ЧЕБЫШЕВСКИЙ СБОРНИК
Том 00. Выпуск 00.

УДК 519.16 DOI 10.22405/2226-8383–00-00-1-14

Выравнивание последовательностей со структурой специального типа[1]

Хазиев Георгий Андреевич — магистр, Институт проблем передачи информации им. А.А. Харкевича Российской академии наук (Москва).
e-mail: khaziev@yandex.ru
Селиверстов Александр Владиславович — к.ф.-м.н., Институт проблем передачи информации им. А.А. Харкевича Российской академии наук (Москва).
e-mail: slvstv@iitp.ru
Любецкий Василий Александрович — д.ф.-м.н., Институт проблем передачи информации им. А.А. Харкевича Российской академии наук (Москва).
e-mail: lyubetsk@iitp.ru

Аннотация

Предложен новый алгоритм выравнивания аминокислотных последовательностей, который учитывает структуру белка, участвующую в образовании димеров или систем из большего числа субъединиц. Этот алгоритм имеет прикладное значение в биоинформатике для предсказания эволюции структур белков. Однако тот же метод применим к анализу любых последовательностей, в состав которых входят повторяющиеся блоки фиксированной длины наряду с бесструктурными участками. Большинство известных подходов к таким задачам основано либо на предсказании структуры одной последовательности, либо на выравнивании последовательностей без использования структуры. Наш подход объединяет эти методы. В частности, наш алгоритм основан на той же идее, что и алгоритм Масека–Патерсона и алгоритмы для выравнивания сжатых последовательностей. Работая с белками, мы рассматриваем блоки длины семь. Но аналогично можно рассмотреть блоки другой длины. Важно, что потенциальный структурный элемент выделяется составом небольшой окрестности. Хотя окончательный выбор делается с учётом глобального выравнивания последовательностей. Также обсуждается возможность ускорения алгоритма при дополнительном условии. Это дополнительное условие состоит в отделимости друг от друга структурированных участков. Наш алгоритм позволяет уточнить предсказания структуры белка. Поскольку время работы малое, можно проводить вычисления с большими наборами белков, в частности, при изучении эволюции эукариотических белков. Выравнивание белков позволяет быстро выделять наиболее важные структуры, которые остаются консервативными в ходе эволюции. В частности, это полезно для исследования белков, структура которых меняется в результате изменения внешних условий или взаимодействия с другими белками.

Ключевые слова: последовательность, белок, гептадный повтор, домен, структура, редакционное расстояние, оптимизация, вычислительная сложность, биоинформатика.

Библиография: 33 название.

Для цитирования:
Г. А. Хазиев, А. В. Селиверстов, В. А. Любецкий. Выравнивание последовательностей со структурой специального типа // Чебышевский сборник, , т. 00, вып. 00, с. 1–14.

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

CHEBYSHEVSKII SBORNIK

Vol. 00. No. 00.

UDC 519.16 DOI 10.22405/2226-8383–00-00-1-14

Sequence alignment using a special-type structure

Khaziev, Georgii A. — Master, Institute for Information Transmission Problems of the Russian Academy of Sciences (Kharkevich Institute) (Moscow).

e-mail: khaziev@iitp.ru

Seliverstov, Alexandr V. — Candidate of Physical and Mathematical Sciences, Institute for Information Transmission Problems of the Russian Academy of Sciences (Kharkevich Institute) (Moscow).

e-mail: slvstv@iitp.ru

Lyubetsky, Vassily A. — Doctor of Physical and Mathematical Sciences, Institute for Information Transmission Problems of the Russian Academy of Sciences (Kharkevich Institute) (Moscow).

e-mail: lyubetsk@iitp.ru

Abstract

We propose a new algorithm for amino acid sequence alignment that uses the protein structure involved in the formation of dimers or systems of a larger number of subunits. The algorithm has practical application in bioinformatics for predicting the evolution of protein structures. However, the same method is applicable to the analysis of any sequences that contain repeating blocks of fixed length along with unstructured regions. Most known approaches to solve such problems are based either on the prediction of a single-sequence structure or on the alignment of sequences without using the structure. Our approach combines these methods. In particular, our algorithm is based on the same idea as the Masek–Paterson algorithm and algorithms for aligning compressed sequences. When working with proteins, we consider blocks of length seven. But blocks of other lengths can be considered similarly. It is important that the potential structural element is distinguished by the composition of a small neighborhood. Although the final choice is made taking into account the global alignment of sequences. The possibility of accelerating the algorithm under an additional condition is also discussed. This additional condition consists in the separability of structured sections from each other. Our algorithm allows us to refine protein structure predictions. Because its running time is small, one can perform computations over large sets of proteins, in particular, for studying the evolution of eukaryotic proteins. The protein alignment allows us to quickly identify the most important structures that remain conserved during evolution. In particular, it is useful for studying proteins with structure changes as a result of changing external conditions or interactions with other proteins.

Keywords: sequence, protein, heptad repeat, domain, structure, edit distance, optimization, computational complexity, bioinformatics.

Bibliography: 33 titles.

For citation:

Khaziev, G. A., Seliverstov, A. V., Lyubetsky, V. A. , “Sequence alignment using a special-type structure” , *Chebyshevskii sbornik*, vol. 00, no. 00, pp. 1–14.

