Natural Computing (2025) 24:973–989
https://doi.org/10.1007/s11047-025-10054-5

The threshold *q*-gram distance: a simple, efficient, and effective distance measure for genomic sequence comparison

Davide Cenzato[1] · **Giuditta Franco**[2] · **Zsuzsanna Lipták**[2] · **Alessio Milanese**[3]

Received: 5 March 2025 / Accepted: 27 August 2025 / Published online: 27 November 2025
© The Author(s) 2025

Abstract

The q–gram distance between two strings s, s', introduced by Ukkonen in 1992, is an alignment-free string similarity measure which can be computed in linear time, as opposed to the quadratic time necessary for alignment/edit distance. It is based on the L_1-distance, or Manhattan-distance, between the multiplicity vectors of fixed-length substrings (so-called *q-grams* or *k-mers*), and has been successfully applied in diverse bioinformatics settings. In this paper, we introduce the *threshold q-gram distance* (T*q*D), a new distance measure which is similar to the q-gram distance but uses reduced information on the multiplicities of the q-grams. The new measure retains the linear time computation of the q-gram distance but requires significantly less space. Storage space and accuracy of the measure can be controlled via a user-defined threshold t, which sets a limit on the maximum value of the integers in the multiplicity vectors. In particular, for $t = 1$, the comparison is made only on the basis of the sets of uniquely occurring q-grams on the one hand, and of repeated q-grams, on the other. We tested the new distance measure, using the benchmarking tool *AFproject* of Zielezinski et al. [Genome Biology, 2019], on several real-life data sets for phylogenetic reconstruction and compared the results with those of other k-mer based distance measures. Our experiments show that the new measure T*q*D compares well to other non-alignment based measures regarding accuracy, while requiring substantially less memory than the classic q-gram distance.

Keywords Alignment-free distance measures · Dictionary based discrimination methods · Genomic sequence analysis · k-mer counts · q-gram distance

1 Introduction

The advent of next-generation sequencing (NGS) technologies has greatly increased the amount and variety of biological data available (Stephens et al. 2015), and gave rise to new computational challenges in bioinformatics. It has become even more pressing to design efficient algorithms that are able to cope with data abundance and complexity. In this scenario, the reliable and efficient estimation of similarities and dissimilarities among biological strings plays a crucial role. In fact, in many bioinformatics applications, such as pairwise sequence analysis, phylogenetic reconstruction, and genome assembly, the employment of appropriate pairwise string similarity measures is essential.

Alignment-free (AF) string distances have been around for several decades, see the classic surveys (Vinga and Almeida 2003; Mantaci et al. 2008), or the more recent ones (Zielezinski et al. 2017; Luczak et al. 2019; Swain and Vickers 2022). They were originally proposed as alternatives to edit distance (Levenshtein 1966) (whose similarity

✉ Giuditta Franco
giuditta.franco@univr.it

Davide Cenzato
davide.cenzato@unive.it

Zsuzsanna Lipták
zsuzsanna.liptak@univr.it

Alessio Milanese
milanese.alessio@gmail.com

[1] Department of Environmental Sciences, Informatics and Statistics, Ca' Foscari University of Venice, via Torino 155, Mestre, VE 30172, Italy

[2] Department of Computer Science, University of Verona, Strada Le Grazie 15, Verona 37134, Italy

[3] Department of Biology, Institute of Microbiology and Swiss Institute of Bioinformatics, ETH, Wolfgang-Pauli-Strasse 27, Zürich HIT F 41, 8093, Switzerland

variant is usually referred to as pairwise sequence alignment (Needleman and Wunsch 1970)), which requires quadratic time in its original form. Even though many improvements have been suggested since its introduction (Ukkonen 1985; Myers 1986, 1999; Hermelin et al. 2013; Marco-Sola et al. 2020), which often work very well in practice, especially on real-life biological strings, it is also known that the problem itself is unlikely to have an algorithm with strongly subquadratic worst-case running time (Backurs and Indyk 2018). On the other hand, most alignment-free string distances run in worst-case linear time.

Many alignment-free string distances are based on k-mer counts, i.e., they compute the distance of two sequences based on the number of occurrences of short fixed-length strings (k-mers, also known as q-grams or n-words). One classic such distance is the q-gram distance, introduced by Ukkonen in 1992, as an efficient filter for edit distance (Ukkonen 1992). The key points of its success were a clean mathematical characterization, including a relationship between the q-gram distance and the edit distance of two strings, and a simple algorithm scaling linearly in the size of the sequences. In addition, it is easy to understand and can be implemented efficiently. Early tools based on q-gram distance include (Burkhardt et al. 1999; Rasmussen et al. 2006), while many current tools use similar ideas, related to k-mer count, for string similarity computation (Hanada et al. 2014; Sims et al. 2009; Lu et al. 2017; Ondov et al. 2016).

In most scenarios, edit distance is still considered the gold standard, and the quality of any new distance measure is evaluated against how close it is to the edit distance. However, in certain cases, edit distance is not actually the best choice for detecting biological similarity (Arias et al. 2023). For example, some studies show that certain alignment-free distances can be more accurate for phylogenetic reconstruction than edit distance (Sims et al. 2009; Chan et al. 2014). It is also well known that classic edit distance fails in the case of large-scale editing events such as common types of genome rearrangements (inversions, translocations, insertions, fusion and fission), or indeed in some common text editing applications (swaps, large-scale copy-paste edits) (Amir and Levy 2010). String distances based on q-grams can be more appropriate in these scenarios.

In this paper, we introduce a new alignment-free string distance measure, which we call *threshold q-gram distance* (or TqD for short). It is a variant of the q-gram distance of Ukkonen (1992) but incorporates an additional parameter, the threshold, which limits the maximum count for q-gram multiplicities.

Recall that the classic q-gram distance of Ukkonen (1992) is defined as the L_1-distance (or Manhattan-distance) between the q-gram profiles of the two strings, the q-gram profile of a string s being an array containing the multiplicity (number of occurrences) of each q-gram in s. The q-gram distance can be computed in linear time using a sliding window algorithm. Now, TqD is defined as the Hamming distance of a variant of the q-gram profiles, where each entry is capped by $t+1$, with t being the threshold. The reasoning is that we are interested in the exact number of occurrences only up to t, and that $t+1$ stands for 'many', i.e., more than t occurrences. This allows TqD to tune its memory footprint by storing a more compact q-gram profile. We show that TqD inherits several of the nice theoretical properties of the original q-gram distance, while also maintaining a simple linear time computation algorithm. In addition, TqD can scale to larger genomic datasets due to its reduced memory footprint, making it suitable for more memory-intensive applications in bioinformatics. The underlying theory includes the introduction of *threshold De Bruijn graphs*, which can be used to model a string's threshold q-gram profile, the same way De Bruijn graphs are a valuable tool for handling the q-gram profile of a string.

The inspiration to introduce this new distance came from the observation of nature, in terms of slippage events (trinucleotide or dinucleotide expansion or contraction) occurring in some biological string replication and evolutionary processes, and from the analysis of genomic string similarity based on the ratio between repeats and words occurring exactly once (Franco and Milanese 2013; Castellini et al. 2015). Indeed, for threshold $t=0$, TqD uses only the presence or absence of q-grams for computing the distance, while for threshold $t=1$, it distinguishes between q-grams occurring exactly once (unique q-grams), q-grams occurring more than once (repeated q-grams), and q-grams not occurring at all (absent q-grams).

We complement our theoretical analysis by presenting an experimental evaluation of TqD. To this end, we rely on the AFproject (Zielezinski et al. 2019), a web service that provides a benchmarking platform for the comparison of AF distance measures, using a standardized testing procedure. We show that in spite of its simplicity, TqD compares well with 61 AF distance measures based on q-gram counting when performing phylogenetic reconstruction, a critical task in evolutionary studies and genome analysis. This consistent performance highlights TqD's practical value in bioinformatics applications and suggests that it could serve as a viable alternative to classical existing AF methods, especially in scenarios where memory efficiency and scalability are required (Akbari Rokn Abadi et al. 2024).

2 Background

Let Σ be a finite alphabet of cardinality σ. We write strings (or sequences) s over Σ as $s = s[1] \cdots s[n]$, where each $s[i]$, for $1 \leq i \leq n$, is an element of Σ, and the length n of s is denoted $|s|$. The set of all strings over Σ is denoted Σ^*.

A string u of length m is called a *substring* of a string s of length n, $m \leq n$, if there exists a position $1 \leq i \leq n$ such that $s[i] \cdots s[i+m-1] = u$; such a position i is called an *occurrence* of u in s. For brevity, we also write $s[i..j]$ for $s[i] \cdots s[j]$, with $1 \leq i \leq j \leq n$.

For a positive integer q, a *q-gram* is a string over Σ of length q. Let $u_0, \ldots, u_{\sigma^q-1}$ be some enumeration of all q-grams over Σ (e.g. in lexicographic order, for an ordered alphabet Σ). For a string s with length at least q, the *q-gram profile* $P_q(s)$ *of* s is defined as an array of size σ^q, where the ith entry of $P_q(s)$ equals the number of occurrences in string s of the ith q-gram u_i, for $0 \leq i \leq \sigma^q - 1$.

The *q-gram distance* (Ukkonen 1992) between two strings s, s' over Σ is defined as the L_1-distance, or *Manhattan-distance*, between their q-gram profiles: $\mathrm{dist}_q(s, s') = \sum_{i=0}^{\sigma^q-1} |P_q(s)[i] - P_q(s')[i]|$.

Example 1 Let $s = \mathrm{ACACGACAC}$ and $s' = \mathrm{CACAGAC}$. Then the 2-gram distance of s and s' is $\mathrm{dist}_2(s, s') = 4$. The 2-gram profiles of s and s' are as follows, where the 2-grams are listed lexicographically, with $\mathrm{A} < \mathrm{C} < \mathrm{G} < \mathrm{T}$: $P_2(s) = [0, 4, 0, 0, 2, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0]$ and $P_2(s') = [0, 2, 1, 0, 2, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]$.

As was shown in Ukkonen (1992), the q-gram distance is a pseudo-metric: it is non-negative, reflexive, symmetric, and the triangle inequality holds, but two distinct strings can have q-gram distance 0 (thus, it is not a metric). It is easy to see that $\mathrm{dist}_q(s, s') = 0$ if and only if s and s' have the same q-gram profile.

