ACTA ET COMMENTATIONES UNIVERSITATIS TARTUENSIS DE MATHEMATICA
Volume 21, Number 1, June 2017
Available online at http://acutm.math.ut.ee

On expected score of cellwise alignments

Riho Klement and Jüri Lember

Abstract. We consider certain suboptimal alignments of two independent i.i.d. random sequences from a finite alphabet $\mathcal{A} = \{1, \dots, K\}$, both sequences having length n. In particular, we focus on so-called *cellwise* alignments, where in the first step so many 1-s as possible are aligned. These aligned 1-s define *cells* and the rest of the alignment is defined so that the already existing alignment of 1-s remains unchanged. We show that as n grows, for any cellwise alignment, the average score of a cell tends to the expected score of a random cell, a.s. Moreover, we show that a large deviation inequality holds. The second part of the paper is devoted to calculating the expected score of certain cellwise alignment referred to as *priority letter alignment*. In this alignment, inside every cell first all 2-s are aligned. Then all 3-s are aligned, but in such way that the already existing alignment of 2-s remains unchanged. Then we continue with 4-s and so on. Although easy to describe, for K bigger than 3 the exact formula for expected score is not that straightforward to find. We present a recursive formula for calculating the expected score.

1. Introduction

Throughout this paper, $X := (X_1, X_2, \dots, X_n)$ and $Y := (Y_1, Y_2, \dots, Y_n)$ are two random vectors, usually referred to as sequences, so that all random variables X_i and Y_i, $i = 1, \dots, n$ take their values in a fixed finite alphabet $\mathcal{A} = \{1, \dots, K\}$. We study the properties of certain similarity measures of X and Y. The problem of measuring the similarity of two sequences is central to many areas of applications including computational molecular biology [2, 3, 5, 14, 18] and computational linguistics, e.g., [10, 11, 12]. A popular measure of similarity is the length of the *longest common subsequence* (LCS). A longest common subsequence of X and Y is any common subsequence that has the longest possible length. Let L_n be the length of LCS. Formally,

Received December 6, 2016.

2010 *Mathematics Subject Classification.* 05A05; 60C05; 60F10.

Key words and phrases. Longest common subsequence; optimal alignment; large deviation inequality.

http://dx.doi.org/10.12697/ACUTM.2017.21.10

L_n is the biggest k such that there exist two subsets of indices – *an alignment* – $\{i_1, \ldots, i_k\}, \{j_1, \ldots, j_k\} \subset \{1, \ldots, n\}$ satisfying $i_1 < i_2 < \ldots < i_k$, $j_1 < j_2 < \ldots < j_k$, and $X_{i_1} = Y_{j_1}, X_{i_2} = Y_{j_2}, \ldots, X_{i_k} = Y_{j_k}$. Any alignment corresponding to maximal k is called *optimal alignment.* LCS (or equivalently, an optimal alignment) is typically not unique, but all of them can be found by dynamic programming algorithm called *Smith–Waterman* algorithm, see, e.g., [2, 5] with complexity $O(n^2)$. However, for very long sequences, this complexity can be still too high and so one seeks for common subsequences having the length close to L_n. Besides the computational cheapness those *suboptimal* common subsequences (sometimes called suboptimal alignments) are often analytically easily tractable [16, 13]. This is a clear advantage, since it is well known that L_n, although easy to define, is very difficult to analyze.

A straightforward way to define a suboptimal common subsequence is the following: choose a letter, say $1 \in \mathcal{A}$. Now going from left to right, align as many 1-s in both sequences as possible. The result is a subsequence with the length $N_1^x \wedge N_1^y$, where N_1^x and N_1^y are the numbers of 1-s in X and Y, respectively. The aligned pairs of 1-s divide the sequences into *cells*, where the first cell consists of pieces of X and Y-sequences up to the first aligned pair of 1-s (including the 1-s); the second cell consists of the pieces of X and Y-sequences up to the second aligned pair of 1-s and so on (see Figure 1).

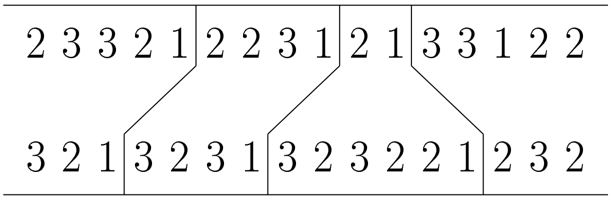

FIGURE 1. An example of cells.

Such alignment is meaningful if the 1-s have relatively high frequency in both sequences. However, after aligning 1-s, one can improve the whole alignment by aligning also the letters $\{2, \ldots, K\}$ inside every cell. This means that aligning the rest of the letters should not disturb the already existing alignment of 1-s. There are several possible ways to align the letters inside a cell, we shall call all obtained alignments *cellwise* alignments, because they share a common feature – first align 1-s to define cells and then perform any alignment inside the cells. We require that all inside-cell alignments should be performed along the same rule. Let Z_i be the number of aligned letters (including the pair of 1-s) in i-th cell. Then the length of the obtained pairs of common subsequence – *the score* – is

$$B_n := \sum_{i=1}^{N_1^x \wedge N_1^y} Z_i. \tag{1.1}$$

In what follows, we shall study the long run behavior of different cellwise alignments. To be able to distinguish the related (dependent) sequences from unrelated (independent) ones, one has to study the long-run behavior of B_n in the case X and Y are independent. So throughout the paper, we consider the case, where X and Y are both independent i.i.d. sequences, but the distribution of random variables X_i and Y_i can be different. In what follows, the distributions of X_i and Y_j are denoted by $P := (p_1, \ldots, p_K)$ and $Q := (q_1, \ldots, q_K)$, respectively. Note that although X and Y are independent i.i.d. sequences, due to the fact that X and Y have fixed length n, the scores of the cells $Z_1, Z_2, \ldots, Z_{N_1^x \wedge N_1^y}$ are not i.i.d.. However, if instead of the fixed length n, the both sequences had random length up to the m-th cell, then obviously $Z_1, \ldots Z_m$ would be an i.i.d. sequence and laws of large numbers, large deviation inequalities and many other classical results would immediately follow. This observation suggests that similar results could also hold for score B_n, and showing that is one objective of the present paper. In particular, in Section 2 we prove the following large deviation inequality (Theorem 2.2): for every $\Delta > 0$ there exist $A(\Delta) < \infty$, $b(\Delta) > 0$ such that

$$P\big(|\frac{B_n}{n} - \gamma| > \Delta\big) \leq A \exp[-bn], \quad \forall n. \tag{1.2}$$

Here $\gamma > 0$ is a constant that depends on the distributions P and Q, the inside-cell alignment methods, but not on n. Obviously from (1.2) it follows that

$$\frac{B_n}{n} \to \gamma, \quad \text{a.s.} \quad \text{and} \quad \text{in} \quad L_1.$$

The bigger γ, the better is the cellwise alignment so that knowing the value of γ (or being able to calculate it) provides valuable information about the performance of B_n. Another application of (1.2) is the statistical tests of testing independence of X and Y. Knowing the constant A, b and γ, such tests can be easily constructed. Those tests would be non-asymptotical, because they hold for any n, not only for n big enough.

Motivated by above-stated arguments, the second half of the paper, Section 3, deals with exact calculation of the constant γ for certain cellwise alignments. We shall show that

$$\gamma = \frac{EZ_1}{(E\tau_1^x \vee E\tau_1^y)},$$

where EZ_1 is the expected score of the first (and any) cell of infinite sequences $X_1, X_2, \ldots$ and $Y_1, Y_2, \ldots$, $E\tau_1^x$ is the expected length of the first (and any) cell of $X_1, X_2, \ldots$ and $E\tau_1^y$ is the expected length of the first (and any) cell of $Y_1, Y_2, \ldots$ (see the beginning of Section 2 for the formal definition). Since all cells have geometric distribution, their expected lengths are straightforward to find. However, depending on the alignment, to find EZ_1 might be difficult. In the paper, we focus on finding EZ_1 of an alignment called *priority letter alignment*. This alignment first aligns all 2-s inside a cell. Then, it aligns all 3-s without disturbing the already existing alignment of 2-s (and 1-s, because cells are already fixed). Then it aligns 4-s without disturbing the already existing alignments of 1-s, 2-s and 3-s and so on, proceeding always from left to right in a cell or subcell. It turns out that when the number of letters K is bigger than two or three, the exact calculation of EZ_1 might be cumbersome. For example, finding EZ_1 for priority letter alignment is almost trivial if $K = 2$, but the formula gets more and more intransparent when K grows. Therefore, we present it recursively in K. It turns out that in order to show the formula, it is convenient to represent a cell as a tree, this formalism is established in Subsection 3.1. Being able to calculate γ allows us to find the ordering of the alphabet $\mathcal{A}$ such that γ would be maximal. Intuitively, one could expect the ordering be such that

$$p_1 \wedge q_1 \geq p_2 \wedge q_2 \geq \cdots \geq p_K \wedge q_K. \tag{1.3}$$

It turns out that this intuition would fail and we finish the paper with a counterexample showing that it is not always the case.

To conclude the introduction, let us also mention that being able to calculate γ exactly is a clear advantage of using suboptimal common subsequences over LCS. Namely, from subadditivity it follows that when (X, Y) are the first n observations from an ergodic process, then there exists a constant γ^* such that $L_n/n \to \gamma*$, a.s. (see, e.g., [2, 7, 6]). The constant γ^* is sometimes called *Chvatal–Sankoff constant* referring to the seminal paper of Chvatal and Sankoff [4], where the existence of γ^* was observed. However, after more than 40 years of study, the exact value of γ^* is not exactly known even for the simplest case where X and Y are independent i.i.d. Bernoulli sequences with probability 1/2. For an overview of the research related with estimating Chvatal–Sankoff constant as well as for some bounds, see [7]. For a suboptimal alignment, surely $\gamma < \gamma^*$, but if the difference is not that big, then the lower score could be a fair price for computational cheapeness, analytical formulas and well-understandable statistical properties. Besides practical use, knowing a lower bound of γ^* might help researchers restrict the search space of optimal alignments in theoretical studies of the longest common subsequence, see, e.g., [8, 9].

