

УСЕЧЕНИЕ НЕСОВЕРШЕННЫХ ПАЛИНДРОМОВ

© 2026 г. Г. А. Хазиев*, О. А. Зверков*, А. В. Селиверстов*

*Институт проблем передачи информации им. А.А. Харкевича Российской академии наук
127051 Москва, Большой Каретный переулок, д. 19 с. 1
E-mail: khaziev@iitp.ru, zverkov@iitp.ru, slvstv@iitp.ru

Поступила в редакцию 14.08.2025

Палиндромы часто возникают в различных задачах биоинформатики. Обсуждаются алгоритмы вычисления редакционного расстояния между последовательностями и распознавания несовершенных палиндромов. В частности, предложен алгоритм квадратичного времени для вычисления сходства данной последовательности с каким-либо совершенным палиндромом. Время работы меньше, чем при полном переборе допустимых совершенных палиндромов. Этот алгоритм реализован на языке Python. Предложены алгоритмы усечения краёв последовательности, для выделения подстроки, близкой к совершенному палиндрому. Приведены результаты вычислений на прекурсорах микроРНК, подтверждена близость рассмотренных прекурсоров к совершенным палиндромам. Получены эмпирические оценки функции $\text{imp}(x)$, характеризующей несовершенство палиндрома, для псевдослучайных последовательностей над алфавитами из двух и четырёх букв. Также обсуждается применимость нашего метода в биоинформатике, например, для предсказания длинных шпилек на РНК.

Ключевые слова: палиндром, редакционное расстояние, комбинаторная оптимизация, биоинформатика, система компьютерной алгебры

1. ВВЕДЕНИЕ

Алгоритмы выравнивания последовательностей и методы анализа структуры, включая поиск палиндромов, широко применяются, в частности, в биоинформатике. Увеличение размеров экспериментальных данных объясняет актуальность поиска новых алгоритмов, в частности, для быстрого поиска несовершенных палиндромов, которые похожи на совершенные, но могут отличаться от них заменами, вставками и стираниями. Также интересно сравнение несовершенных палиндромов друг с другом.

Широко применяемый алгоритм квадратичного времени для вычисления редакционного расстояния (или расстояния Левенштейна) между двумя последовательностями предложили Нидлман (S.B. Needleman) и Вунш (Ch.D. Wunsch) [1]. Этот метод также применим для обобщённых метрик Левенштейна (V.I. Levenshtein), недавно рассмотренных Янковским (V.O. Yankovskiy) [2]. В 1980 году Масек

(W.J. Masek) и Патерсон (M.S. Paterson) [3] опубликовали алгоритм вычисления редакционного расстояния и поиска наибольшей общей подпоследовательности за время $O(mn/\log_2 n)$ при дополнительном условии $n \geq m \geq \log_2 n$, где через m и n обозначены длины двух последовательностей. Этот алгоритм использует разбиение исходных последовательностей на блоки равной длины по аналогии с работой [4]. В работе [5] предложен алгоритм с оценкой сложности, близкой к оценке для алгоритма Масака–Патерсона, для строк, сжатых методом Лемпеля–Зива (Lempel–Ziv). Позже Тискин (A.V. Tiskin) [6] и Грабовски (S. Grabowski) [7] предложили другие алгоритмы. Известны квантовые алгоритмы, работающие за время, существенно меньшее квадратичного [8]. Перечисление всех наибольших общих подпоследовательностей рассмотрено в работе [9]. Алгоритм приближённого вычисления редакционного расстояния за субквадратичное время

предложен в работе [10]. Поиск наибольших подстрок, то есть наибольших общих подпоследовательностей, элементы которых идут подряд, в частности, с данным числом несовпадений рассмотрен в работах [11, 12]. Также Тискин [13] рассматривал последовательности, которые служат перестановками букв (большого) алфавита. Восстановление циклического слова по набору его фрагментов обсуждается в работе Леонтьева (V.K. Leont'ev) [14]. В работе Ефимова [15] рассматривается перечисление некоторых классов хордовых диаграмм, применимых к топологической классификации структур РНК [16].

Задача поиска наибольшей общей подпоследовательности обобщается на случай любого числа последовательностей. Для m последовательностей длины n каждая известен алгоритм построения множественного выравнивания, время работы которого при фиксированном числе m ограничено сверху функцией типа $O(n^m)$. Этот алгоритм основан на методе динамического программирования. Вариант алгоритма, который позволяет экономить память, предложен в работе [17]. В общем случае возможности для экономии памяти за счёт увеличения времени работы или, наоборот, сокращения времени при использовании дополнительной памяти или большого числа процессоров рассмотрены в работе [18]. На практике используют различные эвристические алгоритмы [19, 20]. Отметим, что эвристические алгоритмы известны для многих задач дискретной оптимизации [21, 22]. Наряду с редакционным расстоянием используют и другие оценки сходства последовательностей [23]. Редакционное расстояние с учётом возможных дубликаций рассмотрено в [24].

Рассмотрим конечный алфавит с заданной инволюцией, определяющей комплементарную букву. Для приложений в биоинформатике удобно считать, что алфавит состоит из четырёх букв $\{A, C, T, G\}$, которые связаны отношением комплементарности: буква A комплементарна к T , буква C комплементарна к G . Будем опускать символ для операции конкатенации. Пустая последовательность комплементарна самой себе. Инверсией будем называть замену последовательности x на комплементарную, обозначаемую через $c(x)$. На конечных последовательностях

инверсия определяется индукцией по длине: $c(xy) = c(y)c(x)$. Комплементарная (reverse complement) последовательность получается одновременной перестановкой букв в обратном порядке и заменой букв на комплементарные буквы. Например, $c(AACT) = AGTT$. В работе [25] рассмотрен метод генерации последовательностей Де Брёйна над бинарным алфавитом, комплементарных самой себе.

Палиндромом (или совершенным палиндромом, perfect palindrome) называется последовательность x , которая комплементарна самой себе: $c(x) = x$. В случае, когда любая буква комплементарна самой себе, получается обычное определение палиндрома. Различные определения палиндрома рассмотрены в [26]. Однако в биоинформатике полезно рассматривать случай, когда никакая буква не бывает комплементарна себе. В этом случае совершенный палиндром имеет чётную длину и имеет вид $zc(z)$. Далее мы всегда будем использовать это ограничение. Последовательность, близкую в метрике Левенштейна к палиндрому, называют несовершенным (imperfect) палиндромом. Поиск несовершенных палиндромов, содержащих лишь ограниченное число замен, вставок и стираний, нарушающих структуру палиндрома, рассмотрен в работе [27].

