Об одном подходе к реализации алгоритмов Нидлмана – Вунша и Джаро – Винклера и их применении в корреляционном анализе сходства митохондриальных ДНК обезьян. Часть I. Общее описание работы

Ли Цзямянь, Му Цзинъюань, Б. Ф. Мельников

***Аннотация*—В исследованиях молекулярной биологии и геномики крайне важно понимать генетические различия между разными видами. Сравнение сходства последовательностей ДНК может предоставить ценную информацию о родственных связях между видами.**

В настоящей статье для сравнения митохондриальных ДНК обезьян использовались два алгоритма – Нидлмана – Вунша и Джаро – Винклера; кроме того, в последующих частях статьи будут приведены подобные сравнения и для других млекопитающих.

Ранее при проведении подобных исследований у настоящей статьи авторов возникла гипотеза о том, что при применении этих двух алгоритмов для анализа сходства одних и тех же пар геномных последовательностей получаются весьма непохожие результаты. Одним из предметов настящей статьи и является описание подхода к тому, как именно мы предлагаем давать численные ответы на подобные вопросы. Такие ответы мы предполагаем давать с помощью применения парной корреляции, о которой будет сказано в следующих частях статьи.

Из результатов настоящей статьи следует необходимость продолжения подробных исследований цепочек ДНК, в частности, на предмет анализа их сходства; то есть подобные задачи остаются и ещё на долгое время останутся весьма актуальными.

***Ключевые слова*—последовательности ДНК, алгоритм Нидлмана – Вунша, алгоритм Джаро – Винклера, матрица расстояний, корреляционный анализ.**

I. ВВЕДЕНИЕ

В исследованиях молекулярной биологии и геномики крайне важно понимать генетические различия между разными видами. Сравнение сходства последовательностей ДНК, особенно митохондриальной ДНК (мт ДНК), может предоставить ценную информацию о родственных связях между видами. И можно сказать, что настоящая статья является продолжением комплекса работ одного из авторов по этой тематике; среди этих статей отметим [1], [2], [3], [4], [5], [6], [7], см. также некоторые ссылки из упомянтых статей.

Статья получена 9 мая 2024 г.

Ли Цзямянь, Университет МГУ – ППИ в Шэньчжэне (lijiamian0804@live.com).

Му Цзинъюань, Университет МГУ – ППИ в Шэньчжэне (xirousang@gmail.com).

Борис Феликсович Мельников, Университет МГУ – ППИ в Шэньчжэне (bormel@smbu.edu.cn).

В настоящей статье использовались два алгоритма – Нидлмана – Вунша ([2], [3], а также, в качестве первоисточника, см. [8]) и Джаро – Винклера ([9] и мн. др.) – для сравнения мт ДНК обезьян; кроме того [1], проведено сравнение с другими млекопитающими – для оценки надёжности результатов алгоритмов в сравнении генов.

Приведём текст, который скорее можно назвать научно-популярным. [2] По общей последовательности ДНК человек фактически стоит обособленно от других человекообразных; причём это но не по «формальному набору генов», а по их распределению по хромосомам. И именно подобные факторы –

- несколько хромосомных аберраций;
- делеция огромного участка;
- переход ещё одного к иной хромосоме – в итоге у людей на одну пару аутосом меньше [3];
- разворот ещё одного участка –

скорее всего, и привёл к радикальному изменению *фенотипа* [4].

Последнее же, т. е. изменение фенотипа, можно описать в первую очередь следующими признаками (также на основе материала, взятого из многочисленных научных и научно-популярных публикаций):

- отсутствие у людей массивной, выступающей вперёд челюсти и, следовательно, существенно иное строение ротовой полости – важнейшего резонатора при речеобразовании;
- существенно иное строение носа (а также и гортани);
- отсутствие шерстяного покрова;
- прямохождение;
- перестроенная работа сальных и потовых желёз;
- перестройка верхней части черепа;

[1] В следующих частях статьи.

[2] Однако можно и упростить это слово-композит, т. е. назвать текст *научным*. Конкретных ссылок на литературу мы приводить не будем: ниже в нескольких абзацах – обощение прочитанного авторами в научной и научно-популярной литературе.

[3] Аутосомами у организмов с хромосомным определением пола называются парные хромосомы, одинаковые у мужских и женских организмов.

[4] Фенотипом мы считаем совокупнось внутренних и внешних особенности, свойства и черт конкретного организма. (В литературе имеются и другие определения.)

- и многое другое, отличающее именно людей от человекообразных вообще.

Однако, как можно понять, всё приведённое выше – это общие размышления, на уровне «по-видимому», не подкреплённые конкретными геномными исследованиями. Но при этом именно такая «неподкреплённость», невозможность (по крайней мере, в ближайшее время) строго доказать приведённые выше зависимости, – как раз она

> *говорит о необходимости продолжать подробные исследования цепочек ДНК, в частности, на предмет анализа их сходства*.

То есть подобные задачи остаются и ещё на долгое время останутся весьма актуальными.

И именно при проведении подобных исследований у авторов возникла гипотеза о том, что

> *при применении двух широко известных алгоритмов для анализа сходства одних и тех же пар геномных последовательностей – конкретно, алгоритмов Нидлмана – Вунша и Джаро – Винклера – получаются весьма непохожие результаты*.

Одним из предметов настящей статьи и является описание подхода к тому, как именно мы предлагаем давать *численные* ответы на подобные вопросы.

Для этого мы рассматриваем матрицы расстояний между геномами. Как и в наших предыдущих работах, мы с помощью т. н. значения badness, специальным образом вычисляемого для каждого из треугольников матрицы [5], после чего – в дополнение к предыдущим работам –

> *считаем парную корреляцию между значениями badness для последовательности полученных треугольников*.

Относительно малые значения полученных вариантов парной корреляции (подробнее см. в следующих частях настоящей статьи) говорят о том, что сделанное предположение является верным, то есть алгоритмы Нидлмана – Вунша и Джаро – Винклера мало связаны друг с другом.

В предлагаемой части I настоящей статьи мы описываем только часть вычислительных экспериментов. В частности, если говорить только о рассматриваемых видах, – мы здесь приводим результаты вычислений для мт ДНК

- 32 вида обезьян, причём ни одна пара видов не принадлежит одному роду; сами рассматриваемые виды обезьян приведены в таблице I, некоторые подробности далее.

