Об одной задаче восстановления матриц расстояний между цепочками ДНК

Б. Ф. Мельников, М. А. Тренина

Аннотация—**На практике достаточно часто встречается необходимость вычисления специальным образом определённых расстояний между последовательностями различной природы. Подобные алгоритмы используются в биоинформатике для сравнения секвенированных генетических цепочек. В силу большой размерности таких цепочек приходится использовать эвристические алгоритмы, которые дают приближённые результаты.**

Существуют различные эвристические алгоритмы определения расстояния между геномами, но очевидным недостатком при расчёте расстояния между одной и той же парой строк ДНК является получение несколько различающихся результатов при использовании различных алгоритмов для расчёта метрик. Поэтому возникает задача оценки качества используемых метрик (расстояний), по результатам которой можно сделать вывод о применимости алгоритма к различным исследованиям.

Кроме того, одной из рассматриваемых в биокибернетике задач является задача восстановления матрицы расстояний между последовательностями ДНК, когда на входе алгоритма известны не все элементы рассматриваемой матрицы. В связи с этим возникает задача, заключающаяся в этом, чтобы разработанный метод сравнительной оценки алгоритмов расчёта расстояний между последовательностями использовать для иной задачи – для восстановления матрицы расстояний между последовательностями ДНК.

В настоящей статье мы рассматриваем возможность применения разработанного и исследованного нами ранее метода сравнительной оценки алгоритмов расчёта расстояний между парой строк ДНК для восстановления частично заполненной матрицы расстояний. Восстановление матрицы происходит в результате осуществления нескольких вычислительных проходов. Оценки неизвестных элементов матрицы специальным образом усредняются с применением т. н. функции риска, и результат этого усреднения рассматривается как получаемое значение неизвестного элемента.

Ключевые слова—**последовательности ДНК, метрика, матрица расстояний, частично заполненная матрица, восстановление, функции риска.**

I. Введение

На практике достаточно часто встречается необходимость вычисления специальным образом определённых расстояний между последовательностями различной природы. Подобные алгоритмы часто применяются и в биоинформатике, они составляют отдельный, очень важный вид задач – поиска расстояния между заданными генетическими последовательностями. Основной сложностью, возникающей при вычислении расстояния между генетическими последовательностями, является очень большая длина такой последовательности. Например, даже для очень коротких митохондриальных ДНК человека (мДНК) длина последовательности превышает 16000 символов, а для обычной ДНК может превышать $3 \cdot 10^8$ символов, [1].

В силу этого алгоритмы, вычисляющие точное значение расстояния между двумя последовательностями, являются неприменимыми, а для оценки расстояния между такими цепочками приходится использовать *эвристические* алгоритмы [2], [3], [4], которые дают приближённые результаты. При этом даже подобные эвристические алгоритмы требуют больших временны́х затрат: например, для построения матрицы порядка 50×50, в которую записываются расстояния, вычисляемые алгоритмом Нидлмана-Вунша, требуется около 28 часов (при тактовой частоте процессора порядка 2 ГГц, см. [5]).

Итак, для определения расстояния между геномами нам необходимы эвристические алгоритмы – причём, по возможности, не требующие слишком больших временны́х затрат. Существуют различные подобные алгоритмы, но очевидным их недостатком является получение несколько *различающихся результатов* при использовании различных эвристических алгоритмов, применённых к подчёту расстояния между *одной и той же парой строк ДНК*. Поэтому возникает задача *оценки качества* используемых метрик (расстояний) – и по результатам, полученным при решении этой задачи, можно делать выводы о применимости конкретного алгоритма подсчёта расстояний к различным прикладным исследованиям. Возможный подход к определению оценки качества метрик был приведён в [5].

Кроме того, одной из рассматриваемых в биокибернетике задач является задача *восстановления* матрицы расстояний между последовательностями ДНК (ниже – просто матрицы ДНК), в которой на входе алгоритма известны *не все* элементы рассматриваемой матрицы [6], [7]. В связи с этим возникает другая задача: использовать разработанный метод сравнительной оценки алгоритмов расчёта расстояний между последовательностями для совершенно иной цели, а именно – для кратко описываемой нами далее *задачи восстановления матрицы расстояний* между последовательностями ДНК. Для этой задачи мы в настоящей статье рассматриваем применение разработанного и исследованного нами ранее *метода сравнительной оценки алгоритмов расчёта расстояний* между парой строк ДНК.

При таком подходе (т. е. при применение метода сравнительной оценки алгоритмов расчёта расстояний к восстановлению матриц) само восстановление происходит в результате осуществления нескольких вычислитель-

Статья получена 3 мая 2018.

Борис Феликсович Мельников, Российский государственный социальный университет (email: bf-melnikov@yandex.ru).

