

Extended Pairwise Sequence Alignment

Eloi Araujo[1(✉)], Fábio V. Martinez[1], Luiz C. Rozante[2], and Nalvo F. Almeida[1]

[1] Faculdade de Computação, Universidade Federal de Mato Grosso do Sul, Campo Grande, MS, Brazil
{francisco.araujo,fabio.martinez,nalvo.almeida}@ufms.br

[2] Centro de Matemática Computação e Cognição, Universidade Federal do ABC, Santo André, SP, Brazil
luiz.rozante@ufabc.edu.br

Abstract. A pairwise sequence alignment is a structure describing a set of editing operations that transforms one given sequence into another given sequence. We consider insertion, deletion, and substitution of symbols as editing operations. Given a fixed function assigning a weight for each editing operation, the weight of an alignment A is the sum of the editing operations described by A. Needleman and Wunsch proposed an algorithm for finding a pairwise sequence alignment of minimum editing weight. However, a sequence of editing operations that transforms one sequence into another cannot always be represented by an alignment. We present a more general structure that allows us to represent any sequence of editing operations that transforms one sequence into another. We also show how to find a minimum weight sequence of editing operations to transform one sequence into another in quadratic time, even if they cannot be represented by an alignment. Additionally, we show that there exists no algorithm to solve the problem with subquadratic running time, unless SETH is false. This approach may be used to explain non-trivial evolutionary models in Molecular Biology, where the triangle inequality does not hold for the distance between the sequences, such as those involving adaptive and back mutations.

Keywords: edit distance · alignment distance · pairwise sequence alignment · extended pairwise sequence alignment

1 Introduction

Sequence comparison is a useful tool for solving problems in several areas such as computational biology, text processing, and pattern recognition. A scoring function is a measure to determine the degree of similarity or dissimilarity between two sequences and can be defined by editing operations that transform one given sequence into another given sequence. Typically, the editing operations considered are insertion, deletion, and substitution of symbols. A well-known scoring function to sequences s and t is the *Levenshtein distance* [13], also called *edit*

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023
O. Gervasi et al. (Eds.): ICCSA 2023, LNCS 13956, pp. 218–230, 2023.
https://doi.org/10.1007/978-3-031-36805-9_15

distance, defined as the minimum number of editing operations that transform s into t. Thus, the edit distance is a way to quantifying the dissimilarity of two strings and, as a consequence, it is used in several applications where the data can be represented by strings.

The pairwise sequence alignment is a structure that allows us to visualize editing operations that transform one sequence into another. It can be obtained putting one sequence above the other in such way that edited symbols are in the same column. In other words, an alignment of sequences s and t is a structure that allows us to identify exactly one editing operation in each symbol of s that ends up turning s into t.

Occasionally, some editing operations are considered more likely to occur than others and thus one can compare the given sequences s and t by computing the *weighted edit distance*, that is, minimizing the sum of weights of the editing operations necessary to transform s into t through a sequence of editing operations given by an alignment. The weighted edit distance is useful, for instance, in applications such as spelling correction systems [6]. The weights of editing operations are given by a scoring matrix γ, where $\gamma_{\mathtt{a}\to\mathtt{b}}, \gamma_{\mathtt{a}\to\sqcup}$ and $\gamma_{\sqcup\to\mathtt{a}}$ are the weights to replace `a` with `b`, and delete and insert `a`, respectively, for any pair of symbols `a` and `b`. Levenshtein distance is a particular case of weighted edit distance considering editing operations with weights $\gamma_{\mathtt{a}\to\mathtt{b}} = 0$ if $\mathtt{a} = \mathtt{b}$ and $\gamma_{\mathtt{a}\to\mathtt{b}} = 1$ if $\mathtt{a} \neq \mathtt{b}$. Needleman and Wunsch [18] proposed an algorithm to determine a pairwise sequence alignment of minimum weight using Levenshtein distance in quadratic time, which is also easily extended to solve the weighted edit distance problem with the same running time.

Other scoring functions have been described based on alignments and individual editing operations. Barton *et al.* [5] used the edit distance in genome assembly and showed that adding the flexibility of bounding the number of gaps inserted in an alignment strengthens the classical pairwise sequence alignment scheme. Another typical scoring function is the so called *normalized edit distance* [17], where the number of editing operations is taken into account when computing the distance between two sequences. Examples of applications of the normalized edit distance are the text reading from street view images [23] and software verification [8]. The *generalized normalized edit distance* of two sequences proposed by [25] is a function that takes into account the weighted edit distance and the lengths of involved sequences. This distance was applied in a handwritten digit recognition study. Another variation is the *contextual normalized edit distance*, where the weight of each editing operation is divided by the length of the string on which the editing operation takes place. It was introduced by [11] and authors showed that this variation is useful in many applications with classification purposes. Consequently, alignments based on editing operations have been a standard way to compare two or even more sequences with a myriad of applications in important topics such as computational biology [1,3,7,12,14,15,18,19,22].

A pairwise sequence alignment represents a set of editing operations where each symbol of s and t is edited once. However, not every set of editing operations can be represented by an alignment, even if it represents one whose sum of

weights of the editing operations is minimum. For example, suppose that $\{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ is an alphabet with three symbols, and $s = \mathtt{a}$ and $t = \mathtt{b}$ are sequences from the alphabet. There exist only three classical ways to align s and t:

$$\begin{bmatrix} \mathtt{a} \\ \mathtt{b} \end{bmatrix}, \qquad \begin{bmatrix} \sqcup & \mathtt{a} \\ \mathtt{b} & \sqcup \end{bmatrix}, \qquad \text{and} \qquad \begin{bmatrix} \mathtt{a} & \sqcup \\ \sqcup & \mathtt{b} \end{bmatrix}.$$

If the weighted editing operations are given by the matrix γ in Fig. 1, we have that the weight of the first alignment is 5, and the weight of both second and third alignments are $2 + 2 = 4$. Hence, the second and third are alignments of sequences $s = \mathtt{a}$ and $t = \mathtt{b}$ with minimum weight.