The study in the present paper continues the one in [1], where several cellwise and other suboptimal alignments of binary sequences ($K = 2$) were considered. The simulations in [1] show that many suboptimal alignments perform rather well, provided that the distributions P and Q are asymmetrical, i.e., $p_1 = q_2 \neq q_1 = p_2$. As it becomes evident from the present paper, the analysis of cell-wise alignments, in particular the exact calculation of γ , becomes more involved when the number of letters K increases.

2. Large deviation inequality

Let us first formally define the cells in one sequence. Let $X_1, X_2, \ldots$ be an i.i.d. sequence from $\mathcal{A}$. Let

$$\tau_0^x = 0, \quad \tau_1^x = \min\{r \geq 1 : X_r = 1\}, \ \ldots, \ \tau_k^x = \min\{r \geq \tau_{k-1} + 1 : X_r = 1\}.$$

We call

$$C_k^x := (X_{\tau_{k-1}^x+1}, \ldots, X_{\tau_k^x}), \quad k = 1, 2, \ldots$$

a X-*cell*. Similarly, we define Y-cells $C_1^y, C_2^y, \ldots$. Let

$$f : \mathcal{A} \times \mathcal{A} \to \mathbb{R}^+$$

be a function that assigns to a pair of cells a non-negative score. Since we are considering common subsequences, clearly the score inside the cell cannot be bigger than the length of X-cell and the length of Y-cell. Therefore, for every $i = 1, 2, \ldots$,

$$f(C_i^x, C_i^y) \leq (\tau_i^x - \tau_{i-1}^x) \wedge (\tau_i^y - \tau_{i-1}^y). \tag{2.1}$$

Since sequences $X_1, X_2, \ldots$ and $Y_1, Y_2, \ldots$ are independent i.i.d. sequences, clearly the random variables $Z_1, Z_2, \ldots$, where $Z_i := f(C_i^x, C_i^y)$ are i.i.d. as well. Recall that N_1^x (respectively, N_1^y) is the number of 1-s in X (respectively, Y) sequence. Formally, $N_1^x = \max\{k : \tau_k^x \leq n\}(N_1^y = \max\{k : \tau_k^y \leq n\})$. Our object of interest is the cellwise score B_n defined as in (1.1).

In what follows, let $G(p)$ stand for geometric distribution with parameter p. Thus, if $\tau \sim G(p)$, then $P(\tau = k) = (1-p)^{k-1}p$. Clearly $\tau_k^x - \tau_{k-1}^x \sim G(p_1)$,$\tau_k^y - \tau_{k-1}^y \sim G(q_1)$, $k = 1, 2, \ldots$, and therefore for any fixed m_1, m_2, it holds

$$P\big(N_1^x > m_2\big) \leq P(\tau_{m_2}^x < n) = P\big(\sum_{i=1}^{m_2} G_i < n\big) = P\big(\sum_{i=1}^{m_2} G_i < \frac{n}{m_2} \cdot m_2\big) \tag{2.2}$$

$$P\big(N_1^x < m_1\big) = P(\tau_{m_1}^x > n) = P\big(\sum_{i=1}^{m_1} G_i > n\big) = P\big(\sum_{i=1}^{m_1} G_i > \frac{n}{m_1} \cdot m_1\big), \tag{2.3}$$

where $G_1, G_2, \ldots$ are i.i.d. random variables with distribution $G(p_1)$. The inequality in (2.2) follows from the observation that if $X_1, \ldots, X_n$ contains strictly more than m_2 1-s, then $\tau_{m_2}^x < n$. The equality in (2.3) follows from the observation that $X_1, \dots, X_n$ contains strictly less than m_1 1-s if and only if $\tau^x_{m_1} > n$.

For bounding the right hand side of (2.2) and (2.3), the following bounds are useful: for any $A > 1$ and $\alpha < 1$,

$$P\Big(\sum_{i=1}^m G_i > \frac{A}{p_1}m\Big) \leq \exp[-C(A)m], \quad P\Big(\sum_{i=1}^m G_i < \frac{\alpha}{p_1}m\Big) \leq \exp[-C(\alpha)m], \tag{2.4}$$

where $C(A) = A - 1 - \ln A$ and $C(\alpha) = \alpha - 1 - \ln \alpha$. For the proof, see [8]. Fix $p_1 > \Delta > 0$, define

$$m_1 = (p_1 - \Delta)n, \quad m_2 = (p_1 + \Delta)n,$$

and use (2.4) to deduce

$$\begin{aligned} P\Big(|N^x_1 - p_1 n| > \Delta n\Big) &\leq \exp[-C(\frac{p_1}{p_1 - \Delta})m_1] + \exp[-C(\frac{p_1}{p_1 + \Delta})m_2] \\ &\leq 2\exp[-D(p_1, \Delta)n], \end{aligned} \tag{2.5}$$

where

$$D(p_1, \Delta) := C(\frac{p_1}{p_1 - \Delta})(p_1 - \Delta) \wedge C(\frac{p_1}{p_1 + \Delta})(p_1 + \Delta).$$

Similarly

$$\begin{aligned} P\Big(|N^y_1 - q_1 n| > \Delta n\Big) &\leq \exp[-C(\frac{q_1}{q_1 - \Delta})m_1] + \exp[-C(\frac{q_1}{q_1 + \Delta})m_2] \\ &\leq 2\exp[-D(q_1, \Delta)n]. \end{aligned} \tag{2.6}$$

From (2.5) and (2.6), it follows that if $\Delta > 0$ is small enough, then

$$P\Big(\big|N^x_1 \wedge N^y_1 - (q_1 \wedge p_1)n\big| > \Delta n\Big) \leq 2\exp[-D(q_1, \Delta)n] + 2\exp[-D(p_1, \Delta)n]. \tag{2.7}$$

To see (2.7), note that when $p_1 < q_1$ and $\Delta < \frac{q_1 - p_1}{2}$, then

$$\{N^x_1 \wedge N^y_1 \neq N^x_1\} \subseteq \{|N^y_1 - q_1| > \Delta n\} \cup \{|N^x_1 - p_1| > \Delta n\}$$

and therefore

$$P\Big(\big|N^x_1 \wedge N^y_1 - (q_1 \wedge p_1)n\big| > \Delta n\Big) \leq P\Big(|N^y_1 - q_1| > \Delta n\Big) + P\Big(|N^x_1 - p_1| > \Delta n\Big).$$

The large deviation inequality (1.2) now follows from the following theorem proven in [15].

Theorem 2.1. *Let $\{Z_n\}_{n \geqslant 1}$ be non-negative random variables so that $EZ_i = \mu_Z$ and for every $\Delta > 0$ there exist constants $A_1(\Delta)$, $A_2(\Delta)$, $B_1(\Delta)$, $B_2(\Delta)$, $N_1(\Delta)$, $N_2(\Delta)$ so that*

$$P\left(\sum_{i=1}^n Z_i - \mu_Z n \leqslant -\Delta n\right) \leqslant A_1(\Delta)\exp[-B_1(\Delta)n], \quad \text{if } n > N_1(\Delta) \tag{2.8}$$

$$P\left(\sum_{i=1}^{n} Z_i - \mu_Z n \geqslant \Delta n\right) \leqslant A_2(\Delta)\exp[-B_2(\Delta)n], \quad \textit{if } n > N_2(\Delta), \tag{2.9}$$

and let $M(n)$ be a non-negative integer valued random variable, that might depend on the sequence $Z_1, Z_2, \ldots, Z_n$, suppose there exists $\mu > 0$ such that for every $\Delta_1 > 0$ there exist constants $A_3(\Delta_1), B_3(\Delta_1), N_3(\Delta_1)$ so that

$$P\left(|M(n) - \mu n| \geqslant \Delta_1 n\right) \leqslant A_3(\Delta_1)\exp[-B_3(\Delta_1)n], \ \textit{if } n \geqslant N_3(\Delta_1). \tag{2.10}$$

Then for all $\Delta > 0$ there exist constants $A(\Delta), B(\Delta)$ and $N(\Delta)$, so that

$$P\left(\left|\sum_{i=1}^{M(n)} Z_i - (\mu_Z \mu)\cdot n\right| > \Delta n\right) \leqslant A(\Delta)\exp[-B(\Delta)n], \ n \geqslant N(\Delta).$$

Moreover,

$$\begin{aligned}
&A(\Delta) := A_1(\varepsilon_1) + A_2(\varepsilon_2) + A_3(\varepsilon_3),\\
&B(\Delta) := \min\left\{\frac{1}{2}\left(\mu - \frac{\Delta}{2\mu_Z}\right)B_1(\varepsilon_1), \left(\mu + \frac{\Delta}{2\mu_Z}\right)B_2(\varepsilon_2), B_3(\varepsilon_3)\right\},\\
&N(\Delta) := \max\left\{N_1(\varepsilon_1), N_2(\varepsilon_2), N_3(\varepsilon_3), \frac{2}{\mu - \frac{\Delta}{2\mu_Z}}, \frac{2}{\mu + \frac{\Delta}{2\mu_Z}}\right\},\\
&\qquad \varepsilon_1 = \frac{\Delta}{4(\mu - \frac{\Delta}{2\mu_Z})}, \quad \varepsilon_2 = \frac{\Delta}{8(\mu + \frac{\Delta}{2\mu_Z})}, \quad \varepsilon_3 = \frac{\Delta}{2\mu_Z}.
\end{aligned}$$

From this theorem our main large deviation inequality almost immediately follows.