В дальнейшем мы приведём аналогичные результаты для митохондриальных ДНК:

- 20 видов обезьян, принадлежащих одному роду – «более близкий», чем рассматриваемый в части I, набор видов;
- 30 видов млекопитающих обезьян, принадлежащих разным отрядам – «более далёкий», чем рассматриваемый в части I, набор видов; в этом наборе, конечно, обезьяна будет только одна – однако менять название статьи вряд ли целесообразно.

При этом указанные здесь значения – 20 и 30 – возможно, будут немного скорректированы; к моменту же представления в печать части I у нас есть результаты вычислений именно для указанных значений, 32, 20 и 30.

Отметим ещё раз, что в результате проведённых вычислений с помощью вычисления коэффициентов парной корреляции подтверждается возникшая ранее гипотеза о том, что при применении двух алгоритмов (Нидлмана – Вунша и Джаро – Винклера) для анализа сходства одних и тех же пар геномных последовательностей получаются весьма непохожие результаты.

Приведём содержание по разделам части I настоящей статьи. Раздел II – краткое описание алгоритма Нидлмана – Вунша (без соответствующей программы, она будет кратко рассмотрена далее). Аналогично, раздел III – краткое описание алгоритма Джаро – Винклера. В разделе IV приведено общее описание нашей программы (бóльшая детализация будет приведена в последующих частях статьи). В разделе V приведён список видов обезьян, используемых для вычислений, даны полученные в результате вычислений матрицы расстояний между митохондриальными ДНК и приведены соответствующие краткие комментарии.

В конце введения отметим следующий факт, относящийся ко всем рассматриваемым нами далее структурам данных (можно сказать, что это относится и к применяемым алгоритмам). При рассмотрении геномов (митохондриальных ДНК) из базы данных [10] мы исключали те из них, которые содержат хотя бы один символ N (по смыслу он обозначает любой из четырёх нуклеотидов A, C, G, T). Однако тех видов, мт ДНК которых *не* содержат такого символа, по нашему мнению, достаточно для целей настоящей статьи.

II. КРАТКОЕ ОПИСАНИЕ АЛГОРИТМА НИДЛМАНА – ВУНША

Алгоритм Нидлмана – Вунша является классическим методом для глобального выравнивания биологических последовательностей – например, таких как генетические цепочки. Он был разработан в 1970 году и представляет собой *динамическое программирование решения задачи оптимизации*, направленной на максимизацию сходства между двумя последовательностями.

Конкретно, алгоритм *методом динамического программирования* строит двумерную матрицу, в которой каждый элемент представляет собой специальную оценку соответствующего положения совпадения. Затем начиная с правого нижнего угла этой матрицы осуществляется возврат, цель которого – найти оптимальный путь сопоставления, а также получить «максимальный балл» и результаты этого сопоставления.

[5] Badness – всегда численное значение «отхода» получаемого треугольника от некоторого остроугольного равнобедленного, см. подробнее процитированные выше статьи. Ранее мы использовали эти значения для получения значения badness всей матрицы – просто складывая их.

Для полного понимания наших алгоритмов посчитаем количество получаемых треугольников. В матрице 32×32 число вычисляемых предварительными («препроцессорными») алгоритмами элементов равно

$$\frac{32 \cdot 31}{2} = 496 \ -$$

такое число объясняет тот факт, что вычисления всех значений подобной матрицы, например с помощью алгоритма Нидлмана – Вунша, занимает на среднем современном компьютере около суток. А число треугольников равно

$$\frac{32 \cdot 31 \cdot 30}{2 \cdot 3} = 4960 .$$

Таблица I
Первая часть вычислительных экспериментов.
Рассматриваемые виды обезьян (номера в алфавитном порядке)

№№	Виды обезьян
1	Allenopithecus nigroviridis
2	Ateles belzebuth
3	Brachyteles arachnoides
4	Cacajao calvus
5	Callimico goeldii
6	Callithrix jacchus
7	Carlito syrichta
8	Cebuella pygmaea
9	Cephalopachus bancanus
10	Cercocebus atys
11	Cercopithecus albogularis
12	Chlorocebus sabaeus
13	Colobus angolensis
14	Erythrocebus patas
15	Galago moholi
16	Gorilla gorilla

№№	Виды обезьян
17	Lagothrix lagotricha
18	Leontopithecus rosalia
19	Macaca fascicularis
20	Macaca fuscata
21	Mandrillus leucophaeus
22	Nasalis larvatus
23	Nycticebus coucang
24	Papio anubis
25	Presbytis melalophos
26	Pygathrix nemaeus
27	Rhinopithecus roxellana
28	Saguinus oedipus
29	Saimiri boliviensis
30	Semnopithecus entellus
31	Tarsius dentatus
32	Theropithecus gelada