Марина Анатольевна Тренина, Тольяттинский государственный университет (email: trenina.m.a@yandex.ru).

ных проходов. На каждом из проходов для некоторых пока незаполненных (неизвестных) элементов матрицы получаются разные оценки; эти оценки специальным образом усредняются – и результат усреднения берётся в качестве значения неизвестного элемента. С физической точки зрения применяемое усреднение даёт положение центра тяжести одномерной системы тел, масса которых задается специальной функцией – функцией риска [8], [9]. Отметим, что ранее функции риска применялись нами в совершенно иных предметных областях – причём всегда были связаны со вспомогательными алгоритмами, относящимися к многокритериальной оптимизации.

Ниже рассматриваемые матрицы, подлежащие восстановлению, мы будем называть *неполностью заполненными матрицами расстояний*. Мы *вводим этот термин для матрицы*, из которой «вычеркнуто» некоторое количество элементов; при этом схожее понятие «разрежённая матрица» не полностью отражает смысл рассматриваемой нами задачи. Мы будем писать термин «неполностью» в одно слово – аналогично «труднорешаемым задачам» в русском переводе монографии [10].

Настоящая статья имеет следующую структуру. В разделе II при описании предварительных све́дений мы приводим возможный подход к реализации вышеупомянутого алгоритма Нидлмана – Вунша.

В разделах III и IV описывается метод восстановления неполностью заполненной матрицы ДНК, разработанный на основе исследуемой раннее методики сравнительного анализа различных алгоритмов вычисления расстояний между последовательностями ДНК. Конкретно, в разделе III приводятся его неформальные обоснования, а в разделе IV формально описывается сам алгоритм восстановления. В нём для вычисления неизвестных элементов матрицы в качестве вспомогательного алгоритма используются функции риска.

Далее мы рассматриваем примеры применения алгоритма. В разделе V приводятся подробные результаты вычислительного эксперимента для малой размерности (матрица 7×7). А в разделе VI даётся краткое описание результатов вычислительного эксперимента для матрицы существенно большей размерности (28×28), а также приведена оценка полученных результатов для различных способов восстановления матрицы ДНК. Мы считаем, что наш подход правилен, в связи со следующим фактом: получаемое нами значение невязки – т.е. тривиальным образом определённого расстояния между двумя матрицами, исходной и восстановленной по неполным данным, – очень мало.

В заключении (раздел VII) даётся краткое обобщение проделанной работы, а также перечислены направления дальнейших исследований в данном направлении.

II. Предварительные све́дения. Подход к реализации алгоритма Нидлмана – Вунша

Алгоритм Нидлмана – Вунша [11] выполняется путём *выравнивания* двух последовательностей символов. Он представляет из себя пример динамического программирования ([12, стр. 299] и др.) и является, по-видимому, первым описанным в литературе приложением динамического программирования к сравнению биологических последовательностей.

Наш вариант этого алгоритма заключается в следующем. Как и в других его интерпретациях, 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

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

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 делецию (штраф 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.

(Ещё отметим, что описание одного из относительно недавно опубликованных удачных алгоритмов подсчёта расстояния между двумя последовательностями ДНК можно найти по ссылке [13]. В настоящее время нами ведутся работы по сравнению двух этих алгоритмов – причём не только для их применения в «обычных» задачах ДНК-анализа, но и в задачах восстановления матриц ДНК, рассматриваемых в настоящей работе.)

III. Об одном методе восстановления матрицы ДНК

В этом разделе приводится один из методов сравнительного анализа различных алгоритмов вычисления расстояний между последовательностями ДНК, и на его основе разрабатывается *метод восстановления неполностью заполненной матрицы*. С целью проведения этого сравнительного анализа мы предлагаем для полученной в результате работы какого-либо алгоритма вычисления расстояний между геномами рассматривать все возможные треугольники, потому что в идеале они должны быть остроугольными равнобедренными.

Предположение о том, что треугольники должны быть остроугольными равнобедренными, возникает на основе примерно таких рассуждений, см. [14]. Согласно данным биологов, шимпанзе (Ш) и бонобо (Б) имели общего предка, жившего около 2–2.5 млн. лет назад, а человек (Ч) с ними обоими разошелся 5.5–7 млн. лет назад, см. [15] и др. В связи с этим возникает вопрос: почему Ч должен быть ближе к Б чем к Ш? Или наоборот – почему он должен быть ближе к Ш чем к Б? Очевидно, что ответ на оба этих вопроса отрицательный, т. е., иными словами, объяснения большей близости существовать не может.