Theorem 2.2. *Let B_n be defined as in* (1.1) *and let*

$$\gamma = (p_1 \wedge q_1)EZ_1.$$

Then for every $\Delta > 0$ there exist $N(\Delta) < \infty$, $A(\Delta) < \infty$ and $b(\Delta) > 0$ such that

$$P\Big(|\frac{B_n}{n} - \gamma| \geq \Delta\Big) \leq A(\Delta)\exp[-b(\Delta)n], \quad n > N(\Delta). \tag{2.11}$$

Proof. The proof is straightforward application of Theorem 2.1. First, we have to show that the random variables $Z_1, Z_2, \ldots$ satisfy large deviation inequalities (2.8) and (2.9). By the theory of large deviations, it suffices to show that the moment generating function $M_Z(t) := E\exp[tZ_1]$ is finite in the neighborhood of 0. This immediately follows from (2.1), because $Z_1 \leq \tau_1$ and so for every $t \geq 0$

$$M_{Z_1}(t) \leq M_{\tau_1}(t).$$

Since $\tau_1 \sim G(p_1)$, we know that $M_{\tau_1}(t) < \infty$, when t is sufficiently close to 0. Thus the assumptions (2.8) and (2.9) are fulfilled with $\mu_Z = EZ_1$. Then take $M(n) = N_1^x \wedge N_1^y$ and note that (2.7) implies (2.10) with $\mu = p_1 \wedge q_1$. Thus all assumptions of Theorem 2.1 are fulfilled and so (2.11) holds. □

Note that when (2.11) holds for $n > N$, then there exist maybe different constants A' and b' so that with these constants (2.11) holds for $n \geq 1$.

3. Priority letter alignment: Recursion for γ

In this section, we study the priority letter alignment described in the introduction and we develop a recursive formula for calculating EZ_1. Before doing that, we need some additional definitions.

3.1. Cells and trees. Let us consider the X-cells on i.i.d. sequence $X = X_1, X_2, \ldots$. Clearly the sequence X is a concatenation of i.i.d. cells: $X = C_1, C_2, \ldots$. Every cell ends with 1 and this is the only 1 in the whole cell. Let us study a cell $C := C_1$ in more details. Let $N_2 - 1$ be the number of 2-s in a cell C. The 2-s in C partition a cell into i.i.d *2-subcells*

$$C = (C_1^2, \cdots, C_{N_2}^2),$$

where

$$C_k^2 := (X_{\tau_{k-1}^2+1}, \ldots, X_{\tau_k^2}), \quad \tau_0^2 := 0, \quad \tau_k^2 = \min\{r \geq \tau_{k-1}^2+1 : X_r \in \{1,2\}\}.$$

A 2-subcell ends with 1 or 2 and the rest of the letters in a 2-subcell are $3, \ldots, K$. Of course, it can be that there are no 2-s in a cell and so the only 2-subcell coincides with the original cell. Similarly, every 2-subcell C_i^2 can be further partitioned into *3-subcells* $C_{i,j}^3$, $j = 1, \ldots, N_{3,i}$, that consists of letters $4, \ldots, K$ (this part can be empty as well) but ends with a letter in $\{1, 2, 3\}$. Every 3-subcell can be further partitioned into 4-subcells and so on. Thus, a $(K-1)$-subcell is nothing but a (possible empty) block of K-s ending with any letter not being K. On figure 2 there is an example of subcells in the first cell of one short sequence in case $K = 4$. 3-subcells are grouped by 2-subcells, groups separated by vertical lines.

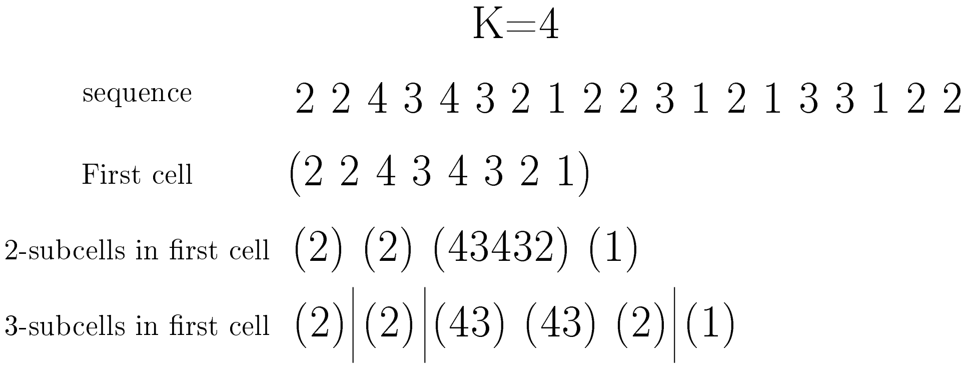

Figure 2. An example of subcells.

Since $X_1, X_2, \dots$ are i.i.d. random variables, we see that the number of 2-subcells in a cell C is geometrically distributed: $N_2 \sim G(\frac{p_1}{p_1+p_2})$, the number of 3-subcells in a 2-subcell C^2 is geometrically distributed $N_3 \sim G(\frac{p_1+p_2}{p_1+p_2+p_3})$. Therefore, for any $l < K$, the number of l-subcells in a $(l-1)$-subcell C^{l-1} is geometrically distributed $N_l \sim G(\frac{p_1+\cdots+p_{l-1}}{p_1+\cdots+p_l})$. Similarly, the number of K-s in a $K-1$-subcell is $N_K - 1$, where $N_K \sim G(1-p_K)$.

In what follows, it is convenient to represent a (random) cell as a (random) tree, where the root has $R_2 := N_2$ children each corresponding to a 2-subcell. The i-th child of the root (node at level 2) has $N_{3,i}$ children, each corresponding to a 3-subcell. The number of nodes at level 3 is $R_3 := \sum_{i=1}^{N_2} N_{3,i}$ and this is the number of 3-subcells. The i-th node at level 3 has $N_{4,i}$ children, all together there are $R_4 := \sum_{i=1}^{R_3} N_{4,i}$ nodes at level four. Recursively, thus,

$$R_l := \sum_{i=1}^{R_{l-1}} N_{l,i}, \quad l = 1, \dots, K-1, \quad (R_0 = 0, R_1 = 1). \tag{3.1}$$

Thus, every l-level node in the tree corresponds to a l-subcell, and if $l < K-1$, the number of its children equals to the number of its $l+1$-subcells. The total number of l-subcells is R_l. The children of the nodes in level $K-1$ are the leaves and the number of children of a node in level $K-1$ is the number of K-s in the corresponding subcell. Unlike other levels, we shall denote the number of children of i-th node in level $K-1$ as $N_{K,i}-1$, so that the total number of leaves is $R_K = \sum_{i=1}^{R_{K-1}}(N_{K,i}-1)$. Note that R_K is the number of K-s in the cell. Note that R_l is the number of $1, \dots, l$ letters in the cell and $R_{K-1}+R_K$ is the length of the cell. The number of l-letters in the cell is $R_l - R_{l-1}$, if $l < K$ (see an example in Figure 3).

Now, a random cell can be modelled as a random tree as follows: at first generate $R_2 = N_2$ children of the root, where $N_2 \sim G(\frac{p_1}{p_1+p_2})$. Then generate R_2 i.i.d. random variables $N_{3,1}, \dots, N_{3,R_2}$ from distribution $G(\frac{p_1+p_2}{p_1+p_2+p_3})$. These are the children of 2-level nodes. All together there are R_3 3-level nodes and then generate R_3 i.i.d. random variables $N_{4,1}, \dots, N_{4,R_3}$ from distribution $G(\frac{p_1+p_2+p_3}{p_1+\cdots+p_4})$ and so on. Thus $N_{l,1}, \dots, N_{l,R_{l-1}}$ are i.i.d. random variables from $G(\frac{p_1+\cdots+p_{l-1}}{p_1+\cdots+p_l})$ and their sum is R_l (recall (3.1)). From (3.1), it easily follows that

$$R_l \sim G(\frac{p_1}{p_1+\cdots+p_l}), \quad l = 1, \dots, K-1.$$

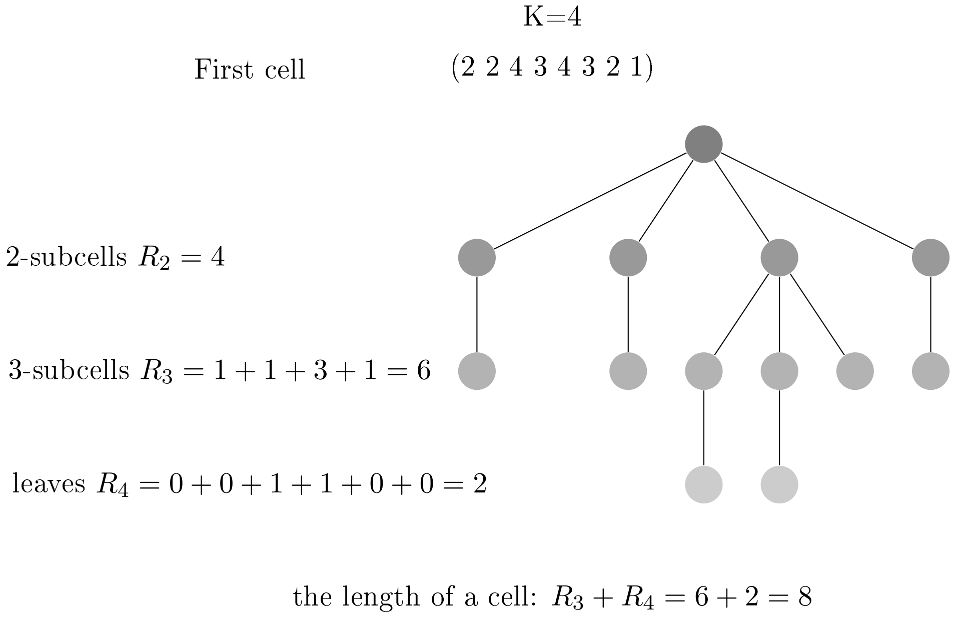
K=4
First cell
(2 2 4 3 4 3 2 1)
2-subcells $R_2 = 4$
3-subcells $R_3 = 1+1+3+1 = 6$
leaves $R_4 = 0+0+1+1+0+0 = 2$
the length of a cell: $R_3 + R_4 = 6+2 = 8$

FIGURE 3. Cell as a tree.