1. Введение

Вычисление (обобщённого) редакционного расстояния, поиск наибольшей общей подпоследовательности и выравнивание последовательностей широко используются в биоинформатике. Обобщённое редакционное расстояние, при котором замены и вставки дают разные вклады в расстояние, легко вычислить за квадратичное время [1]. Известно несколько алгоритмов для точного вычисления редакционного расстояния за меньшее время [2, 3, 4, 5]. Пусть даны две равномерно распределенные случайные двоичные строки равной длины. Математическое ожидание длины наибольшей общей подпоследовательности двух случайных двоичных последовательностей рассмотрено в работе [6]. Аппроксимировать редакционное расстояние можно за меньшее время [7]. Для фиксированной верхней границы на редакционное расстояние известны алгоритмы, время работы которых линейно зависит от длины последовательностей [8]. Программа `MMseqs2` реализует быстрый эвристический алгоритм [9]. Также используют и другие оценки сходства последовательностей [10, 11]. При исследовании последовательностей ДНК в эволюционной биологии часто рассматриваются тандемные дубликации [12, 13].

Для предсказания структуры белка уже создано несколько алгоритмов [14, 15]. И эти алгоритмы часто применяются в биоинформатике [16]. Однако предсказание структуры по одной последовательности может быть трудным для белков с различными альтернативными структурами, формирование которых зависит от небольших изменений внешних условий [17]. Другой причиной изменения структуры может быть модификация отдельных аминокислот.

Менее изучены задачи выравнивания последовательностей с учётом некоторой дополнительной структуры. Примером такой структуры, важной для изучения нуклеотидных последовательностей, служат несовершенные палиндромы, рассмотренные в недавней статье [18]. Примерами таких несовершенных палиндромов могут быть сайты кооперативного связывания гомодимеров транскрипционных факторов с ДНК [19]. Другие примеры связаны с некодирующими РНК, обычно имеющими сложную вторичную структуру [20]. Также рассматривались алгоритмы выравнивания периодических последовательностей [21, 22].Существует связь между паросочетаниями и структурой белка [23].

Мы рассмотрим аминокислотные последовательности со структурой, участвующей в образовании димеров или более сложных комплексов белков, состоящих из нескольких субъединиц. Такие структуры белков были впервые открыты в 1952 году и названы coiled-coil [24]. Известно много типов таких структур, образованных α-спиралями с повторяющимися гидрофобными аминокислотами [25]. Поскольку на один виток α-спирали приходится в среднем 3.6 аминокислот, два витка составляют близкое к целому число аминокислот. Поэтому для взаимодействия двух α-спиралей важно повторение гидрофобных аминокислот с периодом семь. Обычно рассматривают вырожденные повторы участка $abcdefg$, где на позициях a и d расположены гидрофобные аминокислоты, а на позициях b, c и f полярные аминокислоты. Частный случай, когда позиции с периодом семь занимает лейцин, называется лейциновой молнией. Метод для предсказания в одной последовательности такого участка, который входит в состав структуры coiled-coil, был реализован в программе `COILS`, которую предложил Андрей Лупас (Andrei Lupas) [26]. Предсказание coiled-coil с использованием свёрточных нейронных сетей было рассмотрено Фенгом и соавторами [27]. Однако такие предсказания не учитывают выравнивания разных последовательностей.

Мы рассматриваем задачу выравнивания двух аминокислотных последовательностей, при котором вычисляется обобщённое редакционное расстояние и общие потенциальные участки, участвующие в образовании структуры coiled-coil.

2. Предварительные сведения

Аминокислотной последовательностью называется последовательность в двадцатибуквенном алфавите $\{A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y\}$. Из них к гидрофобным неполярным аминокислотам относятся [28]: аланин A, валин V, изолейцин I, лейцин L, фенилаланин F, триптофан W и метионин M. Полярные положительно заряженные при $pH = 7$ аминокислоты: лизин K, аргинин R и гистидин H. Отрицательно заряженные при $pH = 7$ аминокислоты: аспартат D и глутамат E.

Метод Нидлмана–Вунша [1] состоит в последовательном заполнении таблицы (обобщённых) редакционных расстояний между всеми префиксами двух исходных последовательностей. В итоге в нижнем правом углу этой таблицы записано расстояние между последовательностями. Так для двух последовательностей, длины которых равны m и n соответственно, редакционное расстояние вычисляется за время $O(mn)$.

Масек и Патерсон [2] заметили, опираясь на работу [29], что этот метод требует меньшего времени и памяти, если исходные последовательности разбить на блоки, которые могут повторяться. Таблица расстояний заполняется лишь для префиксов, состоящих из целого числа блоков. Если длина блока равна числу p, то получится таблица, содержащая mn/p^2 записей. В последующих работах [3, 4] было показано, что тот же метод применим для вычисления редакционного расстояния между последовательностями, сжатыми методом Лемпеля–Зива.

М. А. Ройтберг и соавторы [30] предлагали заранее разметить на последовательности структурные элементы (для белков это могут быть, например, α-спирали и β-листы), и добавлять при вычислении расстояния штрафы, увеличивающие расстояние, при выравнивании позиций, входящих в разные структурные элементы. К сожалению, эта стратегия может давать неудачные результаты из-за неточностей определения структуры и возможности альтернативных структур, в составе которых одна позиция может входить в структурные элементы разных типов. Однако можно найти оптимальные по Парето выравнивания, учитывающие структуру.