Для ответа на вопрос, насколько «правильной» является матрица, полученная в результате некоторого эвристического алгоритма, мы предлагаем использовать «харак-

теристику отхода» полученных треугольников от «вытянутых равнобедренных» треугольников – т. н. “badness”, ниже будем писать без кавычек. При этом в качестве одного из вариантов badness может использоваться формула

$$\sigma = \frac{\alpha - \beta}{\gamma} \tag{2}$$

где α, β и γ – углы треугольника, причём мы предполагаем, что $\alpha \geq \beta \geq \gamma$ [14]. По мнению авторов настоящей статьи, эта формула наилучшим образом характеризует описанные нами требования (упрощая ситуацию – «насколько» остроугольным равнобедренным является рассматриваемый нами треугольник). Приведём *неформальное* объяснение этого: чем ближе треугольник к равнобедренному, тем меньше у него разность между α и β, и в идеальном случае в числителе получается 0; при этом, согласно сделанным нами допущениям, тупоугольного (или прямоугольного) равнобедренного треугольника получаться не может. Выполнение же свойства остроугольности увеличивает знаменатель. Следовательно, приближение треугольника к равнобедренному треугольному уменьшает в формуле числитель и увеличивает знаменатель, т. е. σ стремится к нулю.

В настоящее время авторы разрабатывают и другие подходы (другие формулы) для вычисления подобных характеристик матрицы ДНК, альтернативных рассматриваемой здесь характеристике σ. В одной из принятых к публикации статей мы рассматриваем подход к сравнению таких характеристик – однако отметим, что рассматриваемая в настоящей статье задача (задача восстановления неполностью заполненных матриц), несомненно, значительно более важна.

При расчёте badness всей матрицы для каждого варианта восстановления можно:

- либо суммировать соответствующие badness по всем возможным треугольникам рассматриваемых матриц;
- либо взять максимальную badness по этим треугольникам.

В дальнейшем мы предполагаем рассмотреть и другие подходы к вычислению badness всей матрицы.

Однако при расчёте этого показателя (badness всей матрицы) может оказаться некоторое (на практике – совсем небольшое) количество треугольников, для которых значение badness может значительно отличаться от других. В частности, могут получаться треугольники, у которых badness равна 1. (Очень редко – конечно, при приемлемых метриках – получаются даже тупоугольные треугольники. Для них мы в практических задачах полагаем значение badness больше 1, и конкретное значение зависит от величины тупого угла.)

Исходя из всего этого, для вычисления badness всей матрицы мы используем специальное усреднение. С физической точки зрения применяемое усреднение даёт положение центра тяжести одномерной системы тел, масса которых задаётся специальной функцией – так называемой функцией риска, см. [16]. Badness для всех треугольников определяет координаты тел, а функция риска – их массы, при этом, чем больше координата, тем меньше ее масса (т. е. чем больше badness, тем меньше его вклад).

Итак, как уже было отмечено выше, мы считаем, что в правильно заполненной матрице расстояний между последовательностями ДНК все возможные построенные треугольники должны быть максимально близкими к равнобедренным остроугольным, и тогда на основании этого вывода можно произвести восстановление матрицы расстояний между строк ДНК, которая сначала имеет некоторое количество неизвестных элементов. Заполнением таких матриц мы и будем заниматься далее.

IV. Строгое описание алгоритма восстановления

Для определения неизвестного элемента мы рассматриваем все возможные треугольники, образованные из элементов этой матрицы, для которых одна из сторон неизвестна. Для каждого такого треугольника из того условия, что он является равнобедренным остроугольным, мы получаем *одно из возможных значений* этой неизвестной стороны. Далее мы специальным образом вычисляем окончательное значение этой стороны (неизвестного элемента). А именно, для её вычисления на основе всех полученных оценок элемент полагается равным среднему арифметическому всех полученных значений; в качестве альтернативного варианта мы можем исключать наибольшее и наименьшее из получаемых значений [1].

При большом количестве пропущенных элементов матрицы треугольников с двумя известными сторонами будет немного, поэтому восстановление матрицы за один проход обычно невозможно. При восстановлении матрицы на втором и последующих проходах можно либо использовать только элементы матрицы последнего прохода, либо же воспользоваться всеми матрицами, полученными на предыдущих проходах. Во втором случае с каждым последующим проходом в матрице становится всё больше элементов, вычисленных приближённо. Поэтому при оценке неизвестного элемента возможно применение аналога функции риска, которая будет корректировать вес элементов *в зависимости от номера прохода*.

При использовании т. н. *статической* функции риска вес элементов с каждым проходом уменьшается с одинаковым коэффициентом, и для оценки неизвестного элемента матрицы используется формула

$$E = \frac{c_0 E_0 + c_1 E_1 + \ldots + c_k E_k}{c_0 + c_1 + \ldots + c_k}, \tag{3}$$

где:

- E_i – где значение элемента матрицы, полученной на i-м проходе;
- $c_0, \ldots, c_k$ – некоторые специально подбираемые коэффициенты.