Finally, for the nodes at level $K-1$ generate again R_{K-1} i.i.d. random variables $N_{K,1}, \ldots, N_{K,R_{K-1}}$ from $G(1-p_K)$, but the number of leaves at node i is not $N_{K,i}$ as in other levels but $N_{K,i}-1$.

Let T be a random tree obtained like that and to specify the distribution, we sometimes write $T(p_1, \ldots, p_K)$. Let T_i^2, $i = 1, \ldots, N_2$ be subtrees at level 2. Clearly they are independent and every subtree is $T(p_1+p_2, p_2, \ldots, p_{K-1})$. This observation is important in recursion.

In what follows, we also consider the random trees, where every node has a different *offspring distribution* for the number of children. So the number of nodes at level 2 is distributed according to law P_1, attached to the root. Given N_2 children, we have thus $P_{2,1}, \ldots P_{2,N_2}$ distributions, each attached to a child. The i-th node at level 2 has $N_{3,i} \sim P_{2,i}$ children, and the j-th of them has distribution $P_{3,i,j}, j = 1, \ldots N_{3,i}$ according to which the number of children are distributed and so on. In this terminology, for a $T(p_1, \ldots, p_K)$-tree, $P_1 = G(\frac{p_1}{P_1+p_2})$, $P_{2,i} = G(\frac{p_1+p_2}{p_1+p_2+p_3})$, all nodes at level three have distribution $G(\frac{p_1+p_2+p_3}{p_1+p_2+p_3+p_4})$ and so on.

3.2. Alignments. Let $X = X_1, X_2, \dots$ and $Y = Y_1, Y_2, \dots$ be two independent i.i.d. sequences with laws $(p_1, \dots, p_K)$ and $(q_1, \dots, q_K)$, respectively. We are interested in calculating the expected score of an alignment $Ef(C^x, C^y)$, where C^x (C^y) is a random cell in X (in Y). For that we represent C^x as a random tree $T^x(p_1, \dots, p_K)$ and C^y as a random tree $T^y(q_1, \dots, q_K)$. In order to facilitate the calculation, we represent the (random) alignment as a (random) tree $T = f(T^x, T^y)$, where the number of aligned l letters ($l < K$) is $R_l - R_{l-1}$ and R_K is the number of aligned K's. Therefore, the score of the alignment is $g(T) := R_{K-1} + R_K$. Thus

$$Ef(C^x, C^y) = ER_{K-1} + ER_K = Eg(T),$$

and the tree-representation allows us to calculate $Eg(T)$ recursively, namely

$$Eg(T) = E\Big(\sum_{i=1}^{R_2} g(T_i^2)\Big). \tag{3.2}$$

We consider closely two alignments. In order to help the reader to understand the tree construction, we first consider so-called priority subcell alignment. After that, we focus on the priority letter alignment which is the main objective of our paper.

In this section, we use the same notation as previously, just adding superscript ^{x} or ^{y} to indicate the X or Y-sequences. For example, $N^x_{l,i}$ stands for the number of l-subcells in the i-th $l-1$ subcell of C^x. Let

$$M_2 := N_2^x \wedge N_2^y, \quad M_{l,i} := N^x_{l,i} \wedge N^y_{l,i}, \; l = 3, \dots, K.$$

3.2.1. *Priority subcell alignment.* Priority subcell alignment is the following: first, we align the maximal number of 2-s, proceeding from left to right. We obtain $R_2 - 1$ aligned letters, where $R_2 := M_2$. Then we align maximal number of 3-s in the first M_2 *2-subcells*, again proceeding from left to right. The number of 3-s we align in the i-th 2-subcell (out of first R_2 subcells) is $M_{3,i} - 1$. In total we align $R_3 - R_2$ letters 3, where

$$R_3 := \sum_{i=1}^{R_2} M_{3,i}.$$

In what follows, we consider these R_3 3-subcells only and so on, always proceeding from left to right during aligning letters in subcells. Thus, after aligning letters $(l-1)$, we have R_{l-1} $(l-1)$-subcells. In i-th $(l-1)$-subcell we align $M_{l,i} - 1$ letters (l), all together we shall have R_l l-subcells, where

$$R_l := \sum_{i=1}^{R_{l-1}} M_{l,i}, \quad l = 3, \dots, K-1. \tag{3.3}$$

Finally, after aligning all letters $K-1$, we end up with R_{K-1} $(K-1)$-subcells and in each of them, we align maximal number of K-letters. Formally, in i-th $(K-1)$-subcell we align $M_{K,i}-1$ letters K. The number of aligned K-s is

$$R_K := \sum_{i=1}^{R_{K-1}} (M_{l,i}-1). \tag{3.4}$$

In terms of trees, this procedure can be described as follows. First recall that the nodes have ordering corresponding to cells. Then take one of the trees and delete all subtrees starting from the nodes having index bigger than M_2. In other words, keep the first M_2 subtrees or, equivalently, M_2 nodes at level 2. Now consider M_2 subtrees $T_1^2,\ldots,T_{M_2}^2$ and for every subtree T_i^2 proceed so: delete all subtrees in level 2 (level 3 in original tree) starting from nodes having indexes bigger than $M_{3,i}$. Then repeat the same procedure for sub-subtrees. We end up with a reduced tree T obtained from two independent original trees $T^x(p_1,\cdots,p_K)$ and $T^y(q_1,\ldots,q_K)$. The number of nodes at level l in the reduced tree is R_l, defined as previously by (3.3) and (3.4). In tree T, all nodes at level l have the same offspring distribution

$$P_l = G(\rho_l),$$

where

$$\rho_l := 1-(\frac{p_{l+1}}{p_1+\cdots+p_{l+1}})(\frac{q_{l+1}}{q_1+\cdots+q_{l+1}}), \quad l=1,\ldots,K-1. \tag{3.5}$$

The score of the alignment $g(T)=R_{K-1}+R_K$ is easy to find recursively

$$Eg(T)=E(M_2)Eg(T^2)=\frac{1}{\rho_1}Eg(T^2)=\frac{1}{\rho_1}\frac{1}{\rho_2}Eg(T^3)=\cdots=\Big(\prod_{l=1}^{K-1}\rho_l\Big)^{-1}.$$

There is an example of priority subcell alignment in Figure 4, where the alignment of one pair of cells is represented both letterwise and in terms of trees in case $K=4$. One can see that there are five 2-subcells (four 2-s) in X and three 2-subcells (two 2-s) in Y which means that we can align two pairs of 2-s and get three aligned 2-subcells (separated by solid lines on the figure). Now we can't align any 3-s in first 2-subcell as there are not any, while we can align one pair of 3-s in the second and third 2-subcell. Now we can't align any more letters, because in the third 2-subcell we have two 3-subcells in X and five 3-subcells in Y and there are no 4-s in first two 3-subcells of Y (separated by dashed lines in the figure).

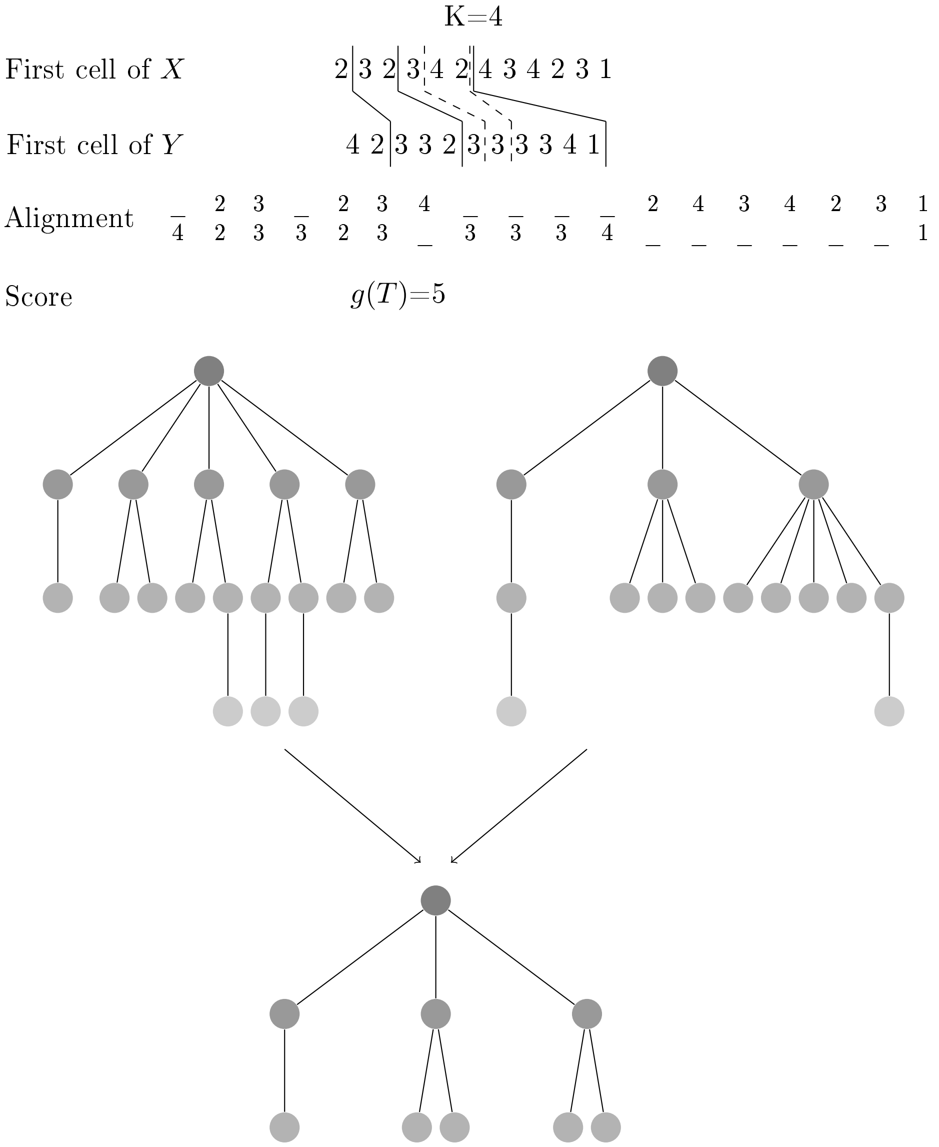

FIGURE 4. Priority subcell alignment.