γ	a	b	c	$\sqcup$
a	0	5	1	2
b	5	3	1	1
c	3	1	0	2
$\sqcup$	2	2	2	

,

Fig. 1. Instance of a scoring matrix γ. The row i and column j represents the weight for editing the symbol i to j.

Notice that none of the alignments above represents the following transformation: "*edit symbol* a *to symbol* c *and then edit symbol* c *to symbol* b". The sum of weights of the editing operations in such transformation is 2, which is smaller than any of those presented before. We can represent this transformation as follows:

$$\begin{bmatrix} \mathtt{a} \\ \mathtt{c} \\ \mathtt{b} \end{bmatrix}.$$

Nevertheless, notice that this structure is not a classical alignment, as shown previously.

Araujo et al. [2] present the *extended alignment problem* that takes into account a set of editing operations regardless of the existence of a usual corresponding alignment to represent them. It is a generalization of the concept of alignment in such way that it can represent a set of sequences of editing operations to transform one given sequence into another given sequence. Consequently, this framework assumes that a single symbol can be edited many times.

Although general substitution models for biological sequences assume each detected mutation as a unique evolutionary event, sometimes it is useful to consider a mutation as a result of a sequence of intermediary steps. In these cases, the classical alignment structure cannot capture such events. It has been shown, for instance, that back mutations ("edit a to b and then b to a") can change the rate of compensatory molecular evolution, where mutations located at two sites, either within a gene or on different genes, are correlated in the

sense that they are individually deleterious but harmless when appropriately combined [16].

Adaptive mutation is another important example of an evolutionary setting that can be explained by such intermediary events. In this case, mutations may arise as an immediate and direct response to selective pressure or even by DNA repair. By suffering a sequence of such pressures, intermediary mutation events may occur [10,20].

This paper is dedicated to the development of a method for finding the sequence of editing operations to transform one given sequence into another given sequence, where the total sum of the single transformation weights is minimum. We use an extended alignment to represent such sequence of operations. We propose a straightforward and efficient algorithm to find an optimal extended pairwise sequence alignment and show that there exists no subquadratic algorithm to perform this task, unless SETH is false.

We organize this paper as follows. Section 2 formalizes the basic concepts necessary for the development of the main results. Section 3 describes how to find a sequence of editing operations of an extended pairwise alignment with minimum weight, presents the running time of the proposed algorithm, and an additional discussion on the complexity of the problem as well. Finally, Sect. 4 concludes showing final remarks.

2 Preliminaries

Let Σ be a finite set of symbols called *alphabet*. We denote a sequence s over Σ by $s(1)\cdots s(n)$, where each $s(j) \in \Sigma$, and we say that $n = |s|$ is its *length*. The set of all sequences over Σ is denoted by Σ^*. A symbol $\sqcup \notin \Sigma$, called *space*, represents insertions and deletions of symbols.

A real-valued *scoring matrix*, whose rows and columns are indexed by symbols of $\Sigma_\sqcup = \Sigma \cup \{\sqcup\}$, stores the weights of editing operations. Given $\mathtt{a}, \mathtt{b} \in \Sigma_\sqcup$ and a scoring matrix γ, we denote by $\gamma_{\mathtt{a}\to\mathtt{b}}$ the entry in row $\mathtt{a}$ and column $\mathtt{b}$ of γ. The value of $\gamma_{\mathtt{a}\to\mathtt{b}}$ is the weight of editing operation of substitution if $\mathtt{a}, \mathtt{b} \in \Sigma$, insertion if $\mathtt{a} = \sqcup$, and deletion if $\mathtt{b} = \sqcup$. The entry $\gamma_{\sqcup\to\sqcup}$ is not defined. See Fig. 1 for an example of scoring matrix.

Given two sequences s and t, a *classical alignment* of s, t is a pair $[s', t']$ of sequences in $\Sigma_\sqcup^*$, where s' and t' are obtained by inserting spaces into s and t, $|s'| = |t'|$, and there is no index j such that $s'(j) = t'(j) = \sqcup$. It is easy to show that if $n = \max\{|s|, |t|\}$, then the *length* $|s'|$ of alignment $[s', t']$ is $O(n)$. The A_γ-score of a classical alignment $[s', t']$ is given by

$$\mathrm{A}_\gamma([s', t']) = \sum_{j=1}^{|s'|} \gamma_{s'(j)\to t'(j)} \,.$$

We define $\mathrm{optA}_\gamma(s, t)$ as the A_γ-score of a classical alignment of s, t with minimum A_γ-score. A classical alignment A of s, t whose A_γ-score is equal to $\mathrm{optA}_\gamma(s, t)$ is called an A_γ-*optimal alignment* of s, t.

Problem Minimum Alignment Score (**MAS**): *Given two sequences s, t in Σ^*, find an A_γ-optimal alignment of s, t, where γ is a fixed scoring matrix.*

MAS is the Levenshtein problem if $\gamma_{\mathtt{a}\to\mathtt{b}} = 0$ when $\mathtt{a} = \mathtt{b}$, and $\gamma_{\mathtt{a}\to\mathtt{b}} = 1$ otherwise, and thus γ is a metric on Σ.