На практике [17] хорошие результаты достигаются, когда для коэффициентов используются формулы

$$c_0 = 1, \quad c_i = p\, c_{i-1}. \tag{4}$$

Согласно [8], [9], [18], функция риска может быть и *динамической*: при использовании последней мы берём усреднение, зависящее от «черновой прикидки» итогового значения: является ли оно «хорошим», «средним» или

[1] Мы также можем использовать функции риска, о которых кратко было сказано выше и немного подробнее будет сказано далее.

«плохим». Кроме того, можем рассматривать и *последовательность* таких динамических функций риска, где на каждом этапе мы в качестве такой «черновой прикидки» опираемся на значение, полученное на предыдущем шаге. В нашем случае для оценки неизвестного элемента матрицы расстояний между строк ДНК используется формула

$$\frac{\sum_{i=1}^{k} a_i f(a_i)}{\sum_{i=1}^{k} f(a_i)}, \tag{5}$$

где $f(x)$ – некоторая специальным образом выбранная убывающая функция.

Алгоритм 1 (Восстановление матрицы с помощью статической функции риска)

Вход: Неполностью определенная матрица $A = a_{ij}$ (все равные нулю элементы вне главной диагонали считаем неизвестными).

Использующиеся вспомогательные переменные: b_i – массив оценок неизвестного элемента.

Описание алгоритма.

Шаг 1: Устанавливаем $s := 1$ – номер прохода.

Шаг 2: Вычисляем h – количество элементов верхнего треугольника, равных нулю.

Шаг 3:
if $a_{ij} = 0$ and $i \neq j$ then
  begin
    $kol := 0$ {считаем количество треугольников, построенных на неизвестном элементе}
    for $k := 0$ to n do begin
      if $k \neq i$ and $k \neq j$ and $a_{ki} \neq 0$ and $a_{kj} \neq 0$ then
        begin
          $kol := kol + 1$; $c_0 := 1$; $c_s := c_{s-1} \cdot p$;

$$E_{ki} := \frac{c_0 E_{ki}^0 + \ldots + c_s E_{ki}^s}{c_0 + \ldots + c_s};$$

$$E_{kj} := \frac{c_0 E_{kj}^0 + \ldots + c_s E_{kj}^s}{c_0 + \ldots + c_s};$$

          if $E_{ki} > E_{kj}$ then $b_{kol} := E_{ki}$ else $b_{kol} := E_{kj}$
        end;
      end;
  end;

$$a_{ij} := \frac{b_1 + \ldots + b_{kol}}{kol}.$$

Шаг 4: Вычисляем h_1 – количество элементов верхнего треугольника, равных нулю после очередного прохода.

Шаг 5:
if $h_1 = 0$ then выход 1;
if $h_1 = h$ then выход 2;
$s := s + 1$;
переход к шагу 2.

Выход 1: Заполненная матрица A.

Выход 2: Матрицу A восстановить невозможно. □

После выполнения алгоритма для проведения сравнительного анализа результатов восстановления матрицы мы используем такой показатель, как невязка; он характеризует отклонение полученной матрицы от исходной. Мы вычисляем невязку на основе естественной метрики

$$d = \frac{\sqrt{\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} (a_{ij} - \widetilde{a_{ij}})^2}}{n(n-1)/2}, \tag{6}$$

где:

- $\widetilde{a_{ij}}$ – элементы матрицы, полученной в результате применения некоторого алгоритма подсчёта расстояний между парой геномов (в нашем случае – алгоритма Нидлмана – Вунша);
- a_{ij} – элементы матрицы, восстановленной в результате работы вышеописанного алгоритма.

V. Подробное описание примера работы с матрицей малой размерности

Оба рассматриваемых нами примера (в этом разделе – для малой размерности 7, а также в следующем, для размерности 28) работают с матрицами, полученными применением алгоритмом Нидлмана – Вунша [19]. Мы применили этот алгоритм к цепочкам мДНК различных животных, взятых из банка данных NCBI [20]; при этом были взяты секвенированные цепочки мДНК для одного представителя каждого из 28 отрядов млекопитающих (классификацию млекопитающих выбираем согласно [1], другие варианты классификации не рассматриваем). В таблице 14 приложения перечислены все выбранные нами виды животных. Повторим, что в настоящем разделе подробно рассматривается пример для первых семи элементов этой таблицы.

Результаты применения рассматриваемых нами алгоритмов (в частности – алгоритма Нидлмана – Вунша) обычно выражаются в «процентах близости». Согласно же нашим предыдущим работам ([14], [19], [21] и др.), нам необходима схожая характеристика («относительная удалённость») – получающаяся вычитанием полученного «процента близости» из 100 и делением на 100. В настоящем разделе в таблицах мы будем использовать эти значения – а в следующих разделах, для бóльших размерностей, данные для удобства в таблицах (см. приложение) будут обозначаться целыми числами, образованными первыми тремя значащими цифрами этих значений.