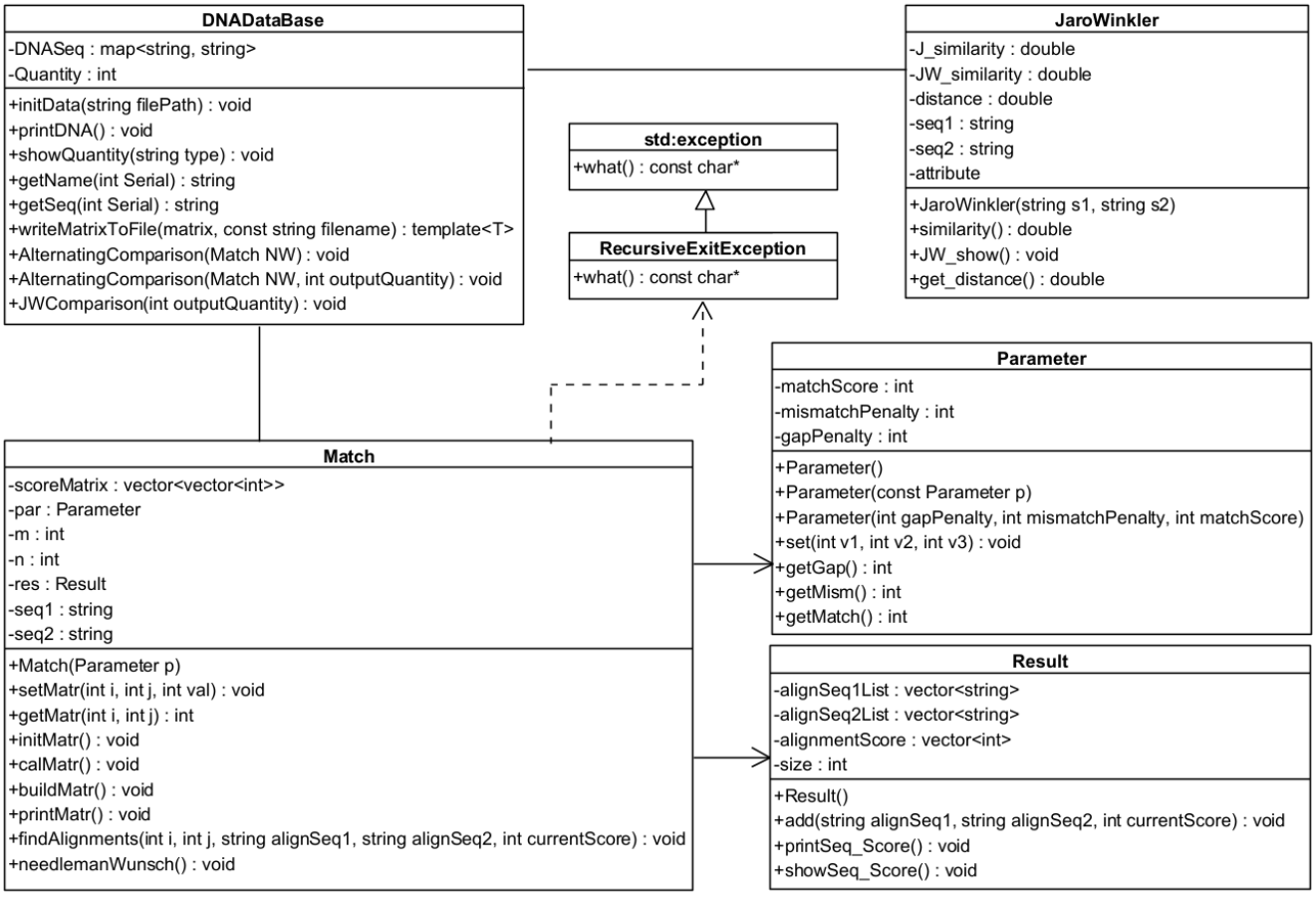

Рис. 1. Общая структура программы

Приведём описание нашего варианта этого алгоритма, близкое к [2], [3]; оно заключается в следующем. Как и в других интерпретациях этого алгоритма, мы считаем заданной матрицу *минимальных расстояний между аминокислотами* (либо между нуклеотидами). В качестве такой матрицы обычно используется матрица т. н. минимальных мутационных расстояний по генетическому коду – также либо между аминокислотами, либо между нуклеотидами; однако отметим, что для последней цели могут использоваться и другие меры.

По заданной матрице расстояний между аминокислотами итеративным образом рассчитывается следующая матрица всех возможных маршрутов

$$s_{ij} = D_{ij} + \max\big(s_{i-1,j-1}, \max_{k<j-1}(s_{j-1,k} - G), \max_{k<i-1}(s_{k,i-1} - G)\big),$$

где:

- s_{ij} – элемент i-й строки j-го столбца строимой матрицы;
- D_{ij} – расстояние между i-й и j-й аминокислотами (или нуклеотидами);
- G – штраф на т. н. «делецию» (т. е. за пропуск аминокислоты).

Затем осуществляется проход по матрице в обратном направлении, по максимальным элементам. Полученный маршрут соответствует оптимальному выравниванию, его значение принимается *в качестве выхода* алгоритма Нидлмана – Вунша.

Итак, основные шаги алгоритма включают:

- инициализацию матрицы;
- вычисление оценок – при котором необходимо учитывать оценки для 3 операций:
  - совпадение,
  - несовпадение
  - и вставка / удаление,

  обычно это достигается за счёт *определенной системы оценок*;
- и обратное отслеживание пути – при котором мы начинаем с правого нижнего угла матрицы и специальным образом двигаемся вверх и влево, определяя оптимальный путь с учётом текущей оценки и оценок соседних позиций.

Временна́я сложность алгоритма Нидлмана – Вунша составляет $O(m \cdot n)$, где m и n – длины двух заданных последовательностей. Хотя этот алгоритм может дать глобально оптимальные результаты сопоставления последовательностей, при работе с крупномасштабными последовательностями его затраты по времени и по памяти велики. Поэтому в практических приложениях часто используются улучшенные алгоритмы – либо эвристические методы сопоставления последовательностей, приводящие к повышению эффективности и точности.

III. КРАТКОЕ ОПИСАНИЕ АЛГОРИТМА ДЖАРО – ВИНКЛЕРА

Алгоритм Джаро–Винклера – один из *эвристических* алгоритмов для сопоставления последовательностей. Иными словами, это алгоритм вычисления сходства между двумя строками, который является *вариацией алгоритма Jaro distance*. Алгоритм Джаро–Винклера вычисляет сходство строк путём сравнения совпадения символов, порядка символов и расстояния между символами. Чем выше *итоговый балл сходства* (называемый обычно Jaro – Winkler similarity), тем больше сходство строк. При этом оценка 0 означает полное отсутствие сходства, а оценка 1 – полное совпадение.

Алгоритм Джаро – Винклера получился из улучшения Винклером алгоритма Джаро. Особенность алгоритма Джаро заключается в вычислении т. н. окна сопоставления (matching window) на основе длины рассматриваемых строк симолов. При сравнении двух строк – если в определённом окне длины появляются одинаковые символы, то эти символы считаются совпадающими, даже если их позиции не совпадают.

Винклер при улучшении алгоритма Джаро уделил особое внимание значению префиксов строк символов. При этом улучшение Винклера первоначально было направлено на анализ строк, введённых человеком, и он (Винклер) полагал, что люди обычно не ошибаются в первых нескольких буквах слова – например, “apple” не будет написано как “paple”, но может быть написано как “appel”. Поэтому он в своей реализации алгоритма увеличил вес первых нескольких символов в строке – считая, что два различных слова, начинающихся с разных символов, вероятнее всего, разные слова, а не одно слово с ошибкой в написании; то есть сходство первых нескольких символов в строках сильнее влияет на оценку сходства.

Приведём пример, который часто встречается в литературе: слова “earth” и “heart” имеют различные значения, и люди обычно не делают ошибок в их написании; Однако люди могут ошибочно написать “earth” как “eartn”, потому что место ошибки в написании находится ближе к концу слова и её труднее заметить. Однако при использовании исходного алгоритма Джаро слова ”eartn”и ”heart”имеют одну и ту же степень сходства с ”earth поскольку в нём нет различий между частями слова. Но если использовать версию алгоритма Джаро – Винклера, то из-за веса префиксов сходство “heart” и “earth” значительно меньше, чем сходство “eartn” и “earth”. Это – особенность алгоритма Джаро – Винклера именно как эвристического алгоритма.