An *extended alignment* of s, t is a n-tuple $[c_1, \ldots, c_n]$, where each c_j is a column that represents a sequence of editing operations of a symbol. Precisely, each c_j is a finite sequence with $m_j = |c_j| \geq 1$ symbols in $\Sigma_\sqcup$, $c_j(1) \neq \sqcup$ or $c_j(m_j) \neq \sqcup$, and it doesn't have consecutive symbols $\sqcup$; the sequence s is obtained by removing symbols $\sqcup$ from $c_1(1) \cdots c_n(1)$, and sequence t is obtained by removing symbols $\sqcup$ from $c_1(m_1) \cdots c_n(m_n)$. For example, the 8-tuple $A = [\mathtt{a}, \mathtt{abcd}, \mathtt{b}\sqcup, \sqcup\mathtt{d}, \mathtt{ca}, \sqcup\mathtt{dabc}, \sqcup\mathtt{abc}, \mathtt{dab}]$ is an extended alignment of the sequences $\mathtt{aabcd}, \mathtt{addaccb}$. Notice that each c_j is a sequence of $m_j - 1$ editing operations. The graphical representation

$$A = \begin{bmatrix} & \mathtt{a} & & & & \sqcup & & \\ & \mathtt{b} & \mathtt{b} & \sqcup & \mathtt{c} & \mathtt{d} & \sqcup & \mathtt{d} \\ \mathtt{a} & \mathtt{c} & \sqcup & \mathtt{d} & \mathtt{a} & \mathtt{a} & \mathtt{b} & \mathtt{a} \\ & \mathtt{d} & & & & \mathtt{b} & \mathtt{c} & \mathtt{b} \\ & & & & & \mathtt{c} & & \end{bmatrix}$$

of the extended alignment A of sequences $\mathtt{aabcd}, \mathtt{addaccb}$ shows the editing operations that are represented by A.

The E_γ-score of a column c_j is the sum of the editing operations represented in c_j, i.e.,

$$\mathrm{E}_\gamma(c_j) = \sum_{i=1}^{m_j - 1} \gamma_{c_j(i)\to c_j(i+1)}\,,$$

and the E_γ-score of an extended alignment $[c_1, \ldots, c_n]$ is

$$\mathrm{E}_\gamma([c_1, \ldots, c_n]) = \sum_{j=1}^{n} \mathrm{E}_\gamma(c_j)\,.$$

We define $\mathrm{optE}_\gamma(s, t)$ as a minimum E_γ-score among all extended alignments of seqeunces s, t. An extended alignment A of s, t whose E_γ-score is equal to $\mathrm{optE}_\gamma(s, t)$ is called an E_γ-*optimal alignment* of s, t.

Problem Minimum Edit Score (**MES**): *Given two sequences s, t in Σ^*, find an E_γ-optimal alignment of s, t, where γ is a fixed scoring matrix.*

We simply say 'alignment' instead of 'classical' or 'extended alignment' if the type of the alignment is obvious from the context.

A scoring matrix γ can be seen as a digraph $D(\gamma)$ with cost on its arcs, where the set of vertices is $V(D) = \Sigma_\sqcup$, the set of arcs is $E(D) = \Sigma_\sqcup \times \Sigma_\sqcup \setminus \{(\sqcup, \sqcup)\}$, and the cost of an arc $(\mathtt{a}, \mathtt{b}) \in E(D)$ is given by $\mathrm{cost}_\gamma(\mathtt{a}, \mathtt{b}) = \gamma_{\mathtt{a}\to\mathtt{b}}$. See Fig. 2.

A sequence of arcs $P = (x_1, x_2), (x_2, x_3), \ldots, (x_{n-1}, x_n)$, for $x_i \in V(D)$, is a *walk* from x_1 to x_n, with *length* $|P| = n - 1$ and *cost* given by $\mathrm{cost}_\gamma(P) =$

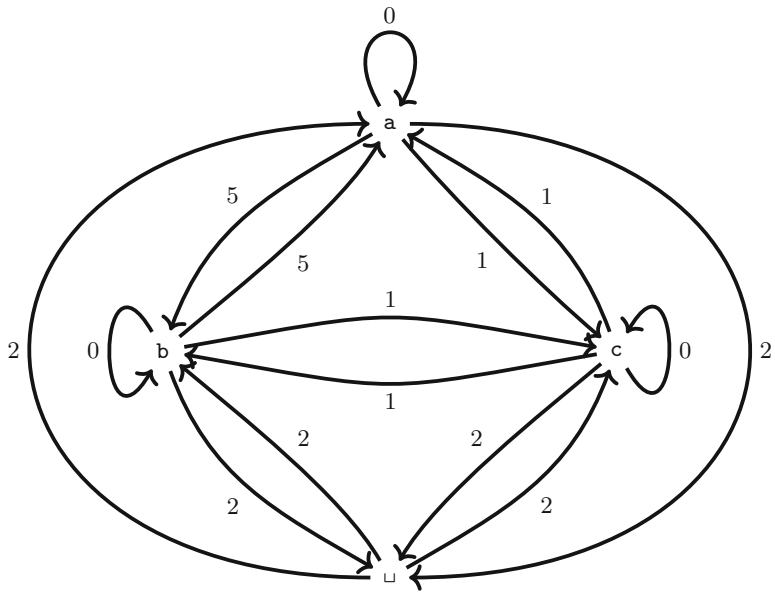


Fig. 2. Digraph $D(\gamma)$ with cost on its arcs given by γ, where γ is the matrix presented in Fig. 1.

$\sum_{i=1}^{n-1} \text{cost}_\gamma(x_i, x_{i+1})$. If $x_1 = x_n$, we say that P is a cycle, and if P is a cycle and $\text{cost}_\gamma(P) < 0$, we say that P is a *negative cycle*. An *optimal walk* from x_1 to x_n in $D(\gamma)$ is a minimum cost walk from x_1 to x_n in $D(\gamma)$ and we denote its cost by $\mathcal{P}_\gamma(x_1, x_n)$. When P and Q are walks from `a` to `c` and from `c` to `b`, respectively, we denote by PQ the walk from `a` to `b` obtained by concatenation of arcs in P and then in Q and we notice that $\text{cost}_\gamma(PQ) = \text{cost}_\gamma(P) + \text{cost}_\gamma(Q)$.

We can easily check that there exists no negative cycle in $D(\gamma)$ if and only if $\text{optE}_\gamma(s, t)$ is well-defined for any $s, t \in \Sigma^*$. Thus, we assume that $D(\gamma)$ does not have a negative cycle as a subgraph.