Обычные тексты часто содержат прямые повторы, но длинные палиндромы в них встречаются редко. При работе с нуклеотидными последовательностями палиндромы, а также инвертированные участки, разделённые промежутком, встречаются чаще. Это обусловлено, в частности, особенностями эволюции ДНК. Например, большой инвертированный повтор присутствует в пластидах многих (но не всех) сосудистых растений. Среди животных такие повторы недавно найдены у волосатиков (Nematomorpha) [28]. Инвертированные повторы известны и у бактерий [29]. Несовершенные палиндромы часто бывают сайтами связывания транскрипционных факторов, что подробно обсуждается в работе [30]. Такие же свойства характерны для генов, кодирующих микроРНК [31, 32]. Мы продолжаем исследование, начатое в работе [33], где рассмотрены биологические примеры несовершенных палиндромов.

В следующем разделе дано описание алгоритма выравнивания скрытого палиндрома. Далее

приведены эффективно проверяемые свойства несовершенных палиндромов, а также некоторые эмпирические оценки.

2. ПРЕДВАРИТЕЛЬНЫЕ РЕЗУЛЬТАТЫ

Элементы последовательностей нумеруются с первого, а не с нулевого. Обозначим через $|x|$ длину последовательности x . Для двух последовательностей x и y обозначим через $\text{dist}(x, y)$ редакционное расстояние между ними, равное наименьшему числу замен, вставок и стираний букв, которые позволяют преобразовать x в y .

Для любых x и y выполнено $\text{dist}(x, y) = \text{dist}(c(x), c(y))$. Для палиндрома $y = wc(w)$ выполнено $\text{dist}(x, wc(w)) = \text{dist}(c(x), wc(w))$. Более того, для любого общего префикса $\text{dist}(wx, wy) = \text{dist}(x, y)$ и любого общего суффикса $\text{dist}(xz, yz) = \text{dist}(x, y)$.

В работе [33] была рассмотрена задача выбора оптимального палиндрома из ансамбля совершенных палиндромов, определяемых второй входной последовательностью, которая обозначена через y . Префиксы палиндромов из этого ансамбля совпадают с префиксами в y . При этом исходные последовательности могут не содержать искомым палиндром целиком. Но в y содержится половина от искомого палиндрома. Первая входная последовательность x нужна для оптимизации. Оптимальный палиндром из ансамбля должен быть на минимальном расстоянии от x . Напомним полученный результат в следующей теореме.

Теорема 1. *Существует алгоритм, который получает на вход две последовательности x и y и вычисляет оптимальное разбиение последовательности y в конкатенацию $y = wz$, при которой достигается минимума редакционное расстояние между x и палиндромом $wc(w)$. Одновременно вычисляется само редакционное расстояние между x и оптимальным палиндромом $wc(w)$. Алгебраическая сложность алгоритма составляет $O(|x||y|)$.*

Доказательство. Для вычисления редакционного расстояния между последовательностями x и y достаточно вычислить значение $f(|x|, |y|)$ функции f , заданной рекуррентными соотноше-

ниями

$$f(j, k) = \begin{cases} j, & j \geq 0, k = 0 \\ k, & j = 0, k \geq 0 \\ \min\{f(j, k-1) + 1, f(j-1, k) + 1, \\ f(j-1, k-1) + s(j, k)\}, & j > 0, k > 0 \end{cases}$$

где значения $s(j, k) \in \{0, 1\}$ зависят от элементов последовательностей x и y :

$$s(j, k) = \begin{cases} 0, & x_j = y_k \\ 1, & x_j \neq y_k \end{cases}$$

Значение $f(j, k)$ равно редакционному расстоянию между префиксами, длины которых равны j и k соответственно. Также можно вычислять редакционные расстояния для суффиксов, начиная с хвостов обеих последовательностей. Но теперь рассмотрим суффиксы для x и $c(y)$. Обозначим через $g(j, k)$ функцию, заданную рекуррентными соотношениями

$$g(j, k) = \begin{cases} j, & j \geq 0, k = 0 \\ k, & j = 0, k \geq 0 \\ \min\{g(j, k-1) + 1, g(j-1, k) + 1, \\ g(j-1, k-1) + r(j, k)\}, & j > 0, k > 0, \end{cases}$$

где значения $r(j, k) \in \{0, 1\}$ зависят от элементов последовательностей x и y :

$$r(j, k) = \begin{cases} 0, & x_{m-j+1} = c(y_k) \\ 1, & x_{m-j+1} \neq c(y_k) \end{cases}$$

Здесь через $c(y_k)$ обозначена буква, комплементарная букве y_k в исходной последовательности y . Редакционное расстояние между x и $c(y)$ равно значению $g(|x|, |y|)$.

Обозначим через h функцию $h(j, k) = f(j, k) + g(|x| - j, k)$. Значение $h(j, k)$ равно сумме двух редакционных расстояний. Первое между двумя префиксами для x и y , а второе между соответствующими суффиксами для x и $c(y)$. При этом x совпадает с конкатенацией рассмотренных префикса и суффикса для x . Искомое редакционное расстояние между x и $wc(w)$ равно минимальному значению h при значениях $0 \leq j \leq |x|$ и $0 \leq k \leq |y|$. При этом значение k , соответствующее этому минимуму, определяет разбиение $y = wz$. Это разбиение и редакционное расстояние вычисляются за время $O(|x||y|)$. Здесь мы отождествляем время работы алгоритма с алгебраической сложностью. $\square$

Алгоритм из теоремы 1 работает быстрее, чем полный перебор допустимых палиндромов $wc(w)$ с выравниванием каждого из них с x . Отметим, что значение функции $g(j, k)$, вычисляемое для последовательностей x и y , численно совпадает со значением функции $f(j, k)$ для последовательностей $c(x)$ и y . Действительно, суффикс для x получается из префикса для $c(x)$ при одновременной замене букв на комплементарные. Также суффикс для $c(y)$ получается из префикса для y при одновременной замене букв на комплементарные.