3. Результаты

Штрафы за вставки и замены аминокислот могут быть выбраны, например, согласно BLOSUM62 и BLOSUM80, которые часто используют в биоинформатике. Гептамером называется слово в данной последовательности длины семь, аминокислоты которого расположены согласно структуре coiled-coil.

Замечание 1. *Обычно вычисляемая в биоинформатике величина, используемая для оценки сходства последовательностей, отличается от редакционного расстояния, поскольку может приниматъ как отрицательные, так и положительные значения. Мы рассматриваем расстояние, удовлетворяющее обычным условиям.*

Теорема 1. *Пусть фиксированы штрафы за вставки и замены аминокислот, а также штрафы за несоответствие гептамеров, участвующих в образовании структуры coiled-coil. Существует алгоритм, который, получив на вход две аминокислотные последовательности с длинами m и n соответственно, за время $O(mn)$ вычисляет обобщенное редакционное расстояние с учётом структуры coiled-coil и оптимальное число выравненных друг под другом гептамеров.*

Доказательство. Разметим в каждой из последовательностей (независимо от их выравнивания между собой) гептамеры — участки по семь аминокислот, соответствующие образцу для структур coiled-coil. Эти участки могут пересекаться между собой или быть разделены

друг от друга любым числом аминокислот. Для всех пар гептамеров из разных последовательностей вычисляется расстояние при выравнивании без делеций. (Здесь видна аналогия с алгоритмом Масека–Патерсона.)

Рассмотрим наряду с основной таблицей расстояний между префиксами последовательностей, вспомогательные таблицы для других вариантов выравнивания.

Пусть известно оптимальное выравнивание (с оптимальной разметкой участков) для префиксов, длины которых меньше числа j для первой или числа k для второй последовательности. Для заполнения новых ячеек таблиц, рассмотрим несколько случаев.

Если хотя бы одна из позиций — j-я в первой или k-я во второй последовательности — не покрыта ранее размеченными гептамерами, то вычисляем расстояние между префиксами, длины которых равны j и k соответственно, как обычно в методе Нидлмана–Вунша.

Если обе позиции — j-я в первой и k-я во второй последовательности — покрыты ранее размеченными гептамерами, то параллельно продолжаем выравнивание двумя путями. На первом пути к префиксу добавляется гептамер, как это делалось в алгоритме Масека–Патерсона. Результаты вносятся во вспомогательную таблицу. На втором пути добавляем дополнительный штраф (за отказ от использования гептамера) и продолжаем заполнение второй вспомогательной таблицы, пока не будет добавлено достаточно много (хотя бы семь) новых позиций. При этом может быть вторичное разветвление. Если на втором пути получено оптимальное расстояние для префиксов достаточно большой длины, то это значение сравнивается со значением на первом пути. Из этих вариантов выбирается меньшее. После того, как такой выбор сделан, значение расстояния записывается в основную таблицу и уже не меняется. Но из-за вторичных разветвлений окончательный выбор может быть отложен. Теорема доказана.

Замечание 2. *Штраф за несоответствие гептамеров из разных последовательностей можно не задавать, но определять подбором до достижения оптимума по Парето.*

Дальнейшее ускорение алгоритма методом Масека–Патерсона позволяет искать решения при дополнительном условии.

Теорема 2. *Пусть фиксированы штрафы за вставки и замены аминокислот, а также штрафы за несоответствие гептамеров, участвующих в образовании структуры coiled-coil. Существует алгоритм, который получает на вход две аминокислотные последовательности, длины которых равны m и n соответственно и кратны некоторому числу p. Пусть выполнены два условия: во-первых, блоки не разделяют начала и концы каких-либо гептамеров, которые могут участвовать в образовании структуры coiled-coil; во-вторых, общее число различных блоков мало, поскольку они повторяются в составе входных последовательностей. Тогда этот алгоритм, делая $O(mn/p^2)$ шагов метода динамического программирования, вычисляет обобщенное редакционное расстояние с учётом структуры coiled-coil и оптимальное число выравненных друг под другом гептамеров.*

Доказательство. По условию длины m и n кратны числу p. Сначала алгоритм вычисляет таблицы оптимальных расстояний для всех пар блоков, встретившихся во входных последовательностях. Число шагов метода динамического программирования для каждой пары блоков ограничено сверху величиной $O(p^2)$. С другой стороны, число встретившихся пар блоков не превосходит числа mn/p^2, но по условию это число значительно меньше верхней границы. Следовательно, время такого препроцессинга мало по сравнению со временем последующей работы алгоритма. Дополнительное условие состоит в том, что блоки не разделяют начала и концы каких-либо гептамеров. Это позволяет за один шаг добавлять к выравниванию блок целиком. Когда известны таблицы расстояний для всех встретившихся пар блоков, вычисление большой таблицы расстояний для исходных последовательностей легко сводится к добавлению блоков. Это позволяет уменьшить число шагов алгоритма динамического

программирования. Считая присоединение блока за один шаг, число шагов будет ограничено сверху функцией $O(mn/p^2)$. Теорема доказана.