Все рассматриваемые нами матрицы являются симметричными относительно главной диагонали, поэтому здесь и далее будем говорить об удалении элементов из верхнего треугольника. Также будем говорить о «проценте обнуления» матрицы.

Итак, сначала мы рассматриваем пример очень малой размерности. Конечно, количество элементов, которые можно убрать из матрицы, зависит от её размера. Применение вычислительных экспериментов показывает, что для матрицы порядка 7×7 *приемлемое* восстановление обычно возможно в том случае, когда процент обнуления матрицы не превосходит 70. (В реальных задачах мы иногда увеличивали это значение до 80%.)

Для примера возьмем квадратную матрицу 7×7 (таблица 1). Отметим ещё раз, что здесь значение 0 соответствует минимально возможному расстоянию между геномами (их совпадению), а значение 1 (по-видимому, недостижимое для реальных геномов) соответствовало *бы* максимально возможному расстоянию между ними

(их «полному несовпадению» – которое определяется поразному для различных алгоритмов определения расстояния между геномами).

Таблица 1. Исходная матрица порядка 7

0	0,299	0,258	0,27	0,3149	0,324	0,285
0,299	0	0,369	0,298	0,2399	0,209	0,3014
0,258	0,369	0	0,292	0,3432	0,339	0,3579
0,27	0,298	0,292	0	0,2936	0,348	0,2923
0,315	0,24	0,343	0,294	0	0,217	0,3177
0,324	0,209	0,339	0,348	0,2168	0	0,3411
0,285	0,301	0.3579	0,292	0,3177	0,341	0

Далее. Из этой матрицы мы удаляем примерно 60% элементов – что для 21 элемента, лежащего выше главной диагонали, составляет 13 удаляемых элементов; оставляем в матрице примерно 40% элементов. *Возможный* (обрабатываемый нами) вариант удаления приведён в таблице 2.

Таблица 2. Неполностью заполненная матрица порядка 7

0	0	0	0	0	0	0,285
0	0	0,369	0	0,2399	0,209	0
0	0,369	0	0	0	0	0
0	0	0	0	0,2936	0,348	0,2923
0	0,24	0	0,294	0	0	0,3177
0	0,209	0	0,348	0	0	0
0,285	0	0	0,292	0,3177	0	0

Будем вычислять неизвестные элементы по всей матрице (найденные значения будем также дублировать, отображая относительно главной диагонали). Например, для неизвестного элемента a_{12} мы рассматриваем все возможные значения k, такие что k отлично от 1 и 2, а элементы a_{k1} и a_{k2} известны. (Далее аналогичные действия будут произведены и для остальных неизвестных элементов a_{ij}.)

В нашем примере не нашлось ни одного значения k, удовлетворяющего этому условию; аналогичная ситуация и для элемента a_{13}. А для следующего элемента a_{14} нашлось единственное значение $k = 7$, удовлетворяющее необходимому условию. По элементам $a_{71} = 0,285$ и $a_{74} = 0,292$ с учётом того, что получающийся треугольник должен быть остроугольным равнобедренным, получаем, что $a_{14} = 0,292$.

Далее. Для элемента a_{15} значение k также равно 7, и из двух чисел $a_{71} = 0,285$ и $a_{75} = 0,3177$ получаем, что $a_{15} = 0,3177$. По мере заполнения матрицы количество возможных значений k будет увеличиваться. Например, для элемента a_{16} не удалось подобрать k, однако для симметричного ему элемента a_{61} (при рассмотрении элементов последовательно по строкам) таких значений имеется несколько.

Результат, получающийся после первого прохода, приведён ниже в таблице 3. Отметим, что после первого прохода осталось небольшое количество элементов, равных 0.

Таблица 3. Матрица, полученная после первого прохода

0	0	0	0,292	0,3177	0,334	0,285
0	0	0,369	0,321	0,2399	0,209	0,3177
0	0,369	0	0,369	0,3691	0,369	0,3691
0,292	0,321	0,369	0	0,2936	0,348	0,2923
0,318	0,24	0,369	0,294	0	0,319	0,3177
0,334	0,209	0,369	0,348	0,319	0	0,3385
0,285	0,318	0,369	0,292	0,3177	0,339	0

Как уже отмечалось, при вычислении на втором проходе для каждого неизвестного элементов подходящих значений k будет больше. Например, для элемента a_{12} получаем следующие варианты:

$$k = 4: \quad a_{14} = 0,2924,\ a_{24} = 0,3209 \ \Rightarrow\ b_1 = 0,3209;$$

$$k = 5: \quad a_{15} = 0,3177,\ a_{25} = 0,2399 \ \Rightarrow\ b_2 = 0,3177;$$

$$k = 6: \quad a_{16} = 0,3336,\ a_{26} = 0,2089 \ \Rightarrow\ b_3 = 0,3336;$$

$$k = 7: \quad a_{17} = 0,285,\ a_{27} = 0,3177 \ \Rightarrow\ b_4 = 0,3177.$$

На основе всех полученных потенциальных значений a_{12} вычисляем среднее арифметическое и получаем присваиваемое значение $a_{12} = 0,3224$.

