are S_1 through S_3.

shortest query sequence prefix by $h_{i,p}$. The query suffix, $Q[p, |Q|]$ denotes the sequence starting at position p, which is absent in subject sequences. The length of the *longest* subsequence starting at $Q[p]$ taken over all subject sequences is denoted, $H_p = \max_{\{1 \le i \le n\}} h_{i,p}$, where h_p is bounded by the query length: $H_p \le |Q| - p + 1$. For example, in Table 6, $H_{1,1} = \mathtt{T} = 1$; $H_{2,1} = \mathtt{T} = 1$; $H_{3,1} = \mathtt{TAG} = 3$; and $H_1 = \max_{\{1,1,3\}} = 3$. Conversely, the longest subject subsequences, which start at $Q[p]$ are found in a subject sequence, are denoted by $S_p = \{S_i \in S | h_{i,p} = H_p\}$. Based on these properties, the authors note that the longest sequence from Table 6 is S_3 (the most similar subject to sequence S_Q).

ADVANTAGES AND DISADVANTAGES OF METHODS

The method that an algorithm uses to gain its statistical data for an analysis is an important part of the whole operation. A fault at this stage would travel throughout the comparison task and upset the conclusion. In this section, we describe the generation of the motif frequency distributions, and we discuss how this initial statistical work may not always be appropriate for a particular data set.

The methods of 'Factor Frequencies' section are powerful methods to use in sequence comparison tasks, as they do not concern the location of the motifs they analyze. Their algorithms are efficient, as they are generally of a linear complexity. They contrast to the general high complexity of the algorithms that are based on dynamic programming. The results of factor frequency methods are adaptable and can be conveniently applied to an analysis by mutual information ('A comparison by frequencies and the Jensen–Shannon Divergence test' section), *k*-mers ('Suffix trees by k-mer frequencies' section) or by CVs ('Composition vectors based on *k*-mer frequencies' and 'A revised string composition method sections').

As the factor frequency methods are generated by word occurrences in a sequence, it important to choose words that are not likely to commonly appear in a sequence. As a general rule in DNA, the shorter the word, then the more likely it will appear randomly in a sequence. In 'BBC by analysis of mutual information' and 'A comparison by frequencies and the Jensen–Shannon Divergence test' section, vectors were created out of pairs of DNA bases. Although this may be an simple way to illustrate the concept, frequencies made up of these short pairs have less meaning than frequencies made up by longer words because any particular DNA pair has a probability of $\frac{1}{4^2} = \frac{1}{16}$ to occur randomly. We note that for sequences that are largely dissimilar, then shorter words (hence shorter CVs) should be used to create the feature frequency distributions. However, longer words, (hence, longer CVs, assuming an exhaustive list of motifs) may be used when the sequences are known to be similar, such as when they are related, [41]. The methods of 'Factor Frequencies' section are well suited for this application using both long and short motifs. They also function well when the location of the motif in the sequence is not important, as in the case of synteny.

Unlike the approaches of 'Factor Frequencies' section where the frequency distributions were generated by user-specified motifs, the methods of 'Data Compression and Dictionaries' section 'choose' their own sizes of words for their sequence comparison task. In the methods proposed by [34] (LZ compression based) and [36] (HGT), the word size is not a parameter set by the researcher. These kinds of algorithms are useful to comparison tasks where it is not clear about the 'correct' kinds of motifs to employ. In the case of factor frequency methods, when designing a list of motifs from which to generate a frequency distribution, an exhaustive list is likely used. As we have previously mentioned, the longer the motif, the larger the exhaustive list. Finding the frequencies of these extra motifs may add additional computational time to the task. Therefore, compression-based methods may be more suitable to comparisons where longer motifs are desirable, such as when the sequences are similar. This might be because the words of varying size and composition will be more similar across related sequences.

