

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
Факультет прикладной математики и информатики  
BELARUSIAN STATE UNIVERSITY  
Faculty of Applied Mathematics and Informatics

# **ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ**

Материалы  
XI Международного научного конгресса по информатике  
(CSIST-2025)

Республика Беларусь  
Минск, 29–31 октября 2025 г.

В двух частях  
Часть 1

# **INFORMATION SYSTEMS AND TECHNOLOGIES**

Proceedings  
of the XI International Scientific Congress on Computer Science  
(CSIST-2025)

Republic of Belarus  
Minsk, October 29–31, 2025

In two parts  
Part 1

Научное электронное издание

Минск, БГУ, 2025

ISBN 978-985-881-852-4 (ч. 1)  
ISBN 978-985-881-851-7

© БГУ, 2025

УДК 004(06)  
ББК 32.97я431

**Редакционная коллегия:**  
академик НАН Беларуси, доктор технических наук,  
профессор *С. В. Абламейко* (гл. ред.);  
доктор педагогических наук,  
профессор *В. В. Казаченок* (зам. гл. ред.);  
кандидат физико-математических наук, доцент *Н. М. Дмитрук*;  
член-корреспондент НАН Беларуси,  
доктор технических наук, профессор *А. В. Тузиков*;  
доктор физико-математических наук, профессор *А. Ю. Харин*

**Рецензенты:**  
академик НАН Беларуси,  
доктор физико-математических наук, профессор *Ю. С. Харин*;  
член-корреспондент НАН Беларуси,  
доктор технических наук, профессор *А. В. Тузиков*;  
доктор физико-математических наук, профессор *Н. А. Лиходед*

**Информационные системы и технологии** = Information Systems and Technologies : материалы XI Междунар. науч. конгр. по информатике (CSIST-2025), Респ. Беларусь, Минск, 29–31 окт. 2025 г. В 2 ч. Ч. 1 / Белорус. гос. ун-т ; редкол.: С. В. Абламейко (гл. ред.) [и др]. – Минск : БГУ, 2025. – 1 электрон. опт. диск (CD-ROM). – Текст : электронный. – ISBN 978-985-881-852-4.

Представлены материалы международного научного конгресса, организованного Белорусским государственным университетом и Объединенным институтом проблем информатики НАН Беларуси.

Рассмотрены вопросы информационной и компьютерной безопасности, биоинформатики и приложений, интеллектуального и статистического анализа данных и принятия решений, оптимизации и надежности систем, параллельной и распределенной обработки данных.

---

**Минимальные системные требования:**  
PC, Pentium 4 или выше; RAM 1 Гб; Windows XP/7/10;  
Adobe Acrobat.  
Оригинал-макет подготовлен в программе Microsoft Word  
На русском и английском языках  
В авторской редакции  
Ответственный за выпуск *С. В. Шолтанюк*  
Подписано к использованию 27.10.2025. Объем 10 МБ  
Белорусский государственный университет.  
Управление редакционно-издательской работы.  
Пр. Независимости, 4, 220030, Минск.  
Телефон: (017) 259-72-40.  
e-mail: [urir@bsu.by](mailto:urir@bsu.by)  
<http://elib.bsu.by/>

## ПОИСК ПЕТЛИ В НЕСОВЕРШЕННОМ ПАЛИНДРОМЕ

Г. А. Хазиев, О. А. Зверков, Л. И. Рубанов, А. В. Селиверстов

*Институт проблем передачи информации имени А. А. Харкевича РАН,  
Москва, Россия, [khaziev@iitp.ru](mailto:khaziev@iitp.ru)*

Палиндромы часто встречаются при анализе нуклеотидных последовательностей. Вопрос автоматического обнаружения несовершенных палиндромов до сих пор остаётся открытым. Мы предлагаем алгоритм `de_shapker` выделения петли в несовершенном палиндроме. Авторы протестировали работу алгоритма на нескольких множествах высококонсервативных элементов (ВКЭ).

**Ключевые слова:** несовершенные палиндромы; шпильки; биоинформатика.

## SEARCHING FOR THE LOOP IN IMPERFECT PALINDROME

G. A. Khaziev, O. A. Zverkov, L. I. Rubanov, A. V. Seliverstov

*Institute for Information Transmission Problems of the Russian Academy of Sciences  
(Kharkevich Institute),  
Moscow, Russia, [khaziev@iitp.ru](mailto:khaziev@iitp.ru)*

Palindromes are often encountered in analysis of nucleotide sequences. The question of automatic detection of imperfect palindromes is still open. We propose the `de_shapker` algorithm for identifying a loop in an imperfect palindrome. The authors tested the algorithm on several sets of highly conserved elements (HCE).