Замечание 3. *Аналогично можно рассматривать разбиение на блоки, имеющие различные длины, в частности, можно ослабить требование, чтобы длины входных последовательностей были кратны одному числу p. К сожалению, при работе с последовательностями над большим алфавитом условие малости числа различных блоков не будет выполнено на большой доле последовательностей.*

3.1. Детали алгоритма

Основной трудностью является обработка ситуаций, в которых несколько пар гептамеров накладывается друг на друга. Заметим, что если в позиции i было размечено начало гептамера, то ещё два гептамера могут начинаться в позициях $i+3$ и $i+6$. В таких случаях возникают вторичные разветвления, которые также необходимо рассматривать. Будем называть гептамерным наложением строку длины s, в которой на позициях $1, 4, 7, ...s-6$ размечены начала гептамеров.

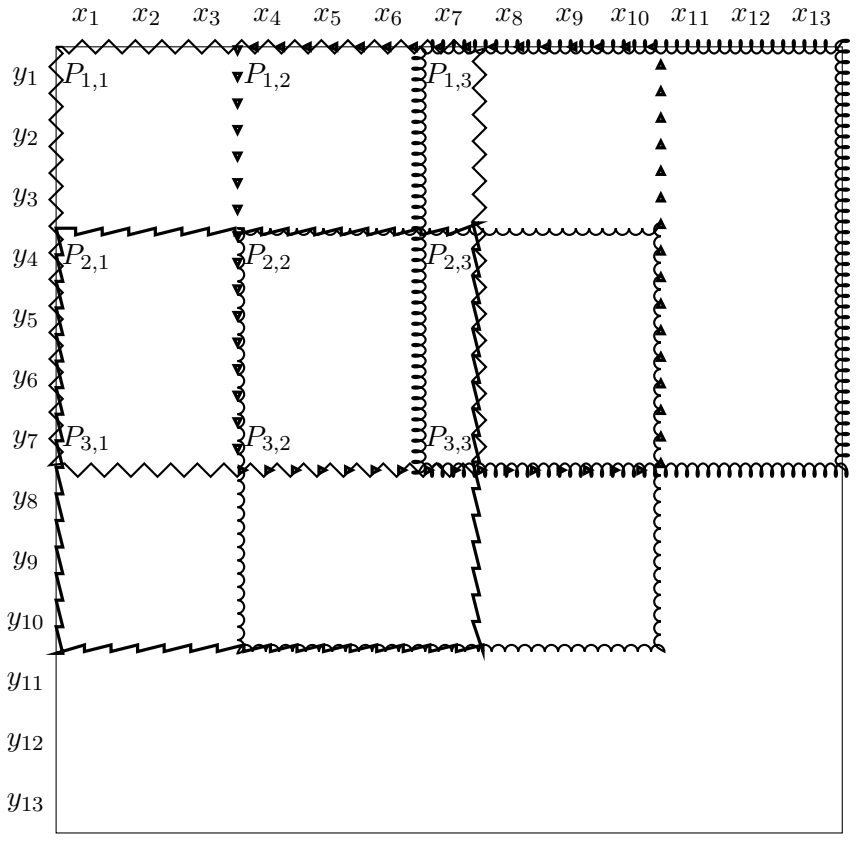

Рис. 1: Выравнивание гептамерных наложений x и y, в индексах $1, 4, 7$ – начала гептамеров, в каждой паре индексов начинается вычисление нового разветвления пары таблиц(на рисунке отмечены первые 5 таблиц), затем выбирается лучшее значение на левом нижнем конце таблицы. Позиции, в которых начинаются новые разветвления отмечены $P_{i,j}$, Все разветвления начинаются внутри первой таблицы.

Заметим, что если выбирается гептамер на позиции i, то гептамеры на позициях $i-3, i-6, i+3, i+6$ не могут быть выбраны. Таким образом, гептамеры внутри наложений можно разбить на непересекающиеся упорядоченные тройки и выравнивать каждую пару троек из

разных последовательностей, так как при выборе одного гептамера в тройке, два других не могут быть выбраны. Тогда, будем называть блоком девять вспомогательных таблиц, содержащих лучшие выравнивания для каждой пары гептамеров из двух троек. Пример таблиц внутри блока изображен на рисунке 1.

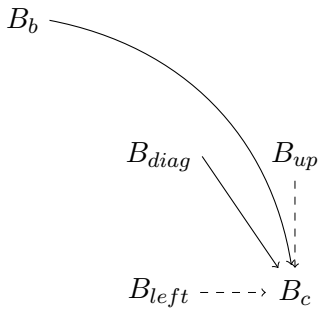

Рис. 2: Блоки участвующие в заполнении блока B_c. Блок B_b содержит таблицу T_{best}, из этого блока, а также из блока B_{diag} необходимо просчитывать два сценария выравнивания, из блоков B_{up} и B_{left} – только игнорирующий гептамеры, эти переходы отмечены пунктиром.

Заметим, что для вычисления каждого следующего блока необходимо знать не более четырёх блоков – блока слева B_{left}, блока сверху B_{up} и блока сверху и слева B_{diag}. Кроме того, необходимо рассматривать таблицу T_{best} с лучшим расстоянием для последних гептамеров в тройках, находящихся на два блока левее и два блока выше от вычисляемого. От каждого из блоков и таблицы достраиваются таблицы для всех пар гептамеров в вычисляемом блоке, а затем для каждой пары выбирается лучшая из них.

