OXFORD

Briefings in Bioinformatics, 20(4), 2019, 1222–1237

doi: 10.1093/bib/bbx161
Advance Access Publication Date: 6 December 2017
Paper

A survey and evaluations of histogram-based statistics in alignment-free sequence comparison

Brian B. Luczak, Benjamin T. James and Hani Z. Girgis

Corresponding author. Hani Z. Girgis, Tandy School of Computer Science, The University of Tulsa, 800 South Tucker Drive, Tulsa, OK 74104, USA.
E-mail: hani-girgis@utulsa.edu

Abstract

Motivation: Since the dawn of the bioinformatics field, sequence alignment scores have been the main method for comparing sequences. However, alignment algorithms are quadratic, requiring long execution time. As alternatives, scientists have developed tens of alignment-free statistics for measuring the similarity between two sequences. **Results:** We surveyed tens of alignment-free *k*-mer statistics. Additionally, we evaluated 33 statistics and multiplicative combinations between the statistics and/or their squares. These statistics are calculated on two *k*-mer histograms representing two sequences. Our evaluations using global alignment scores revealed that the majority of the statistics are sensitive and capable of finding similar sequences to a query sequence. Therefore, any of these statistics can filter out dissimilar sequences quickly. Further, we observed that multiplicative combinations of the statistics are highly correlated with the identity score. Furthermore, combinations involving sequence length difference or Earth Mover's distance, which takes the length difference into account, are always among the highest correlated paired statistics with identity scores. Similarly, paired statistics including length difference or Earth Mover's distance are among the best performers in finding the *K*-closest sequences. Interestingly, similar performance can be obtained using histograms of shorter words, resulting in reducing the memory requirement and increasing the speed remarkably. Moreover, we found that simple single statistics are sufficient for processing next-generation sequencing reads and for applications relying on local alignment. Finally, we measured the time requirement of each statistic. The survey and the evaluations will help scientists with identifying efficient alternatives to the costly alignment algorithm, saving thousands of computational hours. **Availability:** The source code of the benchmarking tool is available as Supplementary Materials.

Key words: alignment-free k-mer statistics; DNA sequence comparison; k-mer histograms; paired statistics

Introduction

Throughout the past decade in the field of bioinformatics, the shear amount of genomic data being produced has eclipsed the rate that computers can process it. Sequence comparison algorithms are among the most fundamental tools for analyzing the vast amount of DNA sequences. Devised in 1970, the Needleman–Wunsch alignment algorithm [1] was able to align the sequences of two proteins. This algorithm was shown to have extensive applications to determining the similarity between two nucleic acid or amino acid sequences. The alignment method of keeping track of insertions, deletions and substitutions between two sequences has spawned a wave of other 'alignment-based' approaches [2, 3] such as the popular BLAST series [4].

However, because the Needleman–Wunsch-based alignment algorithms are quadratic in terms of the sequence length, they are too costly to compute as the sequence length grows and the

Brian B. Luczak is an undergraduate student at the University of Tulsa (TU), majoring in mathematics and minoring in computer science.
Benjamin T. James is an undergraduate student at TU. He is pursuing majors in computer science and mathematics and minors in bioinformatics and high-performance computing.
Hani Z. Girgis is an Assistant Professor of computer science at TU. He did his postdoctoral work at the National Institutes of Health and Johns Hopkins University, and his undergraduate and graduate studies at the State University of New York at Buffalo. He majored in biology and computer science.
Submitted: 15 August 2017; **Received (in revised form):** 13 October 2017

number of comparisons increases. For example, the deficiencies of alignment-based methods are apparent in next-generation sequencing (NGS) data with millions of reads and the costly task of whole-genomic comparison [5]. Furthermore, rearrangements of entire blocks of base pairs are highly detrimental to the way alignment is calculated [6, 7]. The realization of these issues has led to the development of many efficient 'alignment-free' methods [6], which will be reviewed and evaluated on DNA sequences in this study. Although many methods, such as string compression, chaos theory and universal sequence maps exist [6, 8–10], this article focuses on the widely used method of k-mer frequencies (k-tuples) or feature frequency profiles [5, 11–13]. To use this method of k-mer frequencies, a histogram of k-mer counts is generated for the respective sequences that need to be compared [14, 15]. Next, the two histograms are compared using one of the many statistical similarity/distance measures.

A variety of review papers have discussed some of these methods [6, 16, 17] along with a statistical physics perspective [18]. However, no attempt has been made to review and evaluate the performance of a large number of alignment-free k-mer statistics. Further, the effects of combining multiple statistics together have not been studied yet. To this end, we have evaluated 33 statistics and the multiplicative combinations of every two statistics. One of the most important strengths of these statistics is their speed and relatively low cost [19, 20]; however, they can sometimes be less sensitive [21]. For this reason, we used the identity score obtained by the Needleman–Wunsch global alignment algorithm as the basis for comparison in several experiments. In addition, one experiment was evaluated according to local alignment identity scores. We propose several application-based methods that specifically measure each statistic's effectiveness based on its ability to be used instead of the identity score.

This manuscript is organized as the following. First, the statistics are surveyed. Next, we describe the data used in the evaluation experiments. Then, the evaluation results are presented. Finally, we conclude.

Survey of alignment-free k-mer statistics

Owing to past literature on classifying a comprehensive collection of histogram distances [17], we will be organizing the survey based on statistical families. The list of families includes Minkowski, match/mismatch, intersection, D2, inner product, squared chord, Markov, divergence and a variety of other statisitcs. Figure 1 diagrams these families and shows examples. In this section, a summary of each statistic or family will be included along with some initial thoughts.

Discussion on notation

To start, we define some notation concerning k-mer frequencies and histograms, which will be a primary focus throughout the article. Let s and t denote two sequences with corresponding lengths $len(s)$ and $len(t)$. If we consider the set K as the set of all possible words w determined over the alphabet [A, C, G, T], then the number of all possible words for DNA sequences is 4^k with k representing the length of each word. For the rest of the article, each of these k-length words will be referred to as a k-mer. We associate each sequence s and t with their corresponding histograms or word-count vectors as h_s and h_t as shown in Equation (1).

$$h_x = < c(w_1), c(w_2), c(w_3), \ldots, c(w_{|K|}) > . \quad (1)$$

Here, $c(w_i)$ represents the count of the ith k-mer in sequence x.

Minkowski family

In many areas of math and science, Euclidean distance is one of the most widely known statistics for comparing two sequences, i.e. their corresponding histograms as shown in Equation (2).

$$Euclidean(h_s, h_t) = \sqrt{\sum_{w \in K} (h_s(w) - h_t(w))^2}. \quad (2)$$

In this equation, h_s and h_t are the two histograms of the sequences s and t. Although the concept of Euclidean distance has been around since the Greek era, it was not until Herman Minkowski in the late 19th century that variations of this distance were created [17]. These variations include city block distance, which is also known as Manhattan distance, and a generalized form known as Minkowski distance as shown in Equation (3).

$$Minkowski(h_s, h_t) = \sqrt[p]{\sum_{w \in K} (h_s(w) - h_t(w))^p}. \quad (3)$$

Instead of using the exponents 2 and 1/2 as in Euclidean distance, Minkowski considered a generalized power p. From this general idea, we have city block when $P = 1$ and Chebyshev distance as shown in Equation (4) when $P \to \infty$ [17].

$$Chebyshev(h_s, h_t) = \max_{w \in K} |h_s(w) - h_t(w)|. \quad (4)$$

Additionally, the idea of z-score standardization is important to consider for statistics that are not self-standardizing such as the Minkowski family. For example, creating the standardized histograms h_s^z and h_t^z by using the mean and SD leads to the definition of EuclideanZ as shown in Equation (5).

$$EuclideanZ(h_s, h_t) = Euclidean(h_s^z, h_t^z). \quad (5)$$

Match/mismatch family

Although there are many ways to compare the two histograms, some of the most efficient methods involve simply counting whether the counts match. As it is defined in the Deza Encyclopedia [22], Hamming distance counts how many times the k-mer counts match and then divides by the number of possible k-mers as shown in Equation (6).

$$Hamming(h_s, h_t) = \frac{1}{4^k} \sum_{w \in K} h_s(w) == h_t(w). \quad (6)$$

Here, the symbol $==$ represents logical equality, evaluating to 1 if the two counts are the same and to 0 otherwise. Jaccard distance is simply the same as Hamming except that it only examines the nonzero k-mer counts. Equation (7) describes another statistic from this family referred to as Mismatch distance.

$$Mismatch(h_s, h_t) = \sum_{w \in K} h_s(w) \neq h_t(w). \quad (7)$$

Intersection family

Intersection distance, known as Czekanowski distance [17], is based on the intersection of the frequencies of k-mers divided by the union of the counts. Equation (8) defines this distance.

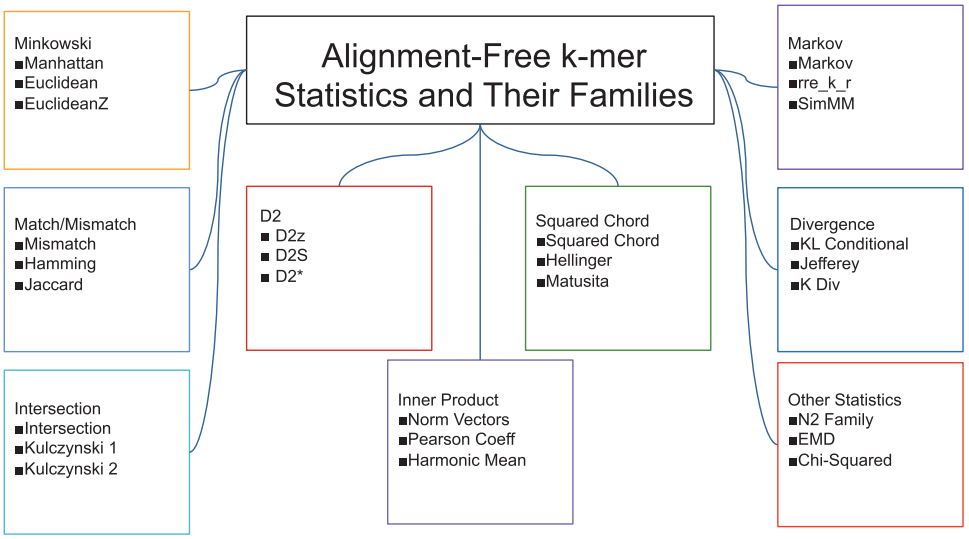

Figure 1. Alignment-free k-mer statistics grouped by statistical families. These families are based on a classification by Cha [17]. This figure provides a visual representation of several alignment-free k-mer statistics from their respective families. Each member of a statistical family shares a common functional element such as a histogram dot-product (Inner Product), minimum/maximum (Intersection), Markov model (Markov family) or overarching radical (Minkowski); although a variety of statistical families exist such as the χ^2 family, many recently developed methods do not fall into any specific category, e.g. N2, DMk and EMD.