Table 7: Summary of the discussed methods in this article

Section	Method	Author	Alignment	Citation
BBC by analysis of mutual information	BBC	Lui *et al.*	Global	[30]
FFP	Oligonucleotide profiling	Arnau *et al.*	Local	[43]
FFP	Feature frequency	Sims *et al.*	Local	[44]
Suffix trees by k-mer frequencies	*k*-mers frequencies	Soars *et al.*	Global	[47]
Composition vectors based on k-mer frequencies	Composition vectors	Lu *et al.*	Global	[32]
A revised string composition method	Composition vectors	Chan *et al.*	Global	[31]
D2 statistic	D_2^* *Statistic*	Reinert *et al.*	Global	[60]
D2 statistic	Improved D_2 statistic	Lui *et al.*	Global	[64]
Data compression and dictionaries	Sequence distance	Otu *et al.*	Global	[34]
Text compression algorithms	DNA compression	Cao *et al.*	Global	[33]
Average common substring	Average common substring	Ulitsky *et al.*	Global	[35]
HGT	ALign. Free local homology	Domazet-Lõo *et al.*	Local	[13, 36]
Biological data and sequence assembly	Sequence assembly	Bonham-Carter *et al.*	Global	[69]
Chromosomal data and phylogeny	Phylogeny	Bonham-Carter *et al.*	Global	[70]

The column 'Alignment' contains the best suggested use of the method.

CONCLUSION

Comparison of sequence data represents a large problem in computational biology research. Discovery is often frustrated by obstacles such as synteny or other forms of genetic recombination, preventing methods of dynamic programming from working effectively. We provide a summary of the methods that we have discussed in Table 7, listed by sections, references and authors. When confronted with a large number of comparison tasks, which are unsuitable for traditional forms of alignment from dynamic programming, these alignment-free methods may be the only feasible approach for completing the tasks to permit discovery. This is because the alignment-free methods do not function based on the location of genes or regions in each sequence. When the location of these regions is not important for the analysis, alignment-free methods like the ones included in the present review may accomplish the goal of comparing genetic sequences.

As there is more sequence data available today than ever before, there are many more projects that depend on sequence comparison. For discovery to be made, this work will have to be done by other technologies such as those based on dynamic programming, which have obvious limitations. Alignment-free methods generally require less computational resources and use algorithms that are typically of linear complexity. These incorporated elements are appropriate for advancing comparative bioinformatics research.

It is our hope that this review provides useful information for researchers who are studying alignment-free methods and are using them in the analysis of genomic sequences and metagenomes. As the mathematical aspects of the aforementioned tools are themselves an obstacle, it is also our hope that this review helps to introduce the reader to some of the more complicated calculations that are associated with these alignment-free tools for discovery. Furthermore, we envisage that this review will serve as a useful reference in identifying open problems and driving future research in sequence comparison.

Key Points

- Dynamic programming methods are inappropriate for the voluminous.
- Statistical methods from information theory and other areas of mathematics are now used to conveniently differentiate sequence data based on extracted motif distributions.
- We review popular methods of comparing word distributions between sequences to infer distance.
- Application of these sequence comparison methods extend to the following: sequence assembly, phylogeny, HGT and many other areas where sequence separation is necessary.

ACKNOWLEDGEMENTS

The authors thank Joe Fitzpatrick for his work in proof-reading our manuscript. In addition, authors thank Janyl Jumadinova for helping us format the manuscript.

FUNDING

UNO-Bioinformatics Core Facility, funded by the grants from the National Center for Research Resources [5P20RR016469] and the National Institute for General Medical Science (NIGMS) [8P20GM103427].