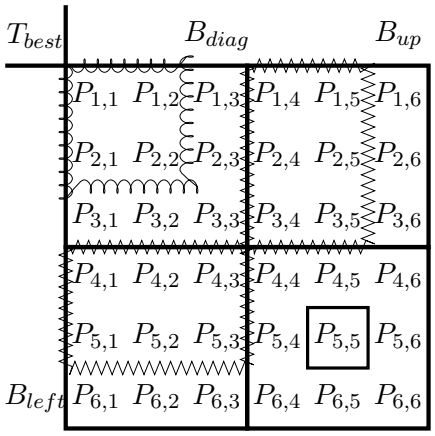

Рис. 3: Таблицы из блока B_{diag}, влияющие на вычисление таблицы $P_{5,5}$(для блока, расположенного справа внизу находятся на расстоянии не менее двух таблиц по вертикали и двух таблиц по горизонтали(обведены спиралью). От этих таблиц, а также от таблицы T_{best} вычисляются оба сценария выравнивания. Из блоков B_{left} и B_{up} используются только таблицы, расположенные не ниже и не левее вычисляемой таблицы (обведены зигзагом).

При этом важно, что при вычислении таблиц от блока B_{diag} и таблицы T_{best} обязательно необходимо вычислить таблицы как для случая, когда к концу таблицы добавляется гептамер, так и для случая, когда гептамеры не используются, но в то же время, от блоков B_{left} и B_{up} достраиваются таблицы с игнорированием гептамеров, причём в блоке B_{left} рассматриваются только таблицы, находящиеся не ниже вычисляемой, а в блоке B_{up} только те таблицы, которые не находятся правее вычисляемой. Игнорирование одного из сценариев выравнивания для

вычисления таблиц от этих блоков обусловлено тем, что если гептамер хотя бы в одной из троек уже использован в выравнивании, то он уже не может быть использован в выравнивании ни с одним другим гептамером. Диаграмма заполнения нового блока изображена на рисунке 2.

Из блока B_{diag} для пары гептамеров будут достраиваться таблицы с использованием гептамеров только от тех пар, гептамеры которых не пересекаются с рассматриваемыми гептамерами. Таблица T_{best} достраивается для всех таблиц вычисляемого блока, предполагая, что в блоке B_{diag} ни одна из пар гептамеров не была использована. На рисунке 3 изображен пример выбора таблиц для вычисления новой таблицы, находящейся в середине блока.

Таким образом, так как число сравнений на каждом шаге равно константе, общая сложность алгоритма остаётся квадратичной. В то же время, так как каждый новый ряд таблицы блоков вычисляется, используя только два предыдущих ряда блоков, одновременно необходимо хранить только три ряда блоков. Пример графа, задающего порядок заполнения новых блоков для двух гептамерных наложений изображен на рисунке 4.

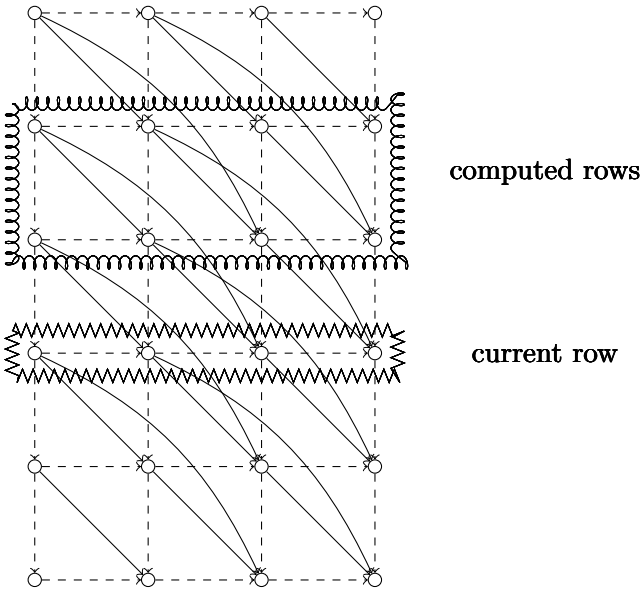

Рис. 4: Граф обхода блоков для двух гептамерных наложений. Зигзагом обведён вычисляемый ряд блоков, для его вычисления используются два верхних ряда блоков.

Замечание 4. *Для вычисления выравнивания двух наложений разной длины можно располагать ряды блоков вдоль более длинного наложения, тогда, каждый ряд будет не большей длины, чем при расположении рядов вдоль короткого наложения, однако, при большой разнице в длине наложений, число блоков в ряду может сильно уменьшиться.*

4. Обсуждение

Заметим, что несмотря на большое количество блоков, которые теоретически могут быть рассмотрены, фактически, в реальных биологических данных длинные наложения являются редкостью. Авторы проанализировали открытый набор данных protein data bank [31, 32], содержащий аминокислотные нуклеотидные последовательности белков, из 89083 уникальных последовательностей, результаты анализа приведены на таблице 4. Гептамеры присутствовали в 75941 последовательностях, из них, в 32521 присутствовали гептамерные наложения, и в 1070 присутствовали наложения длинее 3, которые потребовали бы вычисление более одного блока.