3.2.2. *Priority letter alignment.* The description of priority letter alignment is very simple: At first we align as many 2-s as possible, thus M_2-1 2-s. Then we align as many 3-s as possible, without disturbing the existing alignment of 2-s. While aligning we always proceed again from left to right. The difference with the previously described alignment is that the 3-s outside M_2

first 2-subcells have now chance to be aligned as well. Therefore the score is bigger. Then after aligning 2-s and 3-s, so many 4-s as possible will be aligned, without disturbing the existing alignment of 2-s and 3-s and so on. Thus the score of priority letter alignment is greater or at least equal to the previously considered priority subcell alignment (see Figure 5).

K=4

First cell of X 2 3 2 3 4 2 4 3 4 2 3 1

First cell of Y 4 2 3 3 2 3 3 3 3 3 4 1

Alignment	_	2	3	_	2	3	4	2	4	3	4	2	3	_	_	1
	4	2	3	3	2	3	_	_	_	3	_	_	3	3	4	1

Score $g(T)$=7

FIGURE 5. Priority letter alignment.

In terms of trees, the procedure of building a new tree T out of two trees T^x and T^y is as follows: Suppose $N_2^x > N_2^y$. Recall that last $N_2^x - (N_2^y - 1)$ subtrees (from level 2) from T^x correspond to last $N_2^x - (N_2^y - 1)$ 2-subcells, separated by $N_2^x - N_2^y$ 2-s. Remove these 2-s from the X-sequence. Then these $N_2^x - N_2^y + 1$ subtrees become one subtree. The number of children of the root of this subtree is

$$1 + \sum_{j=N_2^y}^{N_2^x} (N_{3,j}^x - 1), \tag{3.6}$$

that equals to the number of 3-s plus one in the last $N_2^x - N_2^y + 1$ 2-subcells of X-sequence. Note that merging the redundant subtrees into one changes the structure of merged subtrees in every level. For example, in Figure 6 is shown that procedure based on one short pair of cells. One can see, that in original tree T^x nodes a, b and c (all being 4-s in X-sequence) are children of different nodes , but after last three nodes on level 2 of tree T^x become one, nodes a and b are now children of one node, while c has still different node as a parent. In terms of sequence it is explained by the fact, that originally 4-s a and b were separated by 2 which we ignore as it places behind first $M_2 - 1$ 2-s in X. Nodes b and c are separated by a 3 in X-sequence, thus they will remain separated after the first step.

If $N_2^y > N_2^x$, then remove the redundant 2-s in Y-sequence, merging the subtrees of T^y. If $N_2^y = N_2^x$, then leave the subtrees unchanged. After the merging procedure both trees, T^y and (reduced version of) T^x have M_2

First cell of X	2 3 2 3 4 2 4 3 4 2 3 1
First cell of Y	4 2 3 3 2 3 3 3 3 4 1

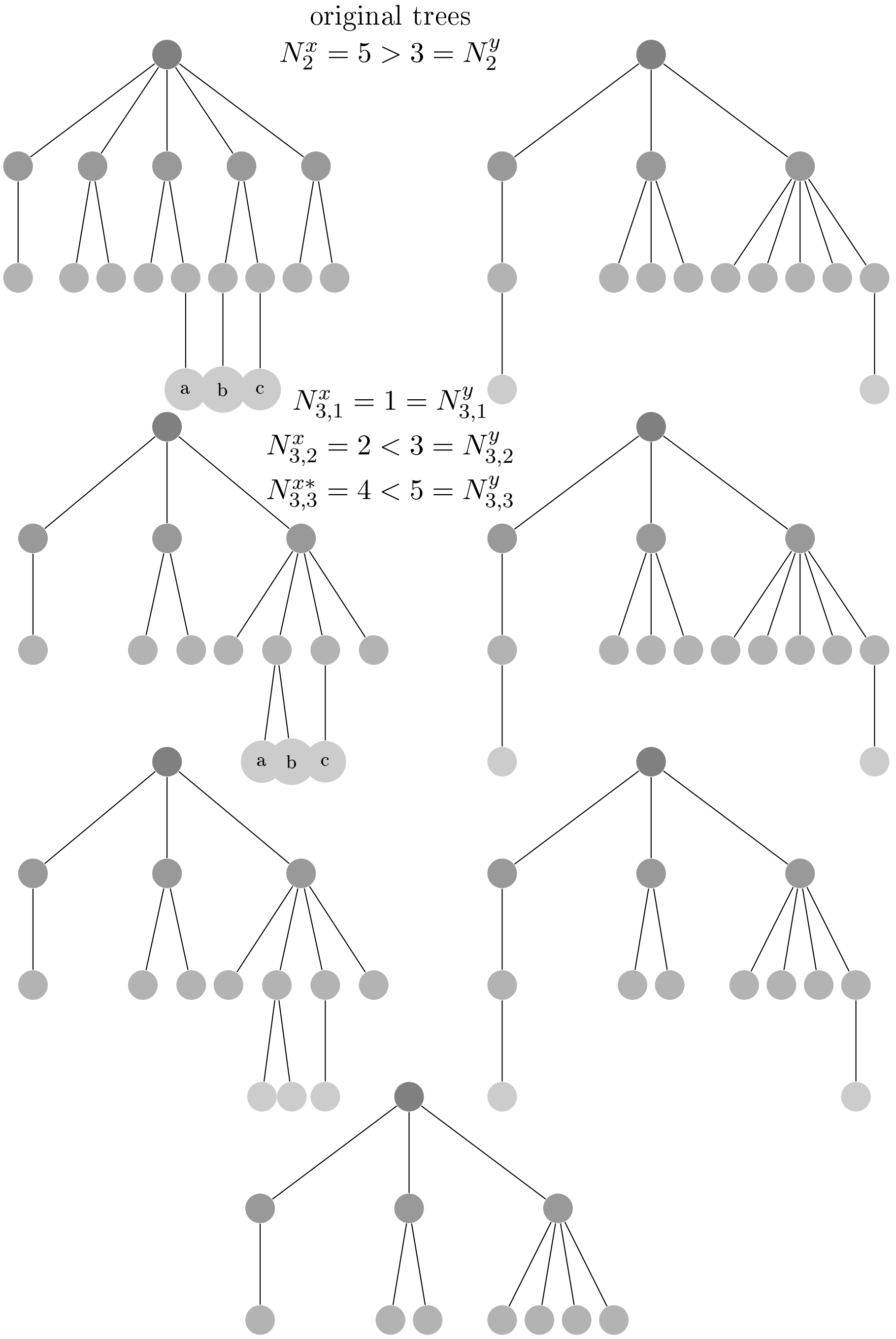
original trees
$N_2^x = 5 > 3 = N_2^y$
a
b
c
$N_{3,1}^x = 1 = N_{3,1}^y$
$N_{3,2}^x = 2 < 3 = N_{3,2}^y$
$N_{3,3}^{x*} = 4 < 5 = N_{3,3}^y$
a
b
c

FIGURE 6. Priority letter alignment represented by trees.

nodes at level 2. Now continue this procedure on level of T^2-subtrees – if, for example $N^x_{3,1} < N^y_{3,1}$, then remove last $N^y_{3,1} - N^x_{3,1}$ 3-s in the first 2-subcell on Y sequence and so the last $N^y_{3,1} - N^x_{3,1} + 1$ subtrees of T^{2y}_1-tree become one. Again, this merging changes all sub-subtrees of T^{2y}_1 as well. Then do the same procedure for the next subtrees, T^{2y}_2 and T^{2x}_2 and so on. After this procedure, in both (reduced) trees T^x and T^y, all nodes at level 2 have equal number of children: i-th node has $M_{3,i}$ children. Then start comparing the subtrees having roots at level 3 and so on. When finished, both reduced trees T^x and T^y are equal, and the output tree is T.

Let us now study the distribution of random output tree

$$T = T(p_1, \dots, p_K; q_1, \dots, q_K)$$

given the two original trees $T^x(p_1, \dots, p_K)$ and $T^y(q_1, \dots, q_K)$ are independent. Recall that T has M_2 nodes at level 2, so that T has M_2 independent subtrees T^2_i, $i = 1, \dots, M_2$, having roots at level 2. The independence comes from the fact that the number of subtrees N^x_2 (N^y_2) is independent of subtrees in T^x (T^y). The difference $N^x_2 - N^y_2$ has influence on the distribution of the last subtree, but the previous subtrees have no influence on the last one. The recursion is based on the observation that when $i = 1, \dots, M_2 - 1$, then the subtrees T^2_i are i.i.d. having the distribution

$$T(p_1 + p_2, p_3 \dots, p_K; q_1 + q_2, q_3, \dots, q_K). \tag{3.7}$$

To see that recall that the 2-level subtrees T^{2x}_i (or T^{2y}_i) are i.i.d. with distribution $T^x(p_1 + p_2, p_3, \dots, p_K)$ (or $T^y(q_1 + q_2, q_3, \dots, q_K)$) and along the first $M_2 - 1$ subtrees the whole alignment procedure is exactly the same as for the whole cell. In particular, these $M_2 - 1$ subtrees T^2_i have offspring distribution $P_{2,i} = G(\rho_2)$, $i = 1, \dots, M_2 - 1$, where ρ_2 is defined as in (3.5). The distribution of the last subtree $T^2_{M_2}$ is different. In what follows, we shall denote that distribution by

$$U^2(p_1, \dots, p_K; q_1, \dots, q_K).$$

When $N^x_2 \geq N^y_2$, then the conditional distribution of U^2 equals to that of

$$T\Big(\frac{p_1}{1-p_2}, \frac{p_3}{1-p_2}, \dots, \frac{p_K}{1-p_2}; q_1 + q_2, q_3, \dots, q_K\Big). \tag{3.8}$$

To see (3.8) recall that removing 2-s means that a letter 2 is not any more an end of a cell. Thus the number of 3-subcells is distributed along $G(\frac{p_1}{p_1+p_3})$, number of 4-subcells is distributed along $G(\frac{p_1+p_3}{p_1+p_3+p_4})$ and so on. So as for the whole tree $T(p_1, \dots, p_K; q_1, \dots, q_K)$ the number of first level subcells is distributed along $G(\frac{p_1}{p_1+p_2})$ and the number of second level subcells is distributed along $G(\frac{p_1+p_2}{p_1+p_2+p_3})$ and so on, then for subtree $T^2_{M_2}$ we need probabilities $a_1, a_3, \dots, a_K$, to be so that

$$\frac{a_1}{a_1+a_3}=\frac{p_1}{p_1+p_3},\ \frac{a_1+a_3+\cdots a_l}{a_1+a_3+\cdots a_{l+1}}=\frac{p_1+p_3+\cdots+p_l}{p_1+p_3+\cdots+p_{l+1}},\ l=3,\dots,K-1. \tag{3.9}$$

Since the distribution of tree is determined by its offspring distributions, we get that U^2 is distributed as

$$T(a_1,a_3,\dots,a_K;q_1+q_2,q_3,\dots,q_K).$$

Now take

$$a_1:=\frac{p_1}{1-p_2},\quad a_l:=\frac{p_l}{1-p_2},\quad l=3,\dots K,$$

and note that (3.9) holds. Similarly, when $N^x_2\leq N^y_2$, then the conditional distribution of U^2 equals to that of

$$T\Big(p_1+p_2,p_3,\dots,p_K;\frac{q_1}{1-q_2},\frac{q_3}{1-q_2},\dots,\frac{q_K}{1-q_2}\Big).$$

Finally, when $N^x_2=N^y_2$, then nothing is changed in level 2 so the conditional distribution of U^2 is exactly the same as the distribution of other subtrees T^2_i i.e. as in (3.7). In this case all subtrees $T^2_1,\dots,T^2_{M_2}$ are i.i.d..

Let us now study the distribution of the subtrees of U^2. Again, consider the case $N^x_2\geq N^y_2$. Then U^2 has M^*_3 subtrees, where (recall (3.6))

$$M^*_3=N^y_{3,M_2}\wedge\big(1+\sum_{j=N^y_2}^{N^x_2}(N^x_{3,j}-1)\big). \tag{3.10}$$

Again, all these subtrees are independent, the first M^*_3-1 of them have the distribution

$$T\Big(\frac{p_1+p_3}{1-p_2},\frac{p_4}{1-p_2},\dots,\frac{p_K}{1-p_2};q_1+q_2+q_3,q_4\dots,q_K\Big). \tag{3.11}$$

Let U^3 be M^*_3-th subtree of U^2. Since the procedure of building the trees is the same in every subcell, we see that under the condition $N^x_2\geq N^y_2$, U^3 has the distribution

$$U^2\Big(\frac{p_1}{1-p_2},\frac{p_3}{1-p_2},\dots,\frac{p_K}{1-p_2};q_1+q_2,q_3,\dots,q_K\Big). \tag{3.12}$$

If $N^x_2=N^y_2$, then U^3 has the distribution

$$U^2(p_1+p_2,p_3,\dots p_K;q_1+q_2,q_3,\dots,q_K). \tag{3.13}$$

Now clearly

$$Eg(T)=(EM_2-1)Eg(T^2)+Eg(U^2),$$
$$Eg(U^2)=E[g(U^2)|N^x_2\geq N^y_2]P(N^x_2\geq N^y_2)$$

$$\begin{aligned}
&\quad + E[g(U^2)|N_2^x \le N_2^y]P(N_2^x \le N_2^y) \\
&\quad - E[g(U^2)|N_2^x = N_2^y]P(N_2^x = N_2^y), \\
E[g(U^2)|N_2^x \ge N_2^y] &= \big(E[M_3^*|N_2^x \ge N_2^y] - 1\big)E[g(T^3)|N_2^x \ge N_2^y] \\
&\quad + E[g(U^3)|N_2^x \ge N_2^y], \\
E[g(U^2)|N_2^x \le N_2^y] &= \big(E[M_3^*|N_2^x \le N_2^y] - 1\big)E[g(T^3)|N_2^x \le N_2^y] \\
&\quad + E[g(U^3)|N_2^x \le N_2^y], \\
E[g(U^2)|N_2^x = N_2^y] &= \big(E[M_3^*|N_2^x = N_2^y] - 1\big)E[g(T^3)|N_2^x = N_2^y] \\
&\quad + E[g(U^3)|N_2^x = N_2^y],
\end{aligned}$$

where T^2 is a random tree with distribution (3.7) and T^3 is distributed as any of the first $M_3^* - 1$ subtrees of U^2. We know that under condition $N_2^x \ge N_2^y$, the subtree T^3 is distributed as (3.11), U^2 as (3.8) and U^3 as (3.12).

Let

$$\begin{aligned}
m_2(p_1,p_2;q_1,q_2) &:= EM_2 = \frac{1}{\rho_1}, \\
m_3^x(p_1,p_2,p_3;q_1,q_2,q_3) &:= E[M_3^*|N_2^x \ge N_2^y], \\
m_3^y(p_1,p_2,p_3;q_1,q_2,q_3) &:= E[M_3^*|N_2^x \le N_2^y] \\
m_3^0(p_1,p_2,p_3;q_1,q_2,q_3) &:= E[M_3^*|N_2^x = N_2^y], \\
p^x(p_1,p_2;q_1,q_2) &:= P(N_2^x \ge N_2^y), \\
p^y(p_1,p_2;q_1,q_2) &:= P(N_2^x \le N_2^y), \\
p^0(p_1,p_2;q_1,q_2) &:= P(N_2^x = N_2^y); \\
g_K(p_1,\dots,p_K;q_1,\dots,q_K) &:= Eg(T), \\
u_K(p_1,\dots,p_K;q_1,\dots,q_K) &:= Eg(U^2), \\
u_K^x(p_1,\dots,p_K;q_1,\dots,q_K) &:= E[g(U^2)|N_2^x \ge N_2^y], \\
u_K^y(p_1,\dots,p_K;q_1,\dots,q_K) &:= E[g(U^2)|N_2^x \le N_2^y], \\
u_K^0(p_1,\dots,p_K;q_1,\dots,q_K) &:= E[g(U^2)|N_2^x = N_2^y].
\end{aligned}$$

With this notation, we have

$$\begin{aligned}
g_K(p_1,\dots,p_K;q_1,\dots,q_K) &= (m_2(p_1,p_2;q_1,q_2) - 1) \\
&\quad \times g_{K-1}(p_1+p_2,p_3,\dots,p_K;q_1+q_2,q_3,\dots,q_K) \\
&\quad + u_K(p_1,\dots,p_K;q_1,\dots,q_K); \\
u_K(p_1,\dots,p_K;q_1,\dots,q_K) &= u_K^x(p_1,\dots,p_K;q_1,\dots,q_K)p^x(p_1,p_2;q_1,q_2) \\
&\quad + u_K^y(p_1,\dots,p_K;q_1,\dots,q_K)p^y(p_1,p_2;q_1,q_2) \\
&\quad - u_K^0(p_1,\dots,p_K;q_1,\dots,q_K)p^0(p_1,p_2;q_1,q_2).
\end{aligned}$$

Using (3.11) and (3.12) we get

$$\begin{aligned}u^x_K(p_1,\ldots,p_K;q_1,\ldots,q_K) &= \big(m^x_3(p_1,p_2,p_3;q_1,q_2,q_3)-1\big)\\ &\times g_{K-2}\Big(\frac{p_1+p_3}{1-p_2},\frac{p_4}{1-p_2},\ldots,\frac{p_K}{1-p_2};q_1+q_2+q_3,q_4\ldots,q_K\Big)\\ &+u_{K-1}\Big(\frac{p_1}{1-p_2},\frac{p_3}{1-p_2},\ldots,\frac{p_K}{1-p_2};q_1+q_2,q_3,\ldots,q_K\Big).\end{aligned}$$

Similarly,

$$\begin{aligned}u^y_K(p_1,\ldots,p_K;q_1,\ldots,q_K) &= \big(m^y_3(p_1,p_2,p_3;q_1,q_2,q_3)-1\big)\\ &\times g_{K-2}\Big(p_1+p_2+p_3,p_4,\ldots,p_K;\frac{q_1+q_3}{1-q_2},\frac{q_4}{1-q_2},\ldots,\frac{q_K}{1-q_2}\Big)\\ &+u_{K-1}\Big(p_1+p_2,p_3,\ldots,p_K;\frac{q_1}{1-q_2},\frac{q_3}{1-q_2},\ldots,\frac{q_K}{1-q_2}\Big),\end{aligned}$$

and from (3.13), it follows that

$$\begin{aligned}u^0_K(p_1,\ldots,p_K;q_1,\ldots,q_K) &= \big(m^0_3(p_1,p_2,p_3;q_1,q_2,q_3)-1\big)\\ &\times g_{K-2}(p_1+p_2+p_3,p_4,\ldots,p_K;q_1+q_2+q_3,q_4,\ldots,q_K)\\ &+u_{K-1}(p_1+p_2,p_3,\ldots,p_K;q_1+q_2,q_3,\ldots,q_K).\end{aligned}$$

Let us calculate $m^{\cdot}_3(p_1,p_2,p_3;q_1,q_2,q_3)$. For this note that under $N^x_2\geq N^y_2$, the difference $N^x_2-N^y_2+1$ has $G(\frac{p_1}{p_1+p_2})$ distribution. Then, because the variables $N^x_{3,j}$ are independent of $N^x_2-N^y_2$ it holds that

$$1+\sum_{j=N^y_2}^{N^x_2}(N^x_{3,j}-1)\sim G\big(\frac{p_1}{p_1+p_3}\big). \tag{3.14}$$

Reader can verify (3.14) via straightforward calculations. Therefore (3.10) is distributed as minimum of two independent geometrically distributed random variables, one having parameter $\frac{q_1+q_2}{q_1+q_2+q_3}$ and another $\frac{p_1}{p_1+p_3}$. Therefore, under $N^x_2-N^y_2\geq 0$ we have $M^*_3\sim G(\rho^x_2)$, where

$$\rho^x_2:=1-\big(\frac{p_3}{p_1+p_3}\big)\big(\frac{q_3}{q_1+q_2+q_3}\big).$$

Similarly, under $N^x_2-N^y_2\leq 0$ we have $M^*_3\sim G(\rho^y_2)$, where

$$\rho^y_2:=1-\big(\frac{q_3}{q_1+q_3}\big)\big(\frac{p_3}{p_1+p_2+p_3}\big).$$

Finally, under $N^x_2-N^y_2=0$, M^*_3 has the same distribution as M_3, thus $G(\rho_2)$. Therefore,

$$\begin{aligned}m^x_3(p_1,p_2,p_3;q_1,q_2,q_3) &= \frac{1}{\rho^x_2},\\ m^y_3(p_1,p_2,p_3;q_1,q_2,q_3) &= \frac{1}{\rho^y_2},\end{aligned}$$

$$m_3^0(p_1,p_2,p_3;q_1,q_2,q_3)=\frac{1}{\rho_2}.$$

Finally, let us find $p^{\cdot}(p_1,p_2;q_1,q_2)$. The reader can easily check that

$$\begin{aligned}
p^0(p_1,p_2;q_1,q_2)&=\frac{p_1q_1}{(p_1+p_2)(q_1+q_2)-p_2q_2},\\
p^y(p_1,p_2;q_1,q_2)&=\frac{p_1(q_1+q_2)}{(p_1+p_2)(q_1+q_2)-p_2q_2},\\
p^x(p_1,p_2;q_1,q_2)&=\frac{q_1(p_1+p_2)}{(p_1+p_2)(q_1+q_2)-p_2q_2}.
\end{aligned}$$

Now all components of our recursion are known and given, for every two $K-1$-dimensional probability vectors $(p'_1,\dots,p'_{K-1})$ and $(q'_1,\dots,q'_{K-1})$, one can calculate

$$g_{K-1}(p'_1,\dots,p'_{K-1};q'_1,\dots,q'_{K-1}),\quad u_{K-1}(p'_1,\dots,p'_{K-1};q'_1,\dots,q'_{K-1}).$$

Using these functions, one can also calculate $g_K(p_1,\dots,p_K;q_1,\dots,q_K)$ and $u_K(p_1,\dots,p_K;q_1,\dots,q_K)$.

Let us specify the beginning of the recursion. Take $K=3$. Then

$$\begin{aligned}
g_3(p_1,p_2,p_3;q_1,q_2,q_3)&=(m_2(p_1,p_2;q_1,q_2)-1)g_2(p_1+p_2,p_3;q_1+q_2,q_3)\\
&\quad+u_3(p_1,p_2,p_3;q_1,q_2,q_3).
\end{aligned}$$

For two level tree, the score of the cell is just the number of nodes, hence

$$\begin{aligned}
g_2(p_1+p_2,p_3;q_1+q_2,q_3)&=m_2(p_1+p_2,p_3;q_1+q_2,q_3),\\
u_3(p_1,p_2,p_3;q_1,q_2,q_3)&=m_3^x(p_1,p_2,p_3;q_1,q_2,q_3)p^x(p_1,p_2;q_1,q_2)\\
&\quad+m_3^y(p_1,p_2,p_3;q_1,q_2,q_3)p^y(p_1,p_2;q_1,q_2)\\
&\quad-m_3^0(p_1,p_2,p_3;q_1,q_2,q_3)p^0(p_1,p_2;q_1,q_2).
\end{aligned}$$

Let us see how the recursion also applies for $K=4$. So,

$$\begin{aligned}
g_4(p_1,p_2,p_3,p_4;q_1,q_2,q_3,q_4)&=\big(m_2(p_1,p_2;q_1,q_2)-1\big)\\
&\quad\times g_3(p_1+p_2,p_3,p_4;q_1+q_2,q_3,q_4)\\
&\quad+u_4(p_1,p_2,p_3,p_4;q_1,q_2,q_3,q_4),\\
u_4(p_1,p_2,p_3,p_4;q_1,q_2,q_3,q_4)&=u_4^x(p_1,p_2,p_3,p_4;q_1,q_2,q_3,q_4)p^x(p_1,p_2;q_1,q_2)\\
&\quad+u_4^y(p_1,p_2,p_3,p_4;q_1,q_2,q_3,q_4)p^y(p_1,p_2;q_1,q_2)\\
&\quad-u_4^0(p_1,p_2,p_3,p_4;q_1,q_2,q_3,q_4)p^0(p_1,p_2;q_1,q_2),\\
u_4^x(p_1,p_2,p_3,p_4;q_1,q_2,q_3,q_4)&=\big(m_3^x(p_1,p_2,p_3;q_1,q_2,q_3)-1\big)\\
&\quad\times g_2\big(\frac{p_1+p_3}{1-p_2},\frac{p_4}{1-p_2};q_1+q_2+q_3,q_4\big)\\
&\quad+u_3\big(\frac{p_1}{1-p_2},\frac{p_3}{1-p_2},\frac{p_4}{1-p_2};q_1+q_2,q_3,q_4\big),
\end{aligned}$$

$$\begin{aligned}
u_4^y(p_1,p_2,p_3,p_4;q_1,q_2,q_3,q_4) = {} & \big(m_3^y(p_1,p_2,p_3;q_1,q_2,q_3)-1\big) \\
& \times g_2\big(p_1+p_2+p_3,p_4;\frac{q_1+q_3}{1-q_2},\frac{q_4}{1-q_2}\big) \\
& + u_3\big(p_1+p_2,p_3,p_4;\frac{q_1}{1-q_2},\frac{q_3}{1-q_2},\frac{q_4}{1-q_2}\big), \\
u_4^0(p_1,p_2,p_3,p_4;q_1,q_2,q_3,q_4) = {} & \big(m_3^0(p_1,p_2,p_3;q_1,q_2,q_3)-1\big) \\
& \times g_2\big(p_1+p_2+p_3,p_4;q_1+q_2+q_3,q_4\big) \\
& + u_3\big(p_1+p_2,p_3,p_4;q_1+q_2,q_3,q_4\big).
\end{aligned}$$

4. Comparing the formulas

Let us compare $EZ_1 =: g_K(p_1,\dots,p_K;q_1,\dots,q_K)$ of priority letter alignment with $EZ_1 =: g'_K(p_1,\dots,p_K;q_1,\dots,q_K)$ of priority subcell alignment. Clearly $g_K \geq g'_K$, but as we see, the difference depends on the distributions. Moreover, we compare both numbers with the best possible cellwise score, namely the expected length of LCS. Let that be $g^*_K(p_1,\dots,p_K;q_1,\dots,q_K)$. This function cannot be calculated recursively, so we estimate it by simulations. In the second part of the section, we study the order of letters $2,\dots,K$ that would give the maximal g_K. We present a counterexample showing that the intuitively best ordering (1.3) does not necessarily give the biggest g_K. Since due to the recursive formula, for any ordering of $(p_1,\dots,p_K)$ and $(q_1,\dots,q_K)$, we can calculate g_K and the corresponding $\gamma = g_K/(E\tau_1^x \vee E\tau_1^y)$. This means that in practice the best ordering can be easily found. Of course the probabilities p_i and q_i might not always be known, but can be easily estimated via relative frequencies.

4.1. Comparison of methods. We first compare two described methods: priority subcell alignment and priority letter alignment. Let $K=4$. During the comparison we always use the intuitively best ordering of letters (1.3). We define symmetrical distribution as follows

$$\begin{array}{ccccc}
 & 1 & 2 & 3 & 4 \\
P & 1/4+\varepsilon & 1/4+\varepsilon & 1/4-\varepsilon & 1/4-\varepsilon \\
Q & 1/4+\varepsilon & 1/4+\varepsilon & 1/4-\varepsilon & 1/4-\varepsilon,
\end{array} \qquad (4.1)$$

where $\varepsilon \in [0,\frac{1}{4}]$. In Figure 7 there are presented theoretical expected values of constants $\gamma = (p_1 \wedge q_1)g_K$ (priority letter alignment) and $\gamma' = (p_1 \wedge q_1)g'_K$ (priority subcell alignment) in case of every $\varepsilon \in [0,\frac{1}{4}]$. One can see that if ε is close to 0, we have a distribution close to uniform distribution and in that case the difference is relatively big. The more ε increases, the less there are 3-s and 4-s in sequences and the smaller is the advantage of priority letter alignment. This is fully understandable, because the case $\epsilon = 1/4$