Если $x = y$, то редакционное расстояние между префиксами из x вычислимо по формуле $f(j, k) = |j - k|$. Это позволяет дополнительно уменьшить вычислительную сложность в этом случае.

3. О НЕСОВЕРШЕННЫХ ПАЛИНДРОМАХ

Определим отклонение последовательности x от совершенного палиндрома величиной

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

Корректность такого определения следует из следующей теоремы.

Теорема 2. $\text{imp}(x) = \text{imp}(c(x))$.

Доказательство. Достаточно проверить, что для любой последовательности $x = wz$ выполнены равенства

$$\text{dist}(x, wc(w)) = \text{dist}(x, c(z)z) = \text{dist}(c(x), c(z)z).$$

Действительно, $\text{dist}(x, wc(w)) = \text{dist}(z, c(w))$, поскольку совпадают префиксы, а также $\text{dist}(x, c(z)z) = \text{dist}(w, c(z))$, поскольку совпадают суффиксы. Кроме того, в силу симметричности функции dist верно, что $\text{dist}(z, c(w)) = \text{dist}(c(w), z)$, а в силу инвариантности относительно одновременной замены обоих аргументов функции dist на комплементарные верно, что $\text{dist}(c(w), z) = \text{dist}(w, c(z))$. Следовательно, выполнено первое равенство $\text{dist}(x, wc(w)) = \text{dist}(x, c(z)z)$. Второе равенство $\text{dist}(x, c(z)z) = \text{dist}(c(x), c(z)z)$ выполнено для любого z , поскольку $c(z)z$ — палиндром (то есть $c(c(z)z) = c(z)z$). $\square$

Алгоритм из теоремы 1 позволяет вычислить значение $\text{imp}(x)$ за время $O(|x|^2)$.

Теорема 3. Для любой последовательности x чётной длины выполнено неравенство

$$\text{imp}(x) \leq \frac{1}{2}.$$

Для любой последовательности x нечётной длины выполнено неравенство

$$0 < \text{imp}(x) \leq \frac{1}{2} + \frac{1}{2|x|}.$$

Доказательство. Для последовательности x чётной длины и префикса w , равного половине от x ,

$$\text{dist}(x, w) = \frac{|x|}{2}.$$

Для последовательности x нечётной длины и префикса w , который ограничен серединой последовательности x , но не содержит середину,

$$\text{dist}(x, w) = \frac{|x| + 1}{2}.$$

В любом из этих случаев верно, что $\text{dist}(x, wc(w)) \leq \text{dist}(x, w)$. Отсюда легко получить искомые неравенства. $\square$

4. ТРИММЕРЫ

Реальные биологические последовательности могут иметь большое значение функции $\text{imp}(x)$, однако, содержать в себе длинную подстроку, близкую к совершенному палиндрому. Для выделения такой подстроки, авторы предлагают новый подход — усечение (trimming), состоящий в усечении префикса или суффикса последовательности. Авторами были разработаны и реализованы шесть алгоритмов усечения на языке Python с использованием библиотеки Numba.

Алгоритмы `pref_trimmer`, `suff_trimmer` и `double_trimmer` действуют из принципа, что если один из оптимальных длин префиксов $|w|$, полученных в выводе алгоритма из теоремы 1, будет заметно отличаться от середины последовательности x , то x содержит префикс или суффикс, который не комплементарен противоположному концу x , и может быть усечён. Данные алгоритмы получают на вход

нуклеотидную последовательность x , упорядоченный по возрастанию список `optimal_lengths` оптимальных длин префиксов $|w|$, полученный в выводе алгоритма из теоремы 1, и число с плавающей точкой `cutoff_condition`, которое определяет, насколько префикс должен отличаться от середины строки.

Функция `pref_trimmer` вычисляет значение

$$rd = \max(\text{optimal_lengths}) - \left\lfloor \frac{|x|}{2} \right\rfloor$$

для оценки разницы между наибольшей длиной префикса и длиной половины последовательности. Так как индексация начинается с нуля, для получения индекса середины последовательности, длину последовательности необходимо поделить пополам и округлить результат вниз. Затем полученная разность `rd` сравнивается со значением $|x| \cdot \text{cutoff_condition}$. Если `rd` не меньше этого значения, то последовательность x усекается на первые `rd` символов.

Алгоритм `suff_trimmer` вычисляет значение

$$ld = \left\lfloor \frac{|x|}{2} \right\rfloor - \min(\text{optimal_lengths})$$

для оценки разницы между наименьшей длиной префикса и длиной половины последовательности. Затем полученное значение сравнивается с $|x| \cdot \text{cutoff_condition}$, аналогично алгоритму `pref_trimmer`. Если `ld` не меньше этого значения, то у последовательности x усекаются `ld` символов.

Алгоритм `double_trimmer`, аналогично алгоритмам `pref_trimmer` и `suff_trimmer` вычисляет значения `rd` и `ld`. Затем сравнивает их с $|x| \cdot \text{cutoff_condition}$ и после этого усекает в x первые `rd` символов, если `rd` не меньше $|x| \cdot \text{cutoff_condition}$, и последние `ld` символов x , если `ld` не меньше $|x| \cdot \text{cutoff_condition}$. Ключевое отличие состоит в том, что оба условия усеечения проверяются на первоначальной строке x , и только затем строка может быть усечена.

Теорема 4. Для совершенного палиндрома x , для любого `cutoff_condition` > 0 усеечение последовательности не будет выполнено.

Доказательство. Пусть x – совершенный палиндром. Тогда его единственное оптимальное

разбиение будет в точности в $\left\lfloor \frac{|x|}{2} \right\rfloor$. Тогда верно $ld = \left\lfloor \frac{|x|}{2} \right\rfloor - \left\lfloor \frac{|x|}{2} \right\rfloor = 0$ и $rd = \left\lfloor \frac{|x|}{2} \right\rfloor - \left\lfloor \frac{|x|}{2} \right\rfloor = 0$. Для ненулевого `cutoff_condition` rd и ld будут меньше $|x| \cdot \text{cutoff_condition}$ и условие усеечения не будет выполнено. $\square$

Теорема 5. Для любого $n \geq 3$, существуют `cutoff_condition` > 0 и x с длиной не менее n , которые удовлетворяют одновременно условиям усеечения префикса и суффикса, и для которых верно, что

$$\text{pref_trimmer}(x_{\text{suff}}) \neq \text{suff_trimmer}(x_{\text{pref}}),$$

где x_{suff} и x_{pref} являются результатами усеечения суффикса и префикса x соответственно.