Длина наибольшего гептамерного наложения	Число последовательностей
1	43420
2	26141
3	5310
4	962
5	78
6	23
7	3
9	1
10	1
11	1
13	1

Таблица 1: Распределение последовательностей по длине наибольшего гептамерного наложения в Protein Data Bank

Замечание 5. *Важно заметить, что рассматриваемая структура является обобщением других принятых в биологической литературе структур, например, структуры, рассмотренной в статье Риса и соавторов [33], в которой, в участке* $abcdefg$*, на позициях* a *и* d *расположены гидрофобные аминокислоты, а на позициях остальных позициях – полярные аминокислоты. Гептамеры, соответсвующие такой структуре не могут содержать внутри себя другие гептамеры, и для них необходимость рассматривать наложения отсутствует.*

5. Заключение

Рассматриваемый нами метод может применяться для уточнения предсказания структуры белка. Малое время работы позволит проводить вычисления с большими наборами белков, в частности, при изучении эволюции эукариотических белков. С одной стороны, выравнивание с учётом структуры позволяет быстро выделять те структуры, которые остаются консервативными в ходе эволюции. В частности, это структуры, которые меняются в результате изменения внешних условий или взаимодействия с другими белками. С другой стороны, полученное редакционное расстояние может отличаться от аналогичного расстояния, вычисляемого вне зависимости от структуры. Это позволит уточнить отношение гомологичности на множестве белков.

СПИСОК ЦИТИРОВАННОЙ 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. https://doi.org/10.1016/0022-2836(70)90057-4

2. 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. https://doi.org/10.1016/0022-0000(80)90002-1

3. 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. https://doi.org/10.1137/S0097539702402007

4. Tiskin A. Semi-local longest common subsequences in subquadratic time // Journal of Discrete Algorithms. 2008. V. 6, № 4. P. 570–581. https://doi.org/10.1016/j.jda.2008.07.001

5. Grabowski S. New tabulation and sparse dynamic programming based techniques for sequence similarity problems // Discrete Applied Mathematics. 2016. V. 212. P. 96–103. https://doi.org/10.1016/j.dam.2015.10.040

6. Tiskin A. The Chvátal–Sankoff problem as a problem of symbolic dynamics // Зап. научн. сем. ПОМИ. 2023. Т. 528. С. 214–237.

7. 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. https://doi.org/10.1145/3422823

8. Kociumaka T., Radoszewski J., Starikovskaya T. Longest common substring with approximately k mismatches // Algorithmica. 2019. V. 81, № 6. P. 2633–2652. https://doi.org/10.1007/s00453-019-00548-x

9. Steinegger M., Söding J. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets // Nat. Biotechnol. 2017. V. 35. P. 1026–1028. https://doi.org/10.1038/nbt.3988

10. Бурдонов И., Максимов А. Двадцать функций подобия двух конечных последовательностей // Программирование. 2023. № 5. С. 3–18. https://doi.org/10.31857/S0132347423050035

11. Lafond M., Lai W., Liyanage A., Zhu B. The longest subsequence-duplicated subsequence and related problems // Information and Computation. 2025. V. 306. №. 105313, P. 1–15. https://doi.org/10.1016/j.ic.2025.105313

12. Kovačević M. On the Maximum Number of Non-Confusable Strings Evolving under Short Tandem Duplications // Probl Inf Transm. 2022. V. 58. P. 111–121 https://doi.org/10.1134/S0032946022020028

13. Goshkoder D., Polyanskii N., Vorobyev I. Codes Correcting Long Duplication Errors // IEEE Transactions on Molecular, Biological, and Multi-Scale Communications. 2024. V. 10, № 2, P. 272-288 https://doi.org/10.1109/TMBMC.2024.3403755

14. Jumper J., Evans R., Pritzel A. *et al.* Highly accurate protein structure prediction with AlphaFold // Nature. 2021. V. 596. P. 583–589. https://doi.org/10.1038/s41586-021-03819-2

15. Rachitskii P., Kruglov I., Finkelstein A. V., Oganov A. R. Protein structure prediction using the evolutionary algorithm USPEX // Proteins. 2023. V. 91, № 7. P. 933–943. https://doi.org/10.1002/prot.26478

16. Цветкова А. Д., Швец Д. Ю., Мусин Х. Г., Кулуев Б. Р. Моделирование структуры белка tRolC *Nicotiana tabacum* и его функциональная связь с другими белками // Математическая биология и биоинформатика. 2024. Т. 19. № 2. С. 322–337. https://doi.org/10.17537/2024.19.322

17. Finkelstein A. V., Bogatyreva N. S., Ivankov D. N., Garbuzynskiy S. O. Protein folding problem: enigma, paradox, solution // Biophys Rev. 2022. V. 14. P. 1255–1272. https://doi.org/10.1007/s12551-022-01000-1

18. Зверков О. А., Селиверстов А. В., Шиловский Г. А. Выравнивание скрытого палиндрома // Математическая биология и биоинформатика. 2024. Т. 19, № 2, С. 427–438. https://doi.org/10.17537/2024.19.427

19. 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 no. e2100191. https://doi.org/10.1002/bies.202100191

20. Назипова Н. Н. Разнообразие некодирующих РНК в геномах эукариот // Математическая биология и биоинформатика. 2021. Т. 16. № 2. С. 256–298. https://doi.org/10.17537/2021.16.256

21. Tiskin A. Periodic string comparison // Kucherov G., Ukkonen E. (eds) *Combinatorial Pattern Matching. CPM 2009.* Lecture Notes in Computer Science, vol. 5577. Springer, Berlin, Heidelberg, 2009. https://doi.org/10.1007/978-3-642-02441-2

22. Золотов Б. А., Гаевой Н. С., Тискин А. В. Алгоритмы сравнения периодических строк // IV Конференция математических центров России: Сборник тезисов. Санкт-Петербург, 2024. С. 143.