The min function allows this statistic to effectively determine the overlap between the two distributions by recording how many of each k-mer are in both sequences. A few statistics in this same family include Kulczynski Similarity 1 and 2 as shown in Equations (9) and (10).

$$Intersection(h_s, h_t) = \sum_{w \in K} \frac{2 * min(h_s(w), h_t(w))}{h_s(w) + h_t(w)}. \quad (8)$$

$$Kulczynski1(h_s, h_t) = \sum_{w \in K} \frac{min(h_s(w), h_t(w))}{|h_s(w) - h_t(w)|}. \quad (9)$$

$$Kulczynski2(h_s, h_t) = A_\mu \sum_{w \in K} min(h_s(w), h_t(w)). \quad (10)$$

Here, A_μ is the scalar value $\frac{4^k(\mu_s+\mu_t)}{2\mu_s\mu_t}$, where μ_s and μ_t represent the mean k-mer counts for h_s and h_t.

χ^2 distance

As defined by Equation (11), χ^2 distance is the sum over all the k-mers of Manhattan distance squared divided by the sum of the two k-mer counts [17].

$$\chi^2(h_s, h_t) = \sum_{w \in K} \frac{(h_s(w) - h_t(w))^2}{h_s(w) + h_t(w)}. \quad (11)$$

Canberra distance

Canberra distance, as in Equation (12), at first glance, is somewhat of a hybrid between Manhattan distance and χ^2 distance [22]. In the original source, absolute value bars are included in the denominator. However, as k-mer counts are always positive, they will not be necessary when comparing sequences.

$$Canberra(h_s, h_t) = \sum_{w \in K} \frac{|h_s(w) - h_t(w)|}{h_s(w) + h_t(w)}. \quad (12)$$

D_2 statistic and its variations

On its own, the D_2 statistic is one of the most intuitive ways to find the similarity between two sequences as shown in Equation (13) [23].

$$D_2(h_s, h_t) = \sum_{w \in K} h_s(w) h_t(w). \quad (13)$$

Although taking the inner product between two histograms is time efficient, the results are not standardized and identical sequences can produce entirely different distances. For example, taking the dot product of the vector (1, 2, 3) with itself yields 14, whereas (1, 1, 1) · (1, 1, 1) yields 3. To fix some of the drawbacks, one method is to use the mean and SD similar to EuclideanZ as shown in Equation (5) [24]. *D2z* is the dot product between the two standardized histogram vectors h_s^z and h_t^z as shown in Equation (14).

$$D2z(h_s, h_t) = \sum_{w \in K} h_s^z(w) h_t^z(w). \quad (14)$$

However, there are some clear ways to improve on the idea [23, 25]. The easiest way is to make a 'self-standardized version of D_2', which will account for any differences in background noise. To describe Reinert's new statistic D_2^S and later D_2^*, we must also define a few other terms. Let $E(w)$ denote the expected probability of w, which is calculated by multiplying the probability of each of the k nucleotides that make up w together. For an additional definition described in the article, let $\tilde{h}$ be the updated histogram, which is calculated according to Equation (15). The final definition of D_2^S can be seen in Equation (16).

$$\tilde{h}_s(w) = h_s(w) - (len(S) - k + 1)E(w). \quad (15)$$

$$D_2^S(h_s, h_t) = \sum_{w \in K} \frac{\tilde{h}_s(w)\tilde{h}_t(w)}{\sqrt{\tilde{h}_s(w)^2 + \tilde{h}_t(w)^2}}. \quad (16)$$

Furthermore, a new word probability measure $\tilde{E}(w)$ is the expected probability of w in the two sequences concatenated

together. This results in the additional statistic D_2^*, which is described by Equations (17) and (18).

$$l = \sqrt{(len(S) - k + 1)(len(T) - k + 1)}. \quad (17)$$

$$D_2^*(h_s, h_t) = \sum_{w \in K} \frac{\tilde{h}_1(w)\tilde{h}_2(w)}{l\tilde{E}(w)}. \quad (18)$$

Reinert ultimately concluded that D_2^S and D_2^* are considerably better at calculating sequence similarity because the D_2 statistic is 'measuring the sum of the departure of each sequence from the background rather than the (dis)similarity between the two sequences' [23].

A few years later, even greater improvements were made in refining the D_2^S and D_2^* statistics by using a pattern transfer model [26]. In one of Wan's papers, it was shown that the 'power of D_2^* and D_2^S approaches a limit that is generally less than 1 when the sequence tends to infinity'. The most effective way to combat this down-side and the irregularities of the expected values for D_2^* and D_2^S is to partition the sequence into b equal subintervals [26]. If we then consider D_{2j}^* and D_{2j}^S as calculating D_2^* and D_2^S over the jth subinterval, this will lead to two new statistics, T^* and T^S as in Equations (19) and (20).

$$T^*(h_s, h_t) = \sum_{j=1}^{b} D_{2j}^*(h_s, h_t) \quad (19)$$

$$T^S(h_s, h_t) = \sum_{j=1}^{b} D_{2j}^S(h_s, h_t) \quad (20)$$

The N2 neighborhood statistic

Our next alignment-free k-mer statistic uses the novel approach of comparing weighted neighborhood counts instead of fixed k-mer frequencies [27]. Because transcription factor binding sites often times do not adhere to a preset combination of k-mers, the adaptable definition of a 'neighborhood region' allows for increased efficiency depending on the types of sequences compared. The set of all words in the neighborhood of w will be defined as $n(w)$. This definition of the neighborhood can vary depending on the particular types of sequences, e.g. tissue-specific enhancers, that are being compared. Equation (21) is the overall weighted word count $c(n(w))$ for that neighborhood.

$$c(n(w)) = \sum_{w \in n(w)} a_w c(w), \quad (21)$$

where a_w is the associated weight for each particular k-mer. If each k-mer contributes equally to the neighborhood, this weight value will be one. However, the weight can be tailored to the particular application if a specific k-mer is more important than others.

Now that we have a vector of all the neighborhood counts associated with every possible word w, the next step is to simply standardize the vectors based on the mean and SD and then divide each vector by its norm to obtain the values $S_{c(n(K))}$ and $T_{c(n(K))}$. The final statistic N2 is the inner product between each of the 'normalized standardized neighborhood count vectors' [27] as in Equation (22).

$$N2(h_s, h_t) = < S_{c(n(K))}, T_{c(n(K))} > . \quad (22)$$

When implementing N2 for a proposed problem, there are a variety of potential neighborhood definitions:

- $n_{rc}(w)$ is the neighborhood of the word and its reverse complement rc.
- $n_{mm}(w)$ is the neighborhood of the word and all words one mismatch away (specified with Hamming distance).
- $n_{mm,\ rc}(w)$ is the neighborhood consisting of both the reverse complement, rc, and one mismatch away, mm.

In addition, we considered $n_r(w)$, which is the word and its reverse because inversion is common in transposons of the same family.

Dinucleotide absolute frequency distance

When using k-mer frequency methods, increasing k should lead to higher accuracy and increased computational cost. In a large variety of other statistics, trying to find a k value that balances the accuracy and the efficiency has been an important problem. Although each of the previous statistics has been dependent on the k value, Zhang and Chen [28] created a novel statistic centered around 2-mers and the idea of di-nucleotide absolute frequency (AFd). Similar to constructing histograms, the first step is to record the frequencies of every 2-mer. Let h_s^p and h_t^p denote two probability histograms for sequences s and t as in Equation (23).

$$h_s^p = \left(\frac{c(AA)}{c(A)}, \frac{c(AC)}{c(A)}, \ldots, \frac{c(TT)}{c(T)}\right). \quad (23)$$

In the above representation, the first element in h_s^p is the count of the first dinucleotide AA divided by the count of its first nucleotide A. If $len(s) < len(t)$, then a sliding window b of base pairs with the length of the smallest sequence s will be considered first. If we let w be a 2-mer in this case, absolute frequency distance with a given window b can be seen in Equation (24).

$$AFd_b(h_s, h_t) = \sum_{w \in K} \left[(h_s^p(w) - h_t^p(w))f_m(h_s^p(w) - h_t^p(w))\right]^2, \quad (24)$$

where $f_m(x)$ is the stabilizing function as defined in Equation (25).

$$f_m(x) = \frac{1}{(1+x)^m}. \quad (25)$$

By adjusting the window b with a sliding percentage, the final distance measure is the minimum of AFd_b under each possible window. There are a variety of potential stabilizing functions; the m value in this case can be optimized to promote performance [28]. For our article and the overall focus on k-mer histograms over locations, we will consider s and t to have approximately the same length and will use $m = 14$ in the stabilizing function [28].

Inner product family

Another example of a common family for histogram/vector comparison is the inner product family. As its name implies, this family of statistics focuses solely on the dot product of two histograms $h_s \cdot h_t$. The dot product can be applied to either vectors of k-mer counts or probabilities [17]. As defined earlier,

consider h_s and h_t as a vector of the counts for each k-mer. This family includes cosine distance as in Equation (26).

$$Co\sin e(h_s, h_t) = 1 - \frac{h_s \bullet h_t}{||h_s||||h_t||} = 1 - \cos(\theta). \quad (26)$$

In this equation, θ could be considered as the 'angle' between the two histogram vectors in 4^k-dimensional space. The inner product of two normalized vectors as in Equation (27) represents the similarity version of this distance.

$$NormVectors(h_s, h_t) = \frac{h_s \bullet h_t}{||h_s||||h_t||} = \cos(\theta). \quad (27)$$

Because a large number of these statistics end up considering θ through the geometric definition of the dot product, they are also referred to as the 'Angle Family'. For example, consider Equation (28), which describes correlation distance.

$$Correlation(h_s, h_t) = 1 - \frac{(h_s - \mu_s) \bullet (h_t - \mu_t)}{||h_s - \mu_s||||h_t - \mu_t||} = 1 - \cos\left(\widehat{\theta}\right). \quad (28)$$

Here, μ_s and μ_h are the means of h_s and h_t. Also, by removing the '1-', this equation simply turns into Pearson's correlation coefficient. In this case, correlation distance is closely related to cosine distance as shown in Equation (26) if we consider the new angle $\widehat{\theta}$ as the angle between the adjusted histogram vectors $h_s - \mu_s$ and $h_t - \mu_t$. Other inner product statistics such as covariance similarity as in Equation (29) also use this idea of mean-adjusted histograms [22].