Доказательство. Возьмём $x = \underbrace{\text{ATA} \dots \text{ATA}}_{\alpha}$, где

$\alpha = 2 \left\lfloor \frac{n}{2} \right\rfloor + 1$, то есть минимальное нечетное число, не меньше n . Для данного x оптимальные разбиения будут в $\left\lfloor \frac{n}{2} \right\rfloor - 1$ и $\left\lfloor \frac{n}{2} \right\rfloor + 1$. Минимальный индекс оптимального разбиения будет соответствовать совершенному палиндрому $\underbrace{\text{AT} \dots \text{AT}}_{\alpha-1}$,

а максимальный индекс – совершенному палиндрому $\underbrace{\text{AT} \dots \text{AT}}_{\alpha+1}$. Пусть `cutoff_condition` $= \frac{1}{10\alpha}$.

Так как это значение удовлетворяет условиям усеечения как суффикса, так и префикса, x_{suff} и x_{pref} могут быть вычислены и будут равны $\underbrace{\text{AT} \dots \text{AT}}_{\alpha-1}$ и $\underbrace{\text{T} \dots \text{ATA}}_{\alpha-1}$, соответственно. Заметим, что обе последовательности являются совершенными палиндромами, и в силу предыдущей теоремы, дальнейшее усеечение не будет выполнено. $\square$

Алгоритмы `pref_GRT`, `suff_GRT`, `double_GRT` пытаются выделить подстроку строки x , близкую к палиндрому, рекурсивно. Они получают на вход строку x , число с плавающей запятой `cutoff_value`, определяющее, какая часть строки будет усечена, число с плавающей запятой `imp_given`, равное значению `imp` на строке x , и целое значение `recursion_depth`, определяющее оставшееся число рекурсивных запусков функции. Если выполняется условие

$$\text{recursion_depth} = 0,$$

то усечение не будет выполнено, и каждый из трёх алгоритмов вернёт входные значения x и `imp_given`.

Алгоритм `pref_GRT` усекает первые `cutoff_value · |x|` символов строки x , если это возможно, и записывает результат в строку x' . Затем вычисляется значение `imp(x')`. Если значение `imp(x')` уменьшилось, относительно значения `imp_given`, то возвращается `pref_GRT` от параметров x' , `imp(x')`, `cutoff_value` и `recursion_depth - 1`. Если же значение не уменьшилось, то возвращается `pref_GRT` от параметров x , `imp_given`, `cutoff_value · 0.5` и `recursion_depth - 1`. Второй вариант выполняется из предположения, что первое усечение x было слишком большим, и поэтому необходимо выполнить усечение меньшего числа символов. Так как в любом из случаев каждый из вызовов уменьшает значение `recursion_depth`, алгоритм обязательно сойдётся за конечное число шагов.

Алгоритм `suff_GRT` усекает последние `cutoff_value · |x|` символов строки x , если это возможно, и записывает результат в строку x' . Затем вычисляется значение `imp(x')`, аналогично алгоритму `pref_GRT`. Если значение `imp(x')` уменьшилось, относительно значения `imp_given`, то возвращается `suff_GRT` от параметров x' , `imp(x')`, `cutoff_value` и `recursion_depth - 1`. Если же значение не уменьшилось, то возвращается `suff_GRT` от параметров x , `imp_given`, `cutoff_value · 0.5` и `recursion_depth - 1`.

Алгоритм `double_GRT` усекает первые и последние `cutoff_value · |x|` символов строки x , если это возможно, результат записывается в x' . Затем, аналогично прошлым алгоритмам, вычисляется значение `imp(x')`. Если значение `imp(x')` уменьшилось, относительно значения `imp_given`, то возвращается `double_GRT` от параметров x' , `imp(x')`, `cutoff_value` и `recursion_depth - 1`. Если же значение не уменьшилось, то возвращается `double_GRT` от параметров x , `imp_given`, `cutoff_value · 0.5` и `recursion_depth - 1`.

Теорема 6. Алгоритмы `pref_GRT`, `suff_GRT` и `double_GRT` возвращают подстроки x со значением функции `imp` не выше, чем `imp(x)`.

Доказательство. На каждом шаге рекурсии алгоритмы начинают обрабатывать новую подстроку x только значение `imp` для неё стало ниже, относительно прошлого шага. Иначе остаётся прошлая строка, со значением `imp` не ниже, чем у x . В худшем случае, если каждое усечение будет приводить к увеличению `imp`, то каждый из алгоритмов вернёт первоначальную строку без изменений. $\square$

5. ЭМПИРИЧЕСКИЕ ОЦЕНКИ

Эмпирические оценки получены для псевдослучайных последовательностей, генерируемых на языке Python командой `random.choices()`. Символы в последовательности независимо и равномерно распределены на множестве $\{A, C, G, T\}$ или $\{A, T\}$. Генерировались выборки последовательностей фиксированной длины. Число последовательностей в каждой выборке менялось от 10000 для длинных последовательностей до миллиона для коротких. Длина последовательности до 10000.

Эмпирическое среднее значение функции `imp(x)` зависит от длины последовательности и на длинах до 10000 аппроксимируется функцией

$$\mu = \alpha + \frac{\sqrt[3]{|x|}}{3|x|},$$

где коэффициент α зависит от алфавита. Для алфавита $\{A, C, G, T\}$ значение $\alpha_{\text{ACGT}} \approx 0.258$. Для алфавита $\{A, T\}$ значение $\alpha_{\text{AT}} \approx 0.144$.

Среднеквадратичное отклонение `imp(x)` оценивалось для последовательностей длины до 100 на выборках по миллиону последовательностей в каждой. Для алфавита $\{A, C, G, T\}$ среднеквадратичное отклонение аппроксимируется быстро убывающей функцией от длины последовательности:

$$\sigma_{\text{ACGT}} = \frac{(|x| - 0.6)^{0.35}}{2.58|x|}$$

с коэффициентом детерминации $R^2 = 0.999$. А для алфавита $\{A, T\}$ — другой тоже быстро убывающей функцией от длины последовательности:

$$\sigma_{\text{AT}} = \frac{(|x| - 0.1)^{0.36}}{2.65|x|}$$

с коэффициентом детерминации $R^2 = 0.99$.