Далее проделываем аналогичные действия для элемента a_{13}, после чего получаем следующую матрицу:

Таблица 4. Восстановленная матрица порядка 7

0	0,322	0,369	0,292	0,3177	0,334	0,285
0,322	0	0,369	0,321	0,2399	0,209	0,3177
0,369	0,369	0	0,369	0,3691	0,369	0,3691
0,292	0,321	0,369	0	0,2936	0,348	0,2923
0,318	0,24	0,369	0,294	0	0,319	0,3177
0,334	0,209	0,369	0,348	0,319	0	0,3385
0,285	0,318	0,369	0,292	0,3177	0,339	0

Как было сказано выше, для анализа результатов восстановления матрицы мы вычисляем заданную обычным образом невязку. Для рассматриваемого примера получаемое значение невязки равно $d = 0,0015$, что, по-видимому, неплохо: отношение невязки к среднему значению элемента матрицы менее $0,005$ (меньше, чем полпроцента).

VI. Результаты восстановления матрицы ДНК

В этом разделе представлены результаты вычислительного эксперимента большей размерности. Как и в предыдущем разделе, мы выбираем по одному геному из 28 отрядов млекопитающих [1] (таблица 14) – но здесь мы рассматриваем представителей всех 28 отрядов.

Как и ранее, мы будем использовать матрицу расстояний, значения в которой изменяются от 0 до 1. При этом в таблицах, приведённых в приложении, данные для удобства обозначены целыми числами, образованными первыми тремя значащими цифрами этих значений (т. е. мы фактически умножаем имеющиеся значения на 1000 и округляем до целых).

Итак, в исходной матрице расстояний мы убрали примерно 63% пар элементов (оставили примерно 37% пар; как и ранее, некоторый элемент верхнего треугольника мы убираем вместе с соответствующим элементом

нижнего). Получившаяся матрица приведена в таблице 9 приложения.

Для проведения сравнительного анализа мы сначала произвели восстановление матрицы на основе использования только элементов той матрицы, которая образовалась на последнем проходе. Как отмечалось в предыдущем разделе возможно применение двух подходов. Результаты восстановления с применением первого подхода, где производилось вычисление среднего арифметического по всем оценкам, полученным по всевозможным треугольникам, построенных на этом элементе с двумя другими известными сторонами, представлены в таблице 10 – по смыслу она соответствует таблице 4 «малого» примера. А в таблице 11 приведеная матрица, полученная с применением второго подхода: из всего множества полученных оценок, если позволяло их количество, исключались наибольший и наименьший элементы, а для оставшихся вычислялось среднее арифметическое.

На основе анализа таблицы 11 – как было сказано выше, полученной путём применения второго подхода, – можно, по-видимому, сделать вывод о недостаточной эффективности этого подхода для восстановления матрицы ДНК. Причём эта недостаточная эффективность проявляется, несмотря на то, что, казалось бы, этот подход должен давать относительно лучший вариант: ведь мы исключаем «крайние ситуации». Однако в матрицах больших размерностей получается большое количество одинаковых элементов; этот факт объясняется тем, что на неизвестном элементе образуется *небольшое* количество треугольников с двумя другими известными сторонами, поэтому исключение наибольшей и наименьшей оценок приводит к тому, что вычисление среднего арифметического производится для очень малого количества оценок. Далее будет приведён подробный анализ полученных результатов путём вычисления невязки – и этот анализ также подтвердит неэффективность второго подхода.

Количество проходов, необходимых для восстановления всей матрицы, зависит от процента пропущенных элементов. Как показали результаты вычислительных экспериментов, если процент пропущенных элементов меньше примерно 55%, то восстановление всей матрицы происходит за 1 проход. Если же это количество превышает примерно 64%, то может оказаться, что потребуется более 2 проходов. Кроме того, в этом случае количество проходов будет зависеть от расположения пропущенных элементов. При отсутствии всех элементов некоторой строки (некоторого столбца) восстановление матрицы вообще невозможно, и поэтому с увеличением процента обнуления матрицы уменьшается вероятность её восстановления.

Для проведения сравнительного анализа различных способов восстановления матрицы мы вычислили невязку, а также – для более полной картины – выделили наибольшее отклонение. Результаты вычислений представлены ниже в таблице 5.