$$Covariance = \frac{(h_s - \mu_s) \bullet (h_t - \mu_t)}{4^k}. \quad (29)$$

Further, Spearman distance as referenced through MATLAB's library is just a variation on correlation distance (and a relative of Pearson's). Spearman distance computes 1− the cosine of the angle between the tied rank vectors minus the tied rank means. Note that this statistic takes a nonlinear time (O(n log n)). Additionally, this family includes harmonic mean as in Equation (30) and similarity ratio [22] as in Equation (31).

$$Harmonic(h_s, h_t) = 2 * \sum_{w \in K} \frac{h_s(w) h_t(w)}{h_s(w) + h_t(w)}. \quad (30)$$

$$SimRatio(h_s, h_t) = \frac{h_s \bullet h_t}{(h_s \bullet h_t) + ||h_s - h_t||}. \quad (31)$$

Gapped k-mer inner product

At this point, we have only focused on comparing k-mer histograms. As the k-mer size increases, the comparisons for sequence similarity get more accurate. At the same time, if one of the sequences is a mutated version of the other, long k-mers common to the two sequences should be infrequent. In one paper that discusses 'gapped k-mers', the problem of having long k-mers can be easily resolved [29]. On its own, the process of computing the gapped k-mer counts is not too complex [30]. If we consider w to be a gapped k-mer with total length k including gaps and g to be the number of gaps in the word, then for DNA sequences, the total number of words $|K| = \binom{k}{k-g} 4^{k-g}$. The next step is to consider the upgraded histograms $\tilde{h}_s$ and $\tilde{h}_t$ of each of the recorded gapped k-mer frequencies for sequences s and t. Then, the article defines a similarity function as shown in Equation (32), which is the normalized inner product between the two upgraded histograms.

$$Gapped(h_s, h_t) = \frac{\tilde{h_s} \bullet \tilde{h_t}}{||\tilde{h_s}||||\tilde{h_t}||}. \quad (32)$$

However, one potential issue is that the number of gapped k-mers will grow extremely quickly as k increases [29]. To increase efficiency, 'the key idea is that only the full [k]-mers present in the two sequences can contribute to the similarity score via all gapped k-mers derived from them' [29]. This idea leads to a revised definition of the inner product given in Equation (33).

$$\tilde{h_s} \bullet \tilde{h_t} = \sum_{m=0}^{k} z_m(h_s, h_t) w_m. \quad (33)$$

Here, m is the number of mismatches between two full k-mers; $z_m(h_s, h_t)$ is the 'mismatch profile', which represents the frequency of the pairs of full k-mers with m mismatches; and w_m is a coefficient determined by Equation (34).

$$w_m = \begin{cases} \binom{k-m}{k-g} & k - m \geq k - g \\ 0 & \text{otherwise.} \end{cases} \quad (34)$$

The article asserts that obtaining $z_m(h_s, h_t)$ can be computationally expensive. However, several methods are described in the article that can effectively reduce the run-time.

Squared chord family

Families such as Minkowski use a radical over the entire summation, whereas a key characterstic of the squared chord family as in Equation (35) is a square root over each histogram independently [17].

$$SquaredChord(h_s, h_t) = \sum_{w \in K} \left(\sqrt{h_s(w)} - \sqrt{h_t(w)}\right)^2. \quad (35)$$

One interesting observation about the squared chord statistic comes from simplification shown in Equation (36).

$$= \sum_{w} h_s(w) + h_t(w) - 2\sqrt{h_s(w) h_t(w)}. \quad (36)$$

There is a well-known mathematical theorem called the Arithmetic–Geometric Mean Inequality, which states that for $a, b \geq 0$; $\frac{a+b}{2} \geq \sqrt{ab}$ [31]. In other words, $h_s(w) + h_t(w) - 2\sqrt{h_s(w) h_t(w)} \geq 0$ always when $h_s(w), h_t(w) \geq 0$. Overall, the squared chord statistic appears to be capturing the variation between the arithmetic and geometric means of the two reported frequency vectors. If the two sequences have the same histogram, the geometric mean and the arithmetic mean will both be the same, resulting in a distance of 0. Equation (37) describes another statistic belonging to the same family [22].

$$Hellinger(h_s, h_t) = \sqrt{2 * \sum_{w \in K} \left(\sqrt{\frac{h_s(w)}{\mu_s}} - \sqrt{\frac{h_t(w)}{\mu_t}}\right)^2}. \quad (37)$$

Here, μ_s and μ_t are the means of h_s and h_t. Next, we discuss other families of alignment-free k-mer statistics that use Markov models.

Markov chain models

The premise for using a Markov chain for sequence similarity comes from the idea of a state machine and conditional probabilities [32, 33]. As we scan along a sequence with a size k window and record frequencies, it is possible to calculate the probability that the kth letter occurs based on the current state of the $k-1$ letters. The log of each probability value for the current state is then summed over the entire sequence until the state reaches the end. There is, however, a mathematically equivalent way to calculate this statistic without looking at the particular sequences themselves and only using the k-mer counts. The first step is to construct the conditional probability table based on each group of words. For example, Equation (38) describes the conditional probability when $k=3$.

$$m_x(AAT) = p_x(T \mid AA) = \frac{c(AAT)}{\sum_{n\in[A,C,G,T]} c(AAn)}. \tag{38}$$

Here, c(AAA) is the frequency of AAA in the sequence x. The next step is to calculate the probability of the second sequence using the conditional probabilities calculated according to the first sequence as shown in Equation (39).

$$d_{h_s}(h_t) = \sum_{w\in K} h_t(w)\ln(m_s(w)). \tag{39}$$

After that, the probability of the first sequence is computed according to the conditional probabilities of the second sequence. The final statistic is the average of $d_{h_s}(h_t)$ and $d_{h_t}(h_s)$ as shown in Equation (40).

$$Markov(h_s, h_t) = \frac{d_{h_s}(h_t) + d_{h_t}(h_s)}{2}. \tag{40}$$

With the success of Markov models in bioinformatics, many variations were created to expand on the idea. Pham and Zuegg [34] invented a new statistic, called SimMM, based on Markov models. Dai, Yang and Wang [35] described SimMM as a 'probabilistic measure based on the concept of comparing the similarity/dissimilarity between two constructed Markov models'. Pham and Zuegg started by defining a helper function as in Equation (41).

$$r(h_s, h_t) = \frac{1}{len(t)}\ln\left(\frac{d_{h_s}(h_t)}{d_{h_t}(h_t)}\right). \tag{41}$$

As the helper function is not symmetric, its average is used in computing the final form of SimMM as shown in Equation (42).

$$SimMM(h_s, h_t) = 1 - e^{\frac{r(h_s,h_t)+r(h_t,h_s)}{2}}. \tag{42}$$

In sum, SimMM involves comparing four conditional probabilities. The final form of the statistic is scaled using an exponential and is subtracted from 1 as shown in Equation (42).

Another Markov-based statistic is the revised relative entropy, which was proposed in 2008 and sought to efficiently integrate Markov models and k-mer frequencies [35]. Let p_s and p_t be the conditional probability models created from sequences s and t. Equations (43–45) define revised relative entropy for a given k-value and Markov model order r.

$$d_1 = \sum_{w\in K} m_s(w)\ln\frac{2 * m_s(w)}{m_s(w) + m_t(w)}. \tag{43}$$

$$d_2 = \sum_{w\in K} m_t(w)\ln\frac{2 * m_t(w)}{m_s(w) + m_t(w)}. \tag{44}$$

$$rre\_k\_r(h_s, h_t) = \frac{d_1 + d_2}{2}. \tag{45}$$

This statistic is largely based on Jensen–Shannon divergence, which is covered in the next section.

Divergence

Similar to Markov chains, a wide variety of divergence statistics use probabilities and effectively compare two sequences by assessing how far apart they are in the log-probability space. For example, consider Conditional Kullback–Liebler Divergence as shown in Equation (46), also known as conditional relative entropy [36].

$$CKL(h_s, h_t) = \sum_{w\in N} p_s(w)\sum_{b\in B} m_s(wb)\ln\left(\frac{m_s(wb)}{m_t(wb)}\right). \tag{46}$$

Here, N is a set of all $(k-1)$-mers; B is a set of the four nucleotides A, C, G and, T; and wb is the word consisting of the $(k-1)$-mer, w, followed by the base b.

Although they do not involve conditional tables, a few other divergence statistics that are commonly used are K as shown in Equation (47), Jensen Shannon as show in Equation (48) and Jeffrey divergence as shown in Equation (49). In the equations describing these divergence statistics, $p_s(w)$ is the probability (not the conditional probability) of w under the histogram of sequence s, and $v(w)$ is the average probability for w over both histograms.

$$K(h_s, h_t) = \sum_{w\in K} p_s(w)\ln\frac{p_s(w)}{v(w)}. \tag{47}$$

$$JenShan(h_s, h_t) = \sum_{w\in K} \ln\frac{p_s(w)^{p_s(w)}p_t(w)^{p_t(w)}}{v(w)^{p_s(w)+p_t(w)}}. \tag{48}$$

$$Jeff(h_s, h_t) = \sum_{w\in K}(p_s(w) - p_t(w)\ln\frac{p_s(w)}{p_t(w)}. \tag{49}$$

Distance measure based on k-tuples

Although it was originally created for a specific clustering algorithm, distance measure based on k-tuples (DMk) is a novel alignment-free k-mer statistic because it makes use of k-mer counts as well as the locations within the sequence [37]. The first step is to define a term related to the density ρ of the ith occurrence of a particular word w as shown in Equation (50).

$$\rho_i(w) = \frac{1}{l_i - l_{i-1}}, 1 \le i \le c(w). \tag{50}$$

Here, l_i is the ith location of word w, and $c(w)$ is the count of w. This ρ statistic captures information about the location where each k-mer occurs as well as information on the previous occurrence. Next, Equation (51) defines $\tilde{\rho}_i$ as a partial sum of the

ρ_i starting from the first occurrence up to the ith occurrence of a particular word.

$$\tilde{\rho}_i(w) = \sum_{n=1}^{i} \rho_n(w), 1 \leq i \leq c(w). \quad (51)$$

One major benefit of this statistic is that given the vector $\tilde{\rho}(w) = \left(\tilde{\rho}_1(w), \tilde{\rho}_2(w), \ldots, \tilde{\rho}_{c(w)}(w)\right)$, one can determine where and how many times w appears in the sequence. Now that we have a vector of densities for each k-mer, the next step is to simply construct a probability distribution vector p_i by dividing each $\tilde{\rho}_i$ by the sum of $\tilde{\rho}$. After that, these values can be further manipulated by applying Shannon's entropy as shown in Equation (52).

$$Shan(w) = -\sum_{i=1}^{c(w)} p_i \log_2 p_i. \quad (52)$$

When this operation is repeated for every k-mer, we have two entropy vectors E_s and E_t for sequences s and t. The final statistic as shown in Equation (53) is then computed using the Euclidean distance between both density histograms:

$$DMk(h_s, h_t) = Euclidean(E_s, E_t). \quad (53)$$

Overall, DMk has been shown to be more effective than count-based statistics because of the integration of both k-mers locations and ordering [37].