References

1. Gusfield D. *Algorithms On Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press, Cambridge, 1997.
2. Knuth DE, Morris JH, Pratt VR. Fast pattern matching in strings. *SIAM J Comput* 1977;**6**:323–50.
3. Boyer RS, Moore SJ. A fast string searching algorithm. *Commun ACM* 1977;**20**:762–72.
4. Horspool RN. Practical fast searching in strings. *Softw Pract Exp* 1980;**10**:501–6.
5. Apostolico A, Giancarlo R. The boyer moore galil string searching strategies revisited. *SiamJ Comput* 1986;**15**:98–105.
6. Ukkonen E. Finding approximate patterns in strings. *J Algor* 1985;**6**:132–7.
7. Navarro G. A guided tour to approximate string matching. *J ACM Comp Surv* 2001;**33**:31–88.
8. Cheng L-L, Cheung DW, Yiu S-M. Approximate string matching in DNA sequences. In: *Procceedings of the Eighth International Conference on Database Systems for Advance Applications,* 2003. pp. 303–10. http://dx.doi.org/10.1109/DASFAA.2003.1192395.
9. Koonin E. The emerging paradigm and open problems in comparative genomics. *Bioinformatics* 1999;**15**:265–66.
10. Wooley J. Trends in computational biology: a summary based on a RECOMB plenary lecture. *J Comput Biol* 1999; **6**:459–74.
11. Li H, Homer N. A survey of sequence alignment algorithms for next-generation sequencing. *Brief Bioinform* 2010;**11**: 473–83.
12. Liao B-Y, Chang Y-J, Ho J-M, *et al*. The unimarker (um) method for synteny mapping of large genomes. *Bioinformatics* 2004;**20**:3156–65.
13. Domazet-Lošo M, Haubold B. Alignment-free detection of horizontal gene transfer between closely related bacterial genomes. *Mob Genet Elements* 2011;**1**:230–5.
14. Berkman PJ, Skarshewski A, Manoli S, *et al*. Sequencing wheat chromosome arm 7BS delimits the 7BS/4AL translocation and reveals homoeologous gene conservation. *Theor Appl Genet* 2012;**124**:423–32.
15. Crameri A, Raillard S-A, Bermudez E, *et al*. DNA shuffling of a family of genes from diverse species accelerates directed evolution. *Nature* 1998;**391**:288–91.
16. Orobitg M, Cores F, Guirado F, *et al*. Enhancing the scalability of consistency-based progressive multiple sequences alignment applications. In: *Parallel & Distributed Processing Symposium (IPDPS), 2012 IEEE 26th International,* 2012; pp. 71–82.
17. Eddy SR. What is dynamic programming? *Nat Biotechnol* 2004;**22**:909–10.
18. Smith T, Waterman M. Identification of common molecular subsequences. *J Mol Biol* 1981;**147**:195–7.
19. Needleman S, Wunsch C. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol* 1970;**48**:443–53.
20. Larkin M, Blackshields G, Brown N, *et al*. Clustalw and clustalx version 2. *Bioinformatics* 2007;**23**:2947–8.
21. Altschul SF, Madden TL, Schäffer AA, *et al*. Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic Acids Res* 1997;**25**:3389–402.