**Keywords:** imperfect palindromes; hairpins; bioinformatics.

### 1. Введение

Последовательность нуклеотидов называется совершенным палиндромом, если она комплементарна сама себе. Последовательность, отличающаяся от совершенного палиндрома, называется несовершенным палиндромом. Ранее авторами были предложены квадратичный алгоритм `palindrome_self_alignment` поиска наименьшего редакционного расстояния между последовательностью  $x$  и совершенным палиндромом, который был получен конкатенацией префикса  $x$  и последовательности, комплементарной этому префиксу [1]. Также авторами была предложена функция

$$\text{imp}(x) = \frac{\min \{ \text{dist}(x, wc(w) \mid x = wz \} }{|x|},$$

где  $\text{dist}(\cdot)$  – редакционное расстояние,  $c(\cdot)$  – комплементарная последовательность, а  $|\cdot|$  – длина последовательности. Данная функция характеризует близость последовательности  $x$  к совершенному палиндрому. Чем ближе строка к совершенному палиндрому, тем ближе к нулю значение функции  $\text{imp}(x)$ . Сложность поиска близких к палиндрому последовательностей в молекулярной биологии состоит в наличие длинных подпоследовательностей внутри близких к палиндрому строк, нарушающих общую палиндромность. Для частичного решения данной проблемы, ранее авторами были предложены алгоритмы усечения, целью которых является выделение длинной подстроки  $x$ , более близкой к совершенному палиндрому, чем  $x$  [2].

## 2. Алгоритм удаления петли

Авторы предлагают алгоритм `de_sharker`, целью которого является выделение подпоследовательности с более низким значением функции  $\text{imp}(x)$  с помощью нахождения некомплементарного себе участка внутри последовательности – петли. В отличие от алгоритмов усечения, которые находили подпоследовательность с более высоким значением  $\text{imp}(x)$  с помощью удаления нуклеотидов на краях последовательности, алгоритм `de_sharker` удаляет участок внутри последовательности. Такой участок характерен для несовершенных палиндромов, встречающихся в биологических задачах. Алгоритм получает на вход строку  $x$ , значение  $\text{imp\_given} = \text{imp}(x)$ , матрицу  $H$ , вычисленную в алгоритме `palindrome_self_alignment` для строки  $x$ , число итераций алгоритма `iteration_counter`, первоначальный размер окна `window_size`, приращение к размеру окна между итерациями `window_delta`, а также стратегию выбора координат начала петли `strategy`. Строка  $x$  записывается в строку `result`. Далее, для всех непрерывных подматриц матрицы  $H$  размера `window_size`  $\times$  `window_size` вычисляется  $l_1$  норма по формуле

$$l_1(A) = \sum_{i=1}^{\text{window\_size}} \sum_{j=1}^{\text{window\_size}} A_{ij},$$

где  $A$  – подматрица  $H$ . Из индексов  $(u, v)$  в матрице  $H$  левого верхнего элемента наименьшей по норме подматрицы выбирается индекс  $s$  начала найденного участка петли. За выбор индекса отвечает параметр `strategy`. Затем, в  $x$  удаляется участок начиная с  $s$ , заканчивая  $s + \text{window\_size}$  и

записывается в строку  $x\_new$ . Для полученной строки вычисляется значение функции  $\text{imp}(x\_new)$ . Если данное значение меньше, чем  $\text{imp\_given}$ , то  $x\_new$  записывается в строку  $\text{result}$  и начинается новая итерация, иначе, алгоритм завершает работу и возвращает строку  $\text{result}$ .

### 3. Результаты тестирования

В некодировующих областях геномов *Homo sapiens*, *Macaca fascicularis*, *Mus musculus*, *Sus scrofa* были выделены высококонсервативные элементы (ВКЭ), находящиеся на первой хромосоме человека с использованием модификации метода, описанного в [3].