Finally, we say that a classical alignment $[s', t']$ *induces* an extended alignment $[s'(1)\ t'(1)\ , \dots\ , s'(n)\ t'(n)]$. Any classical alignment induces an extended alignment with same E_γ-score. This means that

$$\text{A}_\gamma([s', t']) = \text{E}_\gamma([s'(1)\ t'(1)\ , \dots\ , s'(n)\ t'(n)])\,, \tag{1}$$

for any $s, t \in \Sigma^*$. Hence, we have that

$$\text{optE}_\gamma(s, t) \leq \text{optA}_\gamma(s, t)\,. \tag{2}$$

3 Computing Optimal Extended Alignments

In this section, we show how to compute an E_γ-optimal alignment for two given sequences. The main procedure is the preprocessing of the scoring matrix γ in

order to obtain a second scoring matrix δ such that $\text{optE}_\gamma(s,t) = \text{optA}_\delta(s,t)$. Since we know efficient algorithms to solve Problem **MAS** [18], we must find matrix δ. Furthermore, we show an algorithm to obtain an E_γ-optimal alignment of sequences s, t from an A_δ-optimal alignment of s, t.

A scoring matrix δ is defined from the scoring matrix γ as follows. For each pair of symbols $\mathtt{a}, \mathtt{b} \in \Sigma_\sqcup$, we have that

$$\delta_{\mathtt{a}\to\mathtt{b}} = \mathcal{P}_\gamma(\mathtt{a}, \mathtt{b})\,, \tag{3}$$

where $\mathcal{P}_\gamma(\mathtt{a}, \mathtt{b})$ is a minimum cost walk from $\mathtt{a}$ to $\mathtt{b}$ in the digraph $D(\gamma)$.

The scoring matrix δ has the following properties.

Lemma 1. *Let Σ be an alphabet, γ be a scoring matrix such that the digraph $D(\gamma)$ has no negative cycle, and δ be a scoring matrix defined according to Eq. (3). Then,*

1. $\delta_{\mathtt{a}\to\mathtt{a}} = 0$, *and*
2. $\delta_{\mathtt{a}\to\mathtt{b}} \leq \delta_{\mathtt{a}\to\mathtt{c}} + \delta_{\mathtt{c}\to\mathtt{b}}$,

for any symbols $\mathtt{a}, \mathtt{b}, \mathtt{c} \in \Sigma_\sqcup$.

Proof. First, suppose that $\mathtt{a} \in \Sigma_\sqcup$. A walk with length zero has cost equal to zero and thus $\mathcal{P}_\gamma(\mathtt{a}, \mathtt{a}) \leq 0$. On the other hand, since $D(\gamma)$ has no negative cycles, $\mathcal{P}_\gamma(\mathtt{a}, \mathtt{a}) \geq 0$. Therefore, we conclude that $\mathcal{P}_\gamma(\mathtt{a}, \mathtt{a}) = 0$, which implies from Eq. (3) that $\delta_{\mathtt{a}\to\mathtt{a}} = \mathcal{P}_\gamma(\mathtt{a}, \mathtt{a}) = 0$.

Let $\mathtt{a}, \mathtt{b}, \mathtt{c} \in \Sigma_\sqcup$ and suppose that P_1 and P_2 are optimal walks from $\mathtt{a}$ to $\mathtt{c}$ and from $\mathtt{c}$ to $\mathtt{b}$, respectively. By Eq. (3), we have that $\text{cost}_\gamma(P_1) = \mathcal{P}_\gamma(\mathtt{a}, \mathtt{c})$ and $\text{cost}_\gamma(P_2) = \mathcal{P}_\gamma(\mathtt{c}, \mathtt{b})$. It follows that $\text{cost}_\gamma(P_1) = \mathcal{P}_\gamma(\mathtt{a}, \mathtt{c}) = \delta_{\mathtt{a}\to\mathtt{c}}$ and $\text{cost}_\gamma(P_2) = \mathcal{P}_\gamma(\mathtt{c}, \mathtt{b}) = \delta_{\mathtt{c}\to\mathtt{b}}$. Since P_1 and P_2 are walks from $\mathtt{a}$ to $\mathtt{c}$ and from $\mathtt{c}$ to $\mathtt{b}$, P_1P_2 is a walk from $\mathtt{a}$ to $\mathtt{b}$ and $\text{cost}_\gamma(P_1P_2) = \text{cost}_\gamma(P_1) + \text{cost}_\gamma(P_2)$. Hence, since $\mathcal{P}_\gamma(\mathtt{a}, \mathtt{b}) \leq \text{cost}_\gamma(P_1P_2)$ and by Eq. (3) we have $\delta_{\mathtt{a}\to\mathtt{b}} = \mathcal{P}_\gamma(\mathtt{a}, \mathtt{b})$, it follows that

$$\delta_{\mathtt{a}\to\mathtt{b}} = \mathcal{P}_\gamma(\mathtt{a}, \mathtt{b}) \leq \text{cost}_\gamma(P_1P_2) = \text{cost}_\gamma(P_1) + \text{cost}_\gamma(P_2) = \delta_{\mathtt{a}\to\mathtt{c}} + \delta_{\mathtt{c}\to\mathtt{b}}\,.$$

□

Lemma 2. *Let Σ be an alphabet, γ be a scoring matrix such that $D(\gamma)$ has no negative cycles, δ be a scoring matrix defined from γ according to Eq. (3), and sequences $s, t \in \Sigma^*$. Then, there exists an E_δ-optimal alignment $[c_1, \ldots, c_n]$ that is induced by an A_δ-optimal alignment of s, t, i.e., $|c_j| = 2$ for each j.*

Proof. Let $A' = [c'_1, \ldots, c'_n]$ be an E_δ-optimal alignment and $A = [c_1, \ldots, c_n]$ be the extended alignment of s, t induced by the classical alignment

$$[(c'_1(1)c'_2(1)\cdots c'_n(1), c'_1(m_1)c'_2(m_2)\cdots c'_n(m_n))]\,,$$

i.e., $c_j = c'_j(1)c'_j(m_j)$ for each j.