22. Chen C, Rajapakse JC. Grid-enabled blastz: application to comparative genomics. *J VLSI Signal Process Syst Signal Image Video Technol* 2007;**48**:301–9.
23. Kent WJ. Blatthe blast-like alignment tool. *Genome Res* 2002;**12**:656–64.
24. Schadt EE, Linderman MD, Sorenson J, *et al*. Computational solutions to large-scale data management and analysis. *Nat Rev Genet* 2010;**11**:647–57.
25. Chenna R, Sugawara H, Koike T, *et al*. Multiple sequence alignment with the clustal series of programs. *Nucleic Acids Res* 2003;**31**:3497–500.
26. Katoh K, Toh H. Parallelization of the mafft multiple sequence alignment program. *Bioinformatics* 2010;**26**: 1899–900.
27. Hara Y, Imanishi T. Abundance of ultramicro inversions within local alignments between human and chimpanzee genomes. *BMC Evol Biol* 2011;**11**:308.
28. Vinga S, Almeida J. Alignment-free sequence comparison–a review. *Bioinformatics* 2003;**19**:513–23.
29. Mantaci S, Restivo A, Sciortino M. Distance measures for biological sequences: some recent approaches. *Int J Approx Reason* 2008;**47**:109–24.
30. Liu Z, Meng J, Sun X. A novel feature-based method for whole genome phylogenetic analysis without alignment: application to HEV genotyping and subtyping. *Biochem Biophys Res Commun* 2008;**223**:223–30.
31. Chan RH, Chan TH, Yeung HM, *et al*. Composition vector method based on maximum entropy principle for sequence comparison. *IEEE/ACM Trans Comput Biol Bioinform,* 2011. Mar 3.
32. Lu G, Zhang S, Fang X. An improved string composition method for sequence comparison. *BMC Bioinform* 2008;**9**: S15.
33. Cao MD, Dix TI, Allison L, *et al*. A simple statistical algorithm for biological sequence compression. In: *Data Compression Conference,* 2007. pp. 43–52.
34. Out HH, Sayood K. A new sequence distance measure for phylogenetic tree construction. *Bioinformatics* 2003;**19**: 2111–30.
35. Ulitsky I, Burstein D, Tuller T, *et al*. The average common substring approach to phylogenomic reconstruction. *J Comp Bio* 2006;**13**:226–50.
36. Domazet-Lošo M, Haubold B. Alignment-free detection of local similarity among viral and bacterial genomes. *Bioinformatics* 2011;**27**:1466–72.
37. Shannon C. The mathematical theory of communication. *Bell Syst Tech J* 1948;**27**:379–429.
38. Kim I, Watada J, Pedrycz W, *et al*. Pattern clustering with statistical methods using a DNA-based algorithm. *IEEE Trans Nanobioscience* 2012;**11**:100–10.
39. Wang J-D. A comparison study of virus classification by genome sequences. In: *Bioinformatics and Bioengineering (BIBE), 2011 IEEE 11th International Conference,* 2011. pp. 270–3.
40. Zhang L, Meng J, Liu H, *et al*. Clustering DNA methylation expressions using nonparametric beta mixture model. In: *Genomic Signal Processing and Statistics (GENSIPS), 2011 IEEE International Workshop,* 2011. pp. 170–3.
41. Wu T-J, Huang Y-H, Li L-A. Optimal word sizes for dissimilarity measures and estimation of the degree of