Earth Mover's Distance

Earth Mover's distance (EMD) was originally demonstrated to have applications to image databases and to the transportation problem. It focuses on analyzing the 'minimum amount of work that must be performed to transform one distribution into another'[38]. The same principle could also have applications to distributions of k-mers. If we consider h_s as the supply distribution and h_t as the demand, then EMD is effectively measuring the minimum number of k-mer counts that need to be transported from h_s to h_t. In some way, this statistic is similar to Manhattan distance except for the fact that the k-mers or bins of the histogram are no longer being compared one-to-one for both sequences [39, 14]. Thus, each k-mer is not being treated as independent, which should perform well in the context of DNA sequences with strings of interconnected and repetitive regions. Although the statistic normally has a more complicated derivation when considering multiple dimensions, it mathematically simplifies to Equation (54) when dealing with k-mer histograms.

$$EMD(h_s, h_t) = \sum_{w \in K} |a_s(w) - a_t(w)|. \quad (54)$$

In this equation, $a_s(w)$ is the aggregate sum vector of h_s calculated by $a_s(w_i) = h_s(w_1) + h_s(w_2) + \cdots + h_s(w_i)$, where w_1 is the first k-mer. Overall, this statistic largely depends on the location of the k-mer bins in the histogram. For our evaluation, we ordered each k-mer alphabetically. In applications involving NGS data, where all reads have the same length, the order of the histogram can be based on the order of k-mers in one of the sequences.

Length difference

Length difference (LD) is the difference in length between two sequences as in Equation (55).

$$LD(s,t) = |\text{length of s} - \text{length of t}| \quad (55)$$

Although it is a simple statistic, it can be used for reducing the number of sequence comparisons in the case of global alignment. For example, if the minimum desired identity score is 70% and the ratio between the shorter and the longer sequence lengths is <70%, then there is no way that the alignment could happen at that threshold. Therefore, the LD metric is an important measure of sequence similarity.

Materials and methods

Statistics evaluated

The following is a list of the 33 statistics evaluated in this article: Hellinger, Manhattan, Euclidean, χ^2, normalized vectors, harmonic mean, Jeffrey divergence, K-Divergence, Pearson correlation coefficient, squared chord, Kullback–Liebler conditional divergence, Markov similarity, intersection, rre_k_r, D2z, SimMM, EuclideanZ, EMD, Spearman, Jaccard, LD, D_2^S, AFd, mismatch, Canberra, Kulczynski Similarity 1, Kulczynski Similarity 2, similarity ratio, Jensen–Shannon Divergence, D_2^*, N2r, N2rc and N2rrc.

Primarily, we chose these statistics based on having a variety of families as well a good number of the latest alignment-free k-mer statistics. Additionally, we have adopted the criteria that each statistic must require only k-mer frequencies as input. Any statistic that requires locations, specialized k-mers, or information beyond the scope of word histograms will not be considered. Gapped k-mers, T_2^S, T_2^* and DMk will not be evaluated and are included in this article for reference purposes. Because the number of paired combinations can quickly increase, other statistics are mentioned in their respective families solely for review purposes.

Calculating a k-mer histogram

For any particular k value, a k-mer is a k-length sequence of DNA. Because the 'alphabet' for nucleic acids is only four letters (A, C, G and T), each sequence has 4^k potential 'words'. A histogram or word-count vector can be created for each sequence by scanning linearly through each k-window of letters and counting occurrences of each word. Indexing a sequence of k-mers can be implemented efficiently using Horner's rule [40].

Selection of k

The selection of k determines the success of the alignment-free k-mer statistics. The k must lie in a certain range to ensure that the comparison of histograms is a linear process. We used Equation (56) to find k.

$$k = \lceil log_4\left(\frac{1}{n}\sum_{i \in s} len(i)\right) \rceil - 1. \quad (56)$$

Here, n is the number of sequences in the set s. Using too short of a k may not provide enough information, but using too long of a k increases the comparison time and memory (4-fold per increment of 1). Therefore, a too long k might not guarantee linear time for comparing two histograms. For example, consider

two sequences of length 100. Our formula gives $k = 3$. But if k gets larger, such as $k = 7$, the number of comparisons is quadratic ($4^7 > 100^2$), negating the advantages of alignment-free k-mer statistics.

A note on pseudo-counts

When computing each of the statistics, many require pseudo-counts within the histograms to prevent a division by 0. This can be accomplished by adding 1 to each of the entries. In addition, these pseudo-counts are needed to allow events that 'seem' impossible to be able to happen [41]. In general, most statistics that operate on probability distributions are implemented with pseudo-counts. However, there are multiple statistics that require either a combination of both or will function the same, regardless, i.e. the Minkowski family. Overall, if the statistic requires dividing by k-mer frequencies, then pseudo-counts should be used.

Scaling statistics

Although most statistics are efficient at measuring the degree of separation between k-mers, they do not all scale naturally between 0 and 1. To use these statistics in conjunction with others, some method of scaling or standardization should be used. Given a group of raw data points calculated using a particular statistic (such as raw Euclidean distance), the scaled version is computed by subtracting the minimum from each item then dividing by the difference between the maximum and the minimum items. Further, all statistics are represented as similarity measures. Each of the distance data points will be converted to similarities by scaling the results between 0 and 1 and subtracting the scaled version from 1. Similarity statistics are still scaled between 0 and 1 but do not require the conversion.

Data sets for experimentation

For comparing the large variety of alignment-free k-mer statistics, we have decided to obtain our data from three sources. The first source is a study of microbiome in the human [42]. Bacterial samples found in healthy adults were collected from a variety of habitats, such as the skin, gut and oral cavity. Analysis of the various species and their 'trends may ultimately reveal how microbiome changes cause or prevent disease' [42]. We will refer to this set as the microbiome data set. This data set includes the sequences of the 16 S ribosomal RNA gene (200–400 bp long). They are produced by pyrosequencing technology. After obtaining a sample of the paper's data from DNAnexus, we globally aligned each of the sequences. The overall distribution includes 125 250 pairwise comparisons with around 50% of the data having 90% alignment identity score or better. Furthermore, the section of identity scores between 60 and 70% accounted for about 37% of the data. Because many statistics tend to degrade in accuracy once the identity score drops <50%, we believe that this distribution is an effective data set for evaluation. To evaluate the correlation between a statistic and the identity score, we randomly sampled collections of the data at different identity score ranges.

The second source is based on the $p27^{Kip1}$ tumor suppressor gene, which controls 'Ras, one of the most common oncogenic events in human cancer' [43]. Using a database of 139 homologs of this gene and their pairwise comparisons, we can effectively analyze the statistics in an important gene similarity application.

The third source is a collection of third-generation sequencing reads taken from a recent genome assembly project using the PacBio sequencing technology [44]. These reads are long (mostly 10k–25k bp). We selected 100 sequences, resulting in 5050 pair-wise alignments. Most alignments identity scores were in the 40–50% range.

Synthetic sequences of the same length comprise the fourth data set. We generated this data set to (i) eliminate the effect of LD and (ii) mimic an NGS application where the reads are the same length. First, we generated a collection of random DNA templates. Second, we fixed the mutation rate and generated additional sequences by randomly modifying 3% of the base pairs with single point mutations. With this method, all sequences have 200 bp, and the final data set has 50 086 pairwise sequence comparisons.

To model local alignment, we generated the fifth data set. It consists of 209 synthetic sequences generated from the p27 set, combining various sequences using two methods. Using the first method, one sequence or a part of it is inserted into another sequence at a random point. In other words, the first sequence or part of it becomes a substring of another sequence. For example, suppose we have two sequences A and B. We randomly select a subregion of A, possible the entire sequence A. Le us call this part 'A_1'. After that, B is split at a random point into B_1 and B_2. The resulting sequence is $B_1A_1B_2$. Using the second methods, we generate two overlapping sequences by using two original sequences. One original sequence is split into two halves at a random point, and the other is the shared part of the overlapping sequences to which the halves are prefixed or suffixed. For example, suppose we have two sequences: A and B. We split A into A_1 and A_2 at a random point. The two generated sequences are A_1B and BA_2.

A note on statistic pairs

In our evaluation of alignment-free k-mer statistics, we decided to extend our analysis of the single statistics to include multiplicative statistic combinations. If two statistics are multiplied together from different families, their drawbacks and strengths might balance each other out and produce a statistic that better correlates with identity score. One important note is that this combination is not between raw scores. Instead, the process for creating the pairs involves (i) scaling the statistics between 0 and 1, (ii) converting all distances to similarities and (iii) multiplying the statistics together to create a paired combination. Additionally, squared versions of each of the single statistics are included in the pair creation.

Results and discussion

In this section, we will be evaluating each of the alignment-free k-mer statistics based on four criteria: (i) sensitivity and specificity, (ii) linear correlation with identity score at different cutoff values, (iii) k-nearest neighbors and (iv) time efficiency. We will present our current findings and give recommendations as to which statistics perform best under certain applications.