In order to prove this lemma, since $\mathrm{E}_\delta(A') = \mathrm{optE}_\delta(s,t)$, it is enough to show that $\mathrm{E}_\delta(A) \leq \mathrm{E}_\delta(A')$ and to do this, it is enough to prove that $\mathrm{E}_\delta(c_j) \leq \mathrm{E}_\delta(c'_j)$ for an arbitrary $j \in \{1,2,\ldots,n\}$. We consider two cases: $m_j = 1$ and $m_j > 1$.

Suppose that $m_j = 1$ and $c'_j = \mathtt{a}$. Trivially, we have that $\mathrm{E}_\delta(c'_j) = 0$. Furthermore, since $D(\gamma)$ has no negative cycles, we have from Lemma 1 that $\delta_{\mathtt{a}\to\mathtt{a}} = 0$. Thus, $\mathrm{E}_\delta(c_j = \mathtt{aa}) = \delta_{\mathtt{a}\to\mathtt{a}} = 0 = \mathrm{E}_\delta(c'_j)$.

Now, suppose that $m_j > 1$. Since $D(\gamma)$ has no negative cycles, it follows from Lemma 1 that

$$\begin{aligned}
\mathrm{E}_\delta(c_j) = \mathrm{E}_\delta(c'_j(1)c'_j(m_j)) &= \delta_{c'_j(1)\to c'_j(m_j)} \\
&\leq \delta_{c'_j(1)\to c'_j(2)} + \delta_{c'_j(2)\to c'_j(m_j)} \\
&\leq \delta_{c'_j(1)\to c'_j(2)} + \delta_{c'_j(2)\to c'_j(3)} + \delta_{c'_j(3)\to c'_j(m_j)} \leq \cdots \\
&\leq \sum_{i=1}^{m_j-1} \delta_{c'_j(i)\to c'_j(i+1)} = \mathrm{E}_\delta(c'_j)\,.
\end{aligned}$$

□

Now we can show the main results of this paper. Given two sequences s,t, we show that $\mathrm{optE}_\gamma(s,t) = \mathrm{optA}_\delta(s,t)$ and present a straightforward algorithm for finding an E_γ-optimal alignment of s,t.

Theorem 3. *Let Σ be an alphabet, γ be a scoring matrix such that $D(\gamma)$ has no negative cycles, δ be a scoring matrix defined from γ according to Eq. (3), and $s,t \in \Sigma^*$. Then, $optE_\gamma(s,t) = optA_\delta(s,t)$. Furthermore, let $[s',t']$ be an A_δ-optimal alignment of s,t, $|[s',t']| = n$. Then, we also have that*

$$C = [\mathcal{C}_\gamma(s'(1),t'(1)),\ldots,\mathcal{C}_\gamma(s'(n),t'(n))]$$

is an E_γ-optimal alignment of s,t, where $\mathcal{C}_\gamma(x,y)$ is an optimal walk from x to y in $D(\gamma)$.

Proof. From Lemma 2, there exists an E_δ-optimal extended alignment A of s,t, that is the alignment induced by a classical alignment A' of s,t. Since, by Eq. (1), $\mathrm{E}_\delta(A) = \mathrm{A}_\delta(A')$, it follows that

$$\mathrm{optE}_\delta(s,t) = \mathrm{E}_\delta(A) = \mathrm{A}_\delta(A') \geq \mathrm{optA}_\delta(s,t)\,.$$

Moreover, by Eq. (2), we have that $\mathrm{optE}_\delta(s,t) \leq \mathrm{optA}_\delta(s,t)$. Therefore, $\mathrm{optE}_\delta(s,t) = \mathrm{optA}_\delta(s,t)$. Since $[s',t']$ is an A_δ-optimal alignment of s,t, it follows that $A = [s'(1)t'(1),\ldots,s'(n)t'(n)]$ is an E_δ-optimal alignment of s,t.

Now, suppose that $B = [c_1,\ldots,c_r]$ is an E_γ-optimal alignment of s,t. Since $\mathrm{E}_\gamma(c_j)$ is the cost of a walk from $c_j(1)$ to $c_j(m_j)$ in $D(\gamma)$ and $\mathcal{P}_\gamma(c_j(1),c_j(m_j))$ is a cost of an optimal walk from $c_j(1)$ to $c_j(m_j)$ in $D(\gamma)$ for an arbitrary integer j, we have that $\mathrm{E}_\gamma(c_j) \geq \mathcal{P}_\gamma(c_j(1),c_j(m_j))$. The E_δ-score of extended alignment $[c_1(1)c_1(m_1),\ldots,c_r(1)c_r(m_r)]$ of s,t is $\sum_j \delta_{c_j(1)\to c_j(m_j)}$ and, by Eq. (3), $\mathcal{P}_\gamma(c_j(1),c_j(m_j)) = \delta_{c_j(1)\to c_j(m_j)}$. It follows that

$$\begin{aligned}\text{optE}_\gamma(s,t) = \text{E}_\gamma(B) = \sum_j \text{E}_\gamma(c_j) &\geq \sum_j \mathcal{P}_\gamma(c_j(1), c_j(m_j)) = \sum_j \delta_{c_j(1)\to c_j(m_j)} \\ &= \text{E}_\delta\left([c_1(1)c_1(m_1), \ldots, c_r(1)c_r(m_r)]\right) \\ &\geq \text{optE}_\delta(s,t)\,. \end{aligned} \tag{4}$$