Применение второго подхода на первом проходе даёт меньшее значение максимального отклонения, но в целом невязка больше, причём наибольшее отклонение этой невязки от невязки, полученной для первого подхода, происходит на второй итерации – когда количество треугольников с двумя другими известными вершинами становится больше. Таким образом, при вычислении среднего арифметического всех «предварительных значений» элемента значение невязки значительно меньше – по-видимому, практически всегда.

Таблица 5. Сравнение невязки различных подходов при восстановлении матрицы

	1-й подход		2-й подход	
	$max\ d_{ij}$	d	$max\ d_{ij}$	d
1-й проход	0.2135	0.001939	0.2135	0.003322
2-й проход	0.2356	0.002791	0.350	0.003942

Приведённые результаты получены при восстановлении матрицы с использованием только элементов матрицы последнего прохода. Однако в этом случае чем больше номер прохода, тем менее точны элементы матрицы – поэтому «не учитывается предыстория».

Далее нами представлены результаты, полученные путём применения как статической, так и динамической функций риска. Для статической функции риска наилучший результат был получен для коэффициента $p = 0,9$ (обозначения, связанные с функциями риска, см. выше). А для динамической функции риска нами была подобрана убывающая функция

$$f(x) = 1 - \sqrt{0,1x}. \quad (7)$$

Применение функций риска позволило уменьшить значение невязки – особенно при втором и (при необходимости) дальнейших проходах.

Таблица 6. Сравнение невязки восстановления матрицы с применением статической и динамической функций риска

	Восстановление матрицы с помощью статической функции риска, p=0,95		Восстановление матрицы с помощью динамической функции риска, $f(x)=1-(0,1x)^{1/2}$	
	$max\ d_{ij}$	d	$max\ d_{ij}$	d
1-й проход	0.1852	0.001715	0.1414	0.001689
2-й проход	0.1852	0.001851	0.1414	0.001801

Таким образом, применение функции риска при восстановлении матриц позволяет уменьшить значение невязки, особенно при большом количестве проходов. В таблицах 12 и 13 приложения представлены восстановленные с использованием статической и динамической функций риска матрицы.

А в итоговой таблице 7 приведено количество проходов, которые в вычислительном эксперименте потребовались для различного процента исключённых элементов матрицы. Также приведены значения невязки после восстановления матрицы – при использовании функции риска и без неё.

Таблица 7. Итоговая таблица вычислений

Процент	Кол-во проходов	Восстановление матрицы на основе только матрицы последнего прохода		Восстановление матрицы с помощью статической функции риска, p=0,95		Восстановление матрицы с помощью динамической функции риска, f(x)=1-(0,1x)$^{1/2}$	
		$max\ d_{ij}$	d	$max\ d_{ij}$	d	$max\ d_{ij}$	d
50	1	0.1934	0.001978	0.1856	0.001734	0.1576	0.001678
62	2	0.2135	0.002791	0.1852	0.001851	0.1414	0.001801
65	3	0.2025	0.002934	0.1745	0.002032	0.1356	0.001998

VII. Заключение

Итак, в основе предлагаемого нами метода *восстановления* матрицы расстояний между последовательностями ДНК мы предлагаем использовать подход, который был ранее разработан и применён на практике для сравнительной оценки других алгоритмов – алгоритмов *расчёта расстояний* между такими последовательностями; упрощая, можно сказать, что мы пытаемся добиться выполнения свойства остроугольной равнобедренности для всех образующихся треугольников. При этом лучшие результаты получаются в том случае, когда оценки неизвестных элементов матрицы основываются на использовании функций риска – как статической, так и динамической. Применение описанного метода для заполнения матрицы расстояний между последовательностями ДНК позволит значительно сократить время её заполнения: например, как уже отмечалось выше, для построения матрицы порядка 50×50, в которую записываются расстояния, вычисляемые алгоритмом Нидлмана – Вунша, требуется около 28 часов [2], а при использовании предложенного нами метода – около 2 часов.

Можно сказать, что, аналогично работам [19], [21], наша статья направлена на то, чтобы *предложить советы*:

- для улучшения уже имеющихся, описанных ранее алгоритмов;
- для разработки новых алгоритмов.

В обоих случаях мы имеем в виду алгоритмы вычисления расстояний между последовательностями геномов.

Далее очень кратко опишем некоторые из возможных направлений дальнейшей работы.