Табл. 1. Медиана, среднее и стандартное отклонение числа оптимальных палиндромов для всех последовательностей длины от 1 до 33.

Длина	Среднее числа п-в	Ст. откл. числа п-в	Медиана числа п-в
1	2.00000000	0.00000000	2
2	1.00000000	0.00000000	1
3	2.00000000	0.00000000	2
4	1.25000000	0.66143783	1
5	1.87500000	0.33071891	2
6	1.46875000	0.78995154	1
7	1.81250000	0.49607837	2
8	1.57031250	0.80703541	1
9	1.79687500	0.63565143	2
10	1.62304688	0.80797786	1
11	1.80859375	0.73850856	2
12	1.67285156	0.82407931	1
13	1.81494141	0.80095853	2
14	1.72790527	0.86238824	1
15	1.81933594	0.84241918	2
16	1.77722168	0.90138305	2
17	1.82827759	0.88085232	2
18	1.81813049	0.93374327	2
19	1.84394836	0.91963413	2
20	1.84964943	0.95954043	2
21	1.86333847	0.95640744	2
22	1.87458563	0.98321547	2
23	1.88362551	0.98867566	2
24	1.89578867	1.00637429	2
25	1.90284598	1.01636142	2
26	1.91457456	1.02894317	2
27	1.92048600	1.04040998	2
28	1.93144143	1.05046195	2
29	1.93662634	1.06198630	2
30	1.94664925	1.07064553	2
31	1.95142870	1.08161302	2
32	1.96039251	1.08940265	2
33	1.96500644	1.09965078	2

Для всех последовательностей длины до 33 включительно над алфавитом из букв {A, T} приведены значения среднего, медианы и стандартного отклонения числа оптимальных палиндромов в таблице 5 и функции dist между последовательностью и её оптимальным палиндромом в таблице 5.

Табл. 2. Медиана, среднее и стандартное отклонение значения функции dist между последовательностью и её оптимальным палиндромом для всех последовательностей длины от 1 до 33.

Длина	Среднее dist	Ст. откл. dist	Медиана dist
1	1.00000000	0.00000000	1
2	0.50000000	0.50000000	0.5
3	1.25000000	0.43301270	1
4	1.00000000	0.70710678	1
5	1.50000000	0.61237244	1
6	1.43750000	0.78809501	1.5
7	1.78125000	0.73884940	2
8	1.81250000	0.82679728	2
9	2.08203125	0.83215961	2
10	2.16210938	0.87225565	2
11	2.39257812	0.90148727	2
12	2.50195312	0.92385703	2
13	2.70483398	0.95480695	3
14	2.83605957	0.97558251	3
15	3.01818848	1.00215161	3
16	3.16455078	1.02383613	3
17	3.33285522	1.04672265	3
18	3.48840332	1.06835178	3
19	3.64854050	1.08901735	4
20	3.80833817	1.10956296	4
21	3.96433735	1.12891275	4
22	4.12533617	1.14832757	4
23	4.27956986	1.16650146	4
24	4.44027984	1.18505337	4
25	4.59381080	1.20199494	5
26	4.75365368	1.21987325	5
27	4.90694101	1.23572240	5
28	5.06574126	1.25292791	5
29	5.21896984	1.26791587	5
30	5.37672382	1.28439938	5
31	5.52996125	1.29872442	5
32	5.68673484	1.31445966	6
33	5.83998419	1.32825738	6

6. ЭКСПЕРИМЕНТАЛЬНЫЕ РЕЗУЛЬТАТЫ

Мы рассмотрели около 38.5 тысяч прекурсов микроРНК, доступных в базе данных miRBase [34]. Из рассмотрения были исключены вырожденные последовательности (120 последовательностей). Среднее значение величины

$\text{imp}(x)$ для рассмотренных невырожденных последовательностей равно 0.19175. Медиана равна 0.19186, среднеквадратичное отклонение равно 0.04148.

Это подтверждает высокое сходство большинства этих РНК с палиндромами, что согласуется с образованием этими РНК шпилек, которые важны для формирования микроРНК [31, 32].

При этом длина оптимального палиндрома может заметно отличаться от длины исходной последовательности. Иногда бывает несколько оптимальных решений. Например, у человека для последовательности hsa-mir-200b длины 95 нашлось пять оптимальных палиндромов с длинами 92, 94, 96, 98 и 100.

7. ОБСУЖДЕНИЕ

При отборе последовательностей, близких к совершенному палиндрому, естественно выбирать порог в зависимости от длины последовательности, а также от состава, поскольку для фиксированной длины среди последовательностей x , бедных некоторыми буквами, среднее значение $\text{imp}(x)$ отличается от среднего по всей совокупности последовательностей.

Алгоритм из теоремы 1 легко изменить для вычисления расстояний в обобщённой метрике Левенштейна, рассмотренной Янковским [2]. В этом случае вставки и замены дают разный вклад в расстояние между последовательностями. Теоремы 1 и 2 остаются справедливыми. Но в этом случае используются параметры, значения которых надо выбирать в зависимости от контекста. С другой стороны, результаты вычислений на прекурсорах микроРНК позволяют надеяться на применимость метода в других задачах, когда нужно отобрать последовательности, близкие к палиндромам.

Отметим, что вычисление $\text{imp}(x)$ не сводится к вычислению редакционного расстояния $\text{dist}(x, c(x))$, хотя эта величина тоже может служить для отклонения от совершенного палиндрома. Например, для последовательности x , состоящей из $2k + 1$ одинаковых букв, значения $|x|\text{imp}(x) = k + 1$ и $\text{dist}(x, c(x)) = 2k + 1$ взаимно простые. А для x , состоящей из $2k$ одинаковых букв, значения $|x|\text{imp}(x) = k$ и $\text{dist}(x, c(x)) = 2k$ отличаются в два раза.

Кроме того, алгоритм из теоремы 1 находит положение оптимального префикса в составе исходной последовательности. Например, в случае, когда последовательность $x = wsc(w)z$ равна конкатенации палиндрома и короткого хвоста, префикс w будет оптимальным для нашего алгоритма. При этом оптимальный палиндром короче исходной последовательности. И это также верно при небольших отклонениях от совершенного палиндрома в начале последовательности. С другой стороны, для последовательности $x = ywc(w)$ с достаточно коротким префиксом y оптимальный палиндром будет длиннее исходной последовательности. По теореме 2 величина $\text{imp}(x)$ не зависит от того, расположен ли палиндром в начале или в конце исходной последовательности. Но от этого зависит длина оптимального палиндрома.