The well-known *(unit-cost) edit distance*, or *Levenshtein distance* (Levenshtein 1966), is defined as the minimum number of edit operations converting string s into string s', where edit operations can be the substitution, deletion, or insertion of one character. There is a simple relationship between the q-gram distance $\mathrm{dist}_q(s, s')$ and the edit distance:

Lemma 1 *(q-gram Lemma (Ukkonen 1992)) For two strings s, s' over Σ, and $q > 0$:*

$$\mathrm{dist}_q(s, s') \leq 2q \cdot \mathrm{dist}_{\mathrm{edit}}(s, s').$$

It was conjectured in Ukkonen (1992) and proven in Pevzner (1995) that two strings that have the same q-gram profile can be transformed into one another using a finite number of transformations of two types (called *transposition* resp. *rotation*). The central tool for obtaining this result was a directed multigraph for modeling the q-gram profile of a string, first introduced for this problem in Pevzner (1989), and now usually referred to as *De Bruijn subgraph* or simply *De Bruijn graph of a string*.[1] We give the definition of this graph next. Note that this definition is different from the (classic) De Bruijn graph (De Bruijn 1946), a simple directed graph whose vertex set is the set of *all* $(q-1)$-grams and whose edge set is the set of *all* q-grams; the definition we use here instead is the one common in bioinformatics, specifically in sequencing algorithms (Zerbino and Birney 2008).

Recall that the *De Bruijn graph of order q* of a string s, $G(s, q)$, is a directed multigraph with vertex set equal to the $(q-1)$-length substrings of s, and with edges defined as follows: for every q-length substring u of s, where $u = axb$ with $a, b \in \Sigma$ and $x \in \Sigma^*$, there is a directed edge e from vertex ax to vertex xb, with multiplicity $\mu_G(e)$ equal to the number of occurrences of u in s. As was shown in Pevzner (1989, 1995), every Euler-trail in $G(s, q)$ corresponds to a string with the same q-gram profile as s, and thus with q-gram distance 0 to s. Equivalently, $\mathrm{dist}_q(s, s') = 0$ if and only if $G(s, q) = G(s', q)$.

Example 2 Let $s = \mathrm{ACACGACACG}$ and $q = 3$. The De Bruijn graph $G(s, 3)$ is shown on the left of Fig. 1. There are two other Euler paths in $G(s, 3)$, with the corresponding strings $s' = \mathrm{ACGACACACG}$ and $s'' = \mathrm{ACACACGACG}$. Therefore, $\mathrm{dist}_3(s, s') = \mathrm{dist}_3(s, s'') = \mathrm{dist}_3(s', s'') = 0$.

The notion of *De Bruijn graph of order q* implicitly assumes $q \geq 2$. For this reason in next section we will work with, and define the threshold q-gram distance only for, $q \geq 2$.

3 Threshold q-gram distance

In this section, we introduce the threshold q-gram distance, TqD for short, between two strings, and study some of its properties. First we need the threshold q-gram profile, which caps the entries at $t+1$, with $t+1$ standing for 'more than t' occurrences of the respective q-gram.

[1] According to Dutch spelling, De Bruijn has to be capitalized, as explained in (Van Lint and Wilson 2001, Ch. 8): "We mention a peculiarity concerning the spelling of some Dutch names. When omitting the initials of N. G. de Bruijn, one should capitalize the word 'de' and furthermore the name should be listed under B. Similarly Van der Waerden is correct when the initials are omitted and he should be listed under W." We thank a reviewer for pointing this out.

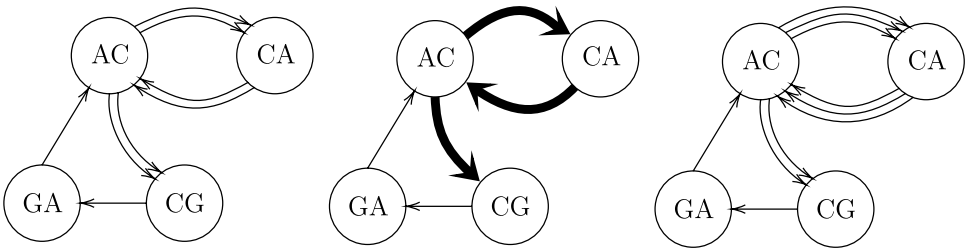

Fig. 1 De Bruijn graph $G(s, 3)$ for $s = \text{ACACGACACG}$ (left) and its threshold graph $H^1(s, 3)$ (middle). On the right, another realization of $H^1(s, 3)$, which is the De Bruijn graph $G(\hat{s}, 3)$ for $\hat{s} = \text{ACACGACACACG}$. Here, $q = 3$ and $t = 1$, see Examples 2 and 4

Definition 1 *(threshold q-gram profile)* Let $t \geq 0$ and $q \geq 2$ be two integers and let s be a string over Σ. The *threshold q-gram profile* of a string s with threshold t is defined as an array $P_q^t(s)$ of length σ^q, where

$$P_q^t(s)[i] = \min(P_q(s)[i], t + 1). \tag{1}$$

In particular, if $t = 0$, then the threshold q-gram profile is a bitvector, with $P_q^0(s)[i] = 1$ if and only if the q-gram u_i occurs in s. For $t = 1$, the set $\{u_i \mid P_q^1(s)[i] = 1\}$ contains all uniquely occurring q-grams of s, and $\{u_i \mid P_q^1(s)[i] = 2\}$ all repeated q-grams of s.

Definition 2 *(threshold q-gram distance)* Let $t \geq 0$ and $q \geq 2$ be two integers and s, s' two strings over Σ. The *threshold q-gram distance* $\text{dist}_q^t(s, s')$ is defined as the Hamming distance of the threshold q-gram profiles of s and s':

$$\text{dist}_q^t(s, s') = |\{i \mid 0 \leq i < \sigma^q, P_q^t(s)[i] \neq P_q^t(s')[i]\}|. \tag{2}$$

In other words, the threshold q-gram distance of s and s' equals the number of q-grams whose multiplicity differs in s and s', when counted with threshold t.

Example 3 Let $s = \text{ACACGACAC}$ and $s' = \text{CACAGAC}$, as in Example 1. Then threshold 2-gram distance of s and s', with threshold $t = 1$, is $\text{dist}_2^1(s, s') = 2$, since the q-gram AC, which occurs four times in s and twice in s', no longer contributes to the distance. The threshold 2-gram profiles are: $P_2^1(s) = [0, 2, 0, 0, 2, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0]$ and $P_2^1(s') = [0, 2, 1, 0, 2, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]$.

Note that even if we allowed $t = \infty$, the threshold q-gram distance would in general differ from the q-gram distance, since the former is the Hamming distance and the latter the Manhattan distance (L_1-distance) of the q-gram profiles. The choice of Hamming distance rather than Manhattan distance in the TqD definition is motivated by typical applications in bioinformatics, where biological strings are compared according to the difference in the number and identity of unique (or scarcely present) vs. (massively) repeating q-grams.

Lemma 2 *Let s, s' be two strings over Σ and $t \geq 0$, $q \geq 2$ two integers. The following hold:*

1. $\text{dist}_q^t(s, s') = 0$ *if and only if* $P_q^t(s) = P_q^t(s')$;
2. dist_q^t *is a pseudo-metric;*
3. $\text{dist}_q^t(s, s') \leq \text{dist}_q(s, s')$;
4. $\text{dist}_q^t(s, s') \leq \text{dist}_q^{t+1}(s, s')$.

Proof Statement *1.* follows directly from the definition of threshold q-gram distance. Statement *2.* holds because the Hamming-distance is a metric, thus reflexivity, symmetry, and transitivity are inherited by TqD; however, since it is possible that two distinct strings have the same threshold q-gram profile (see Example 4), TqD is not a metric. Statements *3.* and *4.* follow respectively from the fact that for each i, $P_q^t(s)[i] \leq P_q(s)[i]$, and from the fact that for all i, $P_q^t(s)[i] = \min(P_q(s)[i], t + 1) \leq \min(P_q(s)[i], t + 2) = P_q^{t+1}(s)[i]$. □

Since for any $t \geq 0$, $\text{dist}_q^t(s, s') \leq \text{dist}_q(s, s')$, as a corollary of Lemma 1 we get a relationship between the threshold q-gram distance and the edit distance:

Corollary 1 *For two strings s, s' over Σ, $q \geq 2$ and $t \geq 0$:* $\text{dist}_q^t(s, s') \leq 2q \cdot \text{dist}_{\text{edit}}(s, s')$.

This implies that the threshold q-gram distance can be used as a filter for the edit distance, similarly to the q-gram distance.

To model the threshold q-gram profile of a string s, we introduce a new graph:

Definition 3 *(De Bruijn threshold graph)* Let s be a string and $t \geq 0$, $q \geq 2$ two integers. The *De Bruijn threshold graph* $H^t(s, q) = (V, E)$ of s is a directed multigraph with vertex set V equal to the $(q-1)$-grams of s, and $E = \{(ax, xb) \mid a, b \in \Sigma, x \in \Sigma^{q-2}, axb \text{ occurs in } s\}$. Let us denote by $\mu_H : E \to \mathbb{N}^+$ the multiplicity of edges, then for an edge e corresponding to q-gram u_i, $\mu_H(e) = P_q^t(s)[i]$.

We refer to e as a *fat edge* if $\mu_H(e) = t + 1$, otherwise e is called a *thin edge*.

For an example of threshold graph with threshould equal to 1, see Fig. 1 (middle). We call a De Bruijn graph G a *realization* of a threshold graph H^t if $V(G) = V(H), E(G) = E(H)$, and for all edges $e \in E(G)$: if $\mu_G(e) \leq t$ then e is a thin edge in H and $\mu_G(e) = \mu_H(e)$, otherwise e is a fat edge in H (now $\mu_G(e) > t$). Clearly, if $G = G(s, q)$ and $H = H^t(s, q)$ then G is a realization of H. See Fig. 1 for an example of a threshold De Bruijn graph and two of its realizations.

Note that two strings with distance 0 need not have the same length, as the following example shows (see also Fig. 1).

Example 4 In Fig. 1, we see two realizations (left and right) of the same threshold graph (middle). The realization on the left is the De Bruijn graph $G(s, 3)$ for $s = \text{ACACGACACG}$, and the one on the right is $G(\hat{s}, 3)$ for $\hat{s} = \text{ACACGACACACG}$. The value of the threshold is $t = 1$. The threshold q-gram distance of the two strings is 0.

In the following, we give a characterization of pairs of strings with threshold q-gram distance equal to 0.

