Refined Upper Bounds on the Size of the Condensed Neighbourhood of Sequences

Cedric Chauve[1,2], Marni Mishna[1], and France Paquet-Nadeau[1]

[1] Department of Mathematics, Simon Fraser University
8888 University Drive, Burnaby, BC, V5A 1S6, Canada
[2] LaBRI, Université de Bordeaux
351 Cours de la Libération
33405 Talence Cedex, France

Abstract. The d-neighbourhood of a sequence s is the set of sequences that are at distance at most d from s, for a given measure of distance and parameter d. The condensed d-neighbourhood is obtained by removing from the neighbourhood any sequence having a prefix also in the neighbourhood. Estimating the maximum size of the condensed neighbourhood over all DNA sequences of a given length k for the Levenshtein distance is a problem related, among others, to the analysis of the BLAST (Basic Local Alignment Search Tool, Altschul et al., 1990). In this work, we analyse recurrences for computing an upper bound to the maximum size of the condensed d-neighbourhood for sequences of length k and provide a simpler asymptotic expression that we conjecture results in a dramatically improved upper bound.

Keywords: sequence neighbourhood, algorithms analysis, analytic combinatorics

1 Introduction

The search for all of the approximate occurrences of a query sequence within a text, known as approximate pattern matching, is a central problem in biological sequence analysis [5]. Algorithms based on the seed-and-extend approach, such as BLAST [1], proceed in two phases, the first one identifying *seeds* that are exact short patterns present both in the query and the text, that are later *extended* through dynamic programming. BLAST performs the first phase, that detects seeds, in two steps, *neighbourhood generation* and *filtration*. The neighbourhood generation step consists in computing the neighbourhood of all or a set of k-mers present in the pattern – actually the *condensed neighbourhood*, a subset of the neighbourhood – which is then filtered to keep only the the neighbourhood sequences that also appear in the text. It follows that the maximum size of the (condensed) neighbourhood plays an important role in the analysis of such algorithms, a topic that has been studied in several papers [5,6].

In this note we determine an asymptotics expression for the upper bound formula for the size of the (k, d)-condensed neighbourhood for the Levenshtein distance and provide experimental evidence this expression is an actual upper bound. In Section 2 we define formally the concepts of neighbourhood and condensed neighbourhood for the Levenshtein distance, and describe previous results on upper bounds for the maximum size of the neighbourhood. Then in Section 3, we describe our improved upper bound and apply it to the asymptotic analysis of the approximate pattern matching algorithm introduced in [5], that serves as a basis for BLAST.

Cedric Chauve, Marni Mishna, France Paquet-Nadeau: Refined Upper Bounds on the Size of the Condensed Neighbourhood of Sequences, pp. 30–40.
Proceedings of PSC 2021, Jan Holub and Jan Žďárek (Eds.), ISBN 978-80-01-06869-4

2 Background

2.1 Neighbourhood and condensed neighbourhood

We remind that the Levenshtein distance between two sequences is the minimum number of edit operations (insertion, deletion, substitution of a single character) needed to transform one sequence into the other. We denote by $d_{Lev}(v, w)$ the Levenshtein distance between two sequences v and w.

Formally, given a sequence w of length k on an alphabet Σ (with $|\Sigma| = s$), the d-neighbourhood of w, denoted by $N(d, w)$, is the set of all sequences on Σ at Levenshtein distance of w at most d:

$$N(d, w) := \{v \mid d_{Lev}(v, w) \leq d\}.$$

The condensed neighbourhood of w, denoted by $CN(d, w)$, is the subset of this neighbourhood comprising sequences that have none of their prefixes in the neighbourhood:

$$CN(d, w) := \{v \mid v \in N(d, w) \text{ and there is no } u \in N(d, w) \text{ that is a prefix of } v\}.$$

The time complexity analysis of approximate pattern matching algorithms requires an estimate of the maximum size of a condensed d-neighbourhood over all sequences w of length k on an alphabet of size s. We denote this maximum size $CN(s, k, d)$:

$$CN(s, k, d) := \max_{w \in \Sigma^k} |CN(d, w)|.$$

In the course of Myer's discussion of the BLAST algorithm in [6], he provides such an upper bound for $CN(s, k, d)$ and asks for a tighter bound. In this work, we do exactly this and provide a better upper bound, by combining the recurrence equations given by Myers in [6] and techniques from basic analytic combinatorics.

2.2 Recurrences and known upper bound

In [6], Myers describes recurrences suitable for both generation and counting of edit scripts of distance at most d, over k symbols taken from an alphabet of size s. Furthermore he showed how these recurrences could be used to bound $CN(s, k, d)$.

Lemma 1 (Myers, [6]). *Let $S(s, k, d)$ be defined by the following trivariate recurrence. If $k \leq d$ or $d = 0$ then*

$$S(s, k, d) := 1,$$

otherwise

$$S(s,k,d) := \begin{cases} S(s,k-1,d) + (s-1)S(s,k-1,d-1) \\ +(s-1)\sum_{j=0}^{d-1} s^j S(s,k-2,d-1-j) \\ +(s-1)^2 \sum_{j=0}^{d-2} s^j S(s,k-2,d-2-j) \\ +\sum_{j=0}^{d-1} S(s,k-2-j,d-1-j) \end{cases}$$

Let

$$T(s,k,d) := S(s,k,d) + \sum_{j=1}^{d} s^j S(s,k-1,d-j).$$

Then

$$CN(s,k,d) \leq T(s,k,d).$$

The term $S(s,k,d)$ counts edit scripts between two sequences over an alphabet of size s (a text and a query, assumed to be of length k) and with a Levenshtein score (edit distance implied by the edit script) at most d. From an edit script, one can generate a word of the neighbourhood by discarding gaps from the text. However, not all edit scripts are considered by the recurrences as some sets of edit scripts are *redundant* and generate the same sequence of the neighbourhood. Nevertheless, the recurrences of Lemma 1 generate redundant sequences from the neighbourhood and can then only be used to provide an upper bound to the maximum size of the condensed neighbourhood. For example, considering an alphabet composed of the single symbol $\{a\}$, $k = 5$ and $d = 2$, the sequence aaa belongs to the condensed neighbourhood of $aaaaa$ and is generated 10 times by the recurrences of Lemma 1. We refer to [6,7] for a detailed explanation of the recurrences and an analysis of the redundancy.

From these recurrences, Myers managed to prove that there exists a function $B(s,k,d,c)$

$$B(s,k,d,c) := \left(\frac{c+1}{c-1}\right)^k c^d s^d$$

such that for any $c \geq 1$

$$S(s,k,d) \leq B(s,k,d,c)$$

$$CN(s,k,d) \leq \frac{c}{c-1} B(s,k,d,c).$$

Moreover, it is not difficult to show that $B(s,k,d,c)$ is minimized when

$$c = c^\star := \epsilon^{-1} + \sqrt{1+\epsilon^{-2}}$$

with $\epsilon = d/k$. This leads to

$$\frac{c^\star}{c^\star - 1} = \frac{1+\sqrt{2}}{\sqrt{2}},$$

and we can deduce from there upper bound on both $S(s,k,d)$ and $CN(s,k,d)$, given in Theorem 1 below.

Theorem 1. *Let*

$$M(s,k,d) := \frac{1+\sqrt{2}}{\sqrt{2}} B(s,k,d,c^\star).$$

Then

$$S(s,k,d) \leq B(s,k,d,c^\star) \text{ and } CN(s,k,d) \leq M(s,k,d).$$

An important remark is that this upper bound involves a term with exponent d and a term with exponent k, while simple experiments suggest that, for given s and d, $CN(s,k,d)$ as a function of k fits to a polynomial of degree d [7].

3 Results

In this section, we use basic singularity analysis techniques from analytic combinatorics to illustrate how to approximate the bounds in Lemma 1, with a quantity we conjecture is an actual upper bound, which we show experimentally for $s = 4$ (the DNA alphabet size), $k \leq 50$ and $d \leq 4$. For more detail on the discussion, please see [7]. Then we apply this approximation to the expected time analysis of the approximate pattern matching algorithm introduced in [5].

3.1 An improved upper bound

Theorem 2. *Let $s, k, d \in \mathbb{N}$. Then asymptotically as $k \to \infty$*

$$T(s,k,d) \simeq \frac{(2s-1)^d k^d}{d!}.$$

To simplify the following discussion, we set

$$A(s,k,d) := \frac{(2s-1)^d k^d}{d!}.$$

We will first show the following limit, valid for positive s, d:

$$\lim_{k\to\infty} \frac{S(s,k,d)}{A(s,k,d)} = 1.$$

This limit can be shown by considering the generating function for the sequence $(S(s,k,d))_{k\geq 0}$. This is the formal power series

$$\mathbf{S}_{s,d}(z) := \sum_{k=1}^{\infty} S(s,k,d) z^k.$$

Analytic combinatorics (as described in [3]) connects the singularities of the series (viewed as function) to the behaviour of its coefficients. In this case, the generating functions $\mathbf{S}_{s,d}(z)$ are Taylor series of rational functions, and hence this is a straightforward.

The recurrences given in Lemma 1 lead immediately to a system of functional equations satisfied by $\mathbf{S}_{s,1}(z), \ldots, \mathbf{S}_{s,d}(z)$, for fixed but arbitrary s and d. The system is easily solvable and determines closed forms for the generating functions as rational functions. We provide an illustration below for $d = 1$, followed by a formal proof.

The recurrences of Lemma 1 translate into $S(s,k,1) = S(s,k-1,1)+(s-1)S(s,k-1,0) + (s-1)S(s,k-2,0) + S(s,k-2,0)$ which simplifies to

$$S(s,k,1) = S(s,k-1,1) + 2s - 1.$$

We convert the coefficient recurrence into a functional equation for $\mathbf{S}_{s,d}(z)$ by multiplying each side by z^k and summing from $k = 1$ to infinity. This gives:

$$\begin{aligned}
\mathbf{S}_{s,1}(z) - 1 = \sum_{k=1}^{\infty} S(s,k,1) z^k &= \sum_{k=1}^{\infty} S(s,k-1,1) z^k + (2s-1) \sum_{k=1}^{\infty} z^k \\
&= z \sum_{k=1}^{\infty} S(s,k-1,1) z^{k-1} + (2s-1) \sum_{k=1}^{\infty} z^k \\
&= z\mathbf{S}_{s,1}(z) + \frac{(2s-1)z}{1-z}.
\end{aligned}$$

From this we compute

$$\mathbf{S}_{s,1}(z) = \frac{(2s-2)z+1}{(1-z)^2}.$$

For any fixed d we can determine a recurrence, and solve it to determine a closed form for the generating function $\mathbf{S}_{s,d}(z)$. Indeed, this can be automated using a system of computer algebra such as Maple [4]. The next values are given in the following table.

d	$\mathbf{S}_{s,d}(z)$
2	$(z^3 - z^2(4s^2 - 4s + 3) + 2z - 1)(z-1)^{-3}$
3	$(z^5(2s^3 - s^2 - 3s + 3) - z^4(4s^3 - 6s^2 + 2s + 3) + z^3(10s^3 - 17s^2 + 11s - 2) + 3z^2 - 3z + 1)(z-1)^{-4}$

Given a rational function, it is straightforward to determine the asymptotic growth of its Taylor series coefficients [3], which, in our case, leads to the following formula for the dominant term of the asymptotic growth of the coefficients $S(s,k,d)$:

$$\lim_{k\to\infty} \frac{S(s,k,d)}{A(s,k,d)} = 1$$

We can extend this approach to the analysis of the size of the condensed neighbourhood, using the relation

$$CN(s,k,d) \leq S(s,k,d) + \sum_{j=1}^{d} s^j S(s,k-1,d-j).$$

given in Lemma 1. Let us denote by $\mathbf{T}_{s,d}(z)$ the ordinary generating function defined by

$$\mathbf{T}_{s,d}(z) := \sum_{k=1}^{\infty} T(s,k,d) z^k.$$

Applying the same technique than previously yields the following

$$\begin{aligned}\mathbf{T}_{s,d}(z) &= \sum_{k=1}^{\infty} S(s,k,d) z^k + \sum_{k=1}^{\infty}\sum_{j=1}^{d} s^j S(s,k-1,d-j) z^k \\ &= \mathbf{S}_{s,d}(z) + z\sum_{k=1}^{\infty}\sum_{j=1}^{d-1} s^j S(s,k-1,d-j) z^k + s^d \sum_{k=1}^{\infty} z^k\end{aligned}$$

which leads immediately to

$$\mathbf{T}_{s,d}(z) = \mathbf{S}_{s,d}(z) + z\left(\sum_{j=1}^{d-1} s^j \left(\mathbf{S}_{s,d-j}(z) - 1\right)\right) + \frac{s^d}{1-z}.$$

Asymptotic analysis of the generating function $\mathbf{T}_{s,d}(z)$ shows that asymptotically, its coefficients are equivalent to the function $A(s,k,d)$ defined above.

We provide at `https://github.com/cchauve/CondensedNeighbourhoods` a Maple session that illustrates this process and shows the bounds claimed in Theorem 2.

We now provide a formal proof. We denote by $[z^k]\mathbf{F}(z)$ the coefficient of z^k in a generating function $F(z)$.

Lemma 2. *Let d be a strictly positive integer. Suppose $P(z)$ is a polynomial such that $P(1) \neq 0$. Then asymptotically, when k becomes large,*

$$[z^k]\frac{P(z)}{(1-z)^{d+1}} \sim \frac{P(1)k^d}{d!}.$$

Proof. This follows from basic coefficient asymptotics of rational functions. The only (and hence dominant) singularity of $\frac{P(z)}{(1-z)^d}$ is at $z = 1$, and it is a pole of order $d+1$. The coefficient asymptotics are a direct consequence of the transfer theorem [2].

Lemma 3.

$$\mathbf{S}_{s,d}(z) = \frac{P_{s,d}(z)}{(1-z)^{d+1}}$$

where $P_{s,d}(z)$ is a polynomial that satisfies $P_{s,d}(1) = (2s-1)^d$.

Proof. We prove the result by induction on d. From the recurrences in Lemma 1, we can write

$$\begin{aligned}\mathbf{S}_{s,d}(z) = z\mathbf{S}_{s,d}(z) + z(s-1)\mathbf{S}_{s,d-1}(z) \\ + (s-1)z^2\mathbf{S}_{s,d-1}(z) + z\mathbf{S}_{s,d-1}(z) + \frac{X(z)}{(1-z)^{d-1}}\end{aligned}$$

where $X(z)$ is a linear combination of the $P_{s,d'}$ for $d' < d-1$. We rearrange to obtain

$$\mathbf{S}_{s,d}(z)(1-z) = \mathbf{S}_{s,d-1}(z)\left((s-1)z + (s-1)z^2 + z\right) + X(z).$$

By induction we have

$$\begin{aligned}\mathbf{S}_{s,d}(z) &= \frac{1}{(1-z)}\left(\frac{P_{s,d-1}(z)}{(1-z)^d}\left((s-1)z + (s-1)z^2 + z\right) + \frac{X(z)}{(1-z)^{d-1}}\right) \\ &= \frac{P_{s,d-1}(z)\left((s-1)z + (s-1)z^2 + z\right) + (1-z)X(z)}{(1-z)^{d+1}}\end{aligned}$$

The numerator, when evaluated at $z = 1$ is equal to $P_{s,d-1}(1)(2s-1) = (2s-1)^d$, upon applying the inductive hypothesis. This proves the claimed result.

Proof (Theorem 2). We know that

$$\mathbf{T}_{s,d}(z) = \mathbf{S}_{s,d}(z) + z\left(\sum_{j=1}^{d-1} s^j\left(\mathbf{S}_{s,d-j}(z) - 1\right)\right) + \frac{s^d}{1-z},$$

hence has a pole at $z = 1$. We know from Lemma 3 that it is a pole of order $d+1$, since the dominant singularity for $\mathbf{T}_{s,d}(z)$ is from $\mathbf{S}_{s,d}(z)$. Consequently, as was the case with $\mathbf{S}_{s,d}(z)$, we can show that

$$\lim_{k\to\infty} \frac{[z^k]\mathbf{T}_{s,d}(z)}{A(s,k,d)} = 1.$$

We illustrate in Fig. 1 the behaviour of the three expressions introduced so far to bound up $CN(s,k,d)$, $T(s,k,d)$, $M(s,k,d)$ and $A(s,k,d)$, which shows for $s=4$, $d \leq 4$ and $k \leq 50$ our asymptotics estimate is an actual upper bound that improves dramatically over the previous known upper bound[1].

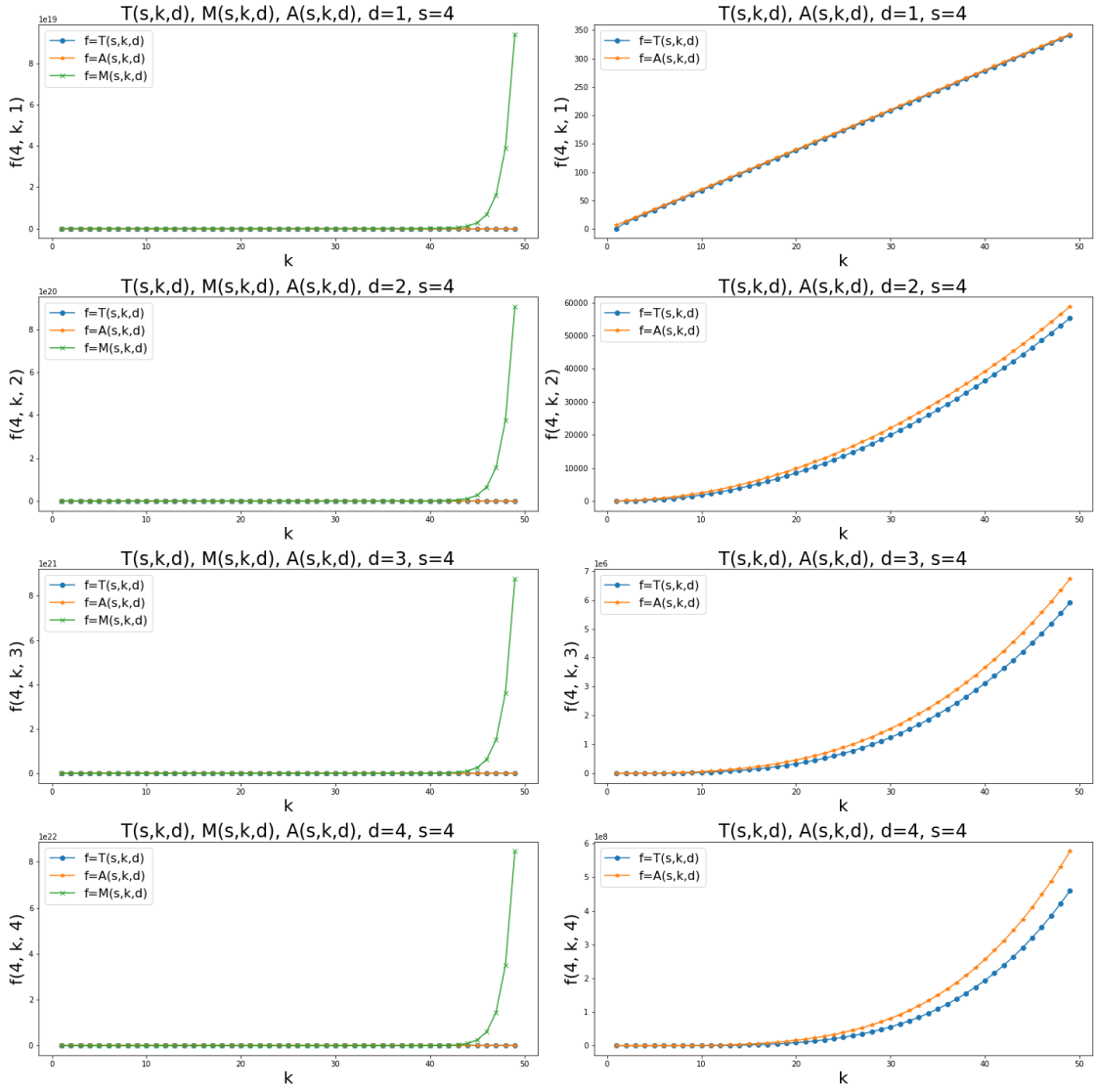

Figure 1. Illustration of the behaviour of $T(s,k,d)$, $M(s,k,d)$ and $A(s,k,d)$ for $s=4$, $d=1,2,3,4$ and $k \geq 30$. (Left): the three functions are shown. (Right): the functions $T(s,k,d)$ and $A(s,k,d)$ are shown.

Our experimental results lead to the proposition and conjecture below.

Proposition 1. *Let* $s \in \{1,\ldots,4\}$, $k \in \{1,\ldots,50\}$, $d \in \{1,\ldots,4\}$. *Then*

$$CN(s,k,d) \leq \frac{(2s-1)^d k^d}{d!}.$$

Conjecture 1 *Let* $s,k,d \in \mathbb{N}$. *Then*

$$CN(s,k,d) \leq \frac{(2s-1)^d k^d}{d!}.$$

[1] The python code used to generate the figures is available in the github repository `https://github.com/cchauve/CondensedNeighbourhoods`.

3.2 Approximate pattern matching expected-time complexity

Bounding the size of condensed neighbourhoods is a key element in the analysis of the time complexity of the approximate pattern matching algorithm described in [5]. The approximate pattern matching problem can be stated as follows: given a (long) text of length n, a (short) pattern of length p, and an integer $e < p$, how can we find in the text all the occurrences of sequences that are at distance at most e from the pattern (e-approximate pattern occurrences).

Myers describes an algorithm that, for a given value k to be discussed later, splits the pattern into p/k non-overlapping substrings of length k (k-mers), then computes for each such k-mer its condensed neighbourhood, searches (through a pre-built index) occurrences of the sequences in these neighbourhoods in the text, and for any such occurrence, tries to extend it into an approximate pattern occurrence. The algorithm is more complex than the high level overview above, but we are interested here in its expected time complexity, under the assumption Conjecture 1 is true.

Expected-time complexity analysis from [6]. Let $\epsilon = e/p$ and so $d = \lceil k\epsilon \rceil$. Assume there exists a function $\alpha(\epsilon)$ such that

$$\frac{1}{\alpha(\epsilon)^k}$$

is an upper bound to the maximum probability, taken over all possible k-mers w, that a random position in a random Bernoulli text is the start of a d-approximate occurrence of w (i.e. belongs to its condensed neighbourhood). Denote this probability by $Pr(k, d)$.

Then, if h is the expected number of e-approximate occurrences of the pattern in the text, the expected-time complexity of the algorithm described in [5] is

$$O\left(e \cdot CN(s,k,d) + n \cdot e \cdot k \cdot Pr(k,d) + h \cdot e \cdot p\right)$$

which gives

$$O\left(e \cdot CN(s,k,d) + \frac{n \cdot e \cdot k}{\alpha(\epsilon)^k} + h \cdot e \cdot p\right)$$

with an optimal value of k being $k = \log_s(n)$.

It can be shown that $Pr(k, d)$ is bounded by $CN(s, k, d)/s^k$. From this, Myers deduced that $Pr(k, d)$ is bounded above by

$$\left(\frac{c^*}{c^* - 1}\right) \frac{B(s,k,d,c^*)}{s^k}$$

to define α by

$$\alpha(\epsilon) := \left(\frac{c^* - 1}{c^* + 1}\right) (c^*)^{-\epsilon} s^{1-\epsilon}.$$

Moreover, if we define

$$pow(\epsilon) := \log_s \left(\frac{c^* + 1}{c^* - 1}\right) + \epsilon \log_s (c^*) + \epsilon,$$

then

$$CN(s,k,d) = O\left(\left(s^{pow(\epsilon)}\right)^k\right) \text{ and } \alpha(\epsilon) = O\left(s^{1-pow(\epsilon)}\right)$$

which implies that the expected time complexity of the algorithm is

$$O\left(e\left(s^k\right)^{pow(\epsilon)}\left(1+k\frac{n}{s^k}\right)+h\cdot e\cdot p\right)$$

which gives, for $k=\log_s(n)$,

$$O\left(e\cdot n^{pow(\epsilon)}\cdot\log_s(n)+h\cdot e\cdot p\right).$$

This is sub-linear in n if $pow(\epsilon)<1$, which Myers showed is true if $\epsilon\leq 1/3$ for $s=4$ (DNA alphabet), i.e. if we are looking for approximate pattern occurrences with roughly a 33% difference rate.

Improved expected-time complexity analysis. The motivation for obtaining a better bound on $CN(s,k,d)$ is to improve the exponential factor (parameterized by ϵ), currently $pow(\epsilon)$. Of particular interest is the question of a bounding function that would intersect the line $y=1$ further than $\epsilon=1/3$ and would thus increase the window of sublinearity of the approximate pattern matching algorithm described in [5].

The key assumption is that, for fixed s and d, $CN(s,k,d)/s^k$ provides an upper bound to $Pr(k,d)$. This leads to the following expected time complexity for the full algorithm, using our improved upper bound $A(s,k,d)$:

$$O\left(e\cdot A(s,k,d)+n\cdot e\cdot k\cdot\frac{A(s,k,d)}{s^k}+h\cdot e\cdot p\right)$$

or equivalently

$$O\left(e\cdot A(s,k,d)\left(1+k\frac{n}{s^k}\right)+h\cdot e\cdot p\right)$$

and we are left with the task to see if there exists a function $f(\epsilon)$ such that

- $A(s,k,d)$ can be expressed as or bounded upon by $s^{k\cdot f(\epsilon)}$, or, if we assume $k=\log_s(n)$, $n^{f(\epsilon)}$, and
- it intersects $y=1$ further than $pow(\epsilon)$.

To evaluate this experimentally, we computed the value of $A(s,k,d)$ and $n^{pow(\epsilon)}$ for $s=4$ and $k=\lceil\log_S(n)\rceil$ for values of n going up to 10^9 (roughly the size of a human genome) and various values of ϵ. We also compared $pow(\epsilon)$ with $\log_n(A(s,k,d))$, taken as a function of ϵ, for $n=10^9$. The results of both computations are shown in Fig. 2.

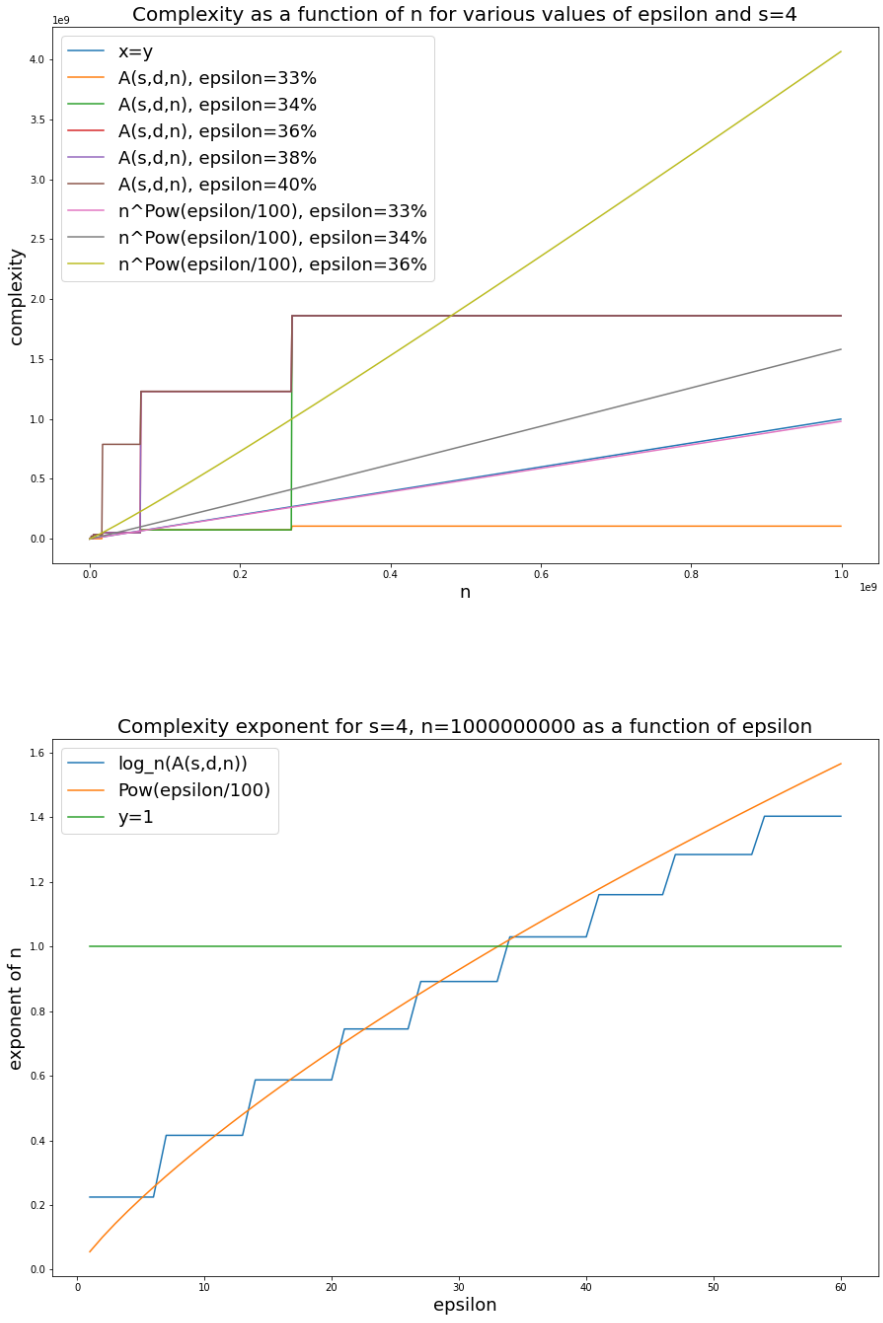

Figure 2. (Top) Illustration of the behaviour of $A(s,k,d)$ and $n^{pow(\epsilon)}$ compared to $f(n) = n$. (Bottom) Illustration of the behaviour of $pow(\epsilon)$ and $\log_n(A(s,k,d))$ for $n = 10^9$ as a function of ϵ.

4 Conclusion

This work contains two main parts. The first one (Theorem 2 and Conjecture 1) suggests a novel upper bound for the size of the condensed neighbourhood. The experimental results illustrated in Fig. 1 suggest strongly that this upper bound is much better than the one provided in [6], although we do not have a formal proof of this claim at the time. The code we provide allows to actually test if our estimate is an actual upper-bound for any given setting defined by s, k, d. Nevertheless, we can observe that our asymtotics expression is very close the actual expression $T(s, k, d)$, although the gap widens as d increases.

The second part addresses the expected-time complexity of the approximate pattern matching algorithm introduced in [5], with the goal to extend the window of sublinearity of the algorithm in terms of the parameter ϵ. We can observe on Fig. 2 that despite being tighter, our upper bound on the size of the condensed neighbourhood does not seem to lead to a much wider window of sublinearity, for a value of n close to the size of a human genome. Indeed, we can see that $\log_n(A(s, k, d))$ intersects with $y = 1$ just right of $pow(\epsilon)$. So we do not seem to obtain a significant improvement, although we obtain a slight one. Note however that the behaviour of the function shown in Fig. 2 based on $A(s, k, d)$ depends significantly on the fact that we impose that k and d are integers and thus take $k = \lceil \log_s(n) \rceil$ and $d = \lceil k\epsilon \rceil$. The graph of the function where any of these expressions is taken as a floating number intersects $y = 1$ much further to $1/3$.

Our results might indicate that extending the window of sublinear behaviour of [5] might require to improve the recurrences of Lemma 1 more than to obtain a tighter bound of $S(s, k, d)$. Indeed, the recurrences of Lemma 1 result in edit scripts that lead to some redundant words, and there might thus be room to obtain better recurrences, on which similar analytic combinatorics techniques could then be applied to obtain tight bounds.

References

1. S. Altschul, W. Gish, W. Miller, E. Myers, and D. Lipman: *Basic local alignment search tool.* Journal of Molecular Biology, 215 1990, pp. 403–410.
2. P. Flajolet and A. M. Odlyzko: *Singularity analysis of generating functions.* SIAM J. Discret. Math., 3(2) 1990, pp. 216–240.
3. P. Flajolet and R. Sedgewick: *Analytic Combinatorics*, Cambridge University Press, New York, NY, USA, 1 ed., 2009.
4. Maplesoft, a division of Waterloo Maple Inc..: *Maple.*
5. E. W. Myers: *A sublinear algorithm for approximate keyword searching.* Algorithmica, 12(4/5) 1994, pp. 345–374.
6. G. Myers: *What's Behind Blast*, in Models and Algorithms for Genome Evolution, Springer, 2013, pp. 3–15, Video presentation at `https://www.youtube.com/watch?v=pVFX3V0Q2Rg`.
7. F. Paquet-Nadeau: *On the maximum size of condensed sequences neighbourhoods under the Levenshtein distance.* MSc Project Report, Department of Mathematics, Simon Fraser University, Burnaby, BC, Canada, URL: `https://github.com/cchauve/CondensedNeighbourhoods`, 2017.