8. БЛАГОДАРНОСТИ

Авторы благодарят Николая Николаевича Васильева за обсуждение.

9. ИСТОЧНИК ФИНАНСИРОВАНИЯ

Работа выполнена в рамках государственного задания ИППИ РАН, утвержденного Минобрнауки России.

СПИСОК ЛИТЕРАТУРЫ

1. Needleman S.B., Wunsch Ch.D. A general method applicable to the search for similarities in the amino acid sequence of two proteins // Journal of Molecular Biology. 1970. v. 48. № 3. p. 443–453. DOI: 10.1016/0022-2836(70)90057-4
2. Янковский В.О. Группы изометрий формальных языков относительно обобщенных метрик Левенштейна // Матем. заметки. 2024. т. 116. № 2. с. 306–315. DOI: 10.4213/mzm13646
3. Masek W.J., Paterson M.S. A faster algorithm computing string edit distances // Journal of Computer and System Sciences. 1980. v. 20. № 1. p. 18–31. DOI: 10.1016/0022-0000(80)90002-1
4. Арлазаров В.Л., Диниц Е.А., Кронрод М.А., Фараджев И.А. Об экономном построении транзитивного замыкания ориентированного графа // Доклады АН СССР. 1970. т. 194. № 3. с. 487–488.

5. *Crochemore M., Landau G.M., Ziv-Ukelson M.* A subquadratic sequence alignment algorithm for unrestricted score matrices // *SIAM Journal on Computing*. 2003. v. 32. № 6. p. 1654–1673. DOI: 10.1137/S0097539702402007
6. *Tiskin A.* Semi-local longest common subsequences in subquadratic time // *Journal of Discrete Algorithms*. 2008. v. 6. № 4. p. 570–581. DOI: 10.1016/j.jda.2008.07.001
7. *Grabowski S.* New tabulation and sparse dynamic programming based techniques for sequence similarity problems // *Discrete Applied Mathematics*. 2016. v. 212. p. 96–103. DOI: 10.1016/j.dam.2015.10.040
8. *Akmal S., Jin C.* Near-optimal quantum algorithms for string problems // *Algorithmica*. 2023. v. 85. p. 2260–2317. DOI: 10.1007/s00453-022-01092-x
9. *Conte A., Grossi R., Punzi G., Uno T.* Enumeration of maximal common subsequences between two strings // *Algorithmica*. 2022. v. 84. p. 757–783. DOI: 10.1007/s00453-021-00898-5
10. *Chakraborty D., Das D., Goldenberg E., Koucký M., Saks M.* Approximating edit distance within constant factor in truly sub-quadratic time // *Journal of the ACM*. 2020. v. 67. № 6. Article 36. DOI: 10.1145/3422823
11. *Kociumaka T., Radoszewski J., Starikovskaya T.* Longest common substring with approximately k mismatches // *Algorithmica*. 2019. v. 81. № 6. p. 2633–2652. DOI: 10.1007/s00453-019-00548-x
12. *Amir A., Charalampopoulos P., Pissis S.P., Radoszewski J.* Dynamic and internal longest common substring // *Algorithmica*. 2020. v. 82. p. 3707–3743. DOI: 10.1007/s00453-020-00744-0
13. *Tiskin A.* Fast distance multiplication of unit-Monge matrices // *Algorithmica*. 2015. v. 71. p. 859–888. DOI: 10.1007/s00453-013-9830-z
14. *Леонтьев В.К.* Восстановление циклических слов по фрагментам // *Проблемы передачи информации*. 2012. т. 48. № 2. с. 121–126.
15. *Ефимов Д.Б.* Перечисление чётных и нечётных хордовых диаграмм // *Дискретный анализ и исследование операций*. 2024. т. 31. № 2, с. 63–79 DOI: <https://doi.org/10.33048/daio.2024.31.767>
16. *Bon M., Vernizzi G., Orland H., Zee A.* Topological classification of RNA structures // *J Mol Biol*. 2008. v. 379. № 4. p. 900–911. DOI: 10.1016/j.jmb.2008.04.033
17. *Hosseininiasab A., van Hoeve W.-J.* Exact multiple sequence alignment by synchronized decision diagrams // *INFORMS Journal on Computing*. 2020. v. 33. № 2. p. 721–738. DOI: 10.1287/ijoc.2019.0937
18. *Бельтюков А.П., Маслов С.Г., Джудакizada М.* Взаимное моделирование последовательных и параллельных словарных вычислений // *Вестн. Удмуртск. ун-та. Матем. Мех. Компьют. науки*. 2024. т. 34. № 2. с. 299–308.
19. *Liu Y., Yuan H., Zhang Q., Wang Z., Xiong S., Wen N., Zhang Y.* Multiple sequence alignment based on deep reinforcement learning with self-attention and positional encoding // *Bioinformatics*. 2023. v. 39. № 11. Article btad636. p. 1–10. DOI: 10.1093/bioinformatics/btad636
20. *Ibrahim M.K., Yusof U.K., Eisa T.A.E., Nasser M.* Bioinspired algorithms for multiple sequence alignment: A systematic review and roadmap // *Appl. Sci*. 2024. v. 14. № 6. Article 2433. DOI: 10.3390/app14062433
21. *Евтушенко Ю.Г., Третьяков А.А.* Метод локализации фиктивных экстремумов в задаче глобальной оптимизации // *Доклады Российской академии наук. Математика, информатика, процессы управления*. 2023. т. 512. № 1. с. 78–80. DOI: 10.31857/S2686954323600222
22. *Зверков О.А., Селиверстов А.В.* Эффективные нижние границы для ранга матрицы и приложения // *Программирование*. 2023. т. 49. № 5. с. 79–86. DOI: 10.31857/S0132347423020176
23. *Бурдонов И., Максимов А.* Двадцать функций подобия двух конечных последовательностей // *Программирование*. 2023. № 5. с. 3–18. DOI: 10.31857/S0132347423050035
24. *Lafond M., Lai W., Liyanage A., Zhu B.* The longest subsequence-duplicated subsequence and related problems // *Information and Computation*. 2025. v. 306. Article 105313 p. 1–15. DOI: 10.1016/j.ic.2025.105313
25. *Chang, Z., Wang, Q.* Efficient generation of odd order de Bruijn sequence with the same complement and reverse sequences // *Des. Codes Cryptogr*. 2025. v. 93, p. 1935–1949. DOI: 10.1007/s10623-025-01580-5
26. *Mieno T., Funakoshi M., Nakashima Y., Inenaga S., Bannai H., Takeda M.* Computing maximal palindromes in non-standard matching models // *Information and Computation*. 2025. v. 304. Article 105283. p. 1–20. DOI: 10.1016/j.ic.2025.105283

