Федеральный исследовательский центр Информатика и Управление Российской академии наук

RUDN university

РЭУ.РФ Российский экономический университет имени Г.В. Плеханова

КОМПЬЮТЕРНАЯ АЛГЕБРА

Материалы 6-й международной конференции

Москва, 23–25 июня 2025 г.

COMPUTER ALGEBRA

6th International Conference Materials

Moscow, June 23–25, 2025

Moscow
2025

UDC 519.6(063)
BBC 22.19;431

Editors: PhD Anna A. Ryabenko, DSc Dmitry S. Kulyabov

Reviewers: PhD A. V. Nesterov, PhD K. P.Lovetskiy

Printed in author's edition

Computer algebra: 6th International Conference Materials. Moscow, 23–25 June, 2025 / ed. A. A. Ryabenko, D. S. Kulyabov. Moscow: RUDN University.

DOI: 10.22363/12585-2025-6-000.
EDN: DZBCNS.

The international conference is organized by Federal Research Center "Computer Science and Control" of RAS, Peoples' Friendship University of Russia and Plekhanov Russian University of Economics. The talks presented at the conference discuss actual problems of computer algebra — the discipline whose algorithms are focused on the exact solution of mathematical and applied problems using a computer.

For scientists, graduate and undergraduate students in mathematics, physics and computer science.

ISBN 978-5-209-12585-3

Scientific publication

Searching for an Imperfect Palindrome

Georgii A. Khaziev[1,*], Alexandr V. Seliverstov[1] and Oleg A. Zverkov[1]

[1]*Institute for Information Transmission Problems of the Russian Academy of Sciences (Kharkevich Institute), Bolshoy Karetny per. 19, build.1, Moscow, 127051, Russian Federation*

Abstract

We consider algorithm for optimizing an imperfect palindrome in the input sequence. Our algorithm runs in quadratic time, i. e., it is faster than exhaustive search of admissible palindromes. Also we propose new approach called "trimming" to find long substrings of a sequence, that are closer to palindrome, than sequence itself.

Keywords

palindrome, edit distance, Python, Numba, bioinformatics, computational complexity

1. Introduction

There are many works devoted to the search for perfect as well as imperfect and degenerate palindromes [1]. We consider imperfect palindromes, i. e., sequences with gaps and mismatches in some positions. On the other hand, the edit distance between two sequences can be calculated using dynamic programming[3], but the algorithm does not involve palindromic structures. The connection between the longest common subsequence problem and the Hecke monoid has been rediscovered many times in different forms [4].

Let us consider nucleotide sequences. The nucleotides $\{\mathsf{A}, \mathsf{C}, \mathsf{G}, \mathsf{T}\}$ form two complementary pairs: $\mathsf{c}(\mathsf{A}) = \mathsf{T}$, $\mathsf{c}(\mathsf{T}) = \mathsf{A}$, $\mathsf{c}(\mathsf{C}) = \mathsf{G}$, and $\mathsf{c}(\mathsf{G}) = \mathsf{C}$. Next, $\mathsf{c}()$ denotes the reverse complement, i. e., $\mathsf{c}(xy) = \mathsf{c}(y)\mathsf{c}(x)$. For example, $\mathsf{c}(\mathsf{AACG}) = \mathsf{CGTT}$. In DNA, a perfect palindrome is an inverted sequence repeat, i. e., reverse complement of itself. Let us omit the concatenation symbol. So, all perfect palindromes are of the type $x\mathsf{c}(x)$, where x denotes a sequence. In particular, a sequence of odd length cannot be any perfect palindrome. There are many examples of nucleotide sequences with perfect as well as imperfect palindromes [5]. Regulatory palindromes are typically imperfect [2].

For two sequences x and y, let $\mathrm{dist}(x, y)$ denote the edit distance. Of course,

$$\begin{aligned} \mathrm{dist}(x, y) &= \mathrm{dist}(\mathsf{c}(x), \mathsf{c}(y)) \\ \mathrm{dist}(wx, wy) &= \mathrm{dist}(x, y) \\ \mathrm{dist}(xz, yz) &= \mathrm{dist}(x, y) \end{aligned}$$

2. Results

Let us denote by $|x|$ the length of x. If sequences x and y coincide, then the edit distance in Theorem 1 vanishes for a perfect palindrome. Let us denote by $\mathrm{imp}(x)$ the ratio of the minimum edit distance to the length of the sequence:

$$\mathrm{imp}(x) = \frac{\min\{\mathrm{dist}(x, w\mathsf{c}(w)) | x = wz\}}{|x|}.$$

6^{th} *International Conference "Computer Algebra", Moscow, June 23–25, 2025*

10.22363/12585-2025-6-013

EDN: CCODUV

*Corresponding author.

georgy.haziev@yandex.ru (G. A. Khaziev)

0009-0003-6466-8595 (G. A. Khaziev); 0000-0003-4746-6396 (A. V. Seliverstov); 0000-0002-8546-364X (O. A. Zverkov)

The ratio shows how imperfect the palindrome is. The correctness of the definition is based on the next theorem.