dissimilarity between DNA sequences. *Bioinformatics* 2005; **21**:4125–32.

42. Dai Q, Li L, Liu X, *et al*. Integrating overlapping structures and background information of words significantly improves biological sequence comparison. *PLoS One* 2011;**6**: e26779.
43. Arnau V, Gallach M, Marín I. Fast comparison of DNA sequences by oligonucleotide profiling. *BMC Res Notes* 2008;**1**:5.
44. Sims GE, Jun SR, Kim SH. Alignment-free genome comparison with feature frequency profiles (ffp) and optimal resolutions. *Proc Natl Acad Sci USA* 2008;**106**:2677–82.
45. Lin J. Divergence measures based on the shannon entropy. *IEEE Trans Inf Theory* 1991;**37**:145–51.
46. Prasad AB, Allard MW, Green ED. Confirming the phylogeny of mammals by use of large comparative sequence data sets. *Mol Biol Evol* 2008;**25**:1795–808.
47. Soares I, Goios A, Amorim A. Sequence comparison alignment-free approach based on suffix tree and l-words frequency. *Sci World J* 2012;**2012**:450124.
48. Hao B, Qi J, Wang B. Prokaryotic phylogeny based on complete genomes without sequence alignment. *Mod Phys Lett B* 2003;**2**:1–4.
49. Qi J, Wang B, Hao B-I. Whole proteome prokaryote phylogeny without sequence alignment: a k-string composition approach. *J Mol Evol* 2004;**58**:1–11.
50. Wu X, Wan X, Wu G, *et al*. Phylogenetic analysis using complete signature information of whole genomes and clustered neighbour-joining method. *Int J Bioinform Res Appl,* 2006;**2**:219–48.
51. Brendel V, Beckmann JS, Trifonov EN. Linguistics of nucleotide sequences: morphology and comparison of vocabularies. *J Biomol Struct Dyn* 1986;**4**:11–21.
52. Hao B, Qi J. Prokaryote phylogeny without sequence alignment: from avoidance signature to composition distance. *J Bioinform Comput Biol* 2004;**2**:1–19.
53. Gentleman J, Mullin R. The distribution of the frequency of occurrence of nucleotide subsequences, based on their overlap capability. *Biometrics* 1989;**45**:35–52.
54. Hua R, Wanga B. Statistically significant strings are related to regulatory elements in the promoter regions of saccharomyces cerevisiae. *Physica A* 2001;**290**:464–74.
55. Yu Z-G, Zhou L-Q, Anh VV, *et al*. Phylogeny of prokaryotes and chloroplasts revealed by a simple composition approach on all protein sequences from complete genomes without sequence alignment. *J Mol Evol* 2005;**60**: 538–45.
56. Forêt S, Wilson SR, Burden CJ. Characterizing the d2 statistic: word matches in biological sequences. *Stat Appl Genet Mol Biol* 2009;**8**:1–21.
57. Waterman M. *Introduction to Computational Biology: Maps, Sequences and Genomes*. Chapman and Hall, 1995.
58. Forêt S, Kantorovitz M, Burden CJ. Asymptotic behavior of k-word matches between two uniformly distributed sequences. *BMC Bioinform* 2006;**7**:S21.
59. Lippert R, Huang H, Waterman M. Distributional regimes for the number of k-word matches between two random sequences. *Proc Natl Acad Sci USA* 2002;**99**:13980–9.
60. Reinert G, Chew D, Sun F, *et al*. Alignment-free sequence comparison (i): Statistics and power. *J Comput Biol* 2009;**16**: 1615–34.
61. Kantorovitz MR, Robinson GE, Sinha S. A statistical method for alignment-free comparison of regulatory sequences. *Bioinformatics* 2007;**23**:i249–55.
62. Wan L, Reinert G, Sun F, *et al*. Alignment-free sequence comparison (ii): theoretical power of comparison statistics. *J Comput Biol* 2010;**17**:1467–90.
63. Song K, Ren J, Zhai Z, *et al*. Alignment-free sequence comparison based on next generation sequencing reads. In: *Research in Computational Molecular Biology*. Springer, 2012. pp. 272–85.
64. Liu X, Wan L, Li J, *et al*. New powerful statistics for alignment-free sequence comparison under a pattern transfer model. *J Theor Biol* 2011;**284**:106–16.
65. Ziv J, Lempel A. A universal algorithm for sequential data compression. *IEEE Trans Inf Theory* 1977;**23**:337–43.
66. Mohammed MH, Dutta A, Bose T, *et al*. Deliminate–a fast and efficient method for loss-less compression of genomic sequences. *Bioinformatics* 2012;**28**:2527–9.
67. Cox AJ, Bauer MJ, Jakobi T, *et al*. Large-scale compression of genomic sequence databases with the burrows–wheeler transform. *Bioinformatics* 2012;**28**:1415–9.
68. Kozanitis C, Saunders C, Kruglyak S, *et al*. Compressing genomic sequence fragments using slimgene. *J Comput Biol* 2011;**18**:401–13.
69. Bonham-Carter O, Ali H, Bastola D. A meta-genome sequencing and assembly preprocessing algorithm inspired by restriction site base composition. In: *Bioinformatics and Biomedicine Workshops (BIBMW), 2012 IEEE International Conference on* IEEE, 2012. pp. 696–703.
70. Bonham-Carter O, Ali H, Bastola DK. A meta-genome sequencing and assembly preprocessing algorithm inspired by restriction site base composition. *BMC Bioinformatics* 2013.
71. Bonham-Carter O, Najjar L, Thapa I, *et al*. Distributions of palindromic proportional content in bacteria. Short Paper. In: *The 8th International Symposium on Bioinformatics Research and Applications (ISBRA 2012)*, Dallas, TX, 2012.
72. Syvanen M. Evolutionary implications of horizontal gene transfer. *Annu Rev Genet* 2012;**46**:341–58.
73. Moreno-Letelier A, Olmedo G, Eguiarte LE, *et al*. "Parallel evolution and horizontal gene transfer of the pst operon in firmicutes from oligotrophic environments. *Int J Evol Biol*. vol. 2011;**2011**.