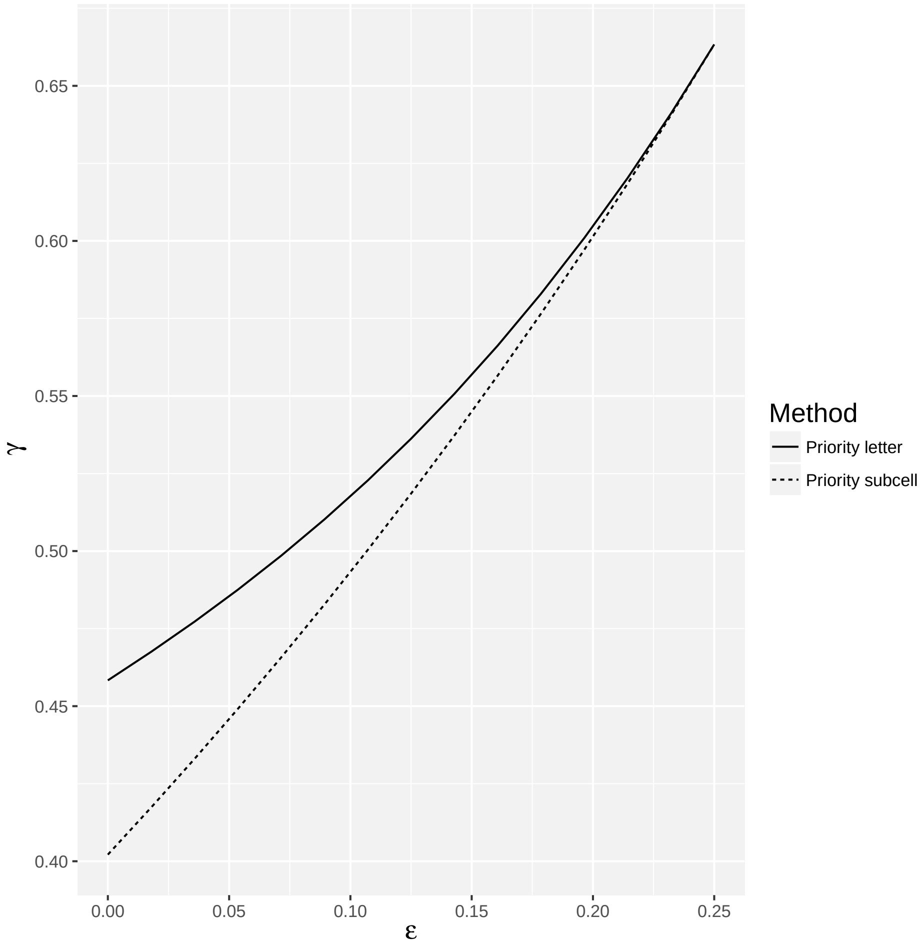

FIGURE 7. Comparison of priority letter and priority subcell alignments.

corresponds to the case $K = 2$ and for binary sequences the priority letter and the priority subcell alignments are the same.

Now we look at the same P and Q, but we try to get an overview on how far from the score of the LCS is our priority letter alignment. To get a better comparison on the methods we add one more alignment – *cellwise LCS*, where we first divide our sequences into cells and then we find LCS in all the pairs of cells. The last cell is "unfinished", because it does not necessarily end with 1. However, it is taken into calculations as well. Asymptotically the effect of the last unfinished cell is negligible. When one sequence has more cells than another, then these redundant cells are considered as the one

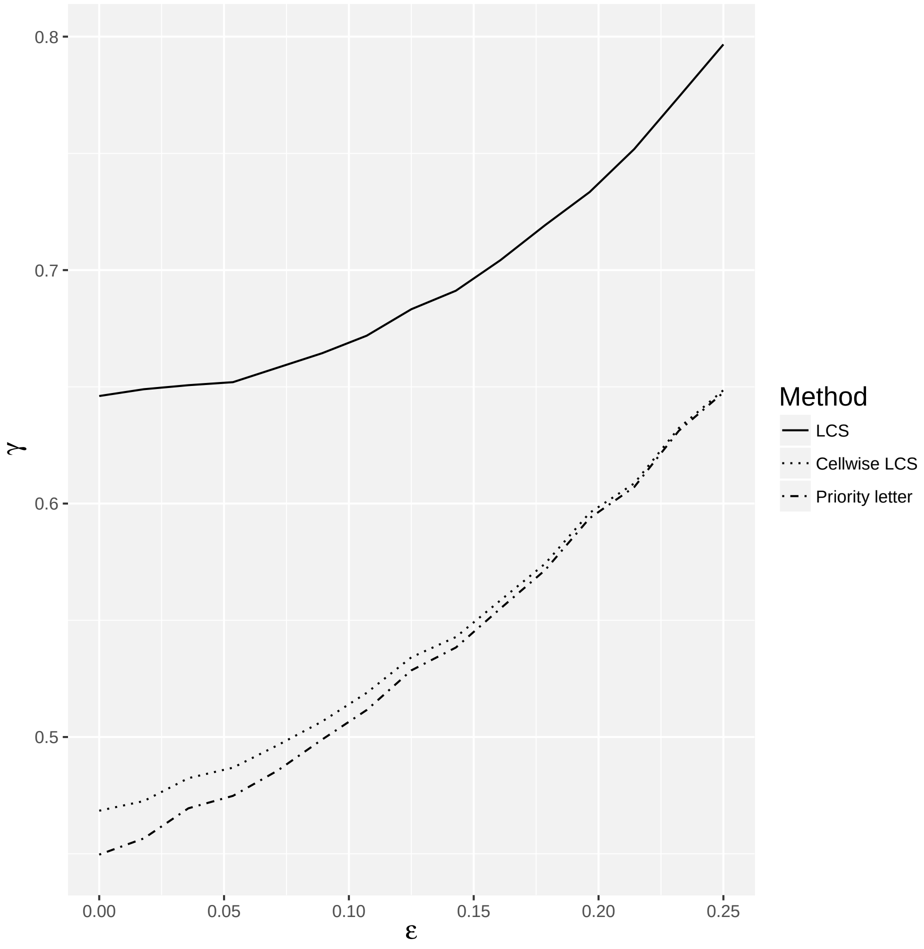

FIGURE 8. Comparison of Priority letter alignment with LCS methods.

big cell and still used in comparison. The sum of all the scores of LCS from every cell is the score of cellwise LCS alignment. We know, that the length of global LCS cannot be smaller than the length of cellwise LCS and priority letter alignment cannot do better than cellwise LCS. In Figure 8 there are presented all three described lines. For 15 different ε value, 30 simulations with sequence length 2000 were made and an average was calculated. One can see, that again the biggest difference between all the methods is when the distribution is close to the uniform distribution. When ε increases, the number of 3-s and 4-s decreases and the number of cells increases which means that the priority letter alignment is basically same as cellwise LCS. The global LCS is still substantially better than the others.

4.2. Ordering of the letters. Studying both of the methods described before (priority subcell and priority letter alignments), one can see, that the ordering of the letters has quite large influence on the score of the alignment. We shall now present a counterexample showing that the ordering (1.3) does not always give the best possible score. Let us have $p = q = (0.45, 0.45, 0.1)$. All possible scores of priority letter alignment are:

Ordering	γ
1 2 3	0.61111
1 3 2	0.61128
3 2 1	0.41548

So ordering 1,3,2 gives us a counterexample that the ordering (1.3) is not always the best and in this case aligning 3-s before 2-s gives better score, although $p_2 \wedge q_2 > p_3 \wedge q_3$. Furthermore, as the score of the priority letter alignment is a continuous function of probabilities, one can found similar counterexample where $p_1 \neq p_2$. However various calculations indicate that letter with the highest $p_i \wedge q_i$ should always be aligned first.

Acknowledgement. Research is supported by Estonian Science foundation Grant 5822 and by institutional research funding IUT34-5 of the Estonian Ministry of Education and Research.

References

[1] S. Barder, J. Lember, H. Matzinger, and M. Toots, *On suboptimal LCS-alignments for independent Bernoulli sequences with asymmetric distributions* Methodol. Comput. Appl. Probab. **14**(2) (2012), 357–382.

[2] K.-M. Chao and L. Zhang, *Sequence Comparison: Theory and Methods*, Springer, 2008.

[3] N. Christianini and M. W. Hahn, *Introduction to Computational Genomics: A Case Studies Approach*, Cambridge University Press, Cambridge, 2007.

[4] V. Chvátal and D. Sankoff, *Longest common subsequences of two random sequences*, J. Appl. Probab. **12**(2) (1975), 306–315.

[5] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison, *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*, Cambridge University Press, Cambridge, 1998.

[6] R. Durrett, *Probability: Theory and Examples*, Cambridge University Press, Cambridge, 2010.

[7] M. Kiwi and J. Soto, *On a speculated relation between Chvatal-Sankoff constants of several sequences*, Combin. Probab. Comput. **18** (2009), 517–532.

[8] J. Lember and, H. Matzinger, *Standard deviation of the longest common subsequence*, Ann. Probab. **37**(3) (2009), 1192–1235.

[9] J. Lember, H. Matzinger, and A. Vollmer, *Optimal alignments of longest common subsequences and their path properties*, Bernoulli, **20**(3) (2014), 1292–1343.

[10] C. Y. Lin and F. J. Och, *Automatic evaluation of machine translation quality using longest common subsequence and Skip-Bigram statistics*, in: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics (ACL'04), 2004, p. 605.

[11] I. D. Melamed, *Automatic evaluation and uniform filter cascades for inducing N-best translation lexicons*, in: Proceedings of the Third Workshop on Very Large Corpora, 1995.

[12] I. D. Melamed, *Bitext maps and alignment via pattern recognition*, Computat. Linguist. **25**(1) (1999), 107–130.

[13] R. Mott and R. Tribe, *Approximate statistics of gapped alignments*, J. Computat. Biol. **6**(1) (1999), 91–112.

[14] P. A. Pevzner, *Computational Molecular Biology: An Algorithmic Approach*, MIT Press, Cambridge, 2000.

[15] M. Toots, *Suboptimal Alignments and Similarity of Random Sequences*, Master thesis, 2011. (unpublished)

[16] M. Vingron, *Near-optimal sequence alignment*, Current Opinion Struct. Biol. **6**(1) (1996), 346–352.

[17] M. S Waterman, *Estimating statistical significance of sequence alignments*, Philos. Trans. R. Soc. Biol. Sci. **344** (1994), 383–390.

[18] M. S. Waterman, *Introduction to Computational Biology: Maps, Sequences and Genomes*, Chapman & Hall / CRC Press, 1995.

Institute of Mathematics and Statistics, University of Tartu, J. Liivi 2, 50409 Tartu, Estonia

E-mail address: riho.klement@ut.ee

E-mail address: juri.lember@ut.ee