



HAL
open science

The complexity of the Approximate Multiple Pattern Matching Problem for random strings

Frédérique Bassino, Tsinjo Rakotoarimalala, Andrea Sportiello

► **To cite this version:**

Frédérique Bassino, Tsinjo Rakotoarimalala, Andrea Sportiello. The complexity of the Approximate Multiple Pattern Matching Problem for random strings. AofA 2020, Jun 2020, Klagenfurt, Austria. 10.4230/LIPIcs.AOFA.2020.24 . hal-03214615

HAL Id: hal-03214615

<https://hal.science/hal-03214615v1>

Submitted on 2 May 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

The complexity of the Approximate Multiple Pattern Matching Problem for random strings

Frédérique Bassino

Université Paris 13, Sorbonne Paris Cité, LIPN, CNRS UMR 7030, 99 av. J.-B. Clément, F-93430 Villetaneuse, France <https://lipn.univ-paris13.fr/~bassino/>
bassino@lipn.univ-paris13.fr

Tsinjo Rakotoarimalala

Université Paris 13, Sorbonne Paris Cité, LIPN, CNRS UMR 7030, 99 av. J.-B. Clément, F-93430 Villetaneuse, France <https://lipn.univ-paris13.fr/~rakotoarimalala/>
rakotoarimalala@lipn.univ-paris13.fr

Andrea Sportiello 

Université Paris 13, Sorbonne Paris Cité, LIPN, CNRS UMR 7030, 99 av. J.-B. Clément, F-93430 Villetaneuse, France <https://lipn.univ-paris13.fr/~sportiello/>
sportiello@lipn.univ-paris13.fr

Abstract

We describe a multiple string pattern matching algorithm which is well-suited for approximate search and dictionaries composed of words of different lengths. We prove that this algorithm has optimal complexity rate up to a multiplicative constant, for arbitrary dictionaries. This extends to arbitrary dictionaries the classical results of Yao [SIAM J. Comput. 8, 1979], and Chang and Marr [Proc. CPM94, 1994].

2012 ACM Subject Classification Formal languages and automata theory; Design and analysis of algorithms

Keywords and phrases Average-case analysis of algorithms, String Pattern Matching, Computational Complexity bounds.

Digital Object Identifier 10.4230/LIPIcs.AOFA.2020.24

Funding *Andrea Sportiello*: Supported by the French ANR-MOST MetAConC.

1 The problem

1.1 Definition of the problem

Let Σ be an alphabet of s symbols, $\xi = \xi_0 \dots \xi_{n-1} \in \Sigma^n$ a word of n characters (the input *text string*), $D = \{w_1, \dots, w_\ell\}$, $w_i \in \Sigma^*$ a collection of words (the *dictionary*). We say that $w = x_1 \dots x_m$ occurs in ξ with final position j if $w = \xi_{j-m+1} \xi_{j-m+2} \dots \xi_j$. We say that w occurs in ξ with final position j , with no more than k errors, if the letters x_1, \dots, x_m can be aligned to the letters $\xi_{j-m'}, \dots, \xi_j$ with no more than k errors of insertion, deletion or substitution type, i.e., it has *Levenshtein distance* at most k to the string $\xi_{j-m'} \dots \xi_j$ (see an example in Figure 1). Let $r_m(D)$ be the number of distinct words of length m in D . We call $\mathbf{r}(D) = \{r_m(D)\}_{m \geq 1}$ the *content* of D , a notion of crucial importance in this paper.

The *approximate multiple string pattern matching problem* (AMPMP), for the datum (D, ξ, k) , is the problem of identifying all the pairs (a, j) such that $w_a \in D$ occurs in ξ with final position j , and no more than k errors (cf. Figure 1). This is a two-fold generalisation of the classical *string pattern matching problem* (PMP), for which the exact search is considered, and the dictionary consists of a single word.

A precise historical account of this problem, and a number of theoretical facts, are presented in Navarro's review [8]. The first seminal works have concerned the PMP. Results



© F. Bassino, T. Rakotoarimalala and A. Sportiello;
licensed under Creative Commons License CC-BY

31st International Conference on Probabilistic, Combinatorial and Asymptotic Methods for the Analysis of Algorithms (AOFA 2020).

Editors: Michael Drmota and Clemens Heuberger; Article No. 24; pp. 24:1–24:15



Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

	5	10	15	20	25	30	35	40	45	50	55
	now . is . the . winter . of . our . discontent . made . glorious . summer										
sourXour.....XouX.....										
intentinteXX.....Xntent.....										
galoreglorX.....										
thereintheSSin.....										

soul: 23, 48; intent: 16, 35; galore: 45; therein: 14.

■ **Figure 1** Typical output of an approximate multiple string pattern matching problem, on an English text (alphabet of 26 characters plus the space symbol `.`). In this case $k = 2$ and $r(D) = (0, 0, 0, 0, 1, 0, 2, 1, 0, \dots)$. The symbols **D**, **S** and ⁱ stand for deletion, substitution and insertion errors, while **X** corresponds to an insertion or a substitution.

44 included the design of efficient algorithms (notably Knuth–Morris–Pratt and Boyer–Moore),
 45 and have led to the far-reaching definition of the Aho–Corasick automata [1, 3, 7, 11]. In
 46 particular, Yao [11] is the first paper that provide rigorous bounds for the complexity of
 47 PMP in random texts. To make a long story short, it is argued that an interesting notion of
 48 complexity is the asymptotic average fraction of text that needs to be accessed (in particular,
 49 at least at this stage, it is *not* the time complexity of the algorithm), and is of order $\ln(m)/m$
 50 for a word of length m . The first works on approximate search, yet again for a single word
 51 (APMP), are the description of the appropriate data structure, in [10, 4], and, more relevant
 52 to our aims here, the derivation of rigorous complexity bounds in Chang and Marr [5]. Yet
 53 again in simplified terms, if we allow for k errors, the complexity result of Yao is deformed
 54 into order $[\ln(m) + k]/m$. More recent works have concerned the case of dictionaries composed
 55 of several words, all of the same length [9],¹ however, also at the light of unfortunate flaws in
 56 previous literature, the rigorous derivation of the average complexity for the MPMP has been
 57 missing even in the case of words of the same length, up to our recent paper [2], where it is
 58 established that the Yao scaling $\ln(m)/m$ is (roughly) modified into $\max_m \ln(mr_m)/m$ (a
 59 more precise expression is given later on). By combining the formula of Chang and Marr for
 60 APMP, and our formula for MPMP, it is thus natural to expect that the AMPMP may have
 61 a complexity of the order $\max_m [\ln(mr_m) + k]/m$. This paper has the aim of establishing a
 62 result in this fashion.