Однако для нашей задачи, заключающейся в первую очередь в анализе степени сходства последовательностей мт ДНК различных видов обезьян, нет биологических доказательств того, что начальные символы генной последовательности оказывают большее влияние на их сходство. Хотя передняя часть мт ДНК содержит специальные функциональные части, такие как промотор (Promoter) и инициаторный кодон (Initiation codon), нет каких-либо биологических данных о том, что при сопоставлении генных последовательностей нужно уделять больше внимания начальной части. Кроме того, в алгоритме Джаро – Винклера префикс определяется как первые 4 символа последовательности, а 4 нуклеотидных пары в гене имеют мало значения, поскольку длина промотора обычно составляет как минимум десятки или сотни нуклеотидов. Возможно, мы можем изменить параметр «количество префиксов с весом», чтобы этот параметр как-то соответствовал бы либо длине промотора, либо длине контрольного региона (Control region) гена – однако длина

каждой части гена различна, и для подобных изменений алгоритма требуется поддержка биологов.

К сказанному приведём такой пример, являющийся, с нашей точки зрения, наиболее близким к рассматриваемой предметной области. Среди выбранных нами обезьян (см. таблицу I) имеются следующие два вида: Allenopithecus nigroviridis и Macaca fascicularis; хотя различие в начальных участках генных последовательностей этих двух видов обезьян значительно – в середине, а именно в области кодирующей части генов, наблюдается высокая степень сходства. А ведь именно эти участки являются ключевыми для определения последовательности аминокислот в синтезируемых белках.

Приведённый пример показывает, что при анализе генных последовательностей, возможно, не следует увеличивать вес начальной части, как это предлагал Винклер. Поэтому, хотя в обычном, «повседневном» сравнении *текстов* алгоритм Джаро – Винклера довольно распространён, мы считаем, что для сравнения генных последовательностей следует использовать первоначальный алгоритм Джаро, а не интерпретацию Джаро – Винклера.

В последующей работе, возможно, стоит рассмотреть улучшение алгоритма Джаро – Винклера, чтобы динамически адаптировать диапазон префиксов к длине конкретной последовательности гена, а также сделать возможным использование отрицательных значений веса, чтобы учесть снижение или повышение веса префикса в общей оценке сходства.

IV. ОБЩЕЕ ОПИСАНИЕ ПРОГРАММЫ

В качестве инструмента для сравнения геномных последовательностей мы предлагаем описываемую далее компьютерную программу. Она включает следующие 5 классов (см. также рис. 1):

- `Parameter` (класс параметров);
- `Result` (класс для результатов выравнивания генов);
- `JaroWinkler` (класс для выполнения алгоритма Джаро – Винклера, как в первоначальном виде, так и с использованием т. н. коррекции Винклера);
- `Match` (класс для выполнения алгоритма Нидлмана – Вунша);
- а также `DNADataBase` (класс для хранения и работы с данными ДНК).

Ниже приведены краткие описания каждой функции этих классов.

A. Класс Parameter

предназначен для установки и получения 3 видов баллов: за совпадение («положительные» баллы), штраф за несовпадение и штраф за вставку / удаление.

В классе реализованы следующие методы:

- `set(int v1, int v2, int v3)`: устанавливает значения для совпадений, несовпадений и вставки / удаления;
- `getMatch()`, `getMism()`, `getGap()`: возвращают соответствующие значения;
- `show()`: отображает значения параметров.

B. Класс Result

предназначен для хранения и отображения результатов выравнивания генов; класс используется для алгоритма Нидлмана – Вунша.

В классе реализованы следующие методы:

- `add(string alignSeq1, string alignSeq2, int currentScore)`: добавляет результат выравнивания генов в контейнер;
- `getDis(int i)`, `getDoubleDis(int i)`: получают значение расстояния, которое представляет собой количество разных символов в двух генных цепях, а также долю количества разных символов;
- `printSeq_Score()`: выводит результат выравнивания, включая две последовательности генов и оценку;
- `showSeq_Score()`: выполняет более наглядный вывод результатов, включающий две последовательности генов, метки совпадения и оценку.

C. Класс Match

предназначен для реализации алгоритма Нидлмана – Вунша.

В классе реализованы следующие методы:

- `setMatr(int i, int j, int val)`: присваивает значение выбранной ячейке матрицы;
- `getMatr(int i, int j)`: возвращает значение выбранной ячейки матрицы;
- `initMatr()`: инициализирует матрицу, учитывая параметры вставки / удаления;
- `calMatr()`: строит матрицу в соответствии с алгоритмом;
- `buildMatr()`: создаёт матрицу;
- `printMatr()`: выводит матрицу;
- `getRes()`: интерфейсный метод для получиения результата;
- `findAlignments(int i, int j, string alignSeq1, string alignSeq2, int currentScore)`: рекурсивно находит наилучшее соответствие;
- `needlemanWunsch()`: выполняет алгоритм Нидлмана – Вунша, включая построение матрицы и динамическое построение результирующего пути.

D. Класс JaroWinkler

предназначен для реализации алгоритма Джаро – Винклера.

В классе реализованы следующие методы:

- `similarity(bool long_tolerance)`: вычисление схожести Джаро – Винклера;
- `JW_show()`: выдача результатов
- `get_distance()`: вычисление значения расстояния Джаро – Винклера.

E. Класс DNADataBase

для хранения и управления данными, полученными при исследовании цепочек ДНК.

В классе реализованы следующие методы:

- `initData(string filePath)`: считывает данные ДНК из файла, создает базу данных;

- `printDNA()`: выводит данные ДНК;
- `showQuantity(string type)`: выводит количество ДНК и количество сравнений;
- `getName(int Serial)`, `getSeq(int Serial)`: возвращает имя и последовательность ДНК по номеру;
- `template<typename T> void writeMatrixToFile( const vector<vector<T>>& matrix, const string& filename)`: записывает матрицу в файл;
- `AlternatingComparison(Match NW)`: сравнивает данные ДНК попарно и выводит результаты в консоли с использованием алгоритма Нидлмана – Вунша;
- `AlternatingComparison(Match NW, int outputQuantity)`: сравнивает данные ДНК попарно и выводит результаты в файлы с использованием алгоритма Нидлмана – Вунша, при этом можно задавать количество выводимых результатов;
- `JWComparison(int outputQuantity)`: сравнивает данные ДНК попарно и выводит результаты в файлы с использованием алгоритма Джаро – Винклера, при этом можно задавать количество выводимых результатов.