23. Efimov D. B. Enumeration of Even and Odd Chord Diagrams //J. Appl. Ind. Math. 2024. V 18, P. 216–226. https://doi.org/10.1134/S1990478924020042

24. Crick F. H. Is α-keratin a coiled coil? // Nature. 1952. V. 170. P. 882–883. https://doi.org/10.1038/170882b0

25. Truebestein L., Leonard T. A. Coiled-coils: The long and short of it // Bioessays. 2016. V. 38, № 9. P. 903–916. https://doi.org/10.1002/bies.201600062

26. Lupas A. Predicting coiled-coil regions in proteins // Current Opinion in Structural Biology. 1997. V. 7, № 3. P. 388–393. https://doi.org/10.1016/S0959-440X(97)80056-5

27. Feng S.-H., Xia C.-Q., Shen H.-B. CoCoPRED: coiled-coil protein structural feature prediction from amino acid sequence using deep neural networks // Bioinformatics. 2022. V. 38, № 3. P. 720–729. https://doi.org/10.1093/bioinformatics/btab744

28. Kyte J., Doolittle R. F. A simple method for displaying the hydropathic character of a protein // Journal of Molecular Biology. 1982. V. 157, № 1. P. 105–132. https://doi.org/10.1016/0022-2836(82)90515-0

29. 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.

30. Ройтберг М. А., Симеоненков М. Н., Таболина О. Ю. Парето-оптимальные выравнивания символьных последовательностей // Биофизика. 1999. Т. 44, № 4. С. 581–594.

31. RCSB Protein Data Bank [Электронный ресурс]. – URL: https://www.rcsb.org (дата обращения: 20.06.2025).

32. Protein Data Set [Электронный ресурс] // Kaggle. – URL: https://www.kaggle.com/datasets/shahir/protein-data-set (дата обращения: 20.06.2025).

33. Rhys, G .G., Wood, C. W., Lang, E. J. M. *et al.* Maintaining and breaking symmetry in homomeric coiled-coil assemblies // Nat Commun 2018. V. 9, № 4132, P. 1–12. https://doi.org/10.1038/s41467-018-06391-y

REFERENCES

1. Needleman, S. B. & Wunsch, Ch. D. 1970, “A general method applicable to the search for similarities in the amino acid sequence of two proteins”, *Journal of Molecular Biology*, vol. 48, no. 3, pp. 443–453. https://doi.org/10.1016/0022-2836(70)90057-4

2. Masek, W. J. & Paterson, M. S. 1980, “A faster algorithm computing string edit distances”, *Journal of Computer and System Sciences*, vol. 20, no. 1, pp. 18–31. https://doi.org/10.1016/0022-0000(80)90002-1

3. Crochemore, M., Landau, G. M. & Ziv-Ukelson, M. 2003, “A subquadratic sequence alignment algorithm for unrestricted score matrices”, *SIAM Journal on Computing*, vol. 32, no. 6, pp. 1654–1673. https://doi.org/10.1137/S0097539702402007

4. Tiskin, A. 2008, “Semi-local longest common subsequences in subquadratic time”, *Journal of Discrete Algorithms*, vol. 6, no. 4, pp. 570–581. https://doi.org/10.1016/j.jda.2008.07.001

5. Grabowski, S. 2016, “New tabulation and sparse dynamic programming based techniques for sequence similarity problems”, *Discrete Applied Mathematics*, vol. 212, pp. 96–103. https://doi.org/10.1016/j.dam.2015.10.040

6. Tiskin, A. 2023, “The Chvátal–Sankoff problem as a problem of symbolic dynamics”, *Zapiski Nauchnykh Seminarov POMI*, vol. 528, pp. 214–237.

7. Chakraborty, D., Das, D., Goldenberg, E., Koucký, M. & Saks M. 2020, “Approximating edit distance within constant factor in truly sub-quadratic time”, *Journal of the ACM*, vol. 67, no. 6, article no. 36. https://doi.org/10.1145/3422823

8. Kociumaka, T., Radoszewski, J. & Starikovskaya, T. 2019, “Longest common substring with approximately k mismatches”, *Algorithmica*, vol. 81, no. 6, pp. 2633–2652. https://doi.org/10.1007/s00453-019-00548-x

9. Steinegger, M. & Söding, J. 2017, “MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets”, *Nat. Biotechnol.*, vol. 35, pp. 1026–1028. https://doi.org/10.1038/nbt.3988

10. Burdonov, I. & Maksimov, A. 2023. “Twenty similarity functions for two finite sequences”, *Programming and Computer Software*, vol. 49, pp. 373–387. https://doi.org/10.1134/S0361768823050031

11. Lafond, M., Lai, W., Liyanage, A., Zhu, B. 2025, “The longest subsequence-duplicated subsequence and related problems” *Information and Computation* vol. 306. no. 105313, pp. 1-15. https://doi.org/10.1016/j.ic.2025.105313

12. Kovačević, M. 2022, “On the Maximum Number of Non-Confusable Strings Evolving under Short Tandem Duplications” *Probl Inf Transm* vol. 58. pp. 111–121 https://doi.org/10.1134/S0032946022020028