63 Of course, the present work uses results, ideas and techniques already presented in [2],
 64 for the PMPM. A main difference is that in [2] we show that, for any dictionary, a slight
 65 modification of an algorithm by Fredriksson and Grabowski [6] is optimal within a constant,
 66 while this is not true anymore for approximate search with Levenshtein distance (we expect
 67 that it remains optimal for approximate search in which only substitution errors are allowed,
 68 although we do not investigate this idea here). As a result, we have to modify this algorithm
 69 more substantially, by combining it with the algorithmic strategy presented in Chang and
 70 Marr [5], and including one more parameter (to be tuned for optimality). This generalised
 71 algorithm is presented in Section 2.2.

72 Also, a large part of our work in [2] is devoted to the determination of a relatively tight
 73 lower bound, while the determination of the upper bound consists of a simple complexity
 74 analysis of the Fredriksson–Grabowski algorithm. Here, instead, we will make considerable
 75 efforts in order to determine an upper bound for the complexity of our algorithm, which is

¹ This is the reason why, before our paper [2], which deals with dictionaries having words of different length, the forementioned notion of “content” of a dictionary did not appear in the literature.

	exact	approximate
single word	$C_{\text{Yao}} \frac{\ln m}{m}$ (Yao)	$C_{\text{CM}} \frac{\ln m + k}{m}$ (Chang and Marr)
dictionary	$C_1^{\text{ex}} \frac{1}{m_{\min}} + C_2^{\text{ex}} \max_m \frac{\ln(sm r_m)}{m}$	$\frac{C_1 k + C_1'}{m_{\min}} + C_2 \max_m \frac{\ln(sm r_m)}{m}$

■ **Table 1** Summary of average complexities for exact and approximate search, for a single word or on arbitrary dictionaries. The results are derived from Yao [11], Chang and Marr [5], our previous paper [2], and the present paper, respectively.

76 the content of Section 2.4, while we will content ourselves of a rather crude lower bound,
 77 derived with small effort in Section 1.3 by combining the results of [5] and [2].

78 1.2 Complexity of pattern matching problems

79 In our previous paper [2] we have established a lower bound for the (exact search) multiple
 80 pattern matching problem, in terms of the size s of the alphabet, and the content $\mathbf{r} = \{r_m\}$
 81 of the dictionary, involving the length m_{\min} of the shortest word in the dictionary, and a
 82 function $\phi(\mathbf{r})$ with the specially simple structure $\phi(\mathbf{r}) = \max_m f(m, r_m)$. More precisely,
 83 calling $\Phi_{\text{aver}}(\mathbf{r})$ (resp. $\Phi_{\text{max}}(\mathbf{r})$) the average over random texts, of the average (res. maximum)
 84 over dictionaries D of content \mathbf{r} , of the asymptotic fraction of text characters that need to
 85 be accessed, we have

86 ► **Theorem 1** (Bassino, Rakotoarimalala and Sportiello, [2]). *Let $s \geq 2$ and $m_{\min} \geq 2$, and*
 87 *define $\kappa_s = 5\sqrt{s}$. For all contents \mathbf{r} , the complexity of the MPMP on an alphabet of size s*
 88 *satisfies the bounds*

$$89 \quad \frac{1}{\kappa_s} \left(\phi(\mathbf{r}) + \frac{1}{2s m_{\min}} \right) \leq \Phi_{\text{aver}}(\mathbf{r}) \leq \Phi_{\text{max}}(\mathbf{r}) \leq 2 \left(\phi(\mathbf{r}) + \frac{1}{2s m_{\min}} \right), \quad (1)$$

90 where

$$91 \quad \phi(\mathbf{r}) := \max_m \frac{1}{m} \ln(sm r_m). \quad (2)$$

92 Note a relative factor $\ln s$ between the statement of the result above, and its original
 93 formulation in [2], due to a slightly different definition of complexity.

94 As we have anticipated, such a result is in agreement with the result of Yao [11], for
 95 dictionaries composed of a single word, which is simply of the form $\ln(m)/m$. Combining
 96 this formula with the complexity result for APMP, derived in Chang and Marr [5], it
 97 is natural to expect that the AMPMP has a complexity whose functional dependence
 98 on k and \mathbf{r} is as in Table 1. Indeed, the bottom-right corner of the table is consistent
 99 both with the entry above it, and the entry at its left. Furthermore, it is easily seen
 100 that, up to redefining the constants, several other natural guesses would have this same
 101 functional form in disguise. Let us give some examples of this mechanism. Write $X \geq$
 102 $a_{L/U}Y + b_{L/U}Z$ as a shortcut for $a_L Y + b_L Z \leq X \leq a_U Y + b_U Z$. Now, suppose that we
 103 establish that $\Phi(\mathbf{r}, k) \geq a_{L/U} (k + 1)/m_{\min} + b_{L/U} \max_m (\ln(mr_m) + k)/m$. Then we also
 104 have $\Phi(\mathbf{r}, k) \geq a'_{L/U} (k + 1)/m_{\min} + b_{L/U} \max_m \ln(mr_m)/m$, with $a'_U = a_U + b_U$ (and all
 105 other constants unchanged). On the other side, if we have $\Phi(\mathbf{r}, k) \geq a_{L/U} (k + 1)/m_{\min} +$
 106 $b_{L/U} \max_m \ln(mr_m)/m$, with $a_L > b_L$, then we also have $\Phi(\mathbf{r}, k) \geq a_{L/U} (k + 1)/m_{\min} +$
 107 $b'_{L/U} \max_m (\ln(mr_m) + k)/m$, with $b'_L = a_L - b_L$.

108 The precise result that we obtain in this paper is the following:

24:4 The complexity of Approximate Multiple Pattern Matching

109 ► **Theorem 2.** For the AMPMP, with k errors and a dictionary D of content $\{r_m\}$, the
 110 complexity rate $\Phi(D)$ is bounded in terms of the quantity

$$111 \quad \tilde{\Phi}(D) := \frac{C_1(k+1)}{m_{\min}} + C_2 \max_m \frac{\ln(sm r_m)}{m} \quad (3)$$

112 as

$$113 \quad \frac{1}{C_1 + \kappa_s C_2} \tilde{\Phi}(D) \leq \Phi(D) \leq \tilde{\Phi}(D), \quad (4)$$

114 with $a = \ln(2s^2/(2s+1))$, $a' = \ln(4s^2 - 1)$, $\kappa_s = 5\sqrt{s}$ (as in Theorem 1) and

$$115 \quad C_1 = \frac{a + 2a'}{a}; \quad C_2 = \frac{2(a + 2a')}{aa'} = \frac{2}{a'} C_1. \quad (5)$$