F. Основная функция main()

инициализирует параметры для алгоритма Нидлмана – Вунша, читает данные ДНК из файла и создает базу данных, а также запускает сравнение последовательностей.

Основные используемые переменные:

- `Parameter par`: экземпляр класса Parameter для установки параметров алгоритма;
- `string filePath`: путь к файлу последовательности ДНК;
- *DNADataBase Monkey*: библиотека ДНК, созданная путем чтения последовательностей ДНК из файлов;
- `quantityComp`: количество пар последовательностей ДНК для сравнения;
- `Match NW`: экземпляр класса Match для выполнения алгоритма Нидлмана – Вунша.

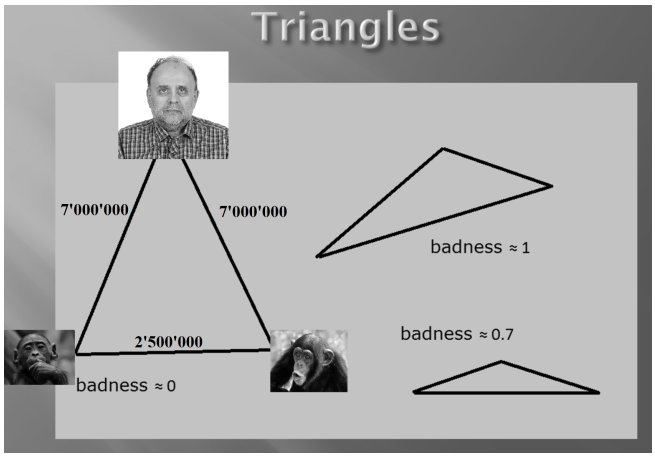

Рис. 2. Три вида, возможные треугольники в матрице расстояний между последовательностями ДНК и примерные соответствующие значения badness

Таблица II
Треугольники и соответствующие значения badness

Sides a, b, c	**Angles** α, β, γ	**Bad. (0)** $(\alpha - \beta)/\gamma$	**Bad. (5)** $(a - b)/c$
1 1 1	60 60 60	0	0
5 5 4	66 66 47	0	0
42 41 28	72 68 39	0.10	0.04
19 18 17	66 60 55	0.11	0.06
10 9 8	72 59 50	0.26	0.13
6 5 5	74 53 53	0.39	0.20
13 12 5	90 67 23	1.00	0.20
5 4 3	90 53 37	1.00	0.33
12 6 5	—	1.09	
20 6 5	—	1.81	

V. ПОЛУЧЕННЫЕ МАТРИЦЫ РАССТОЯНИЙ МЕЖДУ МИТОХОНДРИАЛЬНЫМИ ДНК И КРАТКИЕ КОММЕНТАРИИ

В наших предыдущих публикациях (кроме вышеперечисленных см. также [11]) мы несколько раз приводили естественный пример «о человеке, шимпанзе и бонобо».

Повторим этот пример очень кратко. Человек разошёлся с этими двумя видами около 7 миллионов лет назад (при этом точные сроки значения не имеют), а они между собой – около 2.5 миллионов лет назад. Поэтому любые *приемлемо* вычисленные расстояния между этими тремя геномами должны в идеале образовывать равнобедренный остроугольный треугольник, см. рис. 2. И очень важно, что этот пример может быть применен к любым трем видам. В частности, если случайно случилось так, что все три вида разделились в одно и то же время, то в матрице расстояний ожидается треугольник, близкий к равностороннему.

Как мы уже отмечали, в общем случае *количественную оценку отхода реального треугольника*, получаемого в результате выполнения алгоритма (в нашем случае, например, алгоритма Нидлмана – Вунша) от треугольника остроугольного равнобедренного мы называем badness. Для её вычисления мы использовали несколько возможных алгоритмов; в результате вычислительных экспериментов мы в качестве наиболее удачных определили два варианта, приведённые в таблице II. К ней необходимы такие комментарии:

- кроме реальных треугольников (первые 8 строк таблицы) приведены и две тройки чисел, треугольники не образующие; такие варианы при применении реальных алгоритмов встречаются – но крайне редко;
- номера badness от (0) до (5) совпадают с применявшимися в наших предыдущих работах;
- для всех треугольников мы полагаем $a \geqslant b \geqslant c$ (стороны) и $\alpha \geqslant \beta \geqslant \gamma$ (углы);
- конкретные варианты формул для badness от (1) до (4) мы в настоящей статье не приводим;
- вообще, все дальнейшие вычисления настоящей статьи (таблицы III и IV) выполнены на основе варианта (0) – мы в настоящее время считаем его самым перспективным.