Theorem 1 *Let s be a string over Σ, $q \geq 2$ and $t \geq 0$ two integers, and $H = H^t(s, q)$ its threshold graph. Then for all $s' \in \Sigma^*$, $\text{dist}_q^t(s, s') = 0$ if and only if $G(s', q)$ is a realization of H. Equivalently, $\text{dist}_q^t(s, s') = 0$ if and only if s' corresponds to an Euler-trail in some realization G of $H^t(s, q)$.*

Proof For the first equivalence, note that if $\text{dist}_q^t(s, s') = 0$ then for all i, if $P_q(s)[i] \leq t$ then $P_q(s)[i] = P_q(s')[i]$, and if $P_q(s)[i] > t$ then also $P_q(s')[i] > t$. This implies that $H^t(s, q) = H^t(s', q)$; since $G(s', q)$ is a realization of $H^t(s, q)$, the right side follows. Conversely, let $G = G(s', q)$ and $H = H(s, q)$. We have to show that for every i, $P_q^t(s)[i] = P_q^t(s')[i]$. Let u_i be the ith q-gram. First, if $P_q^t(s)[i] = 0$ then there is no edge in H corresponding to u_i, and since G is a realization of H, this implies that there is no edge in G either, thus $P_q^t(s')[i] = 0$. Next, if $0 < P_q^t(s)[i] \leq t$ then there is an edge e corresponding to u_i in H, and $\mu_H(e) = P_q^t(s)[i]$. Again, since G is a realization of H, this implies that $P_q^t(s')[i] = \mu_G(e) = \mu_H(e) = P_q^t(s)[i]$. Finally, if $P_q^t(s)[i] = t + 1$ then e is a fat edge in H and thus $\mu_G(e) > t$, implying that $P_q^t(s)[i] = t + 1$.

For the second equivalence, note that if a string s corresponds to an Euler-trail in a De Bruijn graph G of order q, then $G = G(s, q)$. □

Since two strings have threshold q-gram distance 0 if and only if their threshold q-gram profiles coincide, Theorem 1 shows that threshold graphs are in one-to-one correspondence with threshold q-gram profiles, the same way as De Bruijn graphs are with q-gram profiles. By Theorem 1, therefore, $\text{dist}_q^t(s, s') = 0$ if and only if s' corresponds to an Euler-trail in some realization G of $H^t(s, q)$.

Since the *threshold* q-gram distance can be 0 even if the q-gram distance is greater than 0, it is an interesting question which type of string transformations, besides transposition and rotation (Ukkonen 1992), keep the threshold q-gram profile (or equivalently, the threshold graph) invariant. In other words, we are interested in an analogous version of the Pevzner theorem (*Repeatedly applying rotation and transposition to a string s we can obtain all the strings with the same q-gram profile of s* (Ukkonen 1992; Pevzner 1995)) for strings with the same De Bruijn threshold graph. In the following we introduce a few concepts and report preliminary results from Milanese (2015) in this context.

Given a string s and its threshold graph $H = H^t(s, q)$, let us call a graph G an *Eulerian realization* of H if it is a realization of H and it contains an Eulerian trail or Eulerian circuit (such as, for example, is the case of both realizations on the left and the right in Fig. 1). Of course, not all realizations of a threshold graph are Eulerian.

For a multigraph G, there is only one corresponding protograph $H(G)$ (by definition), while a threshold graph can have more than one Eulerian realization. We call a threshold graph *Eulerian* if it has at least one Eulerian realization. Clearly, if $H = H^t(s, q)$ for some string s, then H is Eulerian.

Therefore, given a string s, all strings with TqD equal to 0 from s correspond to Eulerian trails in Eulerian realizations of $H^t(q, s)$, and to find what kind of string transformations keep the threshold profile invariant, we can study different Eulerian realizations of the same threshold graph, combining them with the transformations of Ukkonen (1992), which are known to produce all strings that have the same De Bruijn graph. Fig. 2 gives a sketch of this concept.

Given a threshold graph H, its different Eulerian realizations can be expressed by transforming one into the other. The following transformations between Eulerian realizations of the same threshold graph were developed in Milanese (2015): (1) changing the beginning or end of an Eulerian trail, (2) alternating fat cycles, (3) parallel fat paths, and (4) directed fat cycles. It was shown, for example, that an alternating fat cycle in H with certain properties creates more than one realization of H. An alternating fat cycle is a necessary condition for having more than one realization, but it is not sufficient. On the other hand, a directed fat cycle is a necessary and a sufficient condition for an Eulerian threshold graph H to have an infinite number of Eulerian

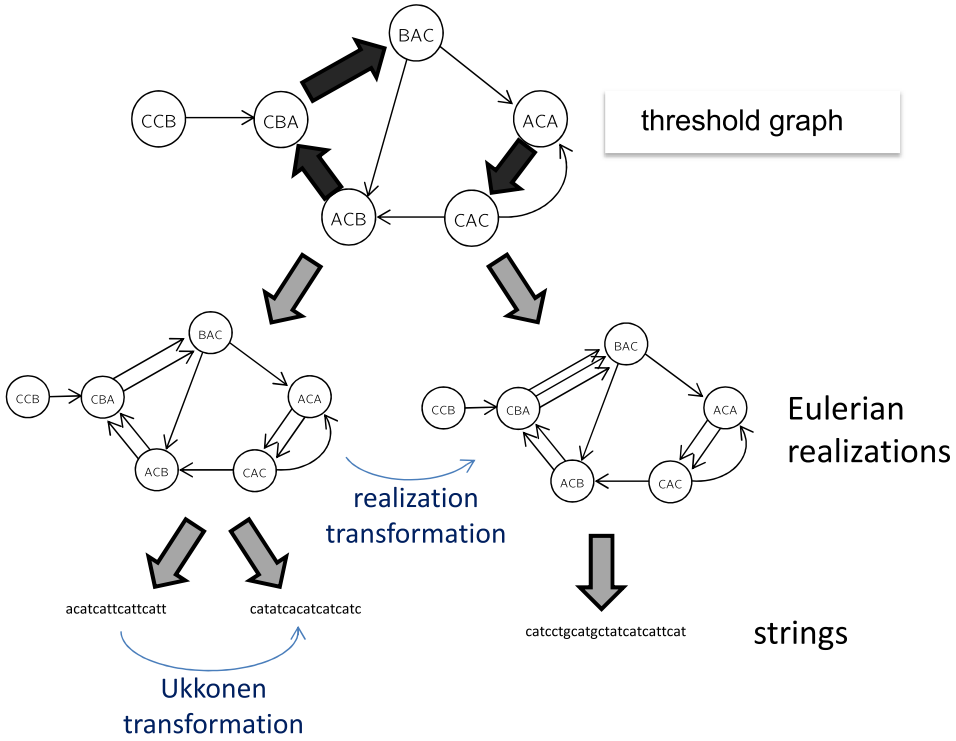

Fig. 2 Representation of the realization transformations and the Ukkonen string transformations (Milanese 2015). In the example, $q = 4$ and $t = 1$.

realizations. A conjecture was also formulated regarding when H has a unique Eulerian realization.

4 Computation of the threshold q-gram distance

As is well known, the q-gram distance can be computed in linear time (Ukkonen 1992): using a sliding window of size q, the q-gram profile of s is computed by incrementing the count of the current q-gram by 1. The crucial trick is to map the alphabet to the set $\{0, 1, \ldots, \sigma - 1\}$, interpret each q-gram as a base-σ number, and use this number as the q-gram's rank in the profile: if the current window contains $s[i..i + q - 1]$, then the rank of this q-gram is $r_i = s[i]\sigma^{q-1} + s[i + 1]\sigma^{q-2} + \cdots + s[i + q - 1]\sigma^0$, where we equate the character $s[i]$ with the corresponding integer in $\{0, 1, \ldots, \sigma - 1\}$. This allows constant-time update from one window to the next, since $r_{i+1} = (r_i - s[i]\sigma^{q-1})\sigma + s[i + q]$ (for more details, see (Ukkonen 1992)). Since there are $n - q + 1$ windows of size q in a string of length n, the profile can thus be computed in $O(n)$ time, for sufficiently small q and σ. Similarly, the q-gram profile of s' can be computed in $O(m)$ time, and their L_1-distance in $O(\sigma^q)$ time. Choosing $q \leq \min(\log_\sigma |s|, \log_\sigma |s'|)$ yields a total running time and storage space of $O(n + m)$. (Several improvements exist, including using one array of size σ^q only; or, as suggested in the original publication (Ukkonen 1992), using two auxiliary linked lists which contain those q-grams that occur in at least one of the two strings, reducing the running time to $O(n + m)$, for any q.)

We modify this algorithm to output the threshold q-gram profile instead of the q-gram profile: slide a q-size window over s and let r_i ($0 \leq r_i \leq \sigma^q - 1$) be the rank of the current content of the window $s[i..i + q - 1]$. Then

$$P_q^t[r_i] \leftarrow \min(P_q^t[r_i] + 1, t + 1). \tag{3}$$

As before, the rank of the current window can be updated in constant time. The threshold q-gram distance of two strings s, s' is then computed by first computing both threshold profiles, and then counting the number of different entries in $O(\sigma^q)$ time.

Therefore, for $q \leq \min(\log_\sigma |s|, \log_\sigma |s'|)$, both the running time and the storage space required are $O(n + m)$. Moreover, it is possible, as for the q-gram distance, to obtain $O(n + m)$ running time even for larger q by using two linked lists L_1 and L_2, which contain only those q-grams that actually occur in s and s', respectively, initializing only these entries of the two profiles to 0, and restricting the final computation to these entries only.

The fundamental difference to the q-gram distance is the space requirement: while the q-gram profile needs $\sigma^q \log_2 n$ bits of space, the threshold q-gram profile requires only $\sigma^q \log_2(t + 1)$ bits, and in particular, for $t = 1$, σ^q bits. To be precise, the q-gram profile occupies

$\sigma^q(\lfloor \log_2 n \rfloor + 1)$ bits, and the threshold q-gram profile occupies $\sigma^q(\lfloor \log_2(t+1) \rfloor + 1)$ bits. We summarize the above by:

Theorem 2 *The threshold q-gram distance between two strings s and s' can be computed in $O(n+m)$ time, using $2\sigma^q \log_2(t+1)$ bits, where n and m are the lengths of s and s', respectively, and t is the threshold.*

Since for large n, m, and q close to $\log n, \log m$, most entries will be 0 or 1, the information lost is small but the space gain significant. We will see experimental evidence of this in the next sections.