A benchmark for evaluating alignment-free k-mer statistics

Before discussing our results, note that the each of these experiments can be duplicated using our evaluation benchmark. This benchmark allows the user to evaluate the 33 statistics on any group of sequences. Ultimately, this benchmark can be

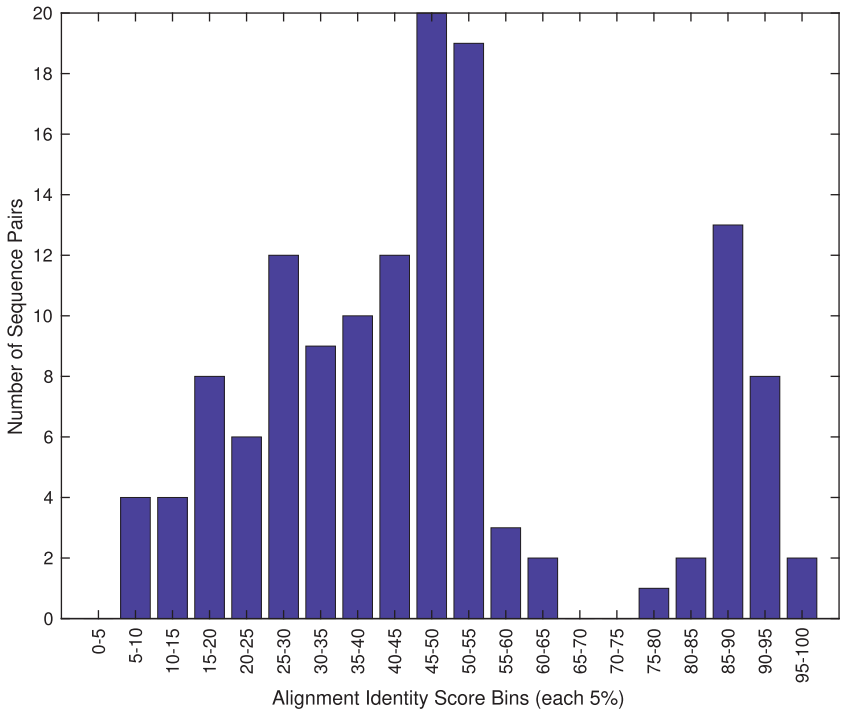

Figure 2. Identity score distribution for the 135 p27 sequences in the sensitivity and specificity experiment. The data set consists of 26 similar sequences (≥70%) and 109 dissimilar sequences (<70%). There are no sequence pairs within 5% of the cutoff line (70%), meaning that classification will be slightly easier for each statistic because the separation between similar and dissimilar sequences is clear.

effectively used for picking ideal statistics for a particular application or compare the effectiveness of new statistics as they are developed. The code of the benchmark is provided as Supplementary File S1.

Sensitivity and specificity

Present in a wide variety of alignment-free papers [27, 34, 35, 45], the purpose of this experiment is to evaluate each statistic's ability to filter through a database of sequences according to the similarity to a query sequence. The task is to correctly classify the similar sequences from the dissimilar ones. A selected p27 query sequence is compared against 109 dissimilar sequences with identity score <70% and 26 similar sequences with identity ≥70%. Identical sequences to the query have been removed because most statistics are able to easily identify identical sequences. The distribution of identity scores for the p27 data can be seen in Figure 2. We used the sensitivity as in Equation (57) and the specificity as in Equation (58) for evaluating the statistics.

$$\text{Sensitivity} = \frac{\text{number of similar sequences correctly identified}}{26}. \tag{57}$$

$$\text{Specificity} = \frac{\text{number of dissimilar sequences correctly identified}}{109}. \tag{58}$$

Figure 3 shows the results of the sensitivity test conducted on each of the 33 single statistics. The graph of the specificity test is available in the Supplementary Files (see Supplementary File S2). Overall from Figure 3, a few statistics are effective at finding the 26 similar p27 genes from a variety of species. The left-most third of the bar graph represents statistics that identified all 26 similar sequences correctly, whereas the middle section got 25 of 26 or 24 of 26 correct. Apart from a few statistics such as Markov, most of the statistics can sufficiently rule out sequences that have identity scores <70%. A likely explanation for this result comes from the graph in Figure 2. Because there is a notable gap in identity scores between 60 and 80%, each of the statistics did not have the difficult task of classifying many points within 5% of the boundary. These results show that a large number of statistics can be effectively used on their own for filtering a database.

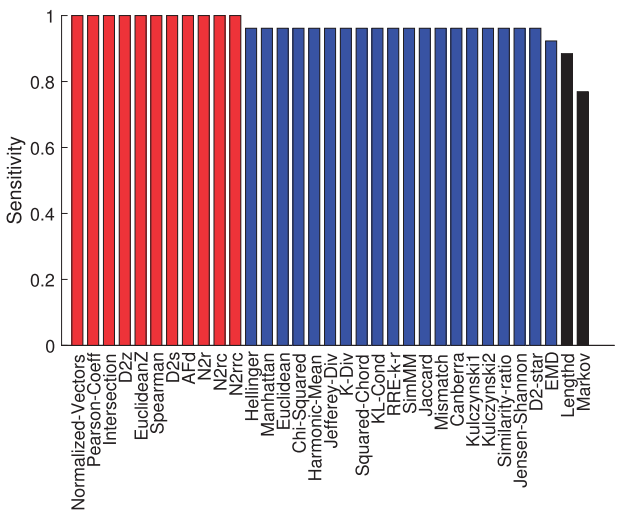

Figure 3. Sensitivity values on the p27 data, each statistic in the left-most third identified all positive sequences correctly (26 of 26). Few statistics such as LD and Markov are not ideal for filtering out sequences that have identity scores < 70%.

Linear correlation with identity scores

For the next experiment, we introduce a new evaluation criterion that calculates a linear correlation with the Needleman–Wunsch alignment algorithm [1] at different identity score ranges. A total of 4735 sequence pairs, referred to as points, were randomly selected from the microbiome data and binned by every 10% identity score, so that each segment of the identity scores is equally represented. Points having identity scores below the threshold are excluded. This evaluation ultimately highlights the most effective alignment-free *k*-mer statistics at different segments of the identity score spectrum. We used cutoff values at 0, 60, 70, 80 and 90%. To consider the multiplicative combinations, squaring each statistic brings the total number to 66. After considering all unique pair combinations, 2211 total statistics were tested. Supplementary Files S3–S7 include the results of these experiments.

The first threshold considered is the entire data set of 4735 comparisons. Figure 4A–E shows the five best statistics ranked by the r^2 ordinary correlation coefficient. These top performers are (i) similarity ratio × EMD, (ii) norm vectors × LD, (iii) *K*-divergence × LD, (iv) SimMM × LD and (v) Jeffrey divergence × LD. All of the top five performers achieved correlation coefficient values of 0.98. EMD appears in combination with an additional statistic at the top spot; however, the recurring pattern of length difference is apparent. Because each of the sequences varied in size between about 200–400 bp, it is clear that trying to globally align two sequences of vastly different lengths will not yield a good alignment. However, LD and EMD on their own only achieve a correlation coefficient of 0.57 and 0.72, respectively. Therefore, the real strength of LD and EMD lies in their ability to combine with other statistics.

For the 60% alignment threshold, our results can be found in Figure 4F–J. In this case, the top five best statistics are (i) norm vectors × EMD, (ii) Pearson coefficient × EMD^2, (iii) D2z × EMD^2, (iv) Pearson coefficient × EMD and (v) D2z × EMD with correlation values all at 0.98. Although EMD appears frequently in the 60% case and LD not all (appeared once among the top 10), these results coincide well with the conclusion that combinations with both EMD and LD can improve alignment-free *k*-mer statistics. Because a collection of around 947 sequence pairs has been removed from the 60% and below section, those points most likely played a large role in helping the correlation coefficient for LD pairs over EMD in the previous experiment.

The top five statistics from the 70% threshold once again see a return of LD (Figure 4K–O). The best correlation coefficients were achieved by (i) $D2z^2$ × LD (ii) Pearson $coefficient^2$ × LD, (iii) $N2r^2$ × LD, (iv) norm $vectors^2$ × LD and (v) Kulczynski2 × LD with values all at 0.98.

For the 80% threshold in Figure 4P–T, the top five best statistics are (i) Manhattan × LD, (ii) Kulczynski2 × LD, (iii) *K*-$divergence^2$ × LD, (iv) intersection × LD and (v) Jefferey $divergence^2$ × LD with r^2 values all at 0.99. In this test, we see the first appearance of Manhattan distance along with LD becoming a dominant statistic when paired together. Once again, however, the correlation coefficient for LD as a single statistic is abysmal. In this case, the r^2 value for LD sits at 0.11, the lowest of all 1711 statistics and combinations tested. Once the identity score gets passed a certain threshold, LD on its own continues to be less useful because most of the sequences in this range are close in length. However, the dual nature of combining this statistic with a more consistent one such as Manhattan or *K*-divergence has been shown to be highly correlated with identity scores.

Finally, for the 90% threshold test (Figure 4U–Y), the best statistics with their r^2 correlation values are (i) N2rrc × LD, (ii) LD $N2rrc^2$ × LD, (iii) D2z × LD, (iv) Pearson coefficient × LD and (v) N2r × LD. The r^2 of the best performing paired statistics was 0.97–0.98, whereas the LD correlation value on its own is a respectable but still low 0.63. Even when all of the points <90% identity score were removed, LD still manages to influence more robust statistics such as Pearson coefficient and increases its correlation from 0.80 to 0.98 when both are combined.

In sum, *using different cutoffs, the best correlations with identity scores are because of paired statistics*. No single statistic was placed among the top best five performing statistics in any of these experiments. A variety of different statistics have been shown to be the most effective at different threshold values. The top performers are application-specific. We observed that they vary based on the input sequences. Nonetheless, paired statistics are consistently the top performers when the input sequences are of variable length. Of all the top statistical combinations from each threshold trial, all 25 involved some combination of either EMD or LD.

K-nearest neighbors application

The purpose of this experiment is to evaluate each statistic based on its ability to identify the *K*-nearest neighbors. Finding the nearest neighbors has many applications in computational biology. Different applications require finding a different number of nearest neighbors. Using the p27 data set, we chose one sequence as the query and then calculated each statistic with the 138 other sequences. As a majority of the statistics included in this article are highly effective at identifying when two sequences are identical, we decided to remove each of the sequences that had an identity score of 100% when compared with the query. Then, we evaluated each statistic on finding the *K*-nearest neighbors for $K = 1$, 5 and 10.

Finding the nearest neighbor

After evaluating the 29 single statistics (Supplementary File S8), we found that LD was the only statistic able to correctly identify the closest neighbor. This result is expected because the database of p27 sequences ranged in length from around 600–7500 bp. However, if the database contains sequences of the same length, e.g. NGS data, the LD will be ineffective on its own. Earlier in this section, the results from the linear correlation test demonstrated that multiple combinations with LD were among the best performers. Therefore, we decided to extend our evaluation to consider multiplicative combinations as well as squares of the statistics. Of 2211 single, squared, or paired statistics, only 35 were able to successfully identify the closest neighbor (Supplementary File S9). Interestingly, all of them involved a multiplication with either LD or EMD and their squares. This idea of combining single statistics with a length modifier has shown to be effective at enhancing the performance of other statistics. For example, multiplying Manhattan and EMD or Manhattan and the square of LD resulted in identifying the nearest neighbor, given that Manhattan cannot identify the closest neighbor on its own. These results can be explained because of EMD's inclusion of a cumulative sum, which naturally captures the length difference; specifically, the last sum difference in EMD is the sequence length difference.