Таблица III
The matrix obtained by applying the Jaro – Winkler's algorithm
to 32 species of monkeys (no more than one species from each genus)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
1	000	541	677	583	592	541	589	536	562	633	465	610	530	370	512	565	545	800	624	640	520	556	548	562	515	570	726	524	511	589	589	540
2	541	000	635	387	342	369	396	381	386	733	600	686	463	542	409	549	349	722	698	708	515	440	401	543	462	455	681	388	452	464	383	532
3	677	635	000	665	676	627	668	626	670	714	728	739	666	678	655	777	617	731	744	760	737	661	663	767	692	680	690	646	648	710	661	753
4	583	387	665	000	334	396	385	384	396	767	630	727	457	579	422	577	383	677	723	733	546	447	411	571	442	434	637	403	474	447	378	568
5	592	342	676	334	000	384	395	321	397	777	644	736	481	584	433	570	375	672	742	751	554	451	421	579	429	444	650	418	498	453	393	562
6	541	369	627	396	384	000	401	319	406	706	581	665	455	528	387	526	383	753	676	675	510	458	381	499	481	457	693	320	436	475	400	527
7	589	396	668	385	395	401	000	397	389	763	630	727	471	580	425	584	392	695	738	741	556	429	346	573	458	451	657	400	488	463	382	573
8	536	381	626	384	321	319	397	000	400	723	595	691	453	537	396	527	345	724	687	696	518	457	392	534	474	457	685	312	448	470	392	526
9	562	386	670	396	397	406	389	400	000	747	585	700	462	561	415	565	390	725	703	722	532	448	403	571	467	469	681	409	477	482	327	546
10	633	733	714	767	777	706	763	723	747	000	628	635	706	661	676	699	723	674	653	678	634	720	693	677	767	758	538	712	697	793	775	656
11	465	600	728	630	644	581	630	595	585	628	000	560	584	462	549	535	594	859	568	579	494	596	589	526	636	608	790	582	560	631	639	464
12	610	686	739	727	736	665	727	691	700	635	560	000	673	610	631	601	687	871	379	381	556	688	669	589	724	706	795	667	646	729	731	571
13	530	463	666	457	481	455	471	453	462	706	584	673	000	535	446	467	449	741	665	678	434	391	454	454	414	402	678	463	454	413	461	448
14	370	542	678	579	584	528	580	537	561	661	462	610	535	000	502	566	545	790	614	627	526	545	539	558	578	549	723	511	492	545	582	539
15	512	409	655	422	433	387	425	396	415	676	549	631	446	502	000	515	400	772	630	642	477	478	395	506	510	483	705	390	437	509	426	493
16	565	549	777	577	570	526	584	527	565	699	535	601	467	566	515	000	529	913	580	571	401	484	548	350	483	461	836	528	513	481	589	379
17	545	349	617	383	375	383	392	345	390	723	594	687	449	545	400	529	000	719	684	701	514	442	391	543	462	461	673	376	443	468	387	503
18	800	722	731	677	672	753	695	724	725	674	859	871	741	790	772	913	719	000	871	884	851	708	759	897	664	690	538	759	763	694	709	874
19	624	698	744	723	742	676	738	687	703	653	568	379	665	614	630	580	684	871	000	366	579	701	682	565	734	711	799	668	647	721	729	547
20	640	708	760	733	751	675	741	696	722	678	579	381	678	627	642	571	701	884	366	000	585	717	688	567	752	718	806	679	656	729	739	551
21	520	515	737	546	554	510	556	518	532	634	494	556	434	526	477	401	514	851	579	585	000	446	515	386	469	462	787	508	498	485	549	344
22	556	440	661	447	451	458	429	457	448	720	596	688	391	545	478	484	442	708	701	717	446	000	438	473	377	369	644	465	471	379	451	469
23	548	401	663	411	421	381	346	392	403	693	589	669	454	539	395	548	391	759	682	688	515	438	000	539	492	478	705	380	451	490	416	528
24	562	543	767	571	579	499	573	534	571	677	526	589	454	558	506	350	543	897	565	567	386	473	539	000	503	479	822	522	509	465	569	372
25	515	462	692	442	429	481	458	474	467	767	636	724	414	578	510	483	462	664	734	752	469	377	492	503	000	346	627	484	486	344	467	486
26	570	455	680	434	444	457	451	457	469	758	608	706	402	549	483	461	461	690	711	718	462	369	478	479	346	000	621	460	453	366	451	471
27	726	681	690	637	650	693	657	685	681	538	790	795	678	723	705	836	673	538	799	806	787	644	705	822	627	621	000	694	699	634	663	805
28	524	388	646	403	418	320	400	312	409	712	582	667	463	511	390	528	376	759	668	679	508	465	380	522	484	460	694	000	389	478	409	525
29	511	452	648	474	498	436	488	448	477	697	560	646	454	492	437	513	443	763	647	656	498	471	451	509	486	453	699	389	000	476	488	500
30	589	464	710	447	453	475	463	470	482	793	631	729	413	545	509	481	468	694	721	729	485	379	490	465	344	366	634	478	476	000	466	479
31	589	383	661	378	393	400	382	392	327	775	639	731	461	582	426	589	387	709	729	739	549	451	416	569	467	451	663	409	488	466	000	541
32	540	532	753	568	562	527	573	526	546	656	464	571	448	539	493	379	503	874	547	551	344	469	528	372	486	471	805	525	500	479	541	000

Average badness $\delta = 0.2429$;

it approximately corresponds to the triangle with the sides 10.5, 9.5, and 8.5.

International Journal of Open Information Technologies, ISSN: 2307-8162 vol. 12, no. 9, 2024

Таблица IV
The matrix obtained by applying the Needleman – Wunsch's algorithm
to 32 species of monkeys (no more than one species from each genus)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
1	000	250	375	260	253	256	283	253	277	156	143	192	197	157	274	216	253	477	206	204	154	187	284	161	188	192	381	263	256	192	281	153
2	250	000	293	184	168	168	267	167	265	250	253	287	258	256	264	240	123	473	289	289	246	247	275	254	245	249	473	180	179	251	267	249
3	375	293	000	322	323	320	371	320	368	373	377	476	380	375	384	375	286	476	474	476	374	372	383	378	377	376	296	327	329	381	372	374
4	260	184	322	000	191	191	271	189	270	258	263	295	264	264	271	258	182	476	297	298	257	258	278	265	258	259	405	196	199	259	270	261
5	253	168	323	191	000	146	268	145	265	255	258	289	259	259	272	251	169	474	292	293	253	253	276	260	250	250	472	169	184	251	269	256
6	256	168	320	191	146	000	276	091	271	254	254	286	261	259	271	249	165	477	286	287	252	253	274	255	255	255	474	163	180	256	273	253
7	283	267	371	272	268	276	000	272	152	283	287	319	286	285	255	279	266	477	319	320	281	277	253	289	276	275	406	273	278	282	177	281
8	253	167	320	189	145	091	272	000	266	251	253	286	257	257	269	247	165	474	289	288	251	254	275	256	253	253	471	163	181	255	269	254
9	277	265	368	270	266	271	152	266	000	275	277	312	279	276	250	272	263	477	311	313	275	271	250	282	272	271	402	270	273	275	172	275
10	156	250	373	258	255	254	283	251	275	000	159	201	202	173	275	212	249	477	191	191	084	190	279	153	191	196	377	260	253	192	279	148
11	143	253	377	263	258	254	287	253	277	159	000	174	202	140	273	215	251	480	205	202	153	191	280	162	191	193	384	264	258	194	283	156
12	192	287	476	295	289	286	319	286	312	201	174	000	244	193	301	246	285	479	160	157	201	236	312	203	235	234	478	291	287	237	316	200
13	197	258	380	264	259	261	286	257	279	202	202	244	000	209	281	227	256	478	246	245	197	167	284	200	176	174	363	266	267	174	283	197
14	157	256	375	264	259	259	285	257	276	173	140	193	209	000	278	225	258	478	221	219	169	200	287	179	200	206	378	270	265	207	282	173
15	274	264	384	271	272	271	255	269	250	275	273	301	281	278	000	267	266	476	301	301	274	273	202	277	273	273	476	271	275	278	249	272
16	216	240	375	258	251	249	279	247	272	212	215	246	227	225	267	000	244	481	245	245	208	217	275	216	219	222	399	254	250	222	275	210
17	253	123	286	182	169	165	266	165	263	249	251	285	256	259	266	245	000	473	288	289	247	248	272	252	252	250	407	179	179	251	267	247
18	477	472	476	476	474	477	476	474	477	477	480	479	478	478	476	482	473	000	480	481	478	479	477	478	475	476	476	477	477	479	474	479
19	206	289	474	297	292	286	319	289	311	191	205	160	246	221	301	245	288	480	000	077	189	234	311	200	237	237	477	296	290	239	317	199
20	204	289	475	298	293	287	320	288	313	191	202	157	245	219	301	245	289	481	077	000	189	236	312	196	236	236	478	293	288	241	316	195
21	154	246	374	257	253	252	281	251	275	084	153	201	197	169	274	208	247	477	189	189	000	185	281	146	187	190	379	256	253	190	276	141
22	187	247	372	258	254	253	277	254	271	190	191	236	167	200	273	217	248	479	234	236	185	000	279	193	142	129	336	264	257	145	271	187
23	284	275	383	278	276	274	253	275	250	279	280	312	284	287	202	275	272	477	311	312	281	279	000	287	282	281	476	276	279	284	253	282
24	161	254	378	265	260	255	289	256	282	153	162	203	200	179	277	216	252	479	200	196	146	193	286	000	199	197	382	264	260	196	286	095
25	188	245	377	258	250	255	276	253	272	191	192	235	176	200	273	219	252	474	237	236	187	142	282	199	000	148	348	267	256	148	272	192
26	192	249	376	259	250	255	275	253	271	196	193	234	174	206	273	222	250	477	237	236	190	129	281	197	148	000	339	264	256	153	276	192
27	381	473	296	405	472	474	406	471	402	377	384	478	363	378	475	399	407	476	477	478	379	336	476	382	348	339	000	477	471	352	403	380
28	263	180	327	196	169	163	273	163	270	260	264	291	266	270	270	254	179	477	296	293	256	264	276	264	267	264	477	000	190	265	273	259
29	256	179	329	199	184	180	278	181	273	253	258	286	267	265	275	250	179	477	290	288	253	257	279	260	256	256	472	190	000	261	275	255
30	192	251	380	259	251	256	282	254	275	192	194	237	174	207	278	222	251	480	239	241	190	145	284	196	148	153	352	265	261	000	279	195
31	281	267	372	270	269	273	177	269	172	280	283	316	283	282	249	275	267	475	317	316	276	272	253	286	272	276	403	273	275	279	000	279
32	153	249	374	261	256	253	281	254	275	148	156	200	197	173	272	210	247	479	199	195	141	187	282	095	192	192	380	259	255	195	279	000

Average badness $\delta = 0.2233$;

it approximately corresponds to the triangle with the sides 11, 10, and 9.

International Journal of Open Information Technologies, ISSN: 2307-8162 vol. 12, no. 9, 2024

Мы суммируем все значения badness всех возможных треугольников матрицы, рис. 3.

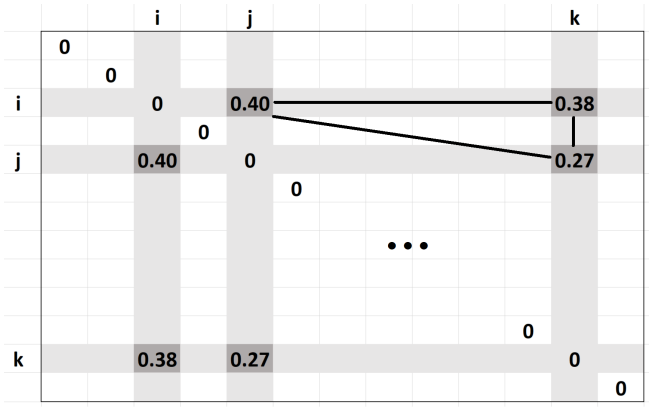

Рис. 3. Один из треугольников матрицы расстояний

Однако такое суммирование (и различные варианты обработки полученных таким образом результатов, в частности, алгоритмы восстановления неполностью заполненной матрицы) – это предмет нескольких предыдущих статей одного из авторов.

Как уже было отмечено, основной предмет *последующих частей настоящей статьи* – описание инструментария для подсчёта парной корреляции между значениями badness для последовательностей треугольников, полученных на основе матриц расстояний между цепочками мт ДНК. Мы имеем в виду не только полученные в вычислительных экспериментах настоящей работы таблицы III и IV, но и любые матрицы, которые могут быть получены таким же способом – для любой пары алгоритмов определения близости двух цепочек ДНК и для любого множества видов.

БЛАГОДАРНОСТИ

Настоящая работа была частично поддержана грантом научной программы китайских университетов “Higher Education Stability Support Program” (раздел “Shenzhen 2022 – Science, Technology and Innovation Commission of Shenzhen Municipality”) – 深圳市 2022 年高等院校稳定支持计划资助项目.

Список литературы

[1] Мельников Б.Ф., Панин А.Г. Параллельная реализация мультиэвристического подхода в задаче сравнения генетических последовательностей // Вектор науки Тольяттинского государственного университета. 2022. № 4 (22). С. 83–86.

[2] Мельников Б.Ф., Тренина М.А., Кочергин А.С. Подход к улучшению алгоритмов расчета расстояний между цепочками ДНК (на примере алгоритма Нидлмана – Вунша) // Известия высших учебных заведений. Поволжский регион. Физико-математические науки. 2018. № 1 (45). С. 46-59.

[3] Мельников Б.Ф., Тренина М.А. Об одной задаче восстановления матриц расстояний между цепочками ДНК // International Journal of Open Information Technologies. 2018. Vol. 6, No. 6. P. 1–13.

[4] Абрамян М.Э., Мельников Б.Ф., Тренина М.А. Реализация метода ветвей и границ для задачи восстановления матрицы расстояний между последовательностями ДНК // Современные информационные технологии и ИТ-образование. 2019. Т. 15, № 1. С. 81–91.

[5] Melnikov B., Chaikovskii D. Some general heuristics in the traveling salesman problem and the problem of reconstructing the DNA chain distance matrix // ACM International Conference Proceeding Series. 2023. P. 361–368.