5 Experimental evaluation in comparison with other alignment-free distance measures

In this section, we present our experimental evaluation of the threshold q-gram distance[2], comparing it with several other AF distance measures based on q-gram counting. To this aim, we rely on AFproject (Zielezinski et al. 2019), an online platform that implements a benchmarking service for AF distance measures. In particular, we apply TqD on five genomic sequence datasets to perform phylogeny reconstruction, and compare our results with those of several other AF competing distance measures included in the AFproject platform.

Given the definition of TqD, to ensure a fair comparison, we decided to exclude certain methods and limit our analysis to distance measures based on q-gram counting. In particular, we included all distance measures that use q-gram counting, with or without frequency, in their computation. Additionally, we only selected methods that take in input a specific q value, thus excluding those that employ dynamic procedures to choose q. Finally, we did not include any method not originally listed on the AFproject if no reference was available. In total, we compare TqD with 61 AF distance measures: 33 distance measures implemented by AFKS (Luczak et al. 2019) (*afd, camberra, chi_squared, d2_star, d2s, d2z, emd, euclidean, eucliadian_z, harmonic_mean, hellinger, intersection, jaccard, jefferey_divergence, jensen_shannon, k_divergence, kl_conditional, kulczynski1, kulczynski2, length_difference, manhattan, markov, mismatch, n2r, n2rc, n2rrc, normalized_vectors, pearson, rre_k_r, sim_mm, simratio, spearman, squared_chord*), 13 distance measures implemented by alfpy (Zielezinski et al. 2017) (*angle_cos_diss, angle_cos_evol, braycurtis, canberra, chebyshev, euclid_norm, euclid_squared, google, jsd, kld, lcc, manhattan, minkowski*), three distance measures implemented by CAFÉ (Lu et al. 2017) (*cvtree, d2_shepp, d2_star*), three distance measures implemented by jD2Stat (Chan et al. 2014) (*D2n, D2S, D2St*), as well as AAF (Fan et al. 2015), FFP (Sims et al. 2009), kWIP (Murray et al. 2017), Mash (Ondov et al. 2016), Skmer (Sarmashghi et al. 2019), PC-MER (Akbari Rokn Abadi et al. 2024), AF_TFIDF (Delibaş 2025), AFM_BTkNG_ (Delibaş et al. 2020) and CD-MAWS (Anjum et al. 2023).

5.1 Methods

The AFproject (http://afproject.org) is a publicly available web-based service for evaluating the performance of tools that implement AF distance measures in five different research scenarios, using twelve biological datasets. These scenarios comprise protein sequence classification, gene tree inference, regulatory sequence identification, genome-based phylogenetics, and horizontal gene transfer.

The AFproject collects the performance of 74 AF methods, organized into 24 different tools. These performances were gathered by asking the developers to run their software on the twelve reference datasets, using the parameters that yield the best outcomes, and then storing the final results in the project's database. This setup offers developers an easy way to benchmark their new AF measures: they simply need to download the datasets, run their tools, and compare their results with those stored in the database. Additionally, the performance of the new methods can be made publicly available on the website, expanding the original collection of AF methods.

The parameters used to assess performance vary depending on the research scenario. In this paper, we focus on phylogeny reconstruction, where the performance of all AF distance measures is evaluated using two tree-based metrics: the Robinson-Foulds distance (RF) (Robinson and Foulds 1981) and the Quartet distance (QD) (Estabrook et al. 1985). In particular, the AFproject website takes in input a matrix of pairwise distances and constructs a phylogenetic tree (custom tree). Then, it calculates the RF and QD distance values between the custom tree and a reference tree, which is regarded as the gold standard for a given dataset[3]. Finally, the resulting values are used to rank the distance measure, which generated the custom tree, against those already listed in the website database. In this study, we present our results based on the RF distance (while we leave the comparison by means of QD-based performance to future work).

[2] We implemented the TqD software in C++, and made it publicly available at https://github.com/davidecenzato/Threshold_q-gram_distance

[3] The trees are generated using the neighbor-joining algorithm from the EMBOSS package (Rice et al. 2000), and the RF distances are computed using the ETE3 toolkit (Huerta-Cepas et al. 2016).

Briefly, the Robinson–Foulds distance (RF) measures the dissimilarity between two tree topologies with the same number of leaves and the same labels (species) on the leaves. In particular, it measures differences in branching patterns, by considering the set of bipartitions of the leaves of the trees, obtained by removing an edge from each tree. The RF-distance equals the cardinality of the symmetric difference of the two sets of bipartitions, i.e. the number of bipartitions that result from one tree but not from the other. The AFproject also employs the normalized version of this distance, the normalized RF measure (nRF). The nRF standardizes the RF distance, such that its value lies between 0 and 1, with 0 for identical tree topologies and 1 for maximally dissimilar topologies. The nRF is an interesting parameter to report in comparisons, as it shows how far we are from the optimal performance (of obtaining the "ideal" trusted reference tree): the lower the distance from the reference tree, the higher the accuracy of the distance measure.

5.2 Datasets

We tested TqD using the five data sets included in the AFproject phylogeny reconstruction research scenario. These include genomic data belonging to different taxonomic groups: bacteria, animals, and plants. We report a summary of the dataset features in Tables 1 and 2.

We divided the datasets into two groups, one containing assembled sequences and the other containing short reads (or fragmented genomes). The first group contains three complete genome datasets: a mitochondrial DNA (mtDNA) sequences dataset of 25 fish species of the suborder Labroidei (Jeffrey 1990), a DNA sequence dataset of 29 bacterial genomes of E. coli/Shigella, and a DNA sequences dataset of 14 plant genomes. The second group includes two simulated short read datasets derived from the bacterial and plant genomes in the first group. The short reads are generated by using the ART software at seven different coverage levels. Thus, instead of single genomic sequences, here we work with FASTA files, each containing the simulated reads for one specific genome and coverage level. Altogether, the fragmented datasets are organized into seven subfolders, with each subfolder containing a collection of FASTA files corresponding to one genome and coverage level combination. Note that coverage is defined as the average number of reads that align to or cover known reference bases. So, the higher the coverage, the more reads the dataset contains.

5.3 Workflow

In the following, we report the workflow we used to test TqD using the AFproject web service:

Table 1 Features of the three full-genome sequence datasets included in our experimental study

Name	No. sequences	Avg. length	Min. length	Max. length	Best nRF
Labroidei fish	25	16,623	16,441	17,045	0.05
E. coli/Shigella	29	4,895,247	4,369,232	5,528,445	0.04
Plants	14	337,515,688	114,396,853	769,291,188	0.09

From left to right, we report the species name, number of sequences, average sequence length, minimum, and maximum sequence length. In the rightmost column, we report the best accuracy (i.e., the smallest nRF distance) by any distance measure included in the AFproject

1. We downloaded the five datasets (details in Section 5.2) falling under the phylogenetic reconstruction research scenario from the AFproject web page.
2. We compute the pairwise TqD distance matrices for the five datasets by using several combinations of q and t. These matrices contain the TqD distances between all distinct pairs of input sequences. Since fragmented datasets contain collections of short reads instead of complete genomic sequences, we compute one TqD profile for each read collection by summing all q-gram occurrence counters across reads (more details in Sect. 6.2). We output all distance matrices in tab-separated value (TSV) format.
3. For each TqD pairwise distance matrix, we compute the normalized Robinson–Foulds (nRF) distance to the corresponding reference tree by using the same procedure used in the AFproject and described in Sect. 5.1. We then submit the matrix with the best parameter configuration—i.e., that provides the smallest distance—to the AFproject website. This generates a benchmark report as the output of the testing procedure.
   This report provides a list of distance measures, ranked in ascending order based on their RF and nRF values. The list includes all the distance measures whose results have been previously collected on the AFproject website, alongside those of TqD.

6 Results

In this section, we present the results of our experimental analysis carried out by comparing TqD with 61 competing q-gram-based distance measures included in the AFproject. We evaluated TqD using six specific values for the threshold, $t = 0, 1, 2, 3, 6, 14$. Thus, while $t = 0$ (and $t = 1$) only store whether a q-gram appears or not (or whether it appears more than once), larger values store all occurrence counters up to t, and set to $t + 1$ all others. The choice of these six values is motivated by the definition of TqD profile, which

Table 2 Features of the two fragmented genome datasets included in our experimental study.

Name	No. seq	Coverage	No. reads	Read length	Best nRF	Best avg. nRF
E. coli/Shigella	29	0.03125	29,557	150	0.50	0.21 ± 0.14
		0.0625	59,116		0.19	
		0.125	118,266		0.23	
		0.25	236,541		0.08	
		0.50	473,081		0.08	
		1	946,169		0.12	
		5	4,730,778		0.12	
Plants	14	0.015625	48,274	150	0.27	0.14 ± 0.08
		0.03125	96,489		0.18	
		0.0625	1,931,268		0.09	
		0.125	3,862,905		0.18	
		0.25	7,725,928		0.00	
		0.50	15,461,718		0.09	
		1	30,903,727		0.09	

From left to right, we report the species name, number of sequences, coverage level, number of simulated reads, read length, and the best accuracy (i.e., smallest nRF distance) among all distance measures included in the AFproject, for each coverage level. In the rightmost column, we report the lowest average nRF value, taken over the seven coverage levels, by any distance measure included in the AFproject

requires $\log_2(t+1)$ bits to encode each q-gram occurrence count. In particular, the thresholds $t = 6$ and $t = 14$ correspond to the maximum values for which we can represent the occurrence counts using 3 and 4 bits, respectively. Additionally, we also test all small threshold values from zero to three to explore how increasing the profile size affects the accuracy of the distance estimation.

Results are arranged in tables, where each row contains the performance of one competitor expressed in terms of (normalized) Robinson-Foulds distance as well as the input parameters generating the corresponding result. The values presented in the tables are the ones provided in the report produced by the AFproject website when submitting a pairwise distance matrix. It is important to note that multiple input parameter combinations may yield the same best RF value, and the AFproject always reports the configurations associated with the minimum amount of computational resources.

Therefore, we report only the combination that allows to compute TqD with the smallest profile. In particular, we always select the parameter pair with the smallest q value. If multiple pairs have the same q value, we chose the one with the smallest threshold t. We also point out that not all 61 distance measures can be computed on all datasets: this is why Table 3 contains results only for 59 of the 61 distance measures, Table 4 for 57 measures, Table 5 for 51, Table 6 and Table 7 for 41.

To rank all methods, we use the two variables *rank* and *count*. The former provides the relative position of each method in terms of RF value, with rank i indicating that the distance measure has the ith smallest RF value in the table. The variable *count* gives, for each distance measure, how many competitors produced a smaller RF value. We introduced this last variable in order to highlight both the absolute accuracy of each competitor and the number of competitors reaching a certain accuracy.