Finding the nearest five neighbors

For the $K = 5$ trial, our results for the singles once again showed that LD was the only one that was able to identify the closest

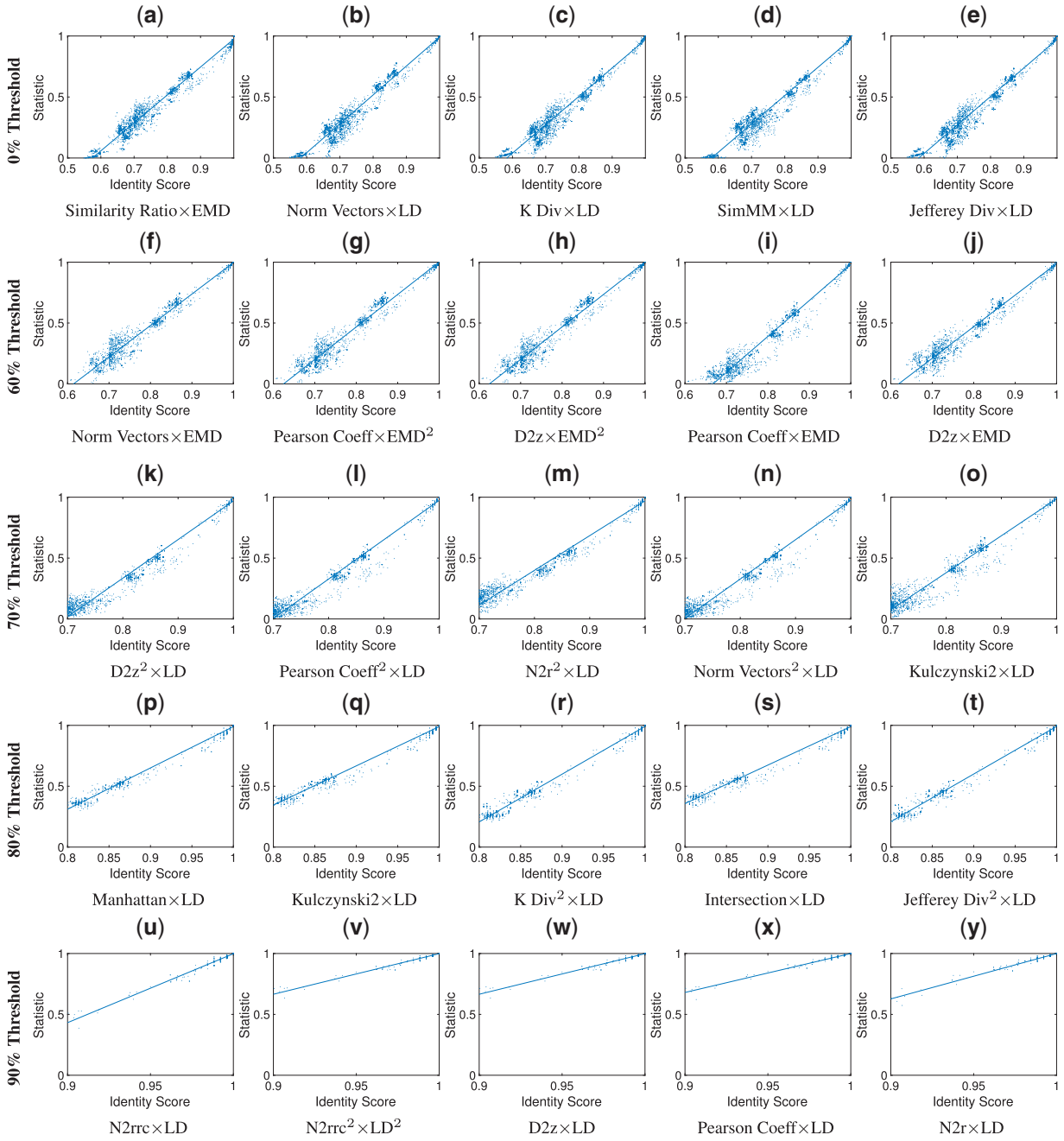

Figure 4. Top five statistics by linear correlation: 0, 60, 70, 80 and 90% threshold. All microbiome data points, i.e. sequence pairs, with identity score greater than the threshold value were used in these experiments. We then computed the ordinary linear correlation coefficient for each statistic. The best statistic pairs were chosen based on their r^2 values. Note that either EMD or LD appears in all top performing statistics.

five neighbors. Unlike the $K=1$ trial, 28 statistics performed well, identifying 4 of the closest 5 neighbors. EMD performed moderately (3 of 5); whereas Markov and AFd were not able to identify any. Although EMD captures the length difference as part of its calculation, it is surprising that EMD was not able to perform more effectively on its own. See Supplementary File S10 for the complete results. To continue our search for accurate methods for identifying the closest five neighbors, we considered the pair combinations (Supplementary File S11). Similar to the $K=1$ trial, all of the top 18 statistics that were able to identify 5 of 5 neighbors included some combination with either EMD or LD. The other statistics that appeared in these pairs were Manhattan, Euclidean, χ^2, harmonic mean, squared chord, Markov, Kulczynski1 and similarity ratio. These results show that a large number of single statistics can identify the majority of the nearest neighbors. Furthermore, combining these statistics with either LD or EMD led to perfect results.

Finding the nearest 10 neighbors

We repeated the previous experiment searching for the closet 10 sequences to the query sequence in the p27 data set. One more time, our evaluation of the single statistics indicates that LD was the best and scored a 10 of 10. It was followed closely behind by both AFd and EMD, which found 8 of 10 correct

neighbors. As EMD identified 3 of 5 and AFd none in the $K = 5$ trial, these high results are interesting. The most reasonable explanation is that both of them are able to identify the closest 10 neighbors but in the incorrect order. Possibly, it ranked the closest five sequences somewhere between 6 and 10, and the ones between 6 and 10 somewhere among the top five. For the additional results, a majority of the statistics (26 of 33) scored 7, whereas Markov once again performed poorly, scoring 2. See Supplementary File S12 for the results of all single statistics. After evaluating the paired combinations, we found that LD, Markov $\times$ LD^2 and Markov $\times$ EMD^2 were able to identify all 10 correct neighbors. One specific observation is that neither EMD nor Markov in the singles test was able to identify all 10 correctly; however, multiplying them together resulted in a perfect identification. Another noteworthy combination was Euclidean $\times$ LD^2 which scored 9 of 10. Supplementary File S13 includes the results of all single/paired statistics.

In sum, if we examine the results over the entire *k*-nearest neighbors experiments on the p27 data set, a large number of the single statistics are effective at identifying a majority of the closest neighbors. However, if the biological application requires perfect identification, then modifying preexisting statistics by LD or EMD can increase the performance overall when the data set includes sequences that differ in length.

K-nearest neighbors with similar length sequences

To further examine the effect of combining both EMD and LD with other statistics, we decided to repeat the experiment on sequences of the microbiome data set because sequences were relatively close in length. This set had 411 sequences, a majority of which had lengths between 220–260 bp. First, we selected a query sequence that was close to the average length. Then, we computed each statistic on the query sequence and each of the 411 sequences in the database. Next, any sequence pairs that had an identity score of 100% were removed to make the test more challenging because almost all of the statistics are effective at identifying identical sequences. After that, in the upper identity range ($\geq$ 87%), only one pair of a particular identity score was kept to separate neighbors and prevent sequence redundancies.

Finding the nearest neighbor

After evaluating each of the 33 single statistics on finding the nearest neighbor, we found that the only ones able to correctly identify the closest were Manhattan, Euclidean, Jaccard and Mismatch. From these initial results, there is a notable contrast between the last experiment, as LD is no longer able to perform well on its own. The small length difference is a likely cause for these results. Supplementary File S14 includes the evaluations of the single statistics. Next, we continued to test the ability of multiplicative combinations. A total number of 233 single/squared/paired statistics were able to correctly identify the closest neighbor (Supplementary File S15). This number is considerably $>$35 statistics able to correctly find the closest sequence in the p27 data set. A large number of the effective combinations included pairs of the five best performing statistics from the singles test as well as EMD and LD among others. In many examples, multiplying two statistics that missed the nearest neighbor resulted in the desired identification. For instance, neither LD nor Jensen–Shannon could identify the closest sequence; however, multiplying these two statistics led to locating the nearest neighbor.

Finding the nearest five neighbors

For the $K = 5$ trial, the top statistics from the singles test include harmonic mean, Jeffrey divergence, *K*-divergence, KL Cond, SimMM, AFd, Kulczynski1, Jensen–Shannon and D2*, all of which identified 3 of 5 correct neighbors. The vast majority, such as Manhattan, Euclidean and mismatch all found 2 of 5. Results at the bottom include Markov and LD, which scored a 0 (Supplementary File S16). Interestingly, after computing each paired statistic, the best two results were achieved by D2*^{2} $\times$ EMD and D2*^{2} $\times$ LD, which scored a 4 of 5 (Supplementary File S17). Although members of the divergence family, such as Jeffery divergence and K divergence, were among the best statistics in the singles test, combinations of these statistics with LD or EMD were surprisingly not among the top performers (3 of 5). A likely explanation is that multiplying statistics by a factor of LD may improve the results for some families while leaving others unaffected, or even decreasing the accuracy. For example, combining AFd with LD lowered the score from 3 of 5 to 0 of 5. However, similar to the p27 data set, combining certain low-performing statistics, such as Markov with LD, both of which scored a 0, can markedly improve accuracy (0/5–3/5). Although we minimized the effect of the length difference, the most striking result is that combinations of either LD or EMD still appeared in the top performing paired statistics.

Finding the nearest 10 neighbors

For our last trial and conclusion of the nearest neighbor experiment, we found that the best statistics for the singles test were EMD and N2r with 7/10 neighbors correctly identified. This result was followed closely by Manhattan, Euclidean, norm vectors, intersection, Jaccard, D2s, mismatch, Kulczynski2, similarity ratio, D2* and N2rrc, which scored a 6 of 10. The rest of the statistics identified only five neighbors correctly, whereas LD and Markov scored a 4 and 3 of 10 (see Supplementary file S18). Once again, if we examine the p27 and this experiment, EMD only shows its true potential as a single statistic in the $K = 10$ trial, suggesting that it can find similar sequences; however, it cannot properly rank them. LD, on the other hand, has failed to achieve repeated perfect scores because of comparable sequence lengths. For the paired test, we found that 126 paired statistics were able to correctly identify 9 of 10 nearest neighbors. Even though we intentionally limited the effect of length difference in this experiment, all 126 best performing combinations involved some multiplication with either LD, LD^2, EMD or EMD^2. See Supplementary File S19 for the complete results. Multiple trials of the nearest neighbor experiment have shown that low-performing statistics, e.g. Canberra (5 of 10), when multiplied by LD (4 of 10) can impressively increase accuracy (9 of 10).