117 1.3 The lower bound

118 Now, let us derive a lower bound of the functional form as in Table 1 for the AMPMP, by
 119 combining our results in [2] for the MPMP and the results in [5] for the APMP. Let us first
 120 observe a simple fact. Suppose that we have two bounds $A^{\text{LB}}(\mathbf{r}, k) \leq \Phi(\mathbf{r}, k) \leq A^{\text{UB}}(\mathbf{r}, k)$
 121 and $B^{\text{LB}}(\mathbf{r}, k) \leq \Phi(\mathbf{r}, k) \leq B^{\text{UB}}(\mathbf{r}, k)$ (with $A^{\text{LB}}(\mathbf{r}, k)$ and $B^{\text{LB}}(\mathbf{r}, k)$ positive). Then, for all
 122 functions $p(\mathbf{r}, k)$, valued in $[0, 1]$, we have

$$123 \quad p(\mathbf{r}, k) A^{\text{LB}}(\mathbf{r}, k) + (1 - p(\mathbf{r}, k)) B^{\text{LB}}(\mathbf{r}, k) \leq \Phi(\mathbf{r}, k) \leq A^{\text{UB}}(\mathbf{r}, k) + B^{\text{UB}}(\mathbf{r}, k).$$

124 We want to exploit this fact by using as bounds $A^{\text{LB/UB}}(\mathbf{r}, k)$ our previous result for the
 125 exact search, and as lower bound $B^{\text{LB}}(\mathbf{r}, k)$ the simple quantity $(k+1)/m_{\min}$. Then, later
 126 on, in Section 2, we will work on the determination of a bound $B^{\text{UB}}(\mathbf{r}, k)$ which has the
 127 appropriate form for our strategy above to apply. Let us discuss why $\Phi(\mathbf{r}, k) \geq (k+1)/m_{\min}$.
 128 We will prove that this quantity is a bound to the minimal density of a *certificate*, over a
 129 single word of length $m = m_{\min}$, and text ξ . A certificate, as described in [11], is a subset
 130 $I \subseteq \{1, \dots, n\}$ such that, for the given text, the characters $\{\xi_i\}_{i \in I}$ imply that no occurrences
 131 of words of the dictionary may be possible, besides the ones which are fully contained in I .
 132 Some reflection shows that: (1) for the interesting case $m > k$, the smallest density $|I|/n$ of
 133 a certificate is realised on a *negative certificate*, that is, on a text ξ with no occurrences of
 134 the word w ; (2) the smallest density is realised, for example, by the text $\xi = bbb \dots b$, and
 135 the word $w = aaa \dots a$; (3) in such a certificate, we must have read at least $k+1$ characters
 136 in every interval of size m , otherwise the alignment of w to this portion of text, in which we
 137 perform all the substitutions on the disclosed characters, would still be a viable candidate.
 138 Note in particular that deletion and insertion errors do not lead to higher lower bounds
 139 (although, for large m , they lead to bounds which are only slightly smaller).

140 As a result, recalling the expression for the lower bound in Theorem 1, by choosing $p(\mathbf{r}, k)$
 141 to satisfy $\frac{p}{1-p} = \frac{\kappa_s C_2}{C_1}$ we have

$$142 \quad \Phi(\mathbf{r}, k) \geq (1-p) \frac{k+1}{m_{\min}} + \frac{p}{\kappa_s} \phi(\mathbf{r}) = \frac{p}{\kappa_s C_2} \left(C_1 \frac{k+1}{m_{\min}} + C_2 \phi(\mathbf{r}) \right) = \frac{C_1 \frac{k+1}{m_{\min}} + C_2 \phi(\mathbf{r})}{C_1 + \kappa_s C_2}.$$

143 This proves the lower bound part of Theorem 2. Note that we could confine all the dependence
 144 from $\{r_m\}$ to the function ϕ (in particular, the choice $\frac{p}{1-p} = \frac{\kappa_s C_2}{C_1}$ only depends on the size
 145 of the alphabet s).

216 Our algorithm accesses the text in three steps, namely, for every block index $b =$
 217 $0, 1, \dots, \lceil n/q \rceil - 1$:

218 ■ We read all the characters ξ_i of the text, for $bq \leq i < bq + L$, that is we read the b -th
 219 block;

220 ■ We consider the possible c -block partial alignments α (with $c = c(\alpha)$) such that $b(\alpha) = b$,
 221 and associated to the intervals of text read so far. If any of these alignments is not
 222 excluded or determined positively, we read also the characters ξ_i for $i = bq - 1, bq - 2, \dots$,
 223 one by one, in this order, up to when all partial alignments are either excluded, or reach
 224 $\varepsilon = 0$. For a given instance of the problem, call $\mathcal{E}_L(b)$ (left-excess at block b) the set of
 225 positions of further characters that we need to access by this second step (with indices
 226 shifted so that the block starts at 1), and $e_L(b) = |\mathcal{E}_L(b)|$.

227 ■ If at the previous step we still have partial alignments which are not excluded, we read
 228 also the characters at positions $i = bq + L, bq + L + 1, \dots$, in this order, up to when all
 229 partial alignments are either excluded, or completed to a full alignment. Similarly to
 230 above, introduce $\mathcal{E}_R(b)$ and $e_R(b) = |\mathcal{E}_R(b)|$ (right-excess at block b).

231 An example with $c(\alpha) = 2$ is in Figure 2. Note that, at all steps, the pattern of the accessed
 232 part of the text consists of some blocks of length L and spacing q , plus one rightmost block
 233 with length $L' \geq L$ and spacing $q' \leq q$. A typical situation within the second step is as
 234 follows (here $c = 5, L = 3, q = 8, L' = 12$ and $q' = 7$):



236 Call $\mathcal{E}(b) = \mathcal{E}_L(b) \cup \mathcal{E}_R(b)$, and $e(b) = e_L(b) + e_R(b)$. Call Ψ_h^{exact} the average over random
 237 texts of the indicator function for the event that $e(b) \geq h$. Clearly, the average complexity
 238 rate of our algorithm is bounded by the expression

$$239 \quad \Phi_{\text{alg}}(D) \leq \frac{L + \mathbb{E}(e(b))}{q} = \frac{L + \sum_{h \geq 1} \Psi_h^{\text{exact}}}{q},$$

240 where the average is taken over random texts, at fixed dictionary. Note that, because of our
 241 choice of range for q and L , $c(\alpha) \geq 1$ for all α , and $c(|w|) \geq 1$ for all w .

242 Let α be a full alignment associated to the block b . Call $\mathcal{E}[\alpha]$ the set of extra positions of
 243 the text (besides the blocks) that we need to access in order to determine the alignment α .
 244 Then clearly $\mathcal{E}(b) = \bigcup_{\alpha} \mathcal{E}[\alpha]$.