6.1 Assembled genome datasets

In Tables 3, 4 and 5, we report a summary of the results for the three assembled genome datasets (see Table 1) of *Labroidei fish*, *Escherichia coli*, and *Plant species*, respectively.

Regarding the *L.fish* dataset, we computed the TqD pairwise distance matrices for q values ranging from 7 to 15 and with six different thresholds, $t = 0, 1, 2, 3, 6, 14$. In Table 3, we observe that despite its simplicity, TqD shows the second-lowest RF value recorded on the AFproject website ($RF = 4$), corresponding to the second-best accuracy ($rank = 2$). Additionally, only 16 out of 59 distance measures show better accuracy than TqD ($count = 16$), reaching an RF value of 2. This result is produced by using the smallest q-gram size we tested, $q = 7$, and the smallest threshold, $t = 0$. We further note that most methods performed very well on this dataset. According to AFproject data, 53 of the 59 q-gram-based methods produced query tree topologies that closely matched the reference Labroidei tree (nRF value less than 0.1), with three methods performing with an nRF value greater than 0.5.

One possible explanation for these results is that mitochondrial DNA shows a significantly higher mutation rate compared to nuclear DNA, which makes it easier to identify similarities through q-gram multiplicities. In addition, since these genomes are quite short, they do not lead to large q-gram occurrence counts, thus minimizing the amount of information lost by TqD profiles. Altogether, this suggests that TqD is a good candidate for building phylogenetic trees in agreement with the reference tree of mitochondrial genomes.

Table 3 Ranking of 59 AF distance measures based on q-gram counting

Rank	Count	Name	Parameters	RF	nRF
1	0	AFKS-d2_star	$q = 8$	2.00	0.05
1	0	AFKS-d2z	$q = 8$	2.00	0.05
1	0	AFKS-euclidean_z	$q = 8$	2.00	0.05
1	0	AFKS-n2r	$q = 8$	2.00	0.05
1	0	AFKS-normalized_vectors	$q = 8$	2.00	0.05
1	0	AFKS-pearson	$q = 8$	2.00	0.05
1	0	AFKS-simratio	$q = 8$	2.00	0.05
1	0	alfpy-angle_cos_diss	$q = 9$	2.00	0.05
1	0	alfpy-angle_cos_evol	$q = 8$	2.00	0.05
1	0	alfpy-euclid_norm	$q = 8$	2.00	0.05
1	0	alfpy-euclid_squared	$q = 8$	2.00	0.05
1	0	alfpy-minkowski	$q = 8$	2.00	0.05
1	0	cafe-d2shepp	$q = 10, m = 0$	2.00	0.05
1	0	jD2Stat-D2S	$q = 10$	2.00	0.05
1	0	Mash	$q = 11$	2.00	0.05
1	0	PC-MER	$q = 15$	2.00	0.05
2	**16**	**TqD**	$q = 7, t = 0$	**4.00**	**0.09**
2	16	AF_TFIDF	$q = 8$	4.00	0.09
2	16	FFP	$q = 8$	4.00	0.09
2	16	Skmer	$q = 21$	4.00	0.09
2	16	23 AFKS methods	$7 \leq q \leq 8$	4.00	0.09
2	16	7 alfpy methods	$7 \leq q \leq 9$	4.00	0.09
2	16	cafe-d2star	$q = 7, m = 1$	4.00	0.09
2	16	cafe-cvtree	$q = 9$	4.00	0.09
2	16	CD-MAWS	$q = 9$	4.00	0.09
3	53	jD2Stat-D2St	$q = 10$	6.00	0.14
4	54	jD2Stat-D2n	$q = 10$	8.00	0.18
4	55	AFM_BTkNG_	$q = 10$	14.00	0.32
5	56	alfpy-chebyshev	$q = 9$	28.00	0.64
6	57	AFKS-emd	$q = 5$	36.00	0.82
7	58	AFKS-length_difference	$q = 5$	44.00	1.00

The methods were ranked according to their best RF values for the reference *L. fish* mitochondrial genomes dataset containing 25 sequences of 16,623 average length

As for the *E. coli/Shigella* bacterial dataset, we compute the TqD pairwise distance matrices for q between 7 and 15, and six different thresholds $t = 0, 1, 2, 3, 6, 14$. In Table 4, we observe that TqD reached the fifth-smallest RF value ($rank = 5$), thus placing it in the middle of the ranking. In particular, only 10 distance measures out of 57 show a better accuracy ($count = 10$), and 23 other competitors had a larger RF value. Again, TqD performs best with small input parameter values: $q = 8$ and $t = 1$.

This dataset, unlike the first one consisting of fish mitochondrial genomes, contains long whole genomic DNA sequences. In this scenario, most distance measures reach much lower accuracy; for instance, no competitor shows an nRF smaller than 0.1, and 45 distance measures have an nRF larger than 0.5 (as in the case of TqD). In addition, we observe a more significant performance gap between the top-scoring AF methods (see Mash, Skmer, FFP, and CD-MAWS) and the other competitors. This can be attributed to the increased complexity of evolutionary relationships between bacterial genomes, which is more effectively captured by the tools employing more sophisticated distance measures and leveraging evolutionary models.

Interestingly, TqD shows performance comparable to or better than most of the AF methods reported in Table 4, even if using a very small q value, $q = 8$, and threshold, $t = 1$. It may suggest that, for bacterial genomes, the exact multiplicity of q-grams does not carry much more information than information just about their presence or absence.

If this is the case, it may indicate that, like in Mash, we do not need to keep the complete q-gram profile to compute precise distance measures.

Finally, for the *plant* genomes dataset, we computed the TqD pairwise distance matrices for q between 7 and 15 and six different thresholds $t = 0, 1, 3, 6, 14$. In Table 5,

Table 4 Ranking of 57 AF distance measures based on q-gram counting

Rank	Count	Name	Parameters	RF	nRF
1	0	Mash	$q = 26$	8.00	0.15
1	0	Skmer	$q = 31$	8.00	0.15
2	2	FFP	$q = 21$	12.00	0.23
2	2	CD-MAWS	$q = 16$	12.00	0.23
3	4	AFKS-n2rrc	$q = 9$	22.00	0.42
4	5	AFKS-n2rc	$q = 12$	26.00	0.50
4	5	alfpy-braycurtis	$q = 25$	26.00	0.50
4	5	alfpy-google	$q = 25$	26.00	0.50
4	5	alfpy-manhattan	$q = 25$	26.00	0.50
4	5	alfpy-euclid_squared	$q = 25$	26.00	0.50
5	**10**	**TqD**	$q = 8, t = 1$	**28.00**	**0.54**
5	10	13 AFKS distance measures	$9 \leq k \leq 12$	28.00	0.54
5	10	5 alfpy distance measures	$8 \leq k \leq 29$	28.00	0.54
5	10	cafe-d2star	$q = 12, m = 0$	28.00	0.54
5	10	3 jD2Stat distance measures	$8 \leq q \leq 14$	28.00	0.54
6	33	15 AFKS distance measures	$q = 9$	30.00	0.58
6	33	alfpy-euclid_norm	$q = 9$	30.00	0.58
6	33	alfpy-minkowski	$q = 9$	30.00	0.58
6	33	cafe-cvtree	$q = 12$	30.00	0.58
6	33	cafe-d2sheep	$q = 12, m = 0$	30.00	0.58
7	52	alfpy-kld	$q = 19$	36.00	0.69
8	53	alfpy-chebyshev	$q = 9$	38.00	0.73
9	54	AFKS-afd	$q = 9$	46.00	0.88
10	55	AFKS-emd	$q = 9$	48.00	0.92
10	56	AFKS-length_difference	$q = 9$	48.00	0.92

The methods were ranked according to the best RF value for the *E. coli/Shigella* dataset containing 29 whole bacterial genomes of 4,895,247 average length

we notice that TqD reaches the fifth-smallest RF value ($rank = 5$), thus again placing it in the middle of the ranking. However, here, only 9 out of 51 competitors showed a higher accuracy ($count = 9$), while 30 other competitors reached a worse RF value. This shows that, on this dataset, TqD performed quite well compared to most of the AF methods, placing in the top part of the ranking.

Again, we observe larger RF values than for the first dataset (the fish sequences); in particular, more than half of all distance measures reach an nRF larger than 0.5, and only one (the Mash distance) has an nRF smaller than 0.1. In this case, TqD reaches its best performance with a higher q-gram length, $q = 13$, and threshold, $t = 0$. This is possibly because plant genomes share larger repetitive regions than bacterial genomes; thus, we need longer q-grams to correctly highlight differences between the genomes. Plant genomes are also the largest sequences included in the AF project datasets, making small q-gram occurrence counters less meaningful due to high redundancy and over-representation of small patterns across the sequences. This suggests that TqD requires larger q values to be effective when working with very long genomes, such as plant genomes.

In Fig. 3, we further analyze the performance of TqD by comparing the nRF values obtained using different parameter configurations for the *E. coli/Shigella* and *plants* datasets. Specifically, we present two heatmaps where the values of q and t are shown on the x- and y-axes, respectively, while the color intensity represents the corresponding nRF value in a scale ranging from the smallest measured value to 1. We observe a consistent trend across both datasets: when the q parameter is too small, $q < 8$ for *E. coli/Shigella* and $q < 13$ for plant genomes, all parameter configurations result in poor performance, as indicated by high color intensities (i.e., high nRF values). Interestingly, for larger values of q, smaller threshold values do not degrade performance. In fact, we consistently get the best results using the three smallest threshold values, suggesting that low threshold values are sufficient and, in some cases, preferable when q is chosen appropriately.

6.2 Fragmented genome datasets

In Tables 6 and 7, we give a summary of the results for the two fragmented genome datasets consisting of short reads of *E. coli* and plant genomes. We computed the TqD pairwise

Table 5 Ranking of 51 AF distance measures based on q-gram counting

Rank	Count	Name	Parameters	RF	nRF
1	0	Mash	$q = 26$	2.00	0.09
2	1	Skmer	$q = 31$	4.00	0.18
2	1	cafe-d2shepp	$q = 10, m = 7$	4.00	0.18
2	1	FFP	$q = 15$	4.00	0.18
3	4	cafe-cvtree	$q = 10$	6.00	0.27
3	4	AFKS-rre_k_r	$q = 13$	6.00	0.27
3	4	cafe-d2star	$q = 9, m = 7$	6.00	0.27
4	7	AFKS-kl_conditional	$q = 12$	8.00	0.36
4	7	alfpy-jsd	$q = 11$	8.00	0.36
5	**9**	**TqD**	$q = 13, t = 0$	**10.00**	**0.45**
5	9	7 AFKS distance measures	$12 \leq q \leq 13$	10.00	0.45
5	9	4 alfpy distance measures	$12 \leq q \leq 13$	10.00	0.45
6	21	AFKS-k_divergence	$q = 12$	12.00	0.55
7	22	17 AFKS distance measures	$q = 12$	14.00	0.64
8	39	AFKS-markov	$q = 13$	16.00	0.73
8	39	3 alfpy distance measures	$q = 6$	16.00	0.73
9	43	AFKS-length_difference	$q = 12$	18.00	0.82
9	43	AFKS-normalized_vectors	$q = 13$	18.00	0.82
9	43	3 alfpy distance measures	$q = 6$	18.00	0.82
10	48	AFKS-simratio	$q = 12$	20.00	0.91
10	48	alfpy-chebyshev	$q = 6$	20.00	0.91
10	48	alfpy-kld	$q = 6$	20.00	0.91

The methods were ranked according to the best RF value for the reference *plants* dataset containing 14 genomes of 337,515,688 average length

Table 6 Ranking of 41 AF distance measures based on q-gram counting

Rank	Count	Name	Parameters	avg. RF	avg. nRF
1	0	Mash	$q = 21, 25, 27, 19, 14, 24, 24$	14.29	0.27 ± 0.18
2	1	AAF	$q = 23, 16, 23, 17, 15, 20, 17$	17.71	0.34 ± 0.19
2	1	Skmer	$q = 31, 31, 31, 31, 31, 31, 31$	17.71	0.34 ± 0.19
3	**3**	**TqD**	$(q, t) = (10, 1), (15, 0), (10, 1), (13, 6), (15, 14), (14, 14), (14, 1)$	**21.43**	**0.41 ± 0.20**
4	4	FFP	$q = 10, 12, 12, 10, 12, 12, 16$	22.29	0.43 ± 0.16
5	5	cafe-cvtree	$q = 12, 12, 12, 12, 12, 12, 12$	22.86	0.44 ± 0.14
5	5	cafe-d2sheep	$(q, m) = (12, 0), (12, 0), (12, 0), (12, 0), (12, 0), (12, 0), (13, 0)$	22.86	0.44 ± 0.16
6	7	AFKS-spearman	$q = 9, 9, 9, 8, 9, 9, 8$	24.00	0.46 ± 0.22
≥ 7	≥ 9	8 AFKS methods	$6 \leq q \leq 9$	≥ 24.57	$\geq 0.47 \pm 0.20$
14	17	kWIP	$q = 9, 10, 19, 10, 25, 10, 18$	27.43	0.53 ± 0.15
≥ 15	≥ 18	10 AFKS methods	$6 \leq q \leq 9$	≥ 28.29	$\geq 0.54 \pm 0.18$
25	28	cafe-d2star	$(q, m) = (12, 0), (13, 0), (12, 1), (12, 0), (13, 1), (13, 0), (13, 0)$	30.00	0.58 ± 0.13
≥ 26	≥ 29	11 AFKS methods	$6 \leq q \leq 9$	≥ 31.14	$\geq 0.60 \pm 0.13$
32	40	AFKS-emd	$q = 6, 6, 6, 6, 6, 6, 6$	48.00	0.92 ± 0.00

The methods were ranked according to the best average RF value for the fragmented *E. coli/Shigella* datasets. We report seven input parameter configurations, one for each coverage level. See Table 2 for all dataset features

matrices with several combinations of q and t values for all seven sub-datasets corresponding to the different coverage levels. We also report the average of the best RF and nRF values over the seven sub-datasets.

Note that, unlike in Sect. 6.1, where we have with complete genomic sequences, this setting requires computing TqD for string collections: these are the simulated reads of the original genomes, rather than single sequences. Since TqD is not originally designed to work with string

Table 7 Ranking of 41 AF distance measures based on q-mer counting

Rank	Count	Name	Parameters	Avg. RF	Avg. nRF
1	0	Mash	$q = 17, 20, 20, 19, 25, 24, 20$	3.14	0.14 ± 0.08
2	1	Skmer	$q = 31, 31, 31, 31, 31, 31, 31$	5.43	0.24 ± 0.09
3	2	cafe-d2shepp	$(q, m) = (9, 7), (10, 8), (9, 7),$ (9, 7), (9, 7), (10, 7), (10, 7)	6.29	0.29 ± 0.17
4	3	cafe-d2star	$(q, m) = (11, 7), (10, 8), (9, 7),$ (9, 7), (9, 7), (9, 7), (9, 7)	7.14	0.32 ± 0.12
5	4	AAF	$q = 18, 22, 22, 22, 22, 28, 25$	8.29	0.37 ± 0.08
6	5	AFKS-rre_k_r	$q = 7, 7, 7, 8, 7, 7, 8$	8.86	0.40 ± 0.11
7	6	kWIP	$q = 10, 10, 14, 10, 15, 14, 15$	9.14	0.42 ± 0.14
7	6	cafe-cvtree	$q = 9, 9, 10, 11, 10, 10, 10$	9.14	0.42 ± 0.23
8	8	FFP	$q = 15, 14, 10, 11, 10, 14, 14$	9.43	0.43 ± 0.08
9	9	AFKS-kl_condition	$q = 8, 8, 7, 8, 7, 7, 7$	10.29	0.47 ± 0.11
10	10	AFKS-k_divergence	$q = 7, 8, 7, 8, 7, 8, 7$	11.14	0.51 ± 0.07
11	11	AFKS-spearman	$q = 7, 7, 7, 8, 7, 7, 7$	11.43	0.52 ± 0.11
12	**12**	**TqD**	$(q, t) = (12, 0), (13, 0), (10, 6),$ (15, 0), (15, 0), (14, 0), (13, 0)	**11.71**	**0.53 ± 0.09**
12	12	AFKS-jensen_shan	$q = 7, 8, 7, 8, 7, 8, 7$	11.71	0.53 ± 0.17
≥ 13	≥ 14	26 AFKS methods	$7 \leq q \leq 8$	≥ 12.00	$\geq 0.55 \pm 0.07$
25	39	AFKS-emd	$q = 7, 7, 7, 7, 7, 7, 7$	18.00	0.82 ± 0.00
25	39	AFKS-length_diff	$q = 7, 7, 7, 7, 7, 7, 7$	18.00	0.82 ± 0.00

The methods were ranked according to the best average RF value for the fragmented plant datasets. We report seven input parameter configurations, one for each coverage level. See Table 2 for all dataset features

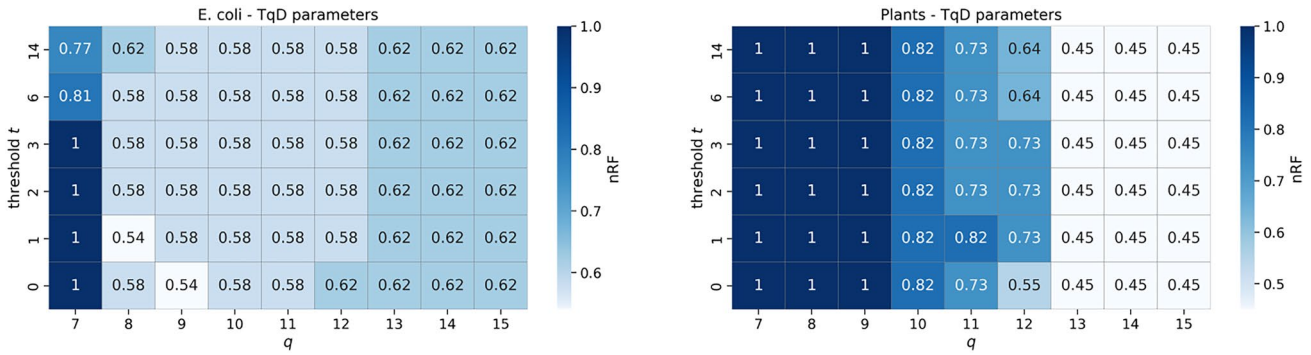

Fig. 3 Normalized Robinson-Foulds (nRF) distances computed using different q and t parameters configurations for *E. coli/Shigella* and *plants* datasets

collections, we implement a simple workaround by generalizing the way we compute the TqD profiles: we construct one single TqD profile for each read collection by summing up all occurrence counters across the reads. Then, given these generalized profiles, we calculate the pairwise TqD values as in the case of individual sequences.

As for the *E. coli* bacterial fragmented dataset, we compute the TqD pairwise distance matrices for q between 7 and 15 and six different thresholds $t = 0, 1, 2, 3, 6, 14$. In Table 6, we note that TqD reaches the third-smallest average RF value (21.43) with a standard deviation of 10.31 ($avg.\ nRF = 0.41 \pm 0.20$). This result ranks T$q$D among the best-performing distance measures for this dataset ($rank = 3$). In particular, only 3 of 41 competitors show a better average accuracy ($count = 3$), while the other 37 competitors had a larger average RF value. The seven input parameter configurations that allow TqD to reach the best result with the minimal computational cost are $(q, t) = (10, 1), (15, 0), (10, 1), (13, 6), (15, 14), (14, 14), (14, 1)$, so higher coverage levels require larger q values. In contrast to the assembled genomes, we needed larger thresholds to obtain the best results with the smallest profile sizes. Specifically, for coverage levels of 0.5 and 1, we obtain the best nRF with $t = 14$. Importantly, we observed that a similar result can be reached by using the two smallest thresholds only: $t = 0$ and $t = 1$. The parameter combinations $(q, t) = (10, 1), (10, 1), (10, 1), (14, 0), (14, 0), (15, 0), (14, 1)$ yield an nRF of 22.00, which is only slightly larger than the best result.

Interestingly, TqD exhibits significantly better performance on fragmented *E. coli* datasets than on complete genomes. The same trend is also shared by other AF methods included in the cafe and AFKS tools (such as cafe-cvtree, AFKS-spearman, and others). One possible explanation for this is that for coverage larger than 1, several q-grams will be overrepresented due to overlapping reads. Unlike other methods relying on classic q-gram profiles, TqD is less affected by this issue because it does not use the exact q-gram multiplicities, thereby mitigating the impact of redundant information. Another explanation could be sequencing errors introduced during the creation of the datasets, which modify the exact q-gram counts distribution, to which again TqD is less sensitive than the other methods. It is also worth noting that in this setting, TqD needs larger q values for all coverage levels to reach its best results than for the complete genomes. We do not have an explanation for this latter phenomenon at present.

Finally, for the plant fragmented dataset, we computed the TqD pairwise distance matrices for q between 7 and 15 and the six different thresholds $t = 0, 1, 2, 3, 6, 14$. In Table 7, we observe that TqD reaches the thirteen-smallest average RF value (11.71) and a standard deviation of 2.07 ($avg.\ nRF = 0.53 \pm 0.09$). This result places T$q$D in the middle of the ranking ($rank = 12$).

However, despite this ranking, when looking at the *count* column, TqD produces a higher average accuracy than 25 out of 40 competitors, thus still placing in the top half of the table.

The seven input parameter configurations that allow TqD to reach the best result with the smallest profile sizes are $(q, t) =$ (12, 0), (13, 0), (10, 6), (15, 0), (15, 0), (14, 0), (13 , 0), so all coverage (except coverage 0.625) require large q values. Also in this case, the two smallest threshold values, $t = 0, 1$, are sufficient to achieve the best RF value for all seven coverage levels. Indeed, using $(q, t) = (11, 1)$ instead of (10, 6) for coverage 0.0625 produces the same average nRF result as reported in Table 7.

In contrast to the fragmented bacterial datasets, the results for plants align more closely with those observed for the complete genomes. In particular, the average RF value is slightly higher than the RF observed for the assembled genomes, primarily due to the penalty introduced by datasets with low coverage. Notably, the individual RF values for all coverage levels (data not shown) never exceed the best one computed for the complete genomes (RF = 10.00), highlighting the consistency of the results across different coverage levels. Unfortunately, in this analysis, we could not test coverage levels greater than 1, which limits our ability to fully evaluate the impact of higher coverage on TqD performance. Future experiments with broader coverage ranges would help provide deeper insights into this topic.

6.3 Discussion

To sum up, we compared TqD with 61 AF distance measures based on q-mer counting using the AFproject web service. We used TqD to compute the pairwise distances for five genomic datasets (including both assembled and fragmented datasets) and compared the accuracy of the phylogenetic tree computed on the basis of these distance matrices. Even though TqD profiles retain only a small part of all information contained in the original q-gram profiles, it turns out to be always competitive with the other distance measures included in the AFproject. In particular, TqD always ranked in the upper half of the tables, and in two cases, for the *L. Fish* mitochondrial sequences and *E. coli/Shigella* fragmented datasets (Tables 3 and 6), it was among the best-performing distance measures. Interestingly, this includes the long genomic sequences datasets of *E. coli* and *plants*, where the amount of information lost by TqD profiles is even more pronounced.

In addition, we observe that in almost all cases, we obtained our best accuracy using the smallest threshold values, namely $t = 1$ resp. $t = 0$. With $t = 1$, the TqD profiles contain the information whether q-grams appear not at all, once, or multiple times, while for $t = 0$, they contain information only on the presence or absence of q-grams, i.e., on the *sets* of q-grams of the input sequences, without frequencies. Both settings allow minimizing the memory footprint, thus enabling TqD to scale to even larger datasets. This finding further suggests that the complete information about q-gram frequencies is not always essential when estimating the evolutionary relationships among species. This is consistent with the results of established tools such as Mash (Ondov et al. 2016) and Skmer (Sarmashghi et al. 2019). These methods consistently rank among the top performers, while being based on computing the Jaccard distance between q-gram *sets*: neither uses information about the q-gram frequencies directly for estimating genomic distances.

It is interesting to note that using the parameter $t = 0$ is equivalent to considering strings simply as sets of q-grams and computing the cardinality of their symmetric difference. To the best of our knowledge, this method has not been used before as a string distance. Indeed, existing tools that rely only on presence or absence of q-grams, such as Mash and Skmer, do also use information about the frequency of q-grams: Mash for filtering, and Skmer for inferring sequencing errors and coverage. Presence and absence of q-grams has also been used for prediction of bacterial resistance (Mahé and Tournoud 2018) and SNP-level relatedness and ribotypes in *C. difficile* (Moore et al. 2022), however, only a small subset of existing q-grams are actually used for the prediction.

Similarly, the variant with $t = 1$, which distinguishes between q-grams that occur once, repeated q-grams, and absent q-grams, has not appeared in the literature before, even though q-gram based approaches in bioinfomatics abound, see e.g. the recent survey (Jenike et al. 2025).

As a further point, our experiments give evidence that TqD is a very flexible measure, since its performance is consistent among all datasets we tested, including the read datasets. All the above suggest that TqD is a valuable alternative to the classic q-gram distance and several other AF distance measures included in the AFproject. It is furthermore easy to compute and can run on large datasets, making it ideal as a preprocessing step before more computationally intensive string alignment tasks.

7 Conclusion

In this paper, we introduced the threshold q-gram distance (TqD), a novel alignment-free measure of strings similarity, firstly and thoroughly analysed in the master thesis (Milanese 2015). TqD is a variant of the well-known q-gram distance, which employs a modified q-gram profile using a threshold to cap the maximum size for the q-gram multiplicities. This new feature allows to save storage space in several practical contexts, while the computation time remains linear (same as the original q-gram distance). Additionally, we introduced the concept of threshold graphs, a generalization of De Bruijn graphs, which helps to group and visualize sequences with the same threshold q-gram profile.

An experimental evaluation of TqD allows us to work on a wide selection of datasets, including genomes from different species present in nature, such as bacteria and plants, and covering a wide range of sequence lengths, from the smallest mitochondria to the longest plant genomes. We also included two fragmented genome datasets of short reads with several coverage levels. We compared TqD to several AF distance measures based on q-gram counting, using the AFproject web platform. In particular, we ran TqD under different parameter settings to compute the pairwise distance matrices for all datasets, which were then uploaded to the AFproject webpage. Our results show that TqD was always competitive when performing phylogenetic reconstruction; in fact, it consistently ranked in the top half of the tables, and in two cases, it ranked among the best scoring methods (on *L. Fish* mitochondrial sequences and *E. coli/Shigella* fragmented datasets).

We also note that, in most cases, TqD achieved its best performance using the smallest threshold values ($t = 0, 1, 2$). This is particularly relevant in practice since, in these cases, TqD only retains a small part of all information of the original q-gram profile. This allows TqD to keep the memory footprint low, which is critical when working on large genomic datasets or using a computer with low hardware specifications or limited storage space. This suggests that TqD could be efficiently incorporated in many bioinformatics routines, where it is necessary to implement a preprocessing step to filter out dissimilar sequences before a heavier string alignment process. In addition, TqD is easy to understand and straightforward to implement, opening up to several optimizations, such as, for example, the possibility to work with threshold q-gram profiles with limited dimensions, given by the cardinality of q-grams occurring in all (or in a large part of) the genomes in a given dataset. By setting $t = 1$, in any pair of strings, TqD counts the differences in uniquely occurring q-grams, repeated q-grams, and absent q-grams.

For future experiments, we have a two-fold objective. On the one hand, we plan to continue and complete this comparison with other q-gram based methods of the AFproject also in terms of nQD (the normalized Quartet Distance). On the other hand, we plan to work on eukaryotes, to compare TqD performance on genome datasets from organisms not so distant in terms of size and evolutionary scale.

A future research direction is to investigate the exact relationship of the TqD distance for different choices of q and t, namely, by applying the algorithm introduced in Haoze et al. (2025) to q-gram distributions limited by the t value, and visualizing our De Bruijn threshold graphs as Chaos Game Representation images.

More ambitious applications could be considered for taxonomic classification of emerging astroviruses, namely, by including TqD as an alternative string similarity measure in the pipeline, in order to counteract the impact of genetic recombination on viral classification proposed in Alipour et al. (2024), or to identify microsatellite instability in cancer cells (Baudrin et al. 2018).

Appendix: Implementation details

We report here some supplementary information, about the code implementation details (developed by Davide Cenzato, from a prior original version by Alessio Milanese).

We tested TqD using such a workflow on selected datasets by computing the pairwise distance matrices to upload (in TSV format) to AFproject webpage. The script takes in input the length of the q-mers, a specific threshold t, and the path to the folder containing the FASTA files. It first computes the TqD profile for each input sequence and stores all profiles on the disk. Then, it iteratively loads one profile pair at a time to calculate the pairwise TqD values by using a linear time scan procedure. This approach optimizes the total running time by avoiding computing the same TqD profiles

multiple times. The final output is a tab-separated values (TSV) file that lists all pairwise sequence comparisons for a given combination of q and t.

We implemented the TqD profiles using static Elias-Fano compressed dictionaries in C++, a compact data structure supporting fast lookup over a list of monotonically increasing integer keys. In particular, we use the q-gram ranks as the keys of the dictionaries and the occurrence counters as the values associated with the keys. In addition, our implementation enables the storage of a single profile to compute TqD for various t values. In fact, given a TqD profile with a threshold of $t = n$, it is possible to derive all profiles with thresholds $t = n'$, where $n' < n$, without having to recompute them from scratch. This is because any occurrence count $c \leq n'$ will remain unchanged in both profiles, while for counts greater than n', we can simply store them as $n' + 1$.

We also provide an alternative Python 3 profile implementation using NumPy arrays and Python dictionaries.

We do not include any extra DNA characters in the profiles (only the four standard DNA characters are allowed: A, C, G, T); thus, we preprocess the input sequences to remove all q-grams that contain the extra characters.

Author contributions All the authors contributed equally to this work.

Funding Open access funding provided by Università degli Studi di Verona within the CRUI-CARE Agreement. Zs.L. is partially funded by the Italian Ministry of University and Research (MUR) PRIN Project PINC, Pangenome INformatiCs: from Theory to Applications (Grant No. 2022YRB97K), and by the INdAM - GNCS Project CUP E53C24001950001. D.C. is funded by ERC StG "REGINDEX: Compressed indexes for regular languages with applications to computational pan-genomics" grant nr 101039208. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

Data availability Data are provided within the manuscript and supplementary information files.

Declarations

Competing interests The authors declare no competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

Akbari Rokn Abadi S, Mohammadi A, Koohi S (2024) PC-mer: an ultra-fast memory-efficient tool for metagenomics profiling and classification. PLoS One 19(8):0307279. https://doi.org/10.1371/journal.pone.0307279

Alipour F, Holmes C, Lu YY, Hill K, Kari L (2024) Leveraging machine learning for taxonomic classification of emerging astroviruses. Front Mol Biosci 10:1305506. https://doi.org/10.3389/fmolb.2023.1305506

Amir A, Levy A (2010) String rearrangement metrics: a survey. In: al., T.E. (ed.) Ukkonen Festschrift. LNCS, vol. 6060, pp. 1–33. https://doi.org/10.1007/978-3-642-12476-1_1

Anjum N, Nabil RL, Rafi RI, Bayzid MS, Rahman MS (2023) CD-MAWS: an alignment-free phylogeny estimation method using cosine distance on minimal absent word sets. IEEE ACM Trans Comput Biol Bioinform 20(1):196–205. https://doi.org/10.1109/TCBB.2021.3136792

Arias PM, Hill KA, Kari L (2023) iDeLUCS: a deep learning interactive tool for alignment-free clustering of DNA sequences. Bioinformatics 39(9):508. https://doi.org/10.1093/bioinformatics/btad508

Backurs A, Indyk P (2018) Edit distance cannot be computed in strongly subquadratic time (unless SETH is false). SIAM J Comput 47(3):1087–1097. https://doi.org/10.1145/2746539.2746612

Baudrin LG, Deleuze J-F, How-Kit A (2018) Molecular and computational methods for the detection of microsatellite instability in cancer. Front Oncol 8:621. https://doi.org/10.3389/fonc.2018.00621

Bruijn NG (1946) A combinatorial problem. Proc Sect Sci 49(7):758–764

Burkhardt S, Crauser A, Ferragina P, Lenhof H, Rivals E, Vingron M (1999) q-gram based database searching using a suffix array (QUASAR). In: Proceedings of the Third Annual International Conference on Research in Computational Molecular Biology, RECOMB 1999, pp. 77–83. https://doi.org/10.1145/299432.299460

Castellini A, Franco G, Milanese A (2015) A genome analysis based on repeat sharing gene networks. Nat Comput 14(3):403–420. https://doi.org/10.1007/S11047-014-9437-6

Chan CX, Bernard G, Poirion O, Hogan J, Ragan M (2014) Inferring phylogenies of evolving sequences without multiple sequence alignment. Sci Rep 4:6504. https://doi.org/10.1038/srep06504

Delibaş E (2025) Efficient TF-IDF method for alignment-free DNA sequence similarity analysis. J Mol Graph Model 137:109011. https://doi.org/10.1016/j.jmgm.2025.109011

Delibaş E, Arslan A, Şeker A, Diri B (2020) A novel alignment-free DNA sequence similarity analysis approach based on top-kn -gram match-up. J Mol Graph Model 100:107693. https://doi.org/10.1016/j.jmgm.2020.107693

Estabrook GF, McMorris FR, Meacham CA (1985) Comparison of undirected phylogenetic trees based on subtrees of four evolutionary units. Syst Zool 34(2):193–200. https://doi.org/10.2307/2413326

Fan H, Ives A, Surget-Groba Y, Cannon C (2015) An assembly and alignment-free method of phylogeny reconstruction from next-generation sequencing data. BMC Genomics 16:522. https://doi.org/10.1186/s12864-015-1647-5

Franco G, Milanese A (2013) An investigation on genomic repeats. In: The Nature of Computation. Logic, Algorithms, Applications - Proceedings of the 9th Conference on Computability in Europe, CiE 2013. LNCS, vol. 7921, pp. 149–160. https://doi.org/10.1007/978-3-642-39053-1_18

Hanada H, Kudo M, Nakamura A (2014) Average-case linear-time similar substring searching by the q-gram distance. Theoret Comput Sci 530:23–41. https://doi.org/10.1016/j.tcs.2014.02.022

Haoze H, Kari L, Millan AP (2025) Bridging Chaos Game Representations and k-mer Frequencies of DNA Sequences. eprint arXiv:2506.22172. https://doi.org/10.48550/arXiv.2506.22172

Hermelin D, Landau GM, Landau S, Weimann O (2013) Unified compression-based acceleration of edit-distance computation. Algorithmica 65(2):339–353. https://doi.org/10.1007/s00453-011-9590-6

Huerta-Cepas J, Serra F, Bork P (2016) Ete 3: reconstruction, analysis, and visualization of phylogenomic data. Mol Biol Evol 33(6):1635–1638. https://doi.org/10.1093/molbev/msw046

Jeffrey HJ (1990) Chaos game representation of gene structure. Nucleic Acids Res 18(8):2163–2170. https://doi.org/10.1093/nar/18.8.2163

Jenike KM, Campos-Domíngez L, Boddé M, Cerca J, Hodson CN, Schatz MC, Jaron KS (2025) k-mer approaches for biodiversity genomics. Genome Res 35:219–230. https://doi.org/10.1101/gr.279452.124

Levenshtein V (1966) Binary codes capable of correcting deletions, insertions, and reversals. Sov Phys-Dokl 10(8):707–710

Lint JH, Wilson RM (2001) A Course in Combinatorics, 2nd edn. Cambridge University Press, Cambridge. https://doi.org/10.1017/CBO9780511987045

Lu YY, Tang K, Ren J, Fuhrman JA, Waterman MS, Sun F (2017) Cafe: accelerated alignment-free sequence analysis. Nucleic Acids Res 45(Webserver Issue):554–559. https://doi.org/10.1093/nar/gkx351

Luczak BB, James BT, Girgis HZ (2019) A survey and evaluations of histogram-based statistics in alignment-free sequence comparison. Brief Bioinform 20(4):1222–1237. https://doi.org/10.1093/bib/bbx161

Mahé P, Tournoud M (2018) Predicting bacterial resistance from whole-genome sequences using k-mers and stability selection. BMC Bioinformatics 19:383. https://doi.org/10.1186/s12859-018-2403-z

Mantaci S, Restivo A, Sciortino M (2008) Distance measures for biological sequences: some recent approaches. Int J Approx Reason 47(1):109–124. https://doi.org/10.1016/j.ijar.2007.03.011

Marco-Sola S, Moure JC, Moreto M, Espinosa A (2020) Fast gap-affine pairwise alignment using the wavefront algorithm. Bioinformatics 37(4):456–463. https://doi.org/10.1093/bioinformatics/btaa777

Milanese A (2015) On a new distance measure for genomic repeat discovery. Master's thesis, University of Verona

Moore MP, Wilcox MH, Walker AS, Eyre DW (2022) K-mer based prediction of *Clostridioides difficile* relatedness and ribotypes. Microb Genom 8:000804. https://doi.org/10.1099/mgen.0.000804

Murray KD, Webers C, Ong CS, Borevitz JO, Warthmann N (2017) kWIP: The k-mer weighted inner product, a de novo estimator of genetic similarity. PLoS Comput Biol 13(9):e1005727. https://doi.org/10.1371/journal.pcbi.1005727

Myers EW (1986) An $O(ND)$ difference algorithm and its variations. Algorithmica 1(1):251–266. https://doi.org/10.1007/BF01840446

Myers G (1999) A fast bit-vector algorithm for approximate string matching based on dynamic programming. J ACM 46(3):395–415. https://doi.org/10.1145/316542.316550

Needleman SB, Wunsch CD (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. J Mol Biol 48(3):443–453. https://doi.org/10.1016/0022-2836(70)90057-4

Ondov B, Treangen T, Melsted P, Mallonee A, Bergman N, Koren S, Phillippy A (2016) Mash: Fast genome and metagenome distance estimation using MinHash. Genome Biol 17(1):132. https://doi.org/10.1186/s13059-016-0997-x

Pevzner PA (1995) DNA physical mapping and alternating Eulerian cycles in colored graphs. Algorithmica 13:77–105. https://doi.org/10.1007/BF01188582

Pevzner PA (1989) ℓ-tuple DNA sequencing: Computer analysis. J Biomol Struct Dyn 7(1):63–73. https://doi.org/10.1080/07391102.1989.10507752

Rasmussen KR, Stoye J, Myers EW (2006) Efficient q-gram filters for finding all epsilon-matches over a given length. J Comput Biol 13(2):296–308. https://doi.org/10.1089/CMB.2006.13.296

Rice P, Longden I, Bleasby A (2000) EMBOSS: The European molecular biology open software suite. Trends Genet 16:276–7. https://doi.org/10.1016/s0168-9525(00)02024-2

Robinson DF, Foulds LR (1981) Comparison of phylogenetic trees. Math Biosci 53(1):131–147. https://doi.org/10.1016/0025-5564(81)90043-2

Sarmashghi S, Bohmann K, Gilbert M, Bafna V, Mirarab S (2019) Skmer: assembly-free and alignment-free sample identification using genome skims. Genome Biol 20(1):34. https://doi.org/10.1186/s13059-019-1632-4

Sims GE, Jun S, Wu GA, Kim S (2009) Alignment-free genome comparison with feature frequency profiles (FFP) and optimal resolutions. Proc Natl Acad Sci U S A 106(8):2677–2682. https://doi.org/10.1073/pnas.0813249106

Stephens ZD, Lee SY, Faghri F, Campbell RH, Zhai C, Efron MJ, Iyer R, Schatz MC, Sinha S, Robinson GE (2015) Big data: Astronomical or Genomical? PLOS Biology, 1–11 https://doi.org/10.1371/journal.pbio.1002195

Swain MT, Vickers M (2022) Interpreting alignment-free sequence comparison: what makes a score a good score? NAR Genom Bioinform 4:lqac062. https://doi.org/10.1093/nargab/lqac062

Ukkonen E (1985) Algorithms for approximate string matching. Inf Control 64(1):100–118. https://doi.org/10.1016/S0019-9958(85)80046-2

Ukkonen E (1992) Approximate string matching with q-grams and maximal matches. Theoret Comput Sci 92(1):191–211. https://doi.org/10.1016/0304-3975(92)90143-4

Vinga S, Almeida JS (2003) Alignment-free sequence comparison–a review. Bioinformatics 19(4):513–523. https://doi.org/10.1093/bioinformatics/btg005

Zerbino DR, Birney E (2008) Velvet: algorithms for de novo short read assembly using de bruijn graphs. Genome Res 18(5):821–829. https://doi.org/10.1101/gr.074492.107

Zielezinski A, Vinga S, Almeida J, Karlowski W (2017) Alignment-free sequence comparison: benefits, applications, and tools. Genome Biol 18:186. https://doi.org/10.1186/s13059-017-1319-7

Zielezinski A, Girgis H, Bernard G, Leimeister C-A, Tang K, Dencker T, Lau A, Röhling S, Choi J, Waterman M, Comin M, Kim S-H, Vinga S, Almeida J, Chan CX, James B, Sun F, Morgenstern B, Karlowski W (2019) Benchmarking of alignment-free sequence comparison methods. Genome Biol 20:144. https://doi.org/10.1186/s13059-019-1755-7

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.