13. Goshkoder, D., Polyanskii, N., Vorobyev, I. 2024, “Codes Correcting Long Duplication Errors” *IEEE Transactions on Molecular, Biological, and Multi-Scale Communications* vol. 10, no. 2, pp. 272-288 https://doi.org/10.1109/TMBMC.2024.3403755

14. Jumper, J., Evans, R., Pritzel, A. & et al. 2021, “Highly accurate protein structure prediction with AlphaFold”, *Nature*, vol. 596, pp. 583–589. https://doi.org/10.1038/s41586-021-03819-2

15. Rachitskii, P., Kruglov, I., Finkelstein, A. V. & Oganov A. R. 2023, "Protein structure prediction using the evolutionary algorithm USPEX", *Proteins*, vol. 91, no. 7, pp. 933–943. https://doi.org/10.1002/prot.26478

16. Tsvetkova, A. D., Shvets, D. Yu., Musin, Kh. G., Kuluev, B. R. 2024, "Modeling of the structure of the tRolC protein of Nicotiana tabacum and its functional relation to other proteins", *Mathematical biology and bioinformatics*, vol. 19, no. 2, pp. 322–337. https://doi.org/10.17537/2024.19.322

17. Finkelstein, A. V., Bogatyreva, N. S., Ivankov, D. N., Garbuzynskiy, S. O. 2022, "Protein folding problem: enigma, paradox, solution", *Biophys Rev.*, vol. 14, pp. 1255–1272. https://doi.org/10.1007/s12551-022-01000-1

18. Zverkov, O., Seliverstov, A. & Shilovsky G. 2024, "Alignment of a hidden palindrome", *Mathematical Biology and Bioinformatics*, vol. 19, no. 2, pp. 427–438. https://doi.org/10.17537/2024.19.427

19. Datta, R. R. & Rister, J. 2022, "The power of the (imperfect) palindrome: sequence-specific roles of palindromic motifs in gene regulation", *Bioessays*, vol. 44, no. 4, article no. e2100191. https://doi.org/10.1002/bies.202100191

20. Nazipova, N. 2021, "Variety of non-coding RNAs in Eukaryotic genomes", *Mathematical biology and bioinformatics*, vol. 16, no. 2, pp. 256–298. https://doi.org/10.17537/2021.16.256

21. Tiskin, A. 2009, "Periodic string comparison", in: Kucherov, G., Ukkonen, E. (eds) *Combinatorial Pattern Matching. CPM 2009.* Lecture Notes in Computer Science, vol. 5577. Springer, Berlin, Heidelberg, 2009. https://doi.org/10.1007/978-3-642-02441-2

22. Zolotov, B. A., Gaevoy, N. S., Tiskin, A. V. 2024, "Algorithms for comparing periodic strings", *IV Conference of Mathematical Centers of Russia: Collection of Abstracts*, St. Petersburg, 2024, p. 143.

23. Efimov, D. B. 2024 "Enumeration of Even and Odd Chord Diagrams", *J. Appl. Ind. Math.*, vol. 18, pp.216–226. https://doi.org/10.1134/S1990478924020042

24. Crick, F. H. 1952, "Is α-keratin a coiled coil?", *Nature*, vol. 170, pp. 882–883. https://doi.org/10.1038/170882b0

25. Truebestein, L. & Leonard, T. A. 2016, "Coiled-coils: The long and short of it", *Bioessays*, vol. 38, no. 9, pp. 903–916. https://doi.org/10.1002/bies.201600062

26. Lupas, A. 1997, "Predicting coiled-coil regions in proteins", *Current Opinion in Structural Biology*, vol. 7, no. 3, pp. 388–393. https://doi.org/10.1016/S0959-440X(97)80056-5

27. Feng, S.-H., Xia, C.-Q., Shen H.-B. 2022, "CoCoPRED: coiled-coil protein structural feature prediction from amino acid sequence using deep neural networks" *Bioinformatics*, vol. 38, no. 3, pp. 720–729. https://doi.org/10.1093/bioinformatics/btab744

28. Kyte, J. & Doolittle, R. F. 1982, "A simple method for displaying the hydropathic character of a protein", *Journal of Molecular Biology*, vol. 157, no. 1, pp. 105–132. https://doi.org/10.1016/0022-2836(82)90515-0

29. Arlazarov, V. L., Dinic, E. A., Kronrod, M. A. & Faradzev, I. A. 1970, "On economic construction of the transitive closure of a directed graph", *Soviet Math. Dokl.*, vol. 11, no. 5, pp. 1209–1210.

30. Roytberg, M. A., Semionenkov, M. N., Tabolina, O. Yu. 1999, “Pareto-optimal alignment of biological sequences”, *Biophysics*, vol. 44, no. 4, pp. 565–577.

31. RCSB PDB (2025) *RCSB Protein Data Bank*. Available at: https://www.rcsb.org (Accessed: 20 June 2025).

32. Kaggle (2018) *Protein Data Set*. Available at: https://www.kaggle.com/datasets/shahir/protein-data-set (Accessed: 20 June 2025).

33. Rhys, G .G., Wood, C. W., Lang, E. J. M. *et al.* 2018, “Maintaining and breaking symmetry in homomeric coiled-coil assemblies” // *Nat Commun*, vol. 9, no. 4132, pp. 1–12. https://doi.org/10.1038/s41467-018-06391-y

Работа выполнена в ИППИ РАН