245 2.3 Proof strategy for the upper bound

246 Our proof strategy is to prove that there exists a choice of parameters L and q , with the
 247 properties that $q = \Theta(m_{\min})$, $L/q = \Theta(\phi(\mathbf{r}(D)))$, and $\mathbb{E}(e(b)) = \Theta(1)$. This last condition
 248 is equivalent to the requirement that Ψ_h^{exact} is a summable series, and we will see that
 249 indeed the first can be bounded by a geometric series, and the second is rather small. Up to
 250 calculating the pertinent multiplicative constants, such a pattern would imply the functional
 251 form of the complexity anticipated in Section 1.2.

252 The idea is that the exact calculation of $\mathbb{E}(e(b))$ or of Ψ_h^{exact} , even at q and L fixed (which
 253 is easier than optimising w.r.t. these parameters), is rather difficult, but we can produce a
 254 simpler upper bound by:

255 ■ For alignments α with $c(\alpha) > 1$, neglect the information coming from the $e(b')$ extra
 256 characters that we have accessed at blocks $b' < b$. This allows to separate the analysis on
 257 the different blocks of text.

258 ■ Naïvely, for different (full) alignments α , we could perform a *union bound*, that is,
 259 $e(b) = |\mathcal{E}(b)| = |\bigcup_{\alpha} \mathcal{E}[\alpha]| \leq \sum_{\alpha} |\mathcal{E}[\alpha]|$, which thus separates the analysis over the
 260 different alignments. We will make an improved version of this bound, namely we use this
 261 bound, not with full alignments, but rather with “classes of equivalent partial alignments”.
 262 As we anticipated, the crucial point is that we count partial alignments instead of full
 263 alignments. A further slight improvement of the bound comes from considering these ‘classes
 264 of equivalent partial alignments’, instead of just the partial alignments. These two facts are
 265 motivated by the same argument, that we now elucidate.

266 Consider the two following notions: (1) Each set $A_h(w)$ of partial alignments is partitioned
 267 into classes I . (2) There is a subset $\bar{A}_h(w) \subseteq A_h(w)$ of alignments, that we shall call *basic*
 268 *alignments*. Now, suppose that the two following properties hold: (i) $I \cap \bar{A}_h(w) \neq \emptyset$
 269 for all classes I of $A_h(w)$. (ii) For each $\alpha \in I$, there exists a $\bar{\alpha} \in I \cap \bar{A}_h(w)$, such that
 270 $\mathcal{E}(\alpha) \subseteq \mathcal{E}(\bar{\alpha})$. In this case it is easily established that the bound above can be improved into
 271 $e(b) = |\mathcal{E}(b)| = |\bigcup_{\alpha} \mathcal{E}[\alpha]| \leq \sum_{\bar{\alpha}} |\mathcal{E}[\bar{\alpha}]|$, where the sum runs only on basic partial alignments.
 272 Thus, calling $\Psi_h := \sum_{w \in D} \sum_{\alpha \in \bar{A}_h(w)} \mathbb{P}(|\mathcal{E}[\bar{\alpha}]| \geq h)$, we have $\Psi_h \geq \Psi_h^{\text{exact}}$.

273 We propose the following definition of basic alignment. Let α be in $A_h(w)$. In the string
 274 u , suppose that we write C_a instead of C , whenever the well-aligned character is a , and
 275 D_a when the deleted character is a (this is clearly just a bijective decoration of u). For
 276 $\alpha \in \bar{A}_h(w)$, we require that there are no occurrences of $C_a I_a$ as factors of u (as these are
 277 equivalent to $I_a C_a$), of $C_a D_a$ (as these are equivalent to $D_a C_a$) and of $I_a D_b$ or $D_b I_a$ (as
 278 these are equivalent to C_a or S_a , depending if $a = b$ or not). If α can be obtained from α' by
 279 a sequence of these rewriting rules, then α and α' are in the same class I .

280 It is easy to see that this definition of basic alignment and classes has the defining
 281 properties above.

282 2.4 Evaluation of an upper bound at q and L fixed

283 Let us call $p_{c,h,\varepsilon'}(w)$ the probability that, for a given word w and parameter ε' , there exists
 284 an alignment $\alpha \in A_h(w)$, to a text consisting of $c - 1$ blocks of length L and one block of
 285 length $L + h$, which is visited by the algorithm (that is, it makes at most k errors), that is,
 286 in particular,

$$287 \Psi_h \leq \sum_{\varepsilon'=0}^{q-1} p_{c,h,\varepsilon'}(w). \quad (6)$$

288 We have the important fact

► Proposition 3.

$$290 p_{c,h,\varepsilon'}(w) \leq \beta s^{-(cL+h)} B_{cL+h+c-1,k} \quad (7)$$

291 for all ε' , where $\beta = \frac{(2s-1)L+k}{(2s-1)L-k}$ and $B_{L,k} = (2s-1)^k \binom{L+k}{k}$.

292 The proof of this proposition is slightly complicated, and is presented in Appendix A. Note
 293 however that for the special case $c = 1$, and with exactly k errors (instead of at most k errors),
 294 the bound $s^{-(L+h)} (2s)^k \binom{L+k}{k}$ can be established trivially. Also note that the bound does
 295 not depend on ε' , and, in particular, it only depends on $h = |\mathcal{E}_L| + |\mathcal{E}_R|$ for the alignments α
 296 at given w and ε' , and not separately on the two summands.

297 We are now ready to evaluate the expressions for the upper bound on the quantity Ψ_h
 298 in (6), in light of (7). Call $R_c = \sum_{m:c(m)=c} r_m = \sum_{m=qc+L-1}^{q(c+1)+L-2} r_m$, and $p_{c,h}$ as q times the

299 RHS of (7) (that is, an upper bound to $\sum_{\varepsilon'=0}^{q-1} p_{c,h,\varepsilon'}(w)$). We have the bound

$$300 \quad \sum_h \Psi_h \leq \sum_c R_c \sum_h p_{c,h} = \sum_c R_c \sum_h \beta q s^{-(cL+h)} B_{cL+h+c-1,k}. \quad (8)$$

302 Recalling that

$$303 \quad \sum_{h \geq 0} s^{-h} \binom{a+k+h}{k} \leq \frac{1}{1 - \frac{1}{s} \frac{a+k+1}{a+1}} \binom{a+k}{k},$$

304 (and that $q < m_{\min}$), substituting in (8) gives

$$305 \quad \begin{aligned} \Phi_{\text{alg}}(D) &\leq \frac{1}{q} \left(L + \beta q \sum_c R_c \frac{1}{1 - \frac{1}{s} \frac{cL+k+c}{cL+c}} s^{-cL} \binom{cL+c-1+k}{k} (2s-1)^k \right) \\ &\leq \frac{1}{q} \left(L + \frac{\beta m_{\min} (2s-1)^k}{1 - \frac{1}{s} \frac{L+k}{L}} \sum_c R_c s^{-cL} \binom{c(L+1)+k}{k} \right). \end{aligned} \quad (9)$$

306 We want to prove that

$$307 \quad \Phi_{\text{alg}}(D) \leq \frac{C_1 k + C'_1}{m_{\min}} + C_2 \max_m \frac{\ln(sm r_m)}{m}, \quad (10)$$

308 with suitable constants C_1, C'_1 and C_2 (it will turn out at the end that we can set $C'_1 = C_1$
309 and C_1, C_2 to be as in Theorem 2, but at this point it is convenient to let them be three
310 separate variables). This would prove the upper bound part of Theorem 2.