Для двух множеств этих ВКЭ был проведён поиск палиндромов с использованием алгоритмов усечения. Затем, к каждой последовательности была применена функция  $\text{de\_shapker}$  с четырьмя различными значениями параметра  $\text{strategy}$ :  $\max(u, v)$ ,  $\min(u, v)$ ,  $\lfloor \text{mean}(u, v) \rfloor$ ,  $\lceil \text{mean}(u, v) \rceil$ . Кроме того, варьировались значения числа итераций: для каждой последовательности выполнялось до 3, 5 и 10 операций. Стартовый размер окна был равен 2, приращение  $\text{window\_delta}$  также был равен 2. Последовательность  $x$  считалась близкой к палиндрому, если для одной из её подпоследовательностей  $x_s$ , полученной в результате анализа, выполнялось  $\text{imp}(x_s) \leq 0.2$ .

В первом множестве из 5689 последовательностей - blk01-12.37.8 с помощью усечения был найден 471 палиндром. Количество палиндромов, найденных с помощью функции  $\text{de\_shapker}$ , указано на рис. 1.

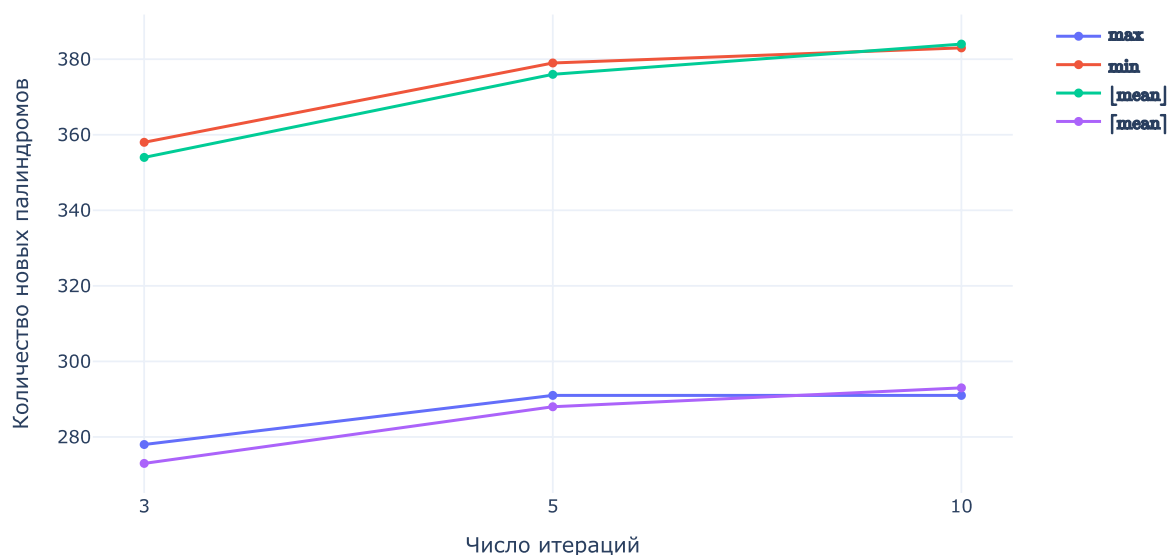


Рис. 1. Зависимость между числом итераций алгоритма  $\text{de\_shapker}$  и количеством найденных палиндромов для множества blk01-12.37.8

Аналогичные эксперименты были проведены для множества blk01-10.39.8. В данном множестве содержится 5862 последовательности, в них с помощью усечения было найдено 475 палиндромов. Количество палиндромов, найденных с помощью функции `de_shapker`, указано на рис. 2.