27. Alzamel M., Hampson C., Iliopoulos C.S., Lim Z., Pissis S., Vlachakis D., Watts S. Maximal degenerate palindromes with gaps and mismatches // Theoretical Computer Science. 2023. v. 978. Article № 114182. p. 1–16. DOI: 10.1016/j.tcs.2023.114182
 28. Nikolaeva O.V., Beregova A.M., Efeykin B.D., Mirolubova T.S., Zhuravlev A.Yu., Ivantsov A.Yu., Mikhailov K.V., Spiridonov S.E., Aleoshin V.V. Expression of hairpin-enriched mitochondrial DNA in two hairworm species (Nematomorpha) // International Journal of Molecular Sciences. 2023. v. 24. № 14. Article 11411. p. 1–17. DOI: 10.3390/ijms241411411
 29. Мирошниченко Л.А., Арефьева Н.А., Джиоев Ю.П., Гусев В.Д., Борисенко А.Ю., Эрдынеев С.В., Букин Ю.С. Структура повторов в геномах сальмонелл // Матем. биология и биоинформ. 2023. т. 18. № 2. с. 602–620. DOI: 10.17537/2023.18.602
 30. Datta R.R., Rister J. The power of the (imperfect) palindrome: sequence-specific roles of palindromic motifs in gene regulation // Bioessays. 2022 V. 44. № 4. Article № e2100191. p. 1–11. DOI: 10.1002/bies.202100191
 31. Гринкевич Л.Н. Роль микроРНК в обучении и долговременной памяти // Вавиловский журнал генетики и селекции. 2020. т. 24. № 8. с. 885–896. DOI: 10.18699/VJ20.687
 32. Назипова Н.Н. Разнообразие некодирующих РНК в геномах эукариот // Математическая биология и биоинформатика. 2021. т. 16. № 2. с. 256–298. DOI: 10.17537/2021.16.256
 33. Зверков О., Селиверстов А., Шиловский Г. Выравнивание скрытого палиндрома // Математическая биология и биоинформатика. 2024. т. 19. № 2. с. 427–438. DOI: 10.17537/2024.19.427
 34. Kozomara A., Birgaoanu M., Griffiths-Jones S. miRBase: from microRNA sequences to function // Nucleic Acids Res. 2019. v. 47. p. D155–D162. DOI: 10.1093/nar/gky1141
- Trimming of imperfect palindromes.**
G. A. Khaziev*, O. A. Zverkov*, A. V. Seliverstov*
** Institute for Information Transmission Problems (Kharkevich Institute), Russian Academy of Sciences*
Bolshoy Karetny per. 19, build. 1, Moscow 127051 Russia
E-mail: khaziev@iitp.ru, zverkov@iitp.ru, slvstv@iitp.ru
- Abstract:** Palindromes are often occurred in a variety of problems in the field of bioinformatics. We discuss algorithms for computation of edit distance and recognition of imperfect palindromes. Particularly, we propose a quadratic-time algorithm for computing similarity between given sequence and some perfect palindrome. The running time of such algorithm is lower, than bruteforcing all reasonable perfect palindromes. This algorithm is implemented in Python. We propose algorithms for trimming termini of sequence, with purpose of computing substring that is close to perfect palindrome. We show the results of computation on microRNA precursors and confirm their similarity to perfect palindromes. We show the empirical estimation on random sequences for two and four letter alphabet of $\text{imp}(x)$ function, that characterizes the imperfectness of the palindrome. We also discuss the applicability of our method in bioinformatics, for example, for predicting long hairpins on RNA.
- Keywords:** palindrome, edit distance, combinatorial optimization, bioinformatics, computer algebra system
- Needleman S.B., Wunsch Ch.D.* A general method applicable to the search for similarities in the amino acid sequence of two proteins // Journal of Molecular Biology. 1970. v. 48. № 3. p. 443–453. DOI: 10.1016/0022-2836(70)90057-4
- Yankovskiy V.O.* Isometry groups of formal languages for generalized Levenshtein distances // Math. Notes. 2024. V. 116. № 2. P. 373–381. DOI: 10.1134/S0001434624070307
- Masek W.J., Paterson M.S.* A faster algorithm computing string edit distances // Journal of Computer and System Sciences. 1980. v. 20. № 1. p. 18–31. DOI: 10.1016/0022-0000(80)90002-1
- Arlazarov V.L., Dinic E.A., Kronrod M.A., Faradzev I.A.* On economic construction of the transitive closure