[6] Abramyan M., Melnikov B., Zhang Y. Some more on restoring distance matrices between DNA chains: reliability coefficients // Cybernetics and Physics. 2023. Vol. 12, No. 4. P. 237–251.

[7] Melnikov B., Chaikovskii D. On the Application of Heuristics of the TSP for the Task of Restoring the DNA Matrix // Frontiers in Artificial Intelligence and Applications. 2024. Vol. 385. P. 36–44.

[8] Needleman S., Wunsch Ch. A general method is applicable to the search for similarities in the amino acid sequence of two proteins // Journal of Molecular Biology. 1970. Vol. 48, No. 3. P. 443–453.

[9] Winkler W. String comparator metrics and enhanced decision rules in the Fellegi – Sunter model of record linkage // Proceedings of the Survey Research Methods Sections, American Statistical Association. 1990. P. 354–359.

[10] NCBI: nucleotide database. URL: http://www.ncbi.nlm.nih.gov/nuccore.

[11] Melnikov B., Pivneva S., Trifonov M. Various algorithms, calculating distances of DNA sequences, and some computational recommendations for use such algorithms // CEUR Workshop Proceedings. 2017. Vol. 1902. P. 43–47.

ЛИ Цзямянь,
аспирант Университета МГУ – ППИ в Шэньчжэне
(http://szmsubit.ru/),
email: lijiamian0804@live.com.

МУ Цзинъюань,
аспирант Университета МГУ – ППИ в Шэньчжэне
(http://szmsubit.ru/),
email: xirousang@gmail.com.

Борис Феликсович МЕЛЬНИКОВ,
профессор Университета МГУ – ППИ в Шэньчжэне
(http://szmsubit.ru/),
email$_1$: bormel@smbu.edu.cn,
email$_2$: bf-melnikov@yandex.ru,
mathnet.ru: personid=27967,
elibrary.ru: authorid=15715,
scopus.com: authorId=55954040300,
ORCID: orcidID=0000-0002-6765-6800.

On an approach to the implementation of the Needleman – Wunsch and Jaro – Winkler algorithms and their application in the correlation analysis of the similarity of mitochondrial DNA of monkeys. Part I

Li Jiamian, Mu Jingyuan, Boris Melnikov

***Abstract*—In molecular biology and genomics research, it is very important to understand the genetic differences between different species. Comparing the similarity of DNA sequences can provide valuable information about the relationships between species.**

In this paper, two algorithms were used to compare the mitochondrial DNA of monkeys, i.e., Needleman – Wunsch and Jaro – Winkler algorithms. In addition, in the following parts of the paper, similar comparisons will be made for other mammals.

Earlier, when conducting such studies, the authors of this paper had a following hypothesis. When using these two algorithms to analyze the similarity of the same pairs of genomic sequences, very different results are obtained. One of the subjects of this paper is a description of the approach to how exactly we propose to give numerical answers to such questions. We propose to give such answers using the use of pair correlation, which will be discussed in the following parts of the paper.

From the results of this paper, it follows that it is necessary to continue detailed studies of DNA chains, in particular, to analyze their similarity. That is, such problems remain and will remain very relevant for a long time.

***Keywords*—DNA sequences, Needleman–Wunsch algorithm, Jaro–Winkler algorithm, distance matrix, correlation analysis.**

References

[1] Melnikov B., Panin A. Parallel implementation of the multiheuristic approach in the task of comparing genetic sequences // Vector of science of Tolyatti State University. 2022. No. 4 (22). P. 83–86 (in Russian).

[2] Melnikov B., Trenina M., Kochergin A. An approach to improving algorithms for calculating distances between DNA chains (using the Needleman – Wunsch algorithm as an example) // News of higher educational institutions. Volga region. Physical and mathematical sciences. 2018. No. 1 (45). P. 46-59 (in Russian).

[3] Melnikov B., Trenina M. On a problem of reconstructing distance matrices between DNA chains // International Journal of Open Information Technologies. 2018. Vol. 6, No. 6. P. 1–13 (in Russian).

[4] Abramyan M., Melnikov B., Trenina M. Implementation of the branch and boundary method for the task of reconstructing the matrix of distances between DNA sequences // Modern information technologies and IT education. 2019. Vol. 15, No 1. P. 81–91 (in Russian).

[5] Melnikov B., Chaikovskii D. Some general heuristics in the traveling salesman problem and the problem of reconstructing the DNA chain distance matrix // ACM International Conference Proceeding Series. 2023. P. 361–368.

[6] Abramyan M., Melnikov B., Zhang Y. Some more on restoring distance matrices between DNA chains: reliability coefficients // Cybernetics and Physics. 2023. Vol. 12, No. 4. P. 237–251.

[7] Melnikov B., Chaikovskii D. On the Application of Heuristics of the TSP for the Task of Restoring the DNA Matrix // Frontiers in Artificial Intelligence and Applications. 2024. Vol. 385. P. 36–44.

[8] Needleman S., Wunsch Ch. A general method is applicable to the search for similarities in the amino acid sequence of two proteins // Journal of Molecular Biology. 1970. Vol. 48, No. 3. P. 443–453.

[9] Winkler W. String comparator metrics and enhanced decision rules in the Fellegi – Sunter model of record linkage // Proceedings of the Survey Research Methods Sections, American Statistical Association. 1990. P. 354–359.

[10] NCBI: nucleotide database. URL: `http://www.ncbi.nlm.nih.gov/nuccore`.

[11] Melnikov B., Pivneva S., Trifonov M. Various algorithms, calculating distances of DNA sequences, and some computational recommendations for use such algorithms // CEUR Workshop Proceedings. 2017. Vol. 1902. P. 43–47.

LI Jiamian,
Post-graduate student of Shenzhen MSU – BIT University, China (`http://szmsubit.ru/`),
email: `lijiamian0804@live.com`.

MU Jingyuan,
Post-graduate student of Shenzhen MSU – BIT University, China (`http://szmsubit.ru/`),
email: `xirousang@gmail.com`.

Boris MELNIKOV,
Professor of Shenzhen MSU – BIT University, China (`http://szmsubit.ru/`),
email$_1$: `bormel@smbu.edu.cn`,
email$_2$: `bf-melnikov@yandex.ru`,
mathnet.ru: `personid=27967`,
elibrary.ru: `authorid=15715`,
scopus.com: `authorId=55954040300`,
ORCID: `orcidID=0000-0002-6765-6800`.