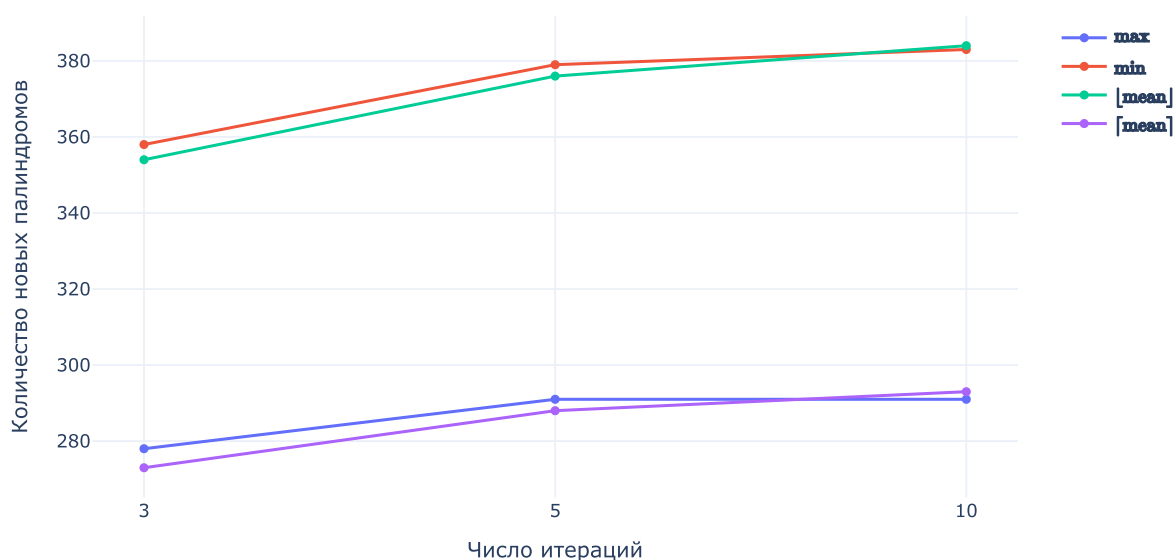


Рис. 2. Зависимость между числом итераций алгоритма `de_shapker` и количеством найденных палиндромов для множества blk01-10.39.8

Из рисунков выше можно сделать два основных вывода. Первый вывод заключается в том, что выбор в качестве параметра `strategy` функции, смещённую к меньшему из значений ( $u$ ,  $v$ ) приводит к большему числу обнаруженных палиндромов. Второй вывод состоит в том, что между 3 и 5 итерациями прирост числа палиндромов сильно больше, чем между 5 и 10, следовательно, при анализе большого количества последовательностей можно ограничиться значением `iteration_counter=5` для экономии времени вычисления.

Кроме того, для найденных палиндромов было вычислен процент оставшейся строки после усечения и применения функции `de_shapker` (табл. 1–2). Из таблиц видно, что даже несмотря на небольшое уменьшение средней доли строки после усечения и удаления петли для каждой отдельной стратегии, в среднем, строки потеряли небольшую долю от своей длины. Таким образом, алгоритмы не выдают в результате анализа слишком короткие последовательности, близкие к палиндромам в силу своей длины, а позволяют определить те последовательности, которые действительно могут содержать длинные подпоследовательности, близкие к палиндромам.

Таблица 1

**Средняя доля строки после усечения в blk01-12.37.8**

| strategy | 3 итерации | 5 итераций | 10 итераций |
|----------|------------|------------|-------------|
| min      | 81,6%      | 80,5%      | 80,5%       |
| max      | 81,5%      | 80,5%      | 80,2%       |
| ⌊ mean ⌋ | 81,5%      | 80,5%      | 80%         |
| ⌈ mean ⌉ | 81,6%      | 80,5%      | 80%         |

*Примечание.* Средний процент от первоначальной строки, оставшийся в результате всех преобразований.

Таблица 2

**Средняя доля строки после усечения в blk01-10.39.8**

| strategy | 3 итерации | 5 итераций | 10 итераций |
|----------|------------|------------|-------------|
| min      | 81,3%      | 80,2%      | 80%         |
| max      | 81%        | 80,3%      | 80%         |
| ⌊ mean ⌋ | 81%        | 80,2%      | 79,9%       |
| ⌈ mean ⌉ | 81,2%      | 80,3%      | 80%         |

*Примечание.* Средний процент от первоначальной строки, оставшийся в результате всех преобразований.

**4. Заключение**

Рассмотренный алгоритм позволяет точнее определять близость строки к совершенному палиндрому. На экспериментальных данных, при выборе стратегии лучше всего себя показали функции минимума и округлённого вниз среднего.

**Библиографические ссылки**

1. Зверков О. А., Селиверстов А. В., Шиловский Г. А. Выравнивание скрытого палиндрома // Математическая биология и биоинформатика. 2024. Т. 19. № 2. С. 427–438. DOI: [10.17537/2024.19.427](https://doi.org/10.17537/2024.19.427).
2. Khaziev G. A., Seliverstov A. V., Zverkov O. A. Searching for an Imperfect Palindrome // Computer algebra : 6th International Conference Materials, Moscow, June 23–25, 2025 / eds.: A. A. Ryabenko, D. S. Kulyabov. Moscow : RUDN University, 2025. P. 62–65.
3. Новый алгоритм поиска несовершенных палиндромов в ДНК / Г. А. Хазиев [и др.] // MCCMB 2025 : Сборник тезисов 12-й Московской конференции по вычислительной молекулярной биологии (MCCMB). С. 621–624. URL: <https://www.mccmb.info/> (дата обращения: 10.09.2025).