311 Note that, if $k/m_{\min} \geq 1/C_1$, the upper bound expression (10) is larger than the trivial
312 bound $\Phi_{\text{alg}}(D) \leq 1$, and there is nothing to prove. So we can assume that $k/m_{\min} < 1/C_1$.

313 2.5 Optimisation of q and L

314 We now have to analyse the expression (9), in order to understand which values of q and L
315 make the bound smaller. The sum over c is the most complicated term. We simplify it by
316 using the fact that, for all $\xi \in \mathbb{R}^+$, $\ln \binom{a+k}{k} \leq k \ln(1 + \xi) + a \ln(1 + \xi^{-1})$, which gives

$$317 \quad \begin{aligned} T &:= m_{\min} (2s-1)^k \sum_c R_c s^{-cL} \binom{c(L+1)+k}{k} \\ &\leq \sum_c \frac{1}{c^2} \exp \left[-c \left(LA - \frac{1}{c} (\ln(R_c m_{\min}) + k \ln((1 + \xi)(2s-1))) - \frac{\ln c^2}{c} - \ln(1 + \xi^{-1}) \right) \right] \\ &= \sum_c \frac{1}{c^2} \exp [-c(LA - \phi'(c) - \ln(1 + \xi^{-1}))], \end{aligned} \quad (11)$$

321 where $A = \ln(s\xi/(1 + \xi))$, $A' = \ln((1 + \xi)(2s-1))$ and

$$322 \quad \phi'(c) = \frac{\ln(c^2 R_c m_{\min}) + k A'}{c}. \quad (12)$$

323 Ultimately, we want to choose L such that T is bounded by a constant, as its summands
324 over c are bounded by a convergent series. With this goal, let c^* be the value maximising
325 the expression $\phi'(c)$, and ϕ^* the value of the maximum. The sum above is then bounded by

$$326 \quad \sum_c \frac{1}{c^2} \exp [-c(LA - \phi^* - \ln(1 + \xi^{-1}))].$$

24:10 The complexity of Approximate Multiple Pattern Matching

327 For any value of ξ such that $A > 0$ (that is, for $\xi > (s-1)^{-1}$), there exists a positive smallest
 328 value of L such that the exponent in the expression above is negative. So we set

$$329 \quad L^* = \left\lceil \frac{\phi^* + \ln(1 + \xi^{-1})}{A} \right\rceil,$$

330 (as the choice of ξ is free, we can tune it at the end so that the ratio is an integer), and
 331 recognise that the RHS of equation (11), specialised to $L = L^*$, is bounded by $\sum_c \frac{1}{c^2} = \pi^2/6$.
 332 Note that

$$333 \quad \phi^* \geq \phi'(1) \geq kA'$$

334 so that

$$335 \quad \frac{L^*}{k} \geq \frac{A'}{A} = \frac{\ln((1 + \xi)(2s - 1))}{\ln(s\xi/(1 + \xi))},$$

336 which implies that we can set $\beta = \frac{2s-1+A/A'}{2s-1-A/A'}$, and

$$337 \quad \frac{1}{1 - \frac{1}{s} \frac{L+k}{L}} \leq \frac{1}{1 - \frac{1}{s}(1 + A/A')} = \frac{1}{1 - \frac{1}{s} \frac{\ln(s\xi(2s-1))}{\ln((1+\xi)(2s-1))}}.$$

338 Now, let us choose $q = \lfloor \frac{m_{\min} - k}{2} \rfloor$, which coincides with the choice of the analogous parameter
 339 in Chang and Marr [5]. This is the largest possible value such that $c(w) \geq 1$ for all $w \in D$.
 340 With this choice,

$$341 \quad \frac{1}{q} \leq \frac{2}{m_{\min}} \frac{C_1}{C_1 - 1}.$$

342 Collecting the various factors calculated above, we get that the expression (9) is bounded by

$$343 \quad \Phi_{\text{alg}}(D) \leq \frac{2}{m_{\min}} \frac{C_1}{C_1 - 1} \left(L^* + \frac{\beta \frac{\pi^2}{6}}{1 - \frac{1}{s}(1 + A/A')} \right).$$

344 We are left with two tasks: choosing suitable values for ξ and C_1 (both of order 1), and
 345 recognising that the expression for L^* (and for ϕ^*) can be related to the quantity $\phi(\mathbf{r})$ in
 346 (2). Let us start from the latter. Note that, as for any $m \geq m_{\min}$

$$347 \quad \frac{m - k}{q} - 2 \leq c(m) \leq \frac{m}{q}$$

348 we can write² $m \leq m_{\min} c(m) \leq s^2 m$, which gives

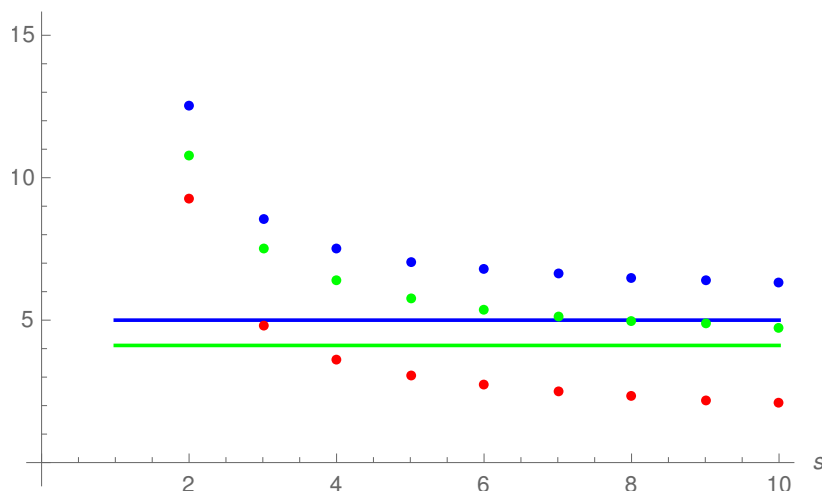
$$349 \quad \max_c \frac{1}{c} \ln(c^2 m_{\min} R_c) \leq \max_m \frac{m_{\min}}{m} \ln(s^2 m^2 r_m) \leq 2m_{\min} \phi(\mathbf{r}).$$