Theorem 1. *There is a quadratic-time algorithm that computes the* $\mathrm{imp}(x)$ *function as well as optimal partition of the input sequence* x *as a concatenation* $x = wz$ *that minimizes the edit distance between* x *and the palindrome* $w\mathrm{c}(w)$.

Theorem 2. $\mathrm{imp}(x) = \mathrm{imp}(\mathrm{c}(x))$.

Proof. For all $x = wz$, because the prefixes match, $\mathrm{dist}(x, w\mathrm{c}(w)) = \mathrm{dist}(z, \mathrm{c}(w))$. And because the suffixes match, $\mathrm{dist}(x, \mathrm{c}(z)z) = \mathrm{dist}(w, \mathrm{c}(z))$. Of course, since $\mathrm{dist}(a, b)$ is symmetric, it is true that $\mathrm{dist}(z, \mathrm{c}(w)) = \mathrm{dist}(\mathrm{c}(w), z) = \mathrm{dist}(w, \mathrm{c}(z))$. Thus, $\mathrm{dist}(x, w\mathrm{c}(w)) = \mathrm{dist}(x, \mathrm{c}(z)z)$. Next, for all z, $\mathrm{dist}(x, \mathrm{c}(z)z) = \mathrm{dist}(\mathrm{c}(x), \mathrm{c}(z)z)$ because $\mathrm{c}(z)z$ is a palindrome. So, the equality $\mathrm{dist}(x, w\mathrm{c}(w)) = \mathrm{dist}(\mathrm{c}(x), \mathrm{c}(z)z)$ holds. □

Theorem 3. *For all even-length sequences it is true that* $\mathrm{imp}(x) \leq 1/2$. *For all odd-length sequences it is true that* $0 < \mathrm{imp}(x) \leq (1 + 1/|x|)/2$.

Substring is a contiguous sequence of characters within a string. The main idea behind the algorithm to select an imperfect palindrome is checking whether one of the optimal lengths of the prefix w of the input sequence x differs significantly from $|x|/2$. If such case occurs, then the algorithm deletes either prefix or suffix by difference between $|w|$ and $|x|/2$.

The software implementation in Python with the Numba library is available at http://lab6.iitp.ru/-/trimmers. Using Numba generally increases performance of Python code. Performance increase in comparison with base Python is achieved using JIT (just-in-time) compilation. While the base Python interpreter compiles programs into bytecode, which is executed on a virtual machine, Numba recompiles this bytecode into machine code with optimizations tailored to the CPU architecture being used. Additionally, a significant performance benefit is gained through Numba's use of type inference, allowing type-specific optimization during compilation.

All three functions `pref_trimmer`, `suff_trimmer`, and `double_trimmer` take as input nucleotide sequence x, sorted list `optimal_lengths` of optimal prefix lengths $|w|$, and floating-point number `cutoff_condition`. Note that $\min($`optimal_lengths`$)$ and $\max($`optimal_lengths`$)$ are the first and last elements of `optimal_lengths`, respectively.

The `pref_trimmer` function trims first rd symbols of x, where

$$rd = \max(\texttt{optimal\_lengths}) - \left\lfloor \frac{|x|}{2} \right\rfloor,$$

when $rd \geq \mathrm{length}(x) \cdot$ `cutoff_condition` is satisfied.

The `suff_trimmer` function trims last ld symbols of x, where

$$ld = \left\lfloor \frac{|x|}{2} \right\rfloor - \min(\texttt{optimal\_lengths}),$$

when $ld \geq |x| \cdot$ `cutoff_condition` is satisfied.

The `double_trimmer` function initially computes

$$rd = \max(\texttt{optimal\_lengths}) - \left\lfloor \frac{|x|}{2} \right\rfloor$$

and

$$ld = \left\lfloor \frac{|x|}{2} \right\rfloor - \min(\texttt{optimal\_lengths}).$$

Subsequently it checks if

$$rd \geq |x| \cdot \texttt{cutoff\_condition}$$

and

$$ld \geq |x| \cdot \texttt{cutoff\_condition}.$$

If the first inequality is satisfied, the function trims the first rd symbols from the string x. Similarly, if the second inequality is satisfied, the function trims the last ld symbols from x.