- of a directed graph // Soviet Math. Dokl. 1970. v. 11. № 5. P. 1209–1210.
- Crochemore M., Landau G.M., Ziv-Ukelson M.* A subquadratic sequence alignment algorithm for unrestricted score matrices // SIAM Journal on Computing. 2003. v. 32. № 6. p. 1654–1673. DOI: 10.1137/S0097539702402007
- Tiskin A.* Semi-local longest common subsequences in subquadratic time // Journal of Discrete Algorithms. 2008. v. 6. № 4. p. 570–581. DOI: 10.1016/j.jda.2008.07.001
- Grabowski S.* New tabulation and sparse dynamic programming based techniques for sequence similarity problems // Discrete Applied Mathematics. 2016. v. 212. p. 96–103. DOI: 10.1016/j.dam.2015.10.040
- Akmal S., Jin C.* Near-optimal quantum algorithms for string problems // Algorithmica. 2023. v. 85. p. 2260–2317. DOI: 10.1007/s00453-022-01092-x
- Conte A., Grossi R., Punzi G., Uno T.* Enumeration of maximal common subsequences between two strings // Algorithmica. 2022. v. 84. p. 757–783. DOI: 10.1007/s00453-021-00898-5
- Chakraborty D., Das D., Goldenberg E., Koucký M., Saks M.* Approximating edit distance within constant factor in truly sub-quadratic time // Journal of the ACM. 2020. v. 67. № 6. Article 36. DOI: 10.1145/3422823
- Kociumaka T., Radoszewski J., Starikovskaya T.* Longest common substring with approximately k mismatches // Algorithmica. 2019. v. 81. № 6. p. 2633–2652. DOI: 10.1007/s00453-019-00548-x
- Amir A., Charalampopoulos P., Pissis S.P., Radoszewski J.* Dynamic and internal longest common substring // Algorithmica. 2020. v. 82. p. 3707–3743. DOI: 10.1007/s00453-020-00744-0
- Tiskin A.* Fast distance multiplication of unit-Monge matrices // Algorithmica. 2015. v. 71. p. 859–888. DOI: 10.1007/s00453-013-9830-z
- Leont'ev V.K.* Reconstruction of cyclic words from their fragments // Problems of Information Transmission. 2012. V. 48. № 2. P. 193–197. DOI: 10.1134/S0032946012020093
- Efimov D.B.* Enumeration of Even and Odd Chord Diagrams // Journal of Applied and Industrial Mathematics. 2024. V. 18, № 2, P. 216–226. DOI: <https://doi.org/10.1134/S1990478924020042>
- Bon M., Vernizzi G., Orland H., Zee A.* Topological classification of RNA structures // J Mol Biol. 2008. V. 379. № 4. P. 900–911. DOI: 10.1016/j.jmb.2008.04.033
- Hosseiniinasab A., van Hoeve W.-J.* Exact multiple sequence alignment by synchronized decision diagrams // INFORMS Journal on Computing. 2020. v. 33. № 2. p. 721–738. DOI: 10.1287/ijoc.2019.0937
- Bel'tyukov A.P., Maslov S.G., Joudakizadeh M.* Mutual modeling of sequential and parallel word computations. Vestn. Udmurtsk. Univ. Mat. Mekh. Komp. Nauki. 2024. v. 34. № 2. p. 299–308. DOI: 10.35634/vm240208
- Liu Y., Yuan H., Zhang Q., Wang Z., Xiong S., Wen N., Zhang Y.* Multiple sequence alignment based on deep reinforcement learning with self-attention and positional encoding // Bioinformatics. 2023. v. 39. № 11. Article btad636. p. 1–10. DOI: 10.1093/bioinformatics/btad636
- Ibrahim M.K., Yusof U.K., Eisa T.A.E., Nasser M.* Bioinspired algorithms for multiple sequence alignment: A systematic review and roadmap // Appl. Sci. 2024. v. 14. № 6. Article 2433. DOI: 10.3390/app14062433
- Evtushenko Y.G., Tret'yakov A.A.* The method of fictitious extrema localization in the problem of global optimization // Doklady Mathematics. 2023. v. 108. № 1. p. 309–311.
- Zverkov O.A., Seliverstov A.V.* Effective lower bounds on the matrix rank and their applications // Programming and Computer Software. 2023. v. 49. № 5. p. 441–447. DOI: 10.1134/S0361768823020160
- Burdonov I., Maksimov A.* Twenty similarity functions for two finite sequences // Programming and Computer Software. 2023. V. 49. P. 373–387. DOI: 10.1134/S0361768823050031
- Lafond M., Lai W., Liyanage A., Zhu B.* The longest subsequence-duplicated subsequence and related problems // Information and Computation. 2025. v. 306. Article 105313 p. 1–15. DOI: 10.1016/j.ic.2025.105313
- Chang, Z., Wang, Q.* Efficient generation of odd order de Bruijn sequence with the same complement and reverse sequences // Des. Codes Cryptogr. 2025. v. 93, p. 1935–1949. DOI: 10.1007/s10623-025-01580-5
- Mieno T., Funakoshi M., Nakashima Y., Inenaga S., Bannai H., Takeda M.* Computing maximal palindromes in non-standard matching models // Information and Computation. 2025. v. 304. Article 105283. p. 1–20. DOI: 10.1016/j.ic.2025.105283
- Alzamel M., Hampson C., Iliopoulos C.S., Lim Z., Pissis S., Vlachakis D., Watts S.* Maximal degenerate palindromes with gaps and mismatches // Theoretical Computer Science. 2023. v. 978. Article № 114182. p. 1–16. DOI: 10.1016/j.tcs.2023.114182
- Nikolaeva O.V., Beregova A.M., Efeykin B.D., Mirolubova T.S., Zhuravlev A.Yu., Ivantsov A.Yu., Mikhailov K.V., Spiridonov S.E., Aleoshin V.V.* Expression of hairpin-enriched mitochondrial DNA in two hairworm species (Nematomorpha) // International Journal of Molecular Sciences. 2023. v. 24. № 14. Article 11411. p. 1–17. DOI: 10.3390/ijms241411411

Miroshnichenko L.A., Arefieva N.A., Dzhioev Yu.P., Gusev V.D., Borisenko A.Yu., Erdynееv S.V., Bukin Yu.S. Repeat structure in Salmonella genomes // Mathematical Biology and Bioinformatics. 2023. v. 18. № 2. p. 602–620. DOI: 10.17537/2023.18.602

Datta R.R., Rister J. The power of the (imperfect) palindrome: sequence-specific roles of palindromic motifs in gene regulation // Bioessays. 2022 V. 44. № 4. Article № e2100191. p. 1-11. DOI: 10.1002/bies.202100191

Grinkevich L.N. The role of microRNAs in learning and long-term memory // Vavilov Journal of Genetics and Breeding. 2020. V. 24. № 8. P. 885–896. DOI: 10.18699/VJ20.687

Nazipova N. Variety of non-coding RNAs in Eukaryotic genomes // Mathematical Biology and Bioinformatics. 2021. v. 16. № 2. p. 256–298. DOI: 10.17537/2021.16.256

Зверков О., Селиверстов А., Шоловский Г. Alignment of a Hidden Palindrome // Mathematical Biology and Bioinformatics. 2024. v. 19. № 2. p. 427–438. DOI: 10.17537/2024.19.427

Kozomara A., Birgaoanu M., Griffiths-Jones S. miRBase: from microRNA sequences to function // Nucleic Acids Res. 2019. v. 47. p. D155–D162. DOI: 10.1093/nar/gky1141

10. FUNDING

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