On the other hand, the extended alignment C obtained from the alignment A replacing each column $s'(j)t'(j)$ by a sequence of symbols $\mathcal{C}_\gamma(s'(j), t'(j))$, is an extended alignment of s, t whose E_γ-score is also $\text{E}_\delta([s'(1)t'(1), \ldots, s'(n)t'(n)]$. Since $[s'(1)t'(1), \ldots, s'(n)t'(n)]$ is an E_δ-optimal alignment of s, t, it follows that

$$\begin{aligned}\text{optE}_\delta(s,t) &= \text{E}_\delta([s'(1)t'(1), \ldots, s'(n)t'(n)] \\ &= \text{E}_\gamma(C = [\mathcal{C}_\gamma(s'(1), t'(1)), \ldots, \mathcal{C}_\gamma(s'(n), t'(n))]) \geq \text{optE}_\gamma(s,t)\,. \end{aligned} \tag{5}$$

From $\text{optE}_\delta(s,t) = \text{optA}_\delta(s,t)$ and Eqs. (4) and (5), we conclude that $\text{optE}_\gamma(s,t) = \text{optA}_\delta(s,t)$. We also conclude by construction that C is an E_γ-optimal alignment of s, t. □

As a consequence of Theorem 3, we can compute $\text{optE}_\gamma(s,t)$ using an algorithm that obtains $\text{optA}_\delta(s,t)$, spending additional time only for computing the scoring matrix δ.

As an example, suppose that

γ	a	b	c	d	␣
a	2	1	3	4	7
b	3	3	−1	1	1
c	2	4	0	5	8
d	9	7	8	0	2
␣	3	7	50	6	

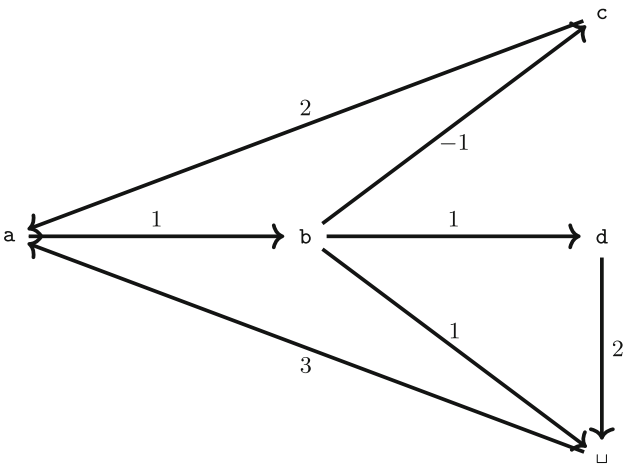
is a scoring matrix and $s = \texttt{addd}$ and $t = \texttt{acddaaa}$ are given sequences. We want to find a minimum weight sequence of editing operations that transforms s into t. Figure 3 shows the subgraph of $D(\gamma)$ containing the arcs (x, y) that are in optimal walks in $D(\gamma)$ and a minimal walks connecting each pair of vertices.

Using Eq. (3) and an algorithm to solve Problem **MAS**, we obtain

	a	b	c	d	␣
a	0	1	0	2	2
b	1	0	−1	1	1
c	2	3	0	4	4
d	5	6	5	0	2
␣	3	4	3	5	0

and $\begin{bmatrix} \texttt{a} & \texttt{␣} & \texttt{d} & \texttt{d} & \texttt{d} & \texttt{␣} \\ \texttt{a} & \texttt{c} & \texttt{d} & \texttt{d} & \texttt{a} & \texttt{a} \end{bmatrix}$,

which are a scoring matrix δ and an A_δ-optimal alignment of $\texttt{addd}, \texttt{acddaaa}$, respectively. Thus, $\text{optA}_\delta(\texttt{addd}, \texttt{acddaaa}) = 11$. Finally, using Theorem 3, we obtain an E_γ-optimal alignment



	a	b	c	d	␣
a	a	a → b	a → b → c	a → b → c	a → b → c
b	b → c → a	b	b → c	b → d	b → ␣
c	c → a	c → a → b	c	c → a → b → d	c → a → b → ␣
d	d → ␣ → a	d → ␣ → a → b	d → ␣ → a → c	d	d → ␣
␣	␣ → a	␣ → a → b	␣ → a → b → c	␣ → a → b → d	

Fig. 3. A subgraph of $D(\gamma)$ containing the edges (x, y) that are optimal walks in $D(\gamma)$. The table below shows a minimal walk connecting vertices which are useful for construct the extended alignment.

$$A = \begin{bmatrix} & ␣ & & & d & \\ a & a & d & d & ␣ & ␣ \\ & b & & & a & a \\ & c & & & & \end{bmatrix}$$

such that $\mathrm{E}_\gamma(A) = 11$.

3.1 Running Time and Complexity

In order to compute matrix δ, it is enough to solve the well-known *all-pairs shortest paths problem* for the digraph $D(\gamma)$. Since $D(\gamma)$ does not have negative cycles, we can use Floyd-Warshall algorithm [9,24], whose time and space complexities are $\Theta(|\Sigma|^3)$, i.e., depend only on the size of the alphabet Σ. Since alphabet Σ does not belong to the instance of **MES**, the preprocessing running time is $\Theta(1)$. In addition, Floyd-Warshall algorithm can also compute a useful structure to obtain optimal walks in $D(\gamma)$ without spending extra time. This structure can be used for obtaining the pairwise extended sequence alignment, again without spending extra time.

Therefore, the running time of the whole process depends only on the running time to solve **MAS**, which is quadratic [21]. This allows us to claim the following result.

Theorem 4. *Problem* **MES** *can be solved in* $O(n^2)$ *time, where* n *is the length of an instance* s, t.

Now we establish a strong relationship between an alignment and an extended alignment of two given sequences.