In sum, the experiment on the microbiome data set led to the following two conclusions: (i) combinations including EMD or LD are often among the best performing paired statistics; and (ii) usually, combining poor or moderately performing statistics with EMD or LD results in improving the performance.

K-nearest neighbors with sequences of the same length

Finding closest sequences is an important step in reducing the redundancy and correcting errors while dealing with the NGS data. Therefore, we repeated the *K*-nearest-neighbors experiment on synthetic data simulating NGS reads. Recall that sequences in this data set are of the same length and have low mutation rate (3%). Supplementary Files S20–S25 include the results of finding the nearest 1, 5 and 10 sequences to a query

sequence. We found that a large number of single statistics are effective in identifying the closest sequence(s), producing perfect results. Further, LD and EMD are no longer effective as expected. A large number of multiplicative combinations are able to achieve perfect results. However, single statistics, such as Manhattan or Mismatch, would be more preferable to multiplicative combinations because of the huge number of NGS reads.

K-nearest neighbors using third-generation sequencing

The newest forms of sequencing, third-generation sequencing, generate reads of much longer length (average >10 000 nucleotides) than the second/NGS sequencing. It is important to evaluate the ability of the statistics to find the nearest neighbor in such data. To this end, 1 query sequence was globally aligned versus 100 of these sequences. Supplementary files S26–S31 show the results of this experiment. For the $k=1$ trial, no single statistic correctly found the nearest neighbor, but 43 paired statistics found the nearest neighbor, demonstrating the effectiveness of paired statistics. There was a good representation of EMD and LD among these pairs. For the $k=5$ trial, two single statistics (EMD and LD) found 5 of 5. All 35 paired statistics that found all five nearest neighbors involved either EMD or LD. However, for the $k=10$ trial, neither have any single statistic nor any paired statistic found 10 of 10 nearest neighbors. The best single statistic was EMD, which found 8 of 10. All paired statistics that found eight nearest neighbors were some single statistic paired with EMD. These results are expected and are similar to the ones obtained on the p27 data set because of the large difference in the length of the sequences in this set. Overall, because of the noticeable differences in length, LD and EMD as single statistics, along with combinations with other statistics, are the most useful for finding nearest neighbors among third-generation reads.

Processing a large number of sequences requires large amount of memory and long time. Using histograms of shorter k-mers reduces the memory and time requirements. Therefore, in the next experiment, we study the effects of the size of the histogram on the ability of a statistic to identify the closest sequences.

K-nearest neighbors using different sized words

Owing to the potential confusion caused by K-nearest neighbors and k-mers, we will be referring to the previously established k-mers as n-mers for this section. Here, we investigated the effects of using shorter and longer n-mers on the nearest neighbor experiments. In the previous experiment on the microbiome data set, we used histograms of 4-mers. In this case, we repeated the three trials of finding the nearest neighbors ($K=1$, $K=5$ and $K=10$) using 2-mers and 3-mers along with 5-mers and 6-mers. Interestingly, the performance of the statistics based on different sized n-mers attained the same accuracy (Supplementary Files S14-S19 and S32-S55). However, the number of single or paired statistics that identified the nearest neighbor(s) declines as the n-mer gets shorter. Table 1 shows the number of single and paired statistics that obtained perfect results with different n-mers. Reducing the size of the n-mer by 1 reduces the memory usage and increases the speed by a factor of 4. As there are 4^n n-mers in a histogram, decreasing n vastly reduces memory usage while also inversely affecting speed. These results show that using shorter n-mers improves both time and space while maintaining the same or very comparable accuracy.

Table 1. Number of single and paired statistics that correctly obtained K-nearest neighbors from the K-nearest neighbor experiments using different n-mers

	$K=1$		$K=5$		$K=10$	
	Singles	Pairs	Singles	Pairs	Singles	Pairs
n=6	24	1395	23	1304	1	108
n=5	11	399	11	355	1	24
n=4	7	129	7	129	1	31
n=3	8	180	8	169	1	23
n=2	8	167	8	162	1	3

Note: Specifically, we repeated the experiments using 2-mers, 3-mers, 4-mers, 5-mers and 6-mers on the same microbiome data set. In these experiments, we used each single (of 33) and paired (of 2211) statistic for finding the nearest K neighbors. The numbers indicate how many statistics found all nearest neighbors. Even though some statistics can find all nearest neighbors, a larger number of statistics achieves perfect results at higher values of n.

Up to this point, all of our experiments are based on global alignment identity scores. Other applications require local alignment scores instead of the global ones. Next, we discuss the results of the K-nearest neighbors application using local alignment scores.

K-nearest neighbors using local alignment

Local alignment is widely used as a sequence similarity measure. Using the synthetic data generated from the p27 set, one query sequence was aligned locally versus all sequences in the synthetic set. Supplementary Files S56–S61 include the results of finding the nearest 1, 5 and 10 sequences to a query sequence. Unlike the applications based on the global alignment, EMD and LD are not effective at all as single statistics of finding the nearest one neighbor; additionally, any combination involving either of them does not do well, as expected. No single or paired statistic was able to find the nearest neighbor. For the $K=5$ trial, a majority of single statistics found 2 of 5. Two paired statistics performed slightly better than single statistics. Specifically, Markov * Spearman^2 and D2s * Markov^2 found 3 of 5. For the $K=10$ trial, nine single statistics found 8/ of 10; These statistics include normalized vectors, harmonic mean, EuclideanZ, Jeffrey divergence, KL-Cond, Jensen–Shannon, Pearson coefficient, D2z and SimMM. Additionally, 618 paired statistics identified 8/10. Many of these paired statistics consist of two single statistics, both of which did not obtain 8 of 10, showing the added value of pairing statistics. In sum, for applications based on local alignments, pairing two weaker single statistics could enhance the performance. However, in practice, single statistics should do just as good as the paired statistics at finding the nearest neighbor(s). Further, as expected, EMD and LD are no longer effective in identifying the nearest neighbors as determined by local alignment identity scores.

Time efficiency

Given that the scope of this article is on alignment-free k-mer methods, one of the most important evaluation criteria is time efficiency. Many local and global alignment algorithms may be needed in certain applications because of their level of accuracy. However, if a particular algorithm requires only a quick similarity search, the statistics discussed in this article highly excel

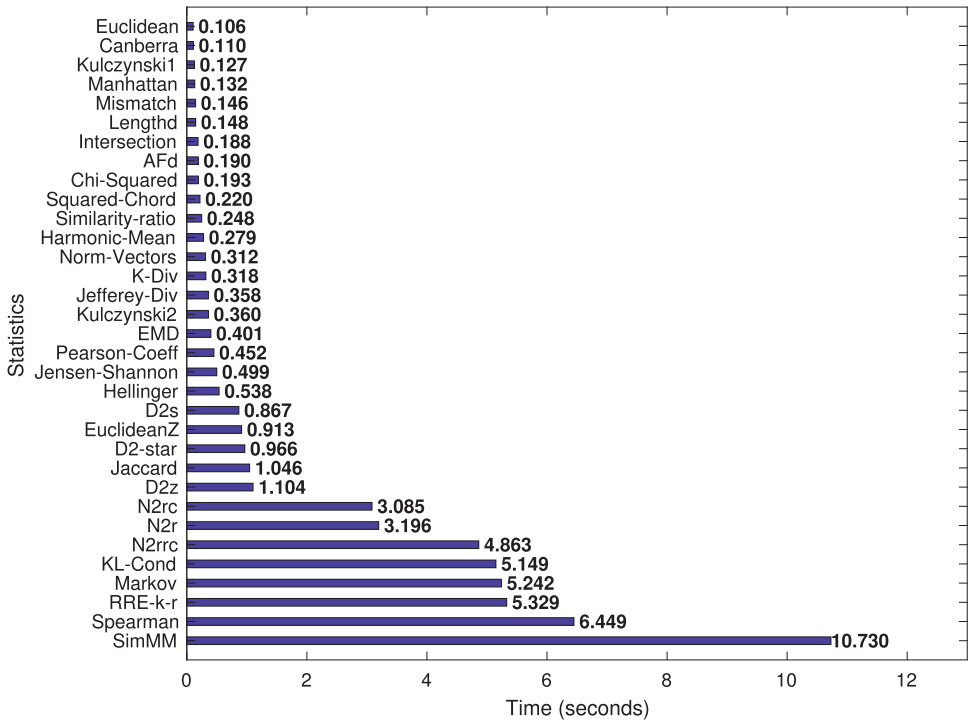

Figure 5. Time to compute similarities for 4735 microbiome sequence pairs. The Minkowski family overall is the most time efficient, but others are comparable; any statistics that use conditional probability such as the Markov family take considerably longer.

at being more time efficient than their alignment-based counterparts.

We ran each statistic on an iMac with a 2.7 GHz Intel Core i5 and 8 GB of DDR3 RAM. After first computing the histograms of 4-mers, the time each statistic took on 28 500 microbiome sequence comparisons was measured (Figure 5). Because the cost of counting n-mers can be minimized with a simple indexing table, which is used by all statistics, we did not consider the additional time needed to construct the histograms.

As expected, many of the statistics can be computed in one line of MATLAB code and are relatively close in terms of time efficiency. However, the slowest statistics are the N2 statistics, KL Cond, revised relative entropy, Markov, Spearman and SimMM. Spearman distance has the added cost of needing to compute the ranks of each of the n-mer frequencies, which can quickly decrease efficiency. The N2 series requires a table to find the locations of other n-mers based on the neighborhood definition. The other four statistics all require conditional probabilities.

When deciding on which alignment-free k-mer statistic to use for a particular application, it is important to take into account the performance/cost ratio. Some statistics such as KL Cond may perform well in terms of accuracy. However, if the database search takes 50 times longer to complete, using a simpler statistic, such as Manhattan, might be a better choice. Although these statistics require linear time, the number of linear operations required can vary and produce important distinctions. For example, the divergence family can take almost three times as long as the Minkowski family.

We have found that multiple combinations of statistics with EMD and LD have been exceptionally useful in multiple experiments. Computing a multiplicative statistic pair will have an added time cost. However, as EMD and LD require a relatively low number of linear operations, their combinations with other efficient statistics can outperform conditional probability methods in both time and accuracy. Alternatively, the best choice for a particular application might be a single statistic, as many are able to perform well in filtering a database and processing NGS reads.