1) Рассмотреть другие возможные формулы для вычисления badness – причём как одного треугольника, так и всей матрицы. В последнем случае необходимо, как и в описанном выше способе, badness всей матрицы считать на основе значений badness каждого из треугольников. Например, в основе вычисления badness одного треугольника можно использовать не только разности углов, но и разности сторон, а также отдельно рассматривать отклонение от равнобедренности и отклонение от остроугольности.
2) Продолжить работу по подбору удачных вариантов динамической функцией риска, дающих для нашей задачи меньшие значения невязки. Согласно [9], динамические функции риска строятся путём квадратичной интерполяции, причём точки для интерполяции выбираются в зависимости от значений статической функции риска. В будущем можно рассмотреть и другие варианты выбора динамических функций риска.
3) Разработать другие подходы для сравнительного анализа различных алгоритмов вычисления расстояний между последовательностями – и описать алгоритмы восстановления матриц на основе этих подходов. В настоящее время нами ведутся работы по сравнению двух из таких алгоритмов – причём как для применения в «обычных» задачах ДНК-анализа, так и в близких к рассматриваемым в настоящей работе задачах восстановления матриц ДНК.
4) Описать и применить на практике методы «учитывания предыстории» при работе алгоритмов восстановления матриц.

Конечно, этими четырьмя направлениями дальнейшая работа ограничиваться не будет.

[2] Несложно посчитать, что количество пар элементов (для которых мы и считаем расстояния) в матрицах размерности 50×50 примерно в $3,25$ раз больше, чем в случае рассматриваемых в настоящей статье «больших» матриц размерности 28×28.

Список литературы

[1] Айала Ф., Кайгер Дж. *Современная генетика*. Пер. с англ. Т. 1. – М.: Мир. 1987. 295 с.

[2] Мельников Б. Ф., Романов Н. В. *Ещё раз об эвристиках для задачи коммивояжёра.* Теоретические проблемы информатики и ее приложений. Т. 4. 2001. С. 81–86.

[3] Melnikov B., Radionov A., Gumayunov V. *Some special heuristics for discrete optimization problems.* In: Proceedings of 8th International Conference on Enterprise Information Systems, ICEIS-2006. Paphos. 2006. P. 360–364.

[4] Мельников Б. Ф., Панин А. Г. *Параллельная реализация мультиэвристического подхода в задаче сравнения генетических последовательностей.* Вектор науки Тольяттинского государственного университета. № 4 (22). 2012. С. 83–86.

[5] Мельников Б. Ф., Пивнева С. В, Трифонов М. А. *Мультиэвристический подход к сравнению качества определяемых метрик на множестве последовательностей ДНК*. Современные информационные технологии и ИТ образование. Т. 13. № 2. 2017. С. 89–96.

[6] Eckes B., Nischt R., Krieg T. *Cell-matrix interactions in dermal repair and scarring.* Fibrogenesis Tissue Repair. No. 3:4. 2010. doi:10.1186/1755-1536-3-4.

[7] Midwood K. S., Williams L. V., Schwarzbauer J. E. *Tissue repair and the dynamics of the extracellular matrix*. The International Journal of Biochemistry & Cell Biology. 2004. Vol. 36. Issue 6. P. 1031–1037.

[8] Мельников Б. Ф., Радионов А. Н. *О выборе стратегии в недетермированных антагонистических играх*. Программирование. № 5. 1998. С. 55–62.

[9] Мельников Б. Ф. *Эвристики в программировании недетерминированных игр.* Известия РАН. Программирование. № 5. 2001. С. 63–80.

[10] Гэри М., Джонсон М. *Вычислительные машины и труднорешаемые задачи.* Пер. с англ. – М.: Мир. 1982. 416 с.

[11] Needleman S., Wunsch Ch. *A general method 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.

[12] Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. *Алгоритмы. Построение и анализ.* М.: Вильямс, 2005. 1296 с.

[13] Pages H., Aboyoun P., Gentleman R., DebRaoy S. *Biostrings: String Objects Representing Biological Sequences and Matching Algorithms* [Электрон. ресурс]. Bioconductor – Электрон. дан. – Режим доступа: https://bioc.ism.ac.jp/packages/2.6/bioc/html/Biostrings.html, свободный

[14] Melnikov B. F., Pivneva S. V., Trifonov M. A. *Comparative analysis of algorithms calculating distances of DNA sequences and some related problems.* Сборник трудов III международной конференции и молодежной школы «Информационные технологии и нанотехнологии (ИТНТ-2017)», Самарский национальный исследовательский университет имени академика С.П. Королева. 2017. С. 1640–1645.

[15] Frans B. M. *Bonobo: The Forgotten Ape.* University of California Press, ISBN 0-520-20535-9; trade paperback. October. 1998. P. 224.

[16] Melnikov B. *Heuristics in programming of nondeterministic games.* Programming and Computer Software. Vol. 27. No. 5. 2001. С. 277–288.

[17] Мельников Б. Ф., Мельникова Е. А. *Подход к программированию недетерминированных игр (Часть I: Описание общих эвристик).* Известия высших учебных заведений. Поволжский регион. Физико-математические науки. № 4 (28). 2013. С. 29–38.

[18] Мельников Б. Ф., Пивнева С. В. *Принятие решений в прикладных задачах с применением динамически подобных функций риска.* