An Algorithm to Calculate the p-value of the Monge-Elkan Distance *

Petr Ryšavý[0000−0002−6597−6616] and Filip Železný[0000−0001−9780−3376]

Department of Computer Science,
Faculty of Electrical Engineering, Czech Technical University in Prague,
Prague, Czech Republic
{petr.rysavy,zelezny}@fel.cvut.cz

Abstract. The Monge-Elkan distance is a straightforward yet popular distance measure used to estimate mutual similarity of two sets of objects. It was initially proposed in the field of databases and it found broad usage in other fields. Nowadays, it is especially relevant to analysis of new-generation sequencing data: the Monge-Elkan distance of two read bags sequenced from two respective sequences is an estimate of the dissimilarity of the sequences. In this paper, we provide an algorithm to calculate the p-value for the Monge-Elkan distance. Given the object-level null distribution, i.e., the distribution of distances between i.i.d.-sampled objects (e.g., reads), the method yields the null distribution of the Monge-Elkan distance, which in turn allows for the calculation of the p-value. We also demonstrate an application on sequencing data, using the Levenshtein distance on the object (i.e., read) level.

Keywords: Monge-Elkan distance · p-value · null distribution

1 Introduction

The Monge-Elkan similarity was proposed in [14]. The paper used the concept to solve the *field matching problem*, i.e., the problem of deciding whether two different fields (say, strings) represent the same entity. For example, *John Doe* and *Doe, John* are likely to represent the same person despite different textual representations. The field matching problem often arises in databases when multiple data sources are combined into a single one.

The paper proposed a simple yet effective recursive algorithm. Denote the fields R_A and R_B. Each of the fields is broken into subfields (say, at positions of spaces), and for each subfield a of R_A, the most similar subfield in R_B is assigned. The similarities between such established pairs of subfields are then averaged to produce the value of similarity between the fields.

Due to the simplicity of the idea, it has found its way into many applications, especially in the approximate string matching field and related tasks as name matching [4], information extraction from health records [9], title matching [6],

* CZ.02.1.01/0.0/0.0/16_019/0000765 "Research Center for Informatics"

business process model matching [1], or toponym matching in geography [18]. There are also generalizations of the original paper as [8].

The Monge-Elkan similarity found its way into bioinformatics as well. The studies [15, 17] used the Monge-Elkan similarity to develop a distance measure applied in analyzing raw read data. Instead of fields in database entries, R_A and R_B represented bags of reads produced by a sequencing machine for two sequences A and B, i.e., the elements of those bags are individual reads. The Monge-Elkan distance then approximates the Levenshtein distance [11] between the sequences A and B without the need for assembly of read data.

A main challenge in the outlined biological use case is the interpretation of the computed distance values. Relative comparisons are not problematic: knowing that the read bag distance is, say, 10 means that the original sequences are more similar than if the distance was 20. However, we also want to be able to establish whether a particular computed distance between two read bags indicates similar input sequences rather than random unrelated sequences. As customary in life sciences, we make the former call if the probability of the latter case, i.e., the *p-value*, is smaller than a threshold (e.g., 0.05). Determining the p-value of the Monke-Elkan distance is however not trivial.

The main contribution of this paper is a theoretic calculation of the p-value for the Monge-Elkan distance. First, we will provide an algorithm for the general version, then apply it to the approach presented in [15, 17], and then we will discuss drawbacks, limitations, and necessary approximations that were done during the calculation. A minor limitation of the presented algorithm is that the similarity between the subfields (reads) must have only a finite set of outcomes, which is true under commonly used measures with limited subfield length. This includes the Levenshtein distance [11], the longest-common-subsequence similarity [19], Jaro-Winkler similarity [20], and others. The restriction to finite domains, together with independence assumptions, will allow us to exploit the probability generating functions, well-known in the statistics field. As a result, the algorithm will be able to calculate the p-value without the need for the standard Barnard's Monte-Carlo sampling [13], which might be infeasible for large bags R_A, R_B.

2 Definition of the Problem

Let U be a universe of elements. Let dst be a distance function on the universe with a finite range $Y \subset \mathbb{R}$, i.e., $\mathsf{dst} : U \times U \mapsto Y$ where $|Y| < \infty$.

Definition 1 (Monge-Elkan distance). *Let R_A, R_B be two bags sampled with replacement from universe U. Then the Monge-Elkan distance*[1] *[14] between R_A*

[1] As we said in the introduction, the original paper [14] defined the Monge-Elkan similarity rather than the distance. We will provide the algorithm for the distance version; however, the ideas might be straightforwardly translated into a similarity version by replacement of min and max operators, and swapping sides of appropriate inequalities.

and R_B, denoted $\mathsf{Dst}(R_A, R_B)$*, is defined as*

$$\mathsf{Dst}(R_A, R_B) = \frac{1}{|R_A|} \sum_{a \in R_A} \min_{b \in R_B} \mathsf{dst}(a, b). \tag{1}$$

The above quantity has the *null distribution* if all reads in $R_A \cup R_B$ are sampled i.i.d. with replacement from U while keeping bag sizes $|R_A|$ and $|R_B|$. The alternative hypothesis states that the bags are similar. Given a value of $\mathsf{Dst}(R_A, R_B)$, we call the alternative hypothesis if the probability of obtaining that value under the null hypothesis is smaller than a threshold.

3 p-value Calculation

First, we can notice that the possible values calculated in (1) are dependent on the range of distance function dst. Therefore, the null distribution of the Monge-Elkan distance is dependent on chosen universum U and distance function dst. As a result, we cannot precompute a generally applicable null distribution of the Monge-Elkan distance. A standard approach in such a case would be based on a random sampling of many bags R_A, and R_B, consequent evaluation of $\mathsf{Dst}(R_A, R_B)$ and approximating the null distribution by this Monte-Carlo approach [13]. In this paper, we will use a probability-generating-function-based approach to evaluate the null distribution. We will break Formula (1) into smaller pieces for which we will calculate the null distribution. Those partial null distributions will be extended to Formula (1).

3.1 Null Distribution of the min Operation

The innermost part of the calculation in (1) is the dst distance. The null distribution of dst is assumed to be given. We also assume that the distance has a finite range as defined in Section 2. The following theorem allows calculating the null distribution of the minimum operation.

Theorem 1. *Let Ω be a finite totally ordered set. Let $p : \Omega \mapsto [0, 1]$ be a probability distribution of a discrete random variable. Suppose that S is a bag of i.i.d. samples taken with replacement from p. Then function $q : \Omega \mapsto [0, 1]$ defined as*

$$q(\omega) = \sum_{i=1}^{|S|} \binom{|S|}{i} \cdot p(\omega)^i \cdot \left(\sum_{\{\omega' \in \Omega | \omega' > \omega\}} p(\omega') \right)^{|S|-i} \tag{2}$$

represents the probability distribution of $\min S$.[2]

[2] Should the last term of the multiplication be in the form of 0^0, then this value is considered 1.

Proof. Suppose that $\min S = \omega$. Then at least one element in S is equal to ω. Denote i the number of elements in Ω equal to ω. There are $\binom{|S|}{i}$ possible choices to select those, and each of them has probability $p(\omega)$. The remaining elements in S have to be bigger than ω, termed in the right multiplicand. The overall result is obtained by summing over all possible values of i.

Corollary 1. *Suppose that distance calculations of* $\mathsf{dst}(a,b)$ *are random i.i.d. for fixed* a*. Then*

$$P\left(\min_{b\in R_B} \mathsf{dst}(a,b) = d\right) = \sum_{i=1}^{|R_B|} \binom{|R_B|}{i} \cdot P(\mathsf{dst}(a,b) = d)^i \cdot P(\mathsf{dst}(a,b) > d)^{|R_B|-i}. \tag{3}$$

We have to be careful when applying Corollary 1. The distribution of the distance is calculated for a fixed a. Consider the situation when the universe $U = \{0, 0.1, 0.2, \ldots, 1.0\}$ and the distance metrics is the absolute difference between two numbers. Then the null distribution looks different in situations when $a = 0.5$ and $a = 0$. In the first case, the maximum distance is 0.5, and in the second case, 1.0. However, if we align the points on a circle so that the coordinates go from 0.0 to 1.0, i.e., that the distance is defined as $\min\{|a-b|, 1-|a-b|\}$, then the null distribution of the distance looks the same for any a, and all we need is to have i.i.d. uniformly selected elements in R_B. Another example of a distance that has the same null distribution for any selection of a is the Hamming distance [7].

The Levenshtein distance [11] is another example where the null distribution depends on a. For string AA, there is only one string of distance 1 if A is inserted, but for string CG, there are three strings of distance 1 if A is inserted. The overall effect of this repetition-based discrepancy needs to be assessed experimentally; nevertheless, with larger alphabet sizes and longer sequences, the effect of repeated symbols will be more negligible.

3.2 Null Distribution of the Sum

Once we have the null distribution for the minimum operation, we need to evaluate the null distribution of the sum. To this end, we will exploit a *probability generating function*, which is a well-established concept in probability theory. The reader is referred to [5] for more details.

Definition 2 (Probability generating function). *Let* Ω *be a finite subset of* $\mathbb{R}$*. Let* $p : \Omega \mapsto [0,1]$ *be a probability distribution of a discrete random variable. Then the* probability generating function *is*

$$\text{genpoly}_p(x) = \sum_{\omega\in\Omega} p(\omega) \cdot x^{\omega}. \tag{4}$$

Usually, the probability generating function is defined only when Ω is a subset of non-negative integers. However, for our application, any finite subset of real

numbers is admissible. For now, we will illustrate the usage of the probability generating functions on a single example. Consider tossing a biased coin that is able to produce three outcomes – 1, 2, and 3. The first one has a probability of $\frac{1}{6}$, the second one $\frac{1}{3}$, and the probability of 3 is equal to $\frac{1}{2}$. Then the generating polynomial is

$$\frac{1}{6}x^1 + \frac{1}{3}x^2 + \frac{1}{2}x^3. \tag{5}$$

We immediately see the probability of an outcome as a coefficient by the respective power of x. Now, let's see what happens when we toss the dice twice and sum the numbers. For example, the sum of 4 can be reached by having the first dice with 1 and the second with 3 with probability of $\frac{1}{6} \cdot \frac{1}{2} = \frac{1}{12}$, both dices showing 2 with probability $\frac{1}{9}$, or, finally, 3 and 1 with probability $\frac{1}{12}$. The probability of seeing 4 is, therefore, $2 \cdot \frac{1}{12} + \frac{1}{9} = \frac{10}{36}$. The previous calculation that we did is exactly what happens to the coefficients if we multiply two polynomials. Consider the second power of the polynomial in (5)

$$\frac{1}{36}x^2 + \frac{4}{36}x^3 + \frac{10}{36}x^4 + \frac{12}{36}x^5 + \frac{9}{36}x^6. \tag{6}$$

We can notice that the power by 4 is equal to $\frac{10}{36}$. To sum it up, the probability generating function allows an easy calculation of the null distribution for a sum of independent variables.

Having this intuition at hand, we can define the generating polynomial for $a \in R_A$ in the calculation of $\min_{b \in R_B} \mathsf{dst}(a, b)$ as

$$\text{genpoly}_a(x) = \sum_{d \in Y} P\left(\min_{b \in R_B} \mathsf{dst}(a, b) = d\right) \cdot x^d. \tag{7}$$

The polynomial for the whole distance in (1) is then the respective power of $\text{genpoly}_a(x)$

$$\text{genpoly}_{R_A}(x) = \left(\text{genpoly}_a(x)\right)^{|R_A|}. \tag{8}$$

This polynomial can then be used to calculate the null distribution of the Monge-Elkan distance. The calculation will have, however, one condition on independence, similarly to dice tosses in (6), that we will discuss later.

Theorem 2. *Assume that the probability* $P\left(\min_{b \in R_B} \mathsf{dst}(a, b) = d\right)$ *is independent of the choice* R_B*. Then the coefficients of the polynomial* $\text{genpoly}_{R_A}(x)$ *represent the null distribution of the Monge-Elkan distance up to a multiplicative term of* $\frac{1}{|R_A|}$.[3]

Proof. Polynomial $\text{genpoly}_{R_A}(x)$ is constructed according to Formula (8). By brute-force multiplication of the product, we get $|Y|^{|R_A|}$ terms, each of them in the form

$$P\left(\min_{b \in R_B} \mathsf{dst}(a, b) = d_1\right) \cdot x^{d_1} \cdot \ldots \cdot P\left(\min_{b \in R_B} \mathsf{dst}(a, b) = d_{|R_A|}\right) \cdot x^{d_{|R_A|}} \tag{9}$$

[3] In other words, if a_d is coefficient by x^d in the polynomial genpoly_{R_A}, then $P\left(\mathsf{Dst}(R_A, R_B) = \frac{1}{|R_A|}d\right) = a_d$.

Each of the powers d_j represents a possible distance selected by the min operator in the sum (1). Their combination represents a possibility of the resulting sum being equal to

$$d = \sum_{j=1}^{|R_A|} d_j. \tag{10}$$

Also, by the independence assumption, the probability of this particular combination is equal to

$$\prod_{j=1}^{|R_A|} P\left(\min_{b \in R_B} \mathsf{dst}(a,b) = d_j\right). \tag{11}$$

The probability $P\left(\mathsf{Dst}(R_A,R_B) = \frac{1}{|R_A|}d\right)$ is equal to the sum of probabilities of all possible combinations of d_j that fulfill equation (10). The probabilities of the combinations are in (11), sum of which which is in turn equal to a_d.

Corollary 2. *Suppose that*

$$\mathrm{genpoly}_{R_A}(x) = a_0 x^{d_0} + a_1 x^{d_1} + a_2 x^{d_2} + \cdots + a_n x^{d_n}. \tag{12}$$

Then the p-value of $\mathsf{Dst}(R_A,R_B) = d$ *is equal to*

$$P(\mathsf{Dst} \leq d) = \sum_{\{d_i | d_i \leq |R_A| d\}} P\left(\mathsf{Dst}(R_A,R_B) = \frac{1}{|R_A|}d_i\right) = \sum_{\{d_i | d_i \leq |R_A| d\}} a_i. \tag{13}$$

3.3 The Independence Assumption and Asymptotic Complexity

We have to look in more detail at the independence assumption in Theorem 2. The fact that the minimum selection should be independent of the set over which we select the minimum is very restrictive and can be satisfied exactly only for very trivial distances dst. The calculation in Theorem 2 corresponds to the situation when for the first element $a \in R_A$, we generate bag R_B and calculate the summand according to the Monge-Elkan distance (1). Then for the second element $a \in R_A$ a *new* set R_B is generated independently, and the summand is calculated. The calculation then follows with the new set R_B for each $a \in R_A$.

However, in the Monge-Elkan distance, the set R_B is kept throughout the calculation. Hence, we are making some error in the p-value calculation. Imagine a space defined by distance function dst. The space looks differently in the case when elements R_B are selected randomly uniformly and in the case when all elements in bag R_B are the same. The first case is, however, common, while the second one is unlikely. Therefore, there are still many situations when Theorem 2 will be applicable as a reasonable approximation.

To evaluate the asymptotic complexity, the starting point is the size of range Y of distance dst. Theorem 1 is applied to calculate the null distribution of the minimum operation. To evaluate the combinatorial coefficients, $\mathcal{O}(|R_B|^2)$ operations are needed. As both probabilities in (3) can be evaluated in constant

time (one by a constant time lookup, the second by using an integral array), the distribution with the same range can be calculated in $\mathcal{O}(|Y|+|R_B|^2)$ operations.

Calculation of the null distribution using Theorem 2 requires power of polynomial of the size $|Y|$ to the power of $|R_A|$. Using a naive implementation, this requires at most $|Y|^{|R_A|}$ multiplications as this is the maximum theoretical possible number of terms in the polynomial multiplication. However, we are calculating the power of a polynomial that requires much less work - there will be $\binom{|R_A|+|Y|-1}{|R_A|}$ terms in the polynomial.

We might use an approach based on fast power calculation for large numbers. Suppose that we are about to calculate the n-th power of a number x. Then we do not need to do n consecutive multiplications. Instead, we calculate powers of $x^1, x^2, x^4, \ldots, x^{\lfloor \log_2 n \rfloor}$ and multiply the respective powers to obtain x^n. As a result, we need only $2 \cdot \log n$ multiplications. In our case, this requires only $\mathcal{O}(\log |R_A|)$ polynomial multiplications. Based on the Fast Fourier transform based algorithm for polynomial multiplication [2], one operation will be at most $\mathcal{O}\left(\binom{|R_A|+|Y|-1}{|R_A|} \cdot \log \binom{|R_A|+|Y|-1}{|R_A|}\right)$. Overall, the runtime complexity is in

$$\mathcal{O}\left(\log |R_A| \cdot \binom{|R_A|+|Y|-1}{|R_A|} \cdot \log \binom{|R_A|+|Y|-1}{|R_A|} + |Y| + |R_B|^2\right). \tag{14}$$

4 Application to Read Data

In paper [15], we proposed using the Monge-Elkan distance to estimate the similarity between sequencing data without the need for sequence assembly. The idea was further extended in [17] or applied to contig data in [16].

In the case of read data, the universum becomes the set of all sequences over alphabet $\Sigma = \{A, C, G, T\}$ of a selected length l. Length l acts as a *read length* - the length of short fragments that are sampled from the DNA sequence. Due to the nature of the sequencing process, those fragments, called reads, have the length of tens to hundreds of symbols. Unfortunately, the information where from the original sequence the reads originated is lost through the process, and *in-silico* reconstruction of the original sequences is usually needed. This reconstruction requires assembly based on prefix-suffix overlaps between the reads.

Further, bags R_A, and R_B represent the bags of reads. The dst in our case is the Levenshtein distance [11] which counts the number of insertions, deletions and substitutions, i.e., the basic evolutionary events. Further the Levenshtein distance is replaced by a slightly modified version [15] that accounts for random locations of the reads by different gap penalty at margins. The usage of the Monge-Elkan distance directly on the read data then avoids the NP-hard problem of assembly. The distance is then made symmetric and rescaled resulting in

$$\underset{\mathsf{MESG}}{\mathsf{Dst}}(R_A, R_B) = \frac{1}{2} \max\{|R_A|, |R_B|\} \cdot (\mathsf{Dst}(R_A, R_B) + \mathsf{Dst}(R_B, R_A)) \tag{15}$$

The reasoning behind the Equation (15) is the following. The alignment between sequences A and B map similar subsequences together. Therefore, read

a should be aligned with the most similar read in R_B. This is done by the Monge-Elkan distance. The results are averaged for all reads in R_A. The distance is then made symmetric in (15). The multiplicative terms at the beginning bring the distance to the proper scale as the Monge-Elkan distance is an average of values no more than read length l while the maximum distance between the original sequences is proportional to $\max\{|R_A|, |R_B|\}$.

4.1 Modification of the p-value Algorithm

From the p-value calculation perspective, the multiplicative terms in (15) are irrelevant as they only apply a linear scale on the null distribution. Therefore, calculating the p-value for the distance in Formula (15) is equivalent to calculating the p-value for[4]

$$|R_B| \sum_{a \in R_A} \min_{b \in R_B} \mathsf{dst}(a,b) + |R_A| \sum_{b \in R_B} \min_{a \in R_A} \mathsf{dst}(a,b). \tag{16}$$

Again, we will assume the independence assumption, which we know will not be fulfilled; however, it might be used as an approximation. Instead of a single sum, we will have an addition of two sums. Therefore, we are able to multiply generating polynomials (under the independence assumption) to obtain the generating polynomial for the new equation (16).

However, one thing remains to solve and that is the different weight of both sums in Equation (16). If we multiply each element in range Y of distance dst, then the null distribution of the multiple is scaled as well. Following the definition of the probability generating function in Definition 2, the multiplication is equivalent to multiplying each exponent in the power of x. Therefore, the resulting generating polynomial is

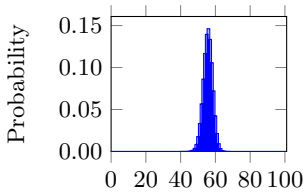
$$\text{genpoly}_{\mathsf{MESG}}(x) = \left(\text{genpoly}_{R_A}(x^{|R_B|})\right)^{|R_A|} \cdot \left(\text{genpoly}_{R_B}(x^{|R_A|})\right)^{|R_B|}. \tag{17}$$

In the formula 17, polynomial genpoly_{R_B} is defined similarly to genpoly_{R_A}, only roles of R_A and R_B are swapped. The coefficients of the polynomial (17) can then be used to estimate the p-value in the same manner as in Theorem 2. There will be an unmet independence assumption which we will describe later.

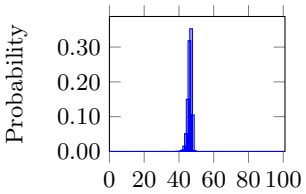
4.2 Runtime Complexity, Independence Assumption, and Null Distribution of dst

In the previous section, we multiplied the generating polynomials for two sums assuming that the sums are independent. In fact, this is not true at all. In the first sum, we are averaging row minima of a matrix, in the second sum, we are averaging row maxima for exactly the same matrix. Therefore, the generating polynomial provides us only an approximation.

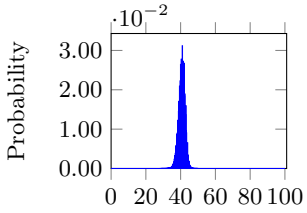
[4] Imagine multiplying the average of $\mathsf{Dst}(R_A, R_B)$ and $\mathsf{Dst}(R_B, R_A)$ by $2|R_A||R_B|$.



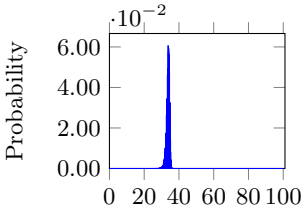
(a) Levenshtein distance dst



(b) Minimum of the Levenshtein distance $\min_{b \in R_B} \mathsf{dst}(\cdot, b)$



(c) Levenshtein distance, margin gaps are not penalized [15]



(d) Theorem 1 applied on the distribution in Figure 1c.

Fig. 1: The null distribution of two distances used to compare read bags with sequencing data and effect of Theorem 1 on the distribution. The distributions were calculated empirically for 10^8 trials assuming random sequences of length $l = 100$. Nevertheless, distances smaller than 38 and bigger than 70 were never registered as their probabilities are very low. The zero probabilities in the calculation could be dealt with using the Laplacian smoothing. The minimum distributions were calculated using Theorem 1 for read bag size of 1,000. The distribution is asymmetric unlike the original one - the smaller distances are preferred and minimum being higher than 50 is unlikely but possible.

Regarding the computational complexity, the runtime is similar to (14), only we do not need to calculate the whole null distribution, only values smaller than d are required, where d is the distance calculated by (16). Therefore the p-value calculation is in

$$\mathcal{O}\left(\log(|R_A||R_B|) \cdot d \cdot \log d + l + |R_A|^2 + |R_B|^2\right). \tag{18}$$

The last component is missing, which is the null distribution of the read-read distance dst. In the case when dst is the Levenshtein distance [11], the null distribution is unknown. The authors of this paper were only able to find previous research that has shown that the null distribution for a related problem of the Longest Common Subsequence [3] problem follows the Tracy-Widom distribution [12]. Therefore, the empirical evaluation of the null distribution for the Levenshtein distance and its modifications that were proposed in [15] is needed. Figure 1 shows null distributions for the Levenshtein distance and the modified distance, together with the effect of min operation in Theorem 1.

5 Experimental Evaluation and Discussion

As the presented approach contains approximation based on the independence assumption that is not true, the null distribution calculated according to Theorem 2 needs to be compared to the ground truth data. To do so, we selected three simple examples of distance functions and evaluated them for various choices of bag sizes as well as different universa. The distances include

- the Levenshtein distance [11] on binary strings of length l.
- Absolute difference between two numbers from $\{0, 1, 2, \ldots, n-1\}$ aligned on a circle. I.e., $\min\{|a-b|, 1-|a-b|\}$ where a and b are numbers in the respective set.

The parameters l and n were selected so that the null distribution of the Monge-Elkan distance could be calculated by mere enumeration of all possible bags R_A and R_B. For simplicity of presentation, $|R_A|$ was set the same as $|R_B|$. The null distribution was then calculated by enumeration of all possible bags and using Theorem 2. Those two distributions were then compared visually as well as using the Kullback-Leibner divergence [10] (KL-divergence, sometimes called relative entropy). In the KL-divergence, the natural logarithm was used.

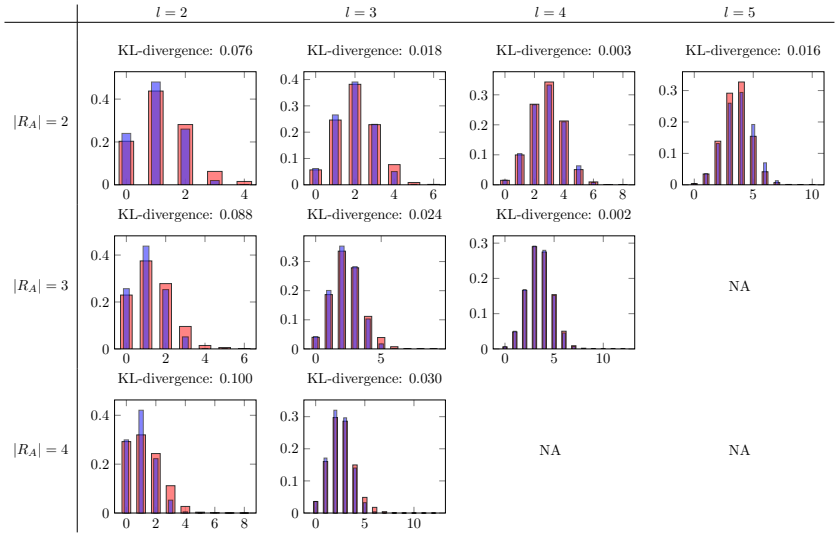
In the case of the string distances, the boundary set for enumeration was $2l|R_A| = 27$, which meant 2^{27} elements in the null distribution at most. In the case of the distance between the numbers, the limit was $n^{2|R_A|} = 10^{10}$, which meant 10^{10} elements in the null distribution at most.

The experimental evaluation is in Figure 2. From the figures we can notice that the KL-divergence is growing with higher bag sizes (the independence assumption is more relied on in the multiplication). It might be expected that the KL-divergence would decrease with universum of more elements (i.e., higher l or n), however, this trend is supported by the data only in the case of string distances.

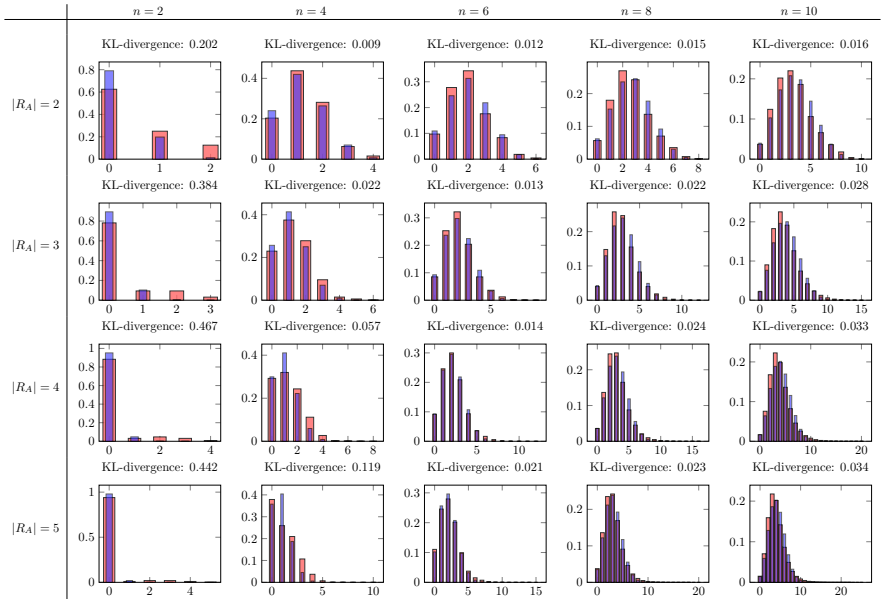
From the plots, we can also notice that the approximation underrepresents the low distances for more cardinal universa U. This is not a desired behavior; however, there remains an open window for future work in modifying the approach so that the p-value cannot be underestimated.

6 Conclusion

We have presented an algorithm to estimate the null distribution of the Monge-Elkan distance that can be used to compare sequence similarity from unassembled read bags. The methodology contains two simplifying assumptions which represent possible sources of error. However, we have confirmed empirically that their detrimental effect is generally not significant. In particular, the KL-divergence between the calculated distribution and the one obtained by Monte-Carlo sampling tends to be negligible. The contributed method thus represents a feasible tool which may even become a necessity when Monte-Carlo sampling is intractable due to slow evaluation of the Monge-Elkan distance.



(a) The Levenshtein distance [11] over binary strings of length l



(b) $\min\{|a-b|, 1-|a-b|\}$ for $a, b \in \{0, 1, 2, \ldots, n-1\}$.

Fig. 2: Comparison of the approximated (blue, narrow) and ground-truth (red, wide) null distribution. The approximated distribution was calculated using Theorem 2 while the ground-truth distribution was calculated enumerating all possible choices of R_A and R_B of the same size. NA means that with the given settings it was not feasible to enumerate the distribution.

References

1. Abdelkader, M.: A method based on WordNet and Monge-Elkan distance for business process model matching. Int. J. Inf. Syst. Model. Des. **9**(4), 37–48 (oct 2018)
2. Cantor, D.G., Kaltofen, E.: On fast multiplication of polynomials over arbitrary algebras. Acta Informatica **28**(7), 693–701 (Jul 1991)
3. Chvátal, V., Sankoff, D.: Longest common subsequences of two random sequences. Journal of Applied Probability **12**(2), 306–315 (1975)
4. Cohen, W.W., Ravikumar, P., Fienberg, S.E.: A comparison of string distance metrics for name-matching tasks. In: Proceedings of the 2003 International Conference on Information Integration on the Web. p. 73–78. IIWEB'03, AAAI Press (2003)
5. Feller, W.: Introduction to probability theory and its applications (1966)
6. Gali, N., Mariescu-Istodor, R., Fränti, P.: Similarity measures for title matching. In: 2016 23rd International Conference on Pattern Recognition (ICPR). pp. 1548–1553 (2016)
7. Hamming, R.W.: Error detecting and error correcting codes. The Bell System Technical Journal **29**(2), 147–160 (April 1950)
8. Jimenez, S., Becerra, C., Gelbukh, A., Gonzalez, F.: Generalized Mongue-Elkan method for approximate text string comparison. In: Gelbukh, A. (ed.) Computational Linguistics and Intelligent Text Processing. pp. 559–570. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)
9. Kaplar, A., Aleksić, A., Stošović, M., Naumović, R., Brković, V., Kovačević, A.: Evaluating string distance metrics for approximate dictionary matching: A case study in serbian electronic health records (2019)
10. Kullback, S., Leibler, R.A.: On Information and Sufficiency. The Annals of Mathematical Statistics **22**(1), 79 – 86 (1951)
11. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. Soviet physics doklady **10**(8), 707 (1966)
12. Majumdar, S.N., Nechaev, S.: Exact asymptotic results for the bernoulli matching model of sequence alignment. Phys. Rev. E **72**, 020901 (Aug 2005)
13. Marriott, F.H.C.: Barnard's Monte Carlo tests: How many simulations? Journal of the Royal Statistical Society. Series C (Applied Statistics) **28**(1), 75–77 (1979)
14. Monge, A.E., Elkan, C.P.: The field matching problem: Algorithms and applications. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. pp. 267–270. KDD'96, AAAI Press (1996)
15. Ryšavý, P., Železný, F.: Estimating sequence similarity from read sets for clustering sequencing data. In: Boström, H., et al. (eds.) Advances in Intelligent Data Analysis XV, pp. 204–214. Springer International Publishing, Cham (2016)
16. Ryšavý, P., Železný, F.: Estimating sequence similarity from contig sets. In: Adams, N., et al. (eds.) Advances in Intelligent Data Analysis XVI. pp. 272–283. Springer International Publishing, Cham (2017)
17. Ryšavý, P., Železný, F.: Estimating sequence similarity from read sets for clustering next-generation sequencing data. Data Mining and Knowledge Discovery **33**(1), 1–23 (Jan 2019)
18. Santos, R., Murrieta-Flores, P., Martins, B.: Learning to combine multiple string similarity metrics for effective toponym matching. International Journal of Digital Earth **11**(9), 913–938 (2018)
19. Wagner, R.A., Fischer, M.J.: The string-to-string correction problem. Journal of the Association for Computing Machinery **21**(1), 168–173 (Jan 1974)
20. Winkler, W.E.: String comparator metrics and enhanced decision rules in the Fellegi-Sunter model of record linkage. (1990)