350 As, of course $\max_c (X(c) + Y(c)) \leq \max_c X(c) + \max_c Y(c)$, we have in particular that

$$351 \quad \phi^* \leq 2m_{\min} \phi(\mathbf{r}) + kA' \qquad L^* \leq \frac{2m_{\min} \phi(\mathbf{r}) + kA' + \ln(1 + \xi^{-1})}{A},$$

² Because $s \geq 2$, and we anticipate that, under our choice, $C_1 \geq 5$, thus

$$m \leq 2(m - k - q) \leq m_{\min} \left(\frac{m - k}{q} - 2 \right) \leq m_{\min} c(m) \leq m_{\min} \frac{m}{q} \leq 2 \left(\frac{C_1}{C_1 - 1} \right) m \leq s^2 m.$$



■ **Figure 3** Plot of the constant $C_1(s)$, $C_1'(s)$ and $C_2(s)$, as given by the expressions in (13) (respectively, in blue, green and red). The asymptotic values are 5, $5\pi^2/12$ and 0 respectively.

353 which thus implies

$$\begin{aligned}
 \Phi_{\text{alg}}(D) &\leq \frac{2}{m_{\min}} \frac{C_1}{C_1 - 1} \left(\frac{2m_{\min}}{A} \phi(\mathbf{r}) + k \frac{A'}{A} + \frac{\beta \frac{\pi^2}{6}}{1 - \frac{1}{s}(1 + A/A')} + \frac{\ln(1 + \xi^{-1})}{A} \right) \\
 &= \frac{2C_1}{C_1 - 1} \left[\frac{A'}{A} \frac{k}{m_{\min}} + \left(\frac{\beta \frac{\pi^2}{6}}{1 - \frac{1}{s}(1 + \frac{A}{A'})} + \frac{\ln(1 + \xi^{-1})}{A} \right) \frac{1}{m_{\min}} + \frac{2}{A} \phi(\mathbf{r}) \right].
 \end{aligned}$$

355 Let us choose $C_1 = 2A'/A + 1$. The expression above simplifies into

$$\Phi_{\text{alg}}(D) \leq \frac{C_1 k}{m_{\min}} + \frac{2A' + A}{AA'} \left[\left(\frac{A\beta \frac{\pi^2}{6}}{1 - \frac{1}{s}(1 + \frac{A}{A'})} + \ln(1 + \xi^{-1}) \right) \frac{1}{m_{\min}} + 2\phi(\mathbf{r}) \right],$$

357 in particular, this justifies the notation C_1 , which in the introduction was chosen to denote
 358 the coefficient in front of the $\frac{k}{m_{\min}}$ summand. Now we shall choose the optimal value of ξ .
 359 The dependence on ξ is mild, provided that we are in the appropriate range $\xi > 1/(s - 1)$.
 360 The choice of ξ , in turns, determines the ratio between the lower and upper bound, which
 361 has the functional form $C_1' + \kappa_s C_2$ (with notations as in the theorem). A choice which is
 362 a good trade-off among the three summands in this expression, and for which the analytic
 363 expression is relatively simple, is to take $\xi = 2s$. Under this choice we have

$$\begin{aligned}
 C_1 &= 1 + 2 \frac{\ln(4s^2 - 1)}{\ln(2s^2/(2s + 1))}, & C_2 &= \frac{4}{\ln(2s^2/(2s + 1))} + \frac{2}{\ln(4s^2 - 1)}, \\
 C_1' &= \frac{C_2}{2} \left[\ln \frac{2s + 1}{2s} + \frac{\beta \pi^2}{6} \frac{s \ln(2s^2/(2s + 1)) \ln(4s^2 - 1)}{(s - 1) \ln(4s^2 - 1) - \ln(2s^2/(2s + 1))} \right]
 \end{aligned}$$

367 or, in a more compact way, calling $a = A|_{\xi=2s} = \ln(2s^2/(2s + 1))$ and $a' = A'|_{\xi=2s} =$
 368 $\ln(4s^2 - 1)$, and substituting back the value of β ,

$$C_1 = \frac{a + 2a'}{a}, \quad C_2 = \frac{a + 2a'}{a} \frac{2}{a'}, \tag{13a}$$

$$C_1' = \frac{a + 2a'}{a} \left(\frac{\ln s - a}{a'} + \frac{\pi^2 (2s - 1)a' + a}{6 (2s - 1)a' - a} \frac{as}{(s - 1)a' - a} \right). \tag{13b}$$

372 The behaviour in s of these constants is depicted in Figure 3.

373 It can be verified that, with our choice of ξ , $C'_1 < C_1$ for all $s \geq 2$.³ we can replace C'_1 by
 374 C_1 in the functional form (10) for the bound on $\Phi_{\text{alg}}(D)$, and thus obtain the statement of
 375 Theorem 2. This concludes our proof.

376 — **References** —

- 377 **1** Alfred V. Aho and Margaret J. Corasick. Efficient string matching: an aid to bibliographic
 378 search. *Communications of the ACM*, 18(6):333–340, 1975.
- 379 **2** Frédérique Bassino, Tsinjo Rakotoarimalala, and Andrea Sportiello. The complexity of the
 380 Multiple Pattern Matching Problem for random strings. In *2018 Proceedings of the Fifteenth
 381 Workshop on Analytic Algorithmics and Combinatorics (ANALCO)*, pages 40–53. SIAM, 2018.
- 382 **3** Robert S Boyer and J Strother Moore. A fast string searching algorithm. *Communications of
 383 the ACM*, 20(10):762–772, 1977.
- 384 **4** William I. Chang and Eugene L. Lawler. Sublinear approximate string matching and biological
 385 applications. *Algorithmica*, 12(4-5):327–344, 1994.
- 386 **5** William I. Chang and Thomas G. Marr. Approximate string matching and local similarity.
 387 In *Combinatorial Pattern Matching - CPM 1994*, volume 807 of *Lecture Notes in Computer
 388 Science*, pages 259–273. Springer, 1994.
- 389 **6** Kimmo Fredriksson and Szymon Grabowski. Average-optimal string matching. *Journal of
 390 Discrete Algorithms*, 7(4):579–594, 2009.
- 391 **7** Donald E. Knuth, James H. Morris, and Vaughan R. Pratt. Fast pattern matching in strings.
 392 *SIAM Journal on Computing*, 6(2):323–350, 1977.
- 393 **8** Gonzalo Navarro. A guided tour to approximate string matching. *ACM Computing Surveys
 394 (CSUR)*, 33(1):31–88, 2001.
- 395 **9** Gonzalo Navarro and Kimmo Fredriksson. Average complexity of exact and approximate
 396 multiple string matching. *Theoretical Computer Science*, 321(2-3):283–290, 2004.
- 397 **10** Peter H. Sellers. The theory and computation of evolutionary distances: pattern recognition.
 398 *Journal of Algorithms*, 1(4):359–373, 1980.
- 399 **11** Andrew Chi-Chih Yao. The complexity of pattern matching for a random string. *SIAM
 400 Journal on Computing*, 8(3):368–387, 1979.