Lemma 5. *Let* γ *and* δ *be scoring matrices such that* $D(\gamma)$ *has no negative cycles and* δ *is obtained from* γ *using Eq.* (3). *Then,* $\delta = \gamma$ *if and only if* $\gamma_{\mathtt{a}\to\mathtt{a}} = 0$ *and* $\gamma_{\mathtt{a}\to\mathtt{b}} \leq \gamma_{\mathtt{a}\to\mathtt{c}} + \gamma_{\mathtt{c}\to\mathtt{b}}$, *for each* $\mathtt{a}, \mathtt{b}, \mathtt{c} \in \Sigma$. *Besides that, for any pair of sequences, there exists an* E_γ*-optimal alignment induced by an* A_δ*-optimal alignment, and the* A_δ*-optimal alignment can be found from the* E_γ*-optimal alignment in* $O(n)$ *time.*

Proof. Suppose that $\gamma = \delta$. We have $\delta_{\mathtt{a}\to\mathtt{a}} = 0$ and $\delta_{\mathtt{a}\to\mathtt{b}} \leq \delta_{\mathtt{a}\to\mathtt{c}} + \delta_{\mathtt{c}\to\mathtt{b}}$ for each $\mathtt{a}, \mathtt{b}, \mathtt{c} \in \Sigma$ from Lemma 1, which implies that $\gamma_{\mathtt{a}\to\mathtt{a}} = 0$ and $\gamma_{\mathtt{a}\to\mathtt{b}} \leq \gamma_{\mathtt{a}\to\mathtt{c}} + \gamma_{\mathtt{c}\to\mathtt{b}}$ for each $\mathtt{a}, \mathtt{b}, \mathtt{c} \in \Sigma$.

Conversely, suppose that $\gamma_{\mathtt{a}\to\mathtt{a}} = 0$ and $\gamma_{\mathtt{a}\to\mathtt{b}} \leq \gamma_{\mathtt{a}\to\mathtt{c}} + \gamma_{\mathtt{c}\to\mathtt{b}}$ for each $\mathtt{a}, \mathtt{b}, \mathtt{c} \in \Sigma$ and, by contradiction, $\gamma_{\mathtt{a}\to\mathtt{a}} \neq \delta_{\mathtt{a}\to\mathtt{a}}$ or $\gamma_{\mathtt{a}\to\mathtt{b}} \neq \delta_{\mathtt{a}\to\mathtt{b}}$ for each $\mathtt{a}, \mathtt{b} \in \Sigma$. Since $\delta_{\mathtt{a}\to\mathtt{a}} = 0$ from Lemma 1 we have that $\delta_{\mathtt{a}\to\mathtt{a}} = 0 = \gamma_{\mathtt{a}\to\mathtt{a}}$. Thus, we assume that $\gamma_{\mathtt{a}\to\mathtt{b}} \neq \delta_{\mathtt{a}\to\mathtt{b}}$. Since $\gamma_{\mathtt{a}\to\mathtt{b}}$ is the cost of edge $(\mathtt{a}, \mathtt{b})$ and $\delta_{\mathtt{a}\to\mathtt{b}}$ is the cost of minimum cost walk from $\mathtt{a}$ to $\mathtt{b}$, $\gamma_{\mathtt{a}\to\mathtt{b}}$ cannot be smaller than $\delta_{\mathtt{a}\to\mathtt{b}}$ which implies, since $\gamma_{\mathtt{a}\to\mathtt{b}} \neq \delta_{\mathtt{a}\to\mathtt{b}}$, that $\gamma_{\mathtt{a}\to\mathtt{b}} > \delta_{\mathtt{a}\to\mathtt{b}}$. Let $(\mathtt{a} = x_1, x_2), (x_2, x_3), \ldots, (x_{m-1}, x_m = \mathtt{b})$ an walk with minimum cost, i.e., $\delta_{\mathtt{a}\to\mathtt{b}} = \gamma_{x_1\to x_2} + \gamma_{x_2\to x_3} + \ldots + \gamma_{x_{m-1}\to x_m}$. Since $\gamma_{\mathtt{a}\to\mathtt{b}} \leq \gamma_{\mathtt{a}\to\mathtt{c}} + \gamma_{\mathtt{c}\to\mathtt{b}}$ for each $\mathtt{a}, \mathtt{b}, \mathtt{c} \in \Sigma$, it follows that

$$\begin{aligned}\delta_{\mathtt{a}\to\mathtt{b}} &= \gamma_{\mathtt{a}\to x_2} + \gamma_{x_2\to x_3} + \ldots + \gamma_{x_{m-1}\to\mathtt{b}} \\ &\geq \gamma_{\mathtt{a}\to x_3} + \ldots + \gamma_{x_{m-1}\to\mathtt{b}} \geq \gamma_{\mathtt{a}\to\mathtt{b}},\end{aligned}$$

contradicting $\gamma_{\mathtt{a}\to\mathtt{b}} > \delta_{\mathtt{a}\to\mathtt{b}}$. Therefore, $\gamma_{\mathtt{a}\to\mathtt{a}} = \delta_{\mathtt{a}\to\mathtt{a}}$ and $\gamma_{\mathtt{a}\to\mathtt{b}} = \delta_{\mathtt{a}\to\mathtt{b}}$ for each $\mathtt{a}, \mathtt{b} \in \Sigma$.

Since $\gamma = \delta$ and δ is obtained from Eq. (3), it follows from Lemma 2, that there exists an E_γ-optimal alignment $A = [s'(1)t'(1), \ldots, s'(m)t'(m)]$ that is induced by an A_γ-optimal alignment $[s'(1)\ldots s'(m), \ldots, t'(1)t'(m)]$ of s, t whose length is $m = O(n)$ and that clearly can be found from A in $O(n)$ time. □

Finally, we present a discussion on the complexity of **MES**. Backurs and Indik [4] showed that the Levenshtein problem cannot be computed in $O(n^{2-\varepsilon})$ time, for some constant $\varepsilon > 0$, unless the Strong Exponential Time Hypothesis (SETH) if false. In this case, the scoring matrix γ is a metric on Σ, which implies that $\gamma_{\mathtt{a}\to\mathtt{a}} = 0$ and $\gamma_{\mathtt{a}\to\mathtt{b}} \leq \gamma_{\mathtt{a}\to\mathtt{c}} + \gamma_{\mathtt{c}\to\mathtt{b}}$, for each $\mathtt{a}, \mathtt{b}, \mathtt{c} \in \Sigma$. Thus, from Lemma 5, we have that an A_γ-optimal alignment can be found from an E_γ-optimal alignment spending $O(n)$ extra time. Therefore, if there exists an algorithm A to solve **MES** in $O(n^{2-\varepsilon})$ time, for some $\varepsilon > 0$, then algorithm A could be used to solve **MAS** in $O(n^{2-\varepsilon}) + O(n)$ time, and therefore SETH is false. Thus, we can state the following.

Theorem 6. *There exists no strongly subquadratic algorithm to solve* **MES***, unless SETH is false.*

4 Conclusion

In this paper we present a more general structure to represent any sequence of editing operations that transforms one given sequence into another given sequence, the so called extended pairwise sequence alignment. We show a straightforward and efficient algorithm to find an optimal extended pairwise sequence alignment and prove that there exists no strongly subquadratic algorithm to solve it, unless SETH is false.

This approach has potential to explain non-trivial evolutionary models in Molecular Biology, in applications such as those involving adaptive and back mutations. Theoretically, this proposed setting can be used especially when the triangle inequality does not hold for the distance between the sequences.

Experiments on simulated and real data for the extended pairwise sequence alignment, in comparison to the classical pairwise sequence alignment, can enrich this work and will be considered in our future steps.

Acknowledgments. The authors thank José Augusto Ramos Soares, Said Sadique Adi, and Vagner Pedrotti for valuable discussions on this topic.

References

1. Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J.: Basic local alignment search tool. J. Mol. Biol. **215**(3), 403–410 (1990)
2. Araujo, E., Martinez, F.V., Higa, C.H.A., Soares, J.: Matrices inducing generalized metric on sequences. Discrete Appl. Math. (2023, to appear)
3. Araujo, E., Rozante, L.C., Rubert, D.P., Martinez, F.V.: Algorithms for normalized multiple sequence alignments. In: Proceedings of ISAAC. LIPIcs, vol. 212, pp. 40:1–40:16 (2021)
4. Backurs, A., Indyk, P.: Edit distance cannot be computed in strongly subquadratic time (unless SETH is false). In: Proceedings of STOC, pp. 51–58 (2015)
5. Barton, C., Flouri, T., Iliopoulos, C.S., Pissis, S.P.: Global and local sequence alignment with a bounded number of gaps. Theor. Comput. Sci. **582**, 1–16 (2015)
6. Chaurasiya, R.K., Londhe, N.D., Ghosh, S.: A novel weighted edit distance-based spelling correction approach for improving the reliability of Devanagari script-based P300 speller system. IEEE Access **4**, 8184–8198 (2016)
7. Chenna, R., et al.: Multiple sequence alignment with the Clustal series of programs. Nucleic Acids Res. **31**(13), 3497–3500 (2003)
8. Fisman, D., Grogin, J., Margalit, O., Weiss, G.: The Normalized Edit Distance with Uniform Operation Costs is a Metric. arXiv:2201.06115 (2022)
9. Floyd, R.: Algorithm 97: shortest path. Commun. ACM **5**(6), 345 (1962)
10. Foster, P.: Adaptive mutation in Escherichia coli. J. Bacteriol. **186**(15), 4846–4852 (2004)
11. de la Higuera, C., Micó, L.: A contextual normalised edit distance. In: Proceedings of ICDEW, pp. 354–361. IEEE (2008)

12. Karplus, K., Barrett, C., Hughey, R.: Hidden Markov models for detecting remote protein homologies. Bioinformatics **14**(10), 846–856 (1998)
13. Levenshtein, V.: Binary codes capable of correcting deletions, insertions and reversals. Sov. Phys. Doklady **10**(8), 707–710 (1966)
14. Lipman, D.J., Altschul, S.F., Kececioglu, J.D.: A tool for multiple sequence alignment. PNAS **86**(12), 4412–4415 (1989)
15. Lipman, D.J., Pearson, W.R.: Rapid and sensitive protein similarity searches. Science **227**(4693), 1435–1441 (1985)
16. Ichinose, M., Iizuka, M., Kusumi, J., Takefu, M.: Models of compensatory molecular evolution: effects of back mutation. J. Theor. Biol. **323**(0), 1–10 (2013)
17. Marzal, A., Vidal, E.: Computation of normalized edit distance and applications. IEEE T. Pattern Anal. **15**(9), 926–932 (1993)
18. Needleman, S.B., Wunsch, C.D.: A general method applicable to the search for similarities in the amino acid sequence of two proteins. J. Mol. Biol. **48**(3), 443–453 (1970)
19. Notredame, C., Higgins, D.G., Heringa, J.: T-Coffee: a novel method for fast and accurate multiple sequence alignment. J. Mol. Biol. **302**(1), 205–217 (2000)
20. Rosenberg, S.: Evolving responsively: adaptive mutation. Nat. Rev. Genet. **2**, 504–515 (2001)
21. Setubal, J.C., Meidanis, J.: Introduction to Computational Molecular Biology. PWS Pub. (1997)
22. Smith, T.F., Waterman, M.S.: Identification of common molecular subsequences. J. Mol. Biol. **147**(1), 195–197 (1981)
23. Sun, Y., et al.: ICDAR 2019 competition on large-scale street view text with partial labeling-RRC-LSVT. In: Proceedings of ICDAR, pp. 1557–1562. IEEE (2019)
24. Warshall, S.: A theorem on Boolean matrices. J. ACM **9**(1), 11–12 (1962)
25. Yujian, L., Bo, L.: A normalized Levenshtein distance metric. IEEE T. Pattern Anal. **29**(6), 1091–1095 (2007)