Conclusion

As evident from the Encyclopedia of Distances, there are many statistics to compare two k-mer histograms [22]. Therefore, we decided to cover a good selection of statistics from each statistical family [17]. Our overall analysis was based on using global alignment identity score as a ground truth. We have evaluated each of the discussed statistics in terms of the following four applications:

1. **Sensitivity and specificity** for simple database filtering at identity scores >70%
2. **Linear correlation** with identity score to isolate useful statistics at various cutoff values
3. ***K*-nearest neighbors** with various K values for clustering
4. **Time efficiency** as a reference for all other applications

In addition, we evaluated the K-nearest neighbors application according to the identity scores obtained by a local alignment algorithm. In terms of data sets, we were able to analyze the statistics in conjunction with microbiome sequences, the $p27^{Kip1}$ suppressor gene, synthetic sequences and 3^{rd} generation reads. The benchmark is general and can be used in evaluating the statistics on additional data sets. In many cases, alignment algorithms might be the best option in scanning a database in terms of accuracy. But if the current experiment is in need of a significantly more time efficient method at the cost of some accuracy, alignment-free k-mer statistics would be the ideal solution. The Needleman–Wunsch algorithm on its own tells a

great deal of information about the relationship between two sequences. However, because of the vast amount of information on faster alignment-free *k*-mer approaches, we can effectively find ideal statistics that will work for particular applications in considerably less time.

Key Points

- A large majority of *k*-mer based alignment-free statistics are effective at simple database filtering and identifying identical DNA sequences.
- Multiplicative combinations of statistics with either Earth Mover's distance or sequence length difference are among the top performers across many applications based on global alignment. Even when sequences are close in length (220–260 bp), multiplicative combinations with Earth Mover's distance or length difference still can frequently enhance the performance of single statistics.
- Reducing the *k*-mer length can decrease the time and space requirements while maintaining comparable accuracy.
- For applications based on sequences of the same length or on local alignment, single statistics are effective in identifying the closest sequence(s). Length-based statistics such as Earth Mover's distance or length difference are no longer effective in such applications.
- An evaluation benchmarking tool is provided. Using the benchmark, 33 statistics, their squares and their multiplicative combinations can be evaluated on any set of DNA sequences, resulting in identifying the ideal statistics for a specific purpose.

Supplementary Data

Supplementary data are available at https://github.com/TulsaBioinformaticsToolsmith/Alignment-Free-Kmer-Statistics.

Acknowledgments

The authors would like to thank the anonymous reviewers whose comments and suggestions improved this manuscript greatly.

Funding

This research was supported by internal funds provided by the College of Engineering and Natural Sciences and the Faculty Research Grant Program at the University of Tulsa. The research results discussed in this publication were made possible in part by funding through the award for project number PS17-015, from the Oklahoma Center for the Advancement of Science and Technology.

References

1. Needleman SB, Wunsch CD. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol* 1970;**48**:443–53.
2. Zhang Z, Schwartz S, Wagner L, *et al*. A greedy algorithm for aligning DNA sequences. *J Comput Biol* 2000;**7**(1–2):203–14.
3. Yano M, Mori H, Akiyama Y, *et al*. CLAST: CUDA implemented large-scale alignment search tool. *BMC Bioinformatics* 2014;**15**:406.
4. Altschul SF, Gish W, Miller W, *et al*. Basic alignment search tool. *J Mol Biol* 1990;**215**(3):403–10.
5. Sims GE, Jun SR, Wu GA, *et al*. Alignment-free genome comparison with feature frequency profiles (ffp) and optimal resolutions. *Proc Natl Acad Sci USA* 2009;**106**(8):2677–82.
6. Vinga S, Almeida J. Alignment-free sequence comparison–a review. *Bioinformatics* 2003;**19**(4):513–23.
7. Borozan I, Watt S, Ferretti V. Integrating alignment-based and alignment-free sequence similarity measures for biological sequence classification. *Bioinformatics* 2015;**31**(9):1396–404.
8. Almeida JS, Vinga S. Universal sequence map (USM) of arbitrary discrete sequences. *BMC Bioinformatics* 2002;**3**(1):6.
9. Almeida JS, Grüneberg A, Maass W, *et al*. Fractal MapReduce decomposition of sequence alignment. *Algorithms Mol Biol* 2012;**7**(1):12.
10. Vinga S, Carvalho AM, Francisco AP, *et al*. Pattern matching through Chaos game representation: bridging numerical and discrete data structures for biological sequence analysis. *Algorithms Mol Biol* 2012;**7**(1):10.
11. Haubold B. Alignment-free phylogenetics and population genetics. *Brief Bioinform* 2014;**15**(3):407.
12. Blaisdell BE. A measure of the similarity of sets of sequences not requiring sequence alignment. *Proc Natl Acad Sci USA* 1986;**83**(14):5155–9.
13. Ren J, Song K, Sun F, *et al*. Multiple alignment-free sequence comparison. *Bioinformatics* 2013;**29**(21):2690–8.
14. Cha SH, Srihari SN. On measuring the distance between histograms. *Pattern Recognit* 2002;**35**:1355–1370.
15. Costa AM, Machado JT, Quelhas MD. Histogram-based DNA analysis for the visualization of chromosome, genome and species information. *Bioinformatics* 2011;**27**(9):1207–14.
16. Bonham-Carter O, Steele J, Bastola D. Alignment-free genetic sequence comparisons: a review of recent approaches by word analysis. *Brief Bioinform* 2014;**15**(6):890–905.
17. Cha SH. Comprehensive survey on distance/similarity measures between probability density functions. *Int J Math Models Methods Appl Sci* 2007;**1**(4):300–7.
18. Chattopadhyay AK, Nasiev D, Flower DR. A statistical physics perspective on alignment- independent protein sequence comparison. *Bioinformatics* 2015;**31**(15):2469–74.
19. Pinello L, Lo Bosco G, Yuan GC. Applications of alignment-free methods in epigenomics. *Brief Bioinform* 2014;**15**(3):419–30.
20. Vinga S. Editorial: alignment-free methods in computational biology. *Brief Bioinform* 2014;**15**(3):341–2.
21. Zharkikh AA, Rzhetsky AY. Quick assessment of similarity of two sequences by comparison of their l-tuple frequencies. *BioSystems* 1993;**30**(1–3):93–111.
22. Deza MM, Deza E. *Encyclopedia of Distances*. Heidelberg, Germany: Springer-Verlag Berlin Heidelberg, 2009.
23. Reinert G, Chew D, Sun F, *et al*. Alignment-free sequence comparison (i): statistics and power. *J Comput Biol* 2009;**16**(12):1615–34.
24. Kantorovitz MR, Robinson GE, Sinha S. A statistical method for alignment-free comparison of regulatory sequences. *Bioinformatics* 2007;**23**(13):i249–55.
25. Lippert RA, Huang H, Waterman MS. Distributional regimes for the number of k-word matches between two random sequences. *Proc Natl Acad of Sci USA* 2002;**99**(22):13980–9.
26. 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**(1):106–16.

27. Göke J, Schulz MH, Lasserre J, *et al*. Estimation of pairwise sequence similarity of mammalian enhancers with word neighbourhood counts. *Bioinformatics* 2012;**28**(5):656–63.
28. Zhang Y, Chen W. A new measure for similarity searching in dna sequences. *MATCH Commun Math Comput Chem* 2011; **65**(2):477–88.
29. Ghandi M, Lee D, Mohammad-Noori M, *et al*. Enhanced regulatory sequence prediction using gapped k-mer features. *PLoS Comput Biol* 2014;**10**(7):e1003711.
30. Leimeister CA, Boden M, Horwege S, *et al*. Fast alignment-free sequence comparison using spaced-word frequencies. *Bioinformatics* 2014;**30**(14):1991–9.
31. Steele J. *The Cauchy-Schwarz Master Class: An Introduction to the Art of Mathematical Inequalities*. Cambridge, UK: Cambridge University Press, 2004.
32. Ghahramani Z. An introduction to Hidden Markov Models and Bayesian networks. *Int J Patt Recogn Artif Intell* 2001;**15**:9–42.
33. Wu TJ, Hsieh YC, Li LA. Statistical measures of dna sequence dissimilarity under markov chain models of base composition. *Biometrics* 2001;**57**(2):441–8.
34. Pham TD, Zuegg J. A probabilistic measure for alignment-free sequence comparison. *Bioinformatics* 2004;**20**(18):3455.
35. Dai Q, Yang Y, Wang T. Markov model plus k-word distributions: a synergy that produces novel statistical measures for sequence comparison. *Bioinformatics* 2008;**24**(20): 2296.
36. Cover TM, Thomas JA. *Joint Entropy and Conditional Entropy*. Hoboken, New Jersey: John Wiley and Sons, Inc, 2006, 16–18.
37. Wei D, Jiang Q, Wei Y, *et al*. A novel hierarchical clustering algorithm for gene sequences. *BMC Bioinformatics* 2012;**13**(1): 174.
38. Rubner Y, Tomasi C, Guibas LJ. A metric for distributions with applications to image databases. In: *Proceedings of the 1998 IEEE International Conference on Computer Vision, IEEE, Bombay, India*. 1998, 59–66.
39. Zhao X, Sandelin A. Gmd: measuring the distance between histograms with applications on high-throughput sequencing reads. *Bioinformatics* 2012;**28**(8):1164–5.
40. Girgis HZ. Red: an intelligent, rapid, accurate tool for detecting repeats de-novo on the genomic scale. *BMC Bioinformatics* 2015;**16**(1):227.
41. Compeau P, Pevzner P. *Bioinformatics Algorithms: An Active Learning Approach*, 2nd edn. La Jolla, California: Active Learning Publishers, 2015.
42. Costello EK, Lauber CL, Hamady M, *et al*. Bacterial community variation in human body habitats across space and time. *Science* 2009;**326**(5960):1694–7.
43. Moeller SJ, Head ED, Sheaff RJ. p27kip1 inhibition of grb2-sos formation can regulate ras activation. *Mol Cell Biol* 2003;**23**(11): 3735–52.
44. Seo JS, Rhie A, Kim J, *et al*. *De novo* assembly and phasing of a korean human genome. *Nature* 2016;**538**(7624):243–7.
45. Song K, Ren J, Reinert G, *et al*. New developments of alignment-free sequence comparison: measures, statistics and next-generation sequencing. *Brief Bioinform* 2014;**15**(3): 343–53.