³ One way to see this is by proving that both $C_1(s)$ and $C'_1(s)$ decrease monotonically as functions on the real interval $[2, +\infty[$, that $\lim_{s \rightarrow \infty} C_1(s) = 5$, that $C_1(s) > C'_1(s)$ for $s \in \{2, 3, \dots, 7\}$, and that $C'_1(8) < 5$.

A Proof of Proposition 3

In this section we evaluate an upper bound to $p_{c,h,\varepsilon'}$, which is the probability that, for a given word w with $c(|w|) = c$, the disclosed text composed of $c - 1$ intervals of size L and one interval of size $L + h$ corresponds to at least one basic alignment α by making no more than k errors. The statement of the result, equation (14) below, is given in Proposition 3.

Let us introduce the recurring quantity

$$B_{L,k} := (2s - 1)^k \binom{L + k}{k}.$$

First, let us analyse the case in which we have a single block, and exactly k errors. For w a word of length m , it is clear that the result depends only on the $m - \varepsilon'$ left-most characters of the word, not on the ε' right-most ones, so we can assume without loss of generality that $\varepsilon' = 0$. Call $H_{L,k}(m)$ the number of different words of length L obtained by transforming the suffixes of w and making exactly k errors. We have

▷ **Proposition A.1.** For all $L \geq k \geq 1$, $H_{L,k} \leq B_{L,k}$.

Proof. Note that the analogous statement with $2s - 1$ replaced by $2s$ in $B_{L,k}$ is trivial, as we have exactly $2s$ types of errors (one deletion, s insertions and $s - 1$ substitutions), and the counting of their possible positions in the string u is a function of the length of the string, bounded from above by the worst case, associated to all insertion errors.

We can gain the factor $2s - 1$ instead of $2s$ by restricting to basic alignments, but this requires a finer analysis involving generating functions. Let us call $f(u, y, z)$ the generating function such that $[u^a y^L z^k] f(u, y, z)$ is the number of basic alignments of length L obtained by transforming a word of length a and making exactly k errors. Calculating $f(u, y, z)$ exactly is a difficult task, and the result would depend on w as a word, not only on $m = |w|$, but we will calculate a simpler upper bound $f_{\text{UB}}(u, y, z)$, which in particular only depends on m . In this context, a generating-function upper bound is an upper bound for partial sums, that is $g \succeq f$ if $\sum_{h=0}^k [u^a y^L z^h] (g(u, y, z) - f(u, y, z)) \geq 0$ for all L and a . Let us construct f_{UB} by starting from $f_0(u, y, z) := \frac{uy}{1-uy}$, which is the generating function f specialised to $z = 0$, and let us introduce the various types of errors one at the time.

The first operation corresponds to allow for *insertion* errors. The restriction to basic alignments, however, brings to a subtlety. For example, starting with a word $w = abcd$, in order to get the alignment $aaabcd$ we can proceed in several ways: $aaabcd$ or $aaabcd$ or by $aaabcd$ (bold letters correspond to insertions). Under the notion of basic alignment we avoid to overcount these manifestly equivalent alignments, as of these expressions we would only keep the latter, $aaabcd$, that is, at the left of a letter a we can only insert letters different from a . On the other hand, at the right end of the word one can insert strings consisting of any character of the alphabet.

Calling f_i the generating function in which insertion errors are allowed, we thus get

$$f_i(u, y, z) = \frac{1}{1 - syz} f(u, y, z) \Big|_{uy \rightarrow uy \left(\frac{1}{1 - (s-1)z} \right)} = \frac{uy}{(1 - syz)(1 - uy - (s - 1)yz)}.$$

We now introduce deletion errors, which, consistently, we allow only on the characters of the initial string (not on the ones which have just been inserted). Thus, any given original character can be either left as is, or deleted. This gives the generating function $f_{i,d}$, with

$$f_{i,d}(u, y, z) = f_i(u, y, z) \Big|_{uy \rightarrow uy + uz} = \frac{uz + uy}{(1 - syz)(1 - uy - uz - (s - 1)yz)}.$$

24:14 The complexity of Approximate Multiple Pattern Matching

442 Finally, for substitution errors, again we can either substitute any initial character with one
 443 of the $s - 1$ other characters of the alphabet, or leave it unchanged, which brings to $f_{i,d,s}$,
 444 with

$$445 \quad f_{i,d,s} = f_{i,d}(u, y, z)|_{uy \rightarrow uy + (s-1)uyz} = \frac{u(syz - yz + y + z)}{(1 - syz)(1 - uz - (s-1)(u+1)yz)}.$$

446 Note that, by this procedure, we have already produced an upper bound, as $f_{i,d,s} \succeq f$ (in
 447 the sense defined above). Note also that it is *not* $f_{i,d,s} = f$, because, for example, we have
 448 overcounted the equivalent cases in which in a word $w = \dots aa \dots$ we have deleted the first
 449 or the second character.

450 If the word w is shorter than $L + k$, we may miss some alignments because they do not
 451 fit in the text interval. As we are evaluating an upper bound, we can restrict to the case in
 452 which w is long enough for this not to happen, and thus sum over all suffixes by just setting
 453 $u = 1$, and conclude that $H_{L,k} \leq [y^L z^k] f'(1, y, z)$. Thus, in order to conclude, we must show
 454 that $[y^L z^k] f'(1, y, z) \leq B_{L,k}$. Let us call

$$455 \quad F_{L,k} = [y^L z^k] \frac{1}{(syz - 1)(2syz - 2yz + y + z - 1)}.$$

456 We can rewrite the inequality above as $H_{L,k} \leq F_{L-1,k} + F_{L,k-1} + (s-1)F_{L-1,k-1}$, and thus,
 457 if we can prove that $F_{L,k} \leq B_{L,k}$, for all pairs of integers $L \geq k$, we could conclude in light
 458 of the fact that

$$459 \quad H_{L,k} \leq B_{L-1,k} + B_{L,k-1} + (s-1)B_{L-1,k-1} = (2s-1)^k \binom{L+k}{k} - R_{L,k},$$

460 where $R_{L,k} = (2s-1)^{k-1} \left(2(s-1) \frac{k-1}{L} \binom{L+k-2}{k-1} \right)$ is indeed easily checked to be non-negative
 461 for all $L \geq k \geq 1$.

462 So, to finish the proof, let us show that $F_{L,k} \leq B_{L,k}$. First,

$$463 \quad \begin{aligned} F_{L,k} &= [y^L z^k] \left(\frac{1}{1 - syz} + \frac{2syz - 2yz + y + z}{1 - 2syz + 2yz - y - z} \right) \\ &= \delta_{L,k} s^k + F_{L-1,k} + F_{L,k-1} + 2(s-1)F_{L-1,k-1}. \end{aligned}$$

464 Since $L \geq k \geq 1$, we have $R_{L,k} \geq \delta_{L,k} s^k$ for $s \geq 2$, and $B_{L,k} \geq F_{L,k} \geq H_{L,k}$.

465 To conclude, we just check the boundary conditions in the recursion above for $F_{L,k}$,
 466 and $B_{L,k}$, which again are in agreement with the inequality. Indeed we have, for $(L, k) \in$
 467 $\{(0, 0), (0, 1), (1, 0)\}$, $F_{0,0} = B_{0,0} = 1$, $B_{0,1} = 2s - 1 \geq 1 = F_{0,1}$ and $B_{1,0} = 4s - 2 \geq 3s =$
 468 $F_{1,0}$. ◀

469 Now we want to deal with the more general case, in which we have more than one block, and
 470 we sum over the number of errors up to k . We will prove a more general statement, in which
 471 we have c blocks of lengths L_1, \dots, L_c , separated by gaps of lengths q_1, \dots, q_{c-1} , which in
 472 particular is so general to allow us to treat in one stroke the case in which we add characters
 473 at the left or at the right of the b -th algorithm block.

474 Similarly to the argument above, in order to produce an upper bound we can set without
 475 loss of generality that $\varepsilon' = 0$, all the q_i 's are larger than k and that m is larger than
 476 $\sum L_i + \sum q_i + k$, as any variant of this would give no more alignments. So, we will call
 477 $p_{L_1, \dots, L_c; k}$ the corresponding quantity, in which the dependence from the q_i 's and m has been
 478 dropped.

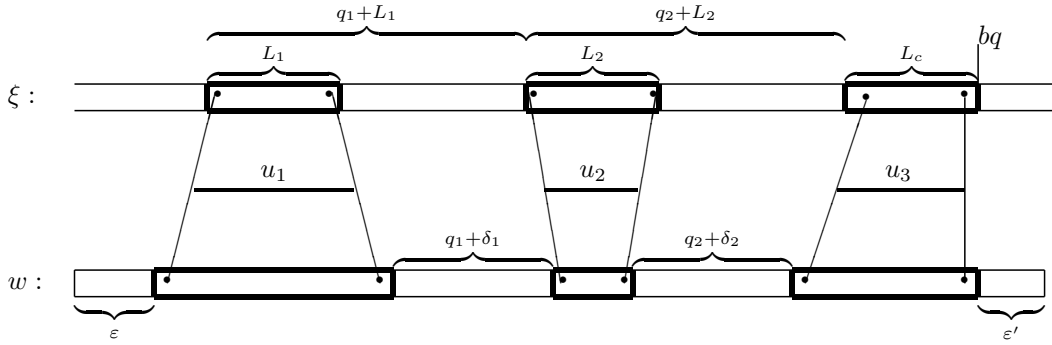


Figure 4 Example of multi-interval alignment analysed for the estimate of $p_{L_1, \dots, L_c; k}$.

479 For multi-block partial alignments, we have parameters $\delta_1, \dots, \delta_{c-1}$ for the offset among
 480 the different consecutive blocks of the partial alignment, and, if we have an offset δ_i in the
 481 alignment of two blocks, we have to perform at least $|\delta_i|$ deletions or insertions errors when
 482 completing the partial alignment to a full one (cf. figure 4).

483 Calling $\bar{L} = \sum_{i=1}^c L_i$, this leads to the following sum

$$484 \quad p_{L_1, \dots, L_c; k} \leq s^{-\bar{L}} \sum_{t=0}^k \sum_{\Delta=0}^t \sum_{\substack{k_1, k_2, \dots, k_c \in \mathbb{N} \\ k_1 + k_2 + \dots + k_c = t - \Delta}} \sum_{\substack{\delta_1, \delta_2, \dots, \delta_{c-1} \in \mathbb{Z} \\ \delta_1 + \delta_2 + \dots + \delta_{c-1} = \Delta}} B_{L_1, k_1} B_{L_2, k_2} \dots B_{L_c, k_c}.$$

485 From the Vandermonde convolution formula, $\sum_{i=0}^k \binom{l_1+i}{i} \binom{l_2+k-i}{k-i} = \binom{l_1+l_2+k+1}{k}$, which implies
 486 $\sum_h B_{L_1, h} B_{L_2, k-h} = B_{L_1+L_2+1, k}$, we can simplify the expression above into

$$487 \quad p_{L_1, \dots, L_c; k} \leq s^{-\bar{L}} \sum_{t=0}^k \sum_{\Delta=0}^t \sum_{\substack{\delta_1, \delta_2, \dots, \delta_{c-1} \in \mathbb{Z} \\ \delta_1 + \delta_2 + \dots + \delta_{c-1} = \Delta}} B_{\bar{L}+c-1, t-\Delta}.$$

488 The sum over the δ_i 's gives

$$489 \quad \sum_{\substack{\delta_1, \delta_2, \dots, \delta_{c-1} \in \mathbb{Z} \\ \delta_1 + \delta_2 + \dots + \delta_{c-1} = \Delta}} 1 = [z^\Delta] \left(\frac{1+z}{1-z} \right)^{c-1}$$

490 that is, by recognising that $B_{L, k-h} \leq B_{L, k} \left(\frac{k}{(2s-1)L} \right)^h$, we get

$$491 \quad p_{L_1, \dots, L_c; k} \leq s^{-\bar{L}} B_{\bar{L}+c-1, k} \left(\frac{(1+z)^{c-1}}{(1-z)^c} \right) \Bigg|_{z = \frac{k}{(2s-1)(\bar{L}+c-1)}}.$$

492 This is all we shall say at this level of generality. Now note that, in our patterns, $\bar{L}+c-1 \geq cL$
 493 (and $k \leq L$), so that, in this range of parameters,

$$494 \quad p_{L_1, \dots, L_c; k} \leq s^{-\bar{L}} B_{\bar{L}+c-1, k} \left(\frac{(1+z)^{c-1}}{(1-z)^c} \right) \Bigg|_{z = \frac{1}{(2s-1)c} \frac{k}{L}} \leq \frac{(2s-1) + \frac{k}{L}}{(2s-1) - \frac{k}{L}} s^{-\bar{L}} B_{\bar{L}+c-1, k}. \quad (14)$$