Theorem 4. *For perfect palindrome x for any* `cutoff_condition` > 0 *no trimming would be performed.*

Proof. Let x be perfect palindrome. Then it's only optimal partition is precisely at $\left\lfloor \frac{|x|}{2} \right\rfloor$. Then, $ld = \left\lfloor \frac{|x|}{2} \right\rfloor - \left\lfloor \frac{|x|}{2} \right\rfloor = 0$ and $rd = \left\lfloor \frac{|x|}{2} \right\rfloor - \left\lfloor \frac{|x|}{2} \right\rfloor = 0$. For non-zero `cutoff_condition` rd and ld will both be less than $|x| \cdot$ `cutoff_condition` so no trimming would be performed. □

Theorem 5. *For any $n \geq 3$, there exist* `cutoff_condition` > 0 *and x with length of at least n, which satisfies both prefix and suffix trimming conditions such that*

$$\texttt{pref\_trimmer}(x_{suff}) \neq \texttt{suff\_trimmer}(x_{pref}),$$

where x_{suff} and x_{pref} are results of suffix and prefix trimming of x, respectively.

Proof. Let $x = \underbrace{\text{ATA...ATA}}_{\alpha}$ where $\alpha = 2\left\lfloor \frac{n}{2} \right\rfloor + 1$, i. e. minimal odd number greater than or equal to n. For that x optimal partitions would be at $\left\lceil \frac{n}{2} \right\rceil - 1$ and $\left\lceil \frac{n}{2} \right\rceil + 1$. Minimal index of optimal partition corresponds to perfect palindrome $\underbrace{\text{AT...AT}}_{\alpha-1}$ and maximal index of optimal partition corresponds to perfect palindrome $\underbrace{\text{AT...AT}}_{\alpha+1}$. Let us choose `cutoff_condition` $= \frac{1}{10\alpha}$. Since this value would satisfy both prefix and suffix trimming condition, x_{suff} and x_{pref} could be computed and will be $\underbrace{\text{AT...AT}}_{\alpha-1}$ and $\underbrace{\text{T...ATA}}_{\alpha-1}$, respectively. Note, that both sequences are perfect palindromes and thus neither suffix nor prefix trimming would be performed for them. □

3. Conclusion

We are confident that the obtained results will be successfully applied to the prediction of imperfect palindromes in nucleotide sequences. In particular, it is crucial for predicting gene expression regulations as well as RNA structures. The implementation of algorithms in Python will enable a wide range of bioinformaticians to apply them in their work. Moreover, low computational complexity allows efficient processing of large datasets.

Author Contributions: Conceptualization, methodology, programming, writing original draft preparation, Georgii A. Khaziev; Proof of Theorem 1, editing, Alexandr V. Seliverstov; Programming, creating the web page, and editing, Oleg A. Zverkov; All authors have read and agreed to the published version of the manuscript.

Funding: The research was carried out within the state assignment of Ministry of Science and Higher Education of the Russian Federation for IITP RAS

Data Availability Statement: Refer to http://lab6.iitp.ru/-/trimmers

Conflicts of Interest: The authors declare no conflict of interest.

Acknowledgments: We thank Vassily A. Lyubetsky for creating friendly and healthy work environment.

References

[1] M. Alzamel, C. Hampson, C. S. Iliopoulos, Z. Lim, S. Pissis, D. Vlachakis, S. Watts, Maximal degenerate palindromes with gaps and mismatches, Theoretical Computer Science 978 (2023) 114182. doi:https://doi.org/10.1016/j.tcs.2023.114182.

[2] R. R. Datta, J. Rister, The power of the (imperfect) palindrome: Sequence-specific roles of palindromic motifs in gene regulation, BioEssays 44 (2022) 2100191. doi:https://doi.org/10.1002/bies.202100191.

[3] S. B. Needleman, C. D. Wunsch, A general method applicable to the search for similarities in the amino acid sequence of two proteins, Journal of Molecular Biology 48 (1970) 443–453. doi:10.1016/0022-2836(70)90057-4.

[4] A. V. Tiskin, The Chvátal–Sankoff problem as a problem of symbolic dynamics, Zap. Nauchn. Sem. POMI 528 (2023) 214–237.

[5] O. A. Zverkov, A. V. Seliverstov, G. A. Shilovsky, Alignment of a hidden palindrome, Mathematical Biology and Bioinformatics 19 (2024) 427–438. doi:10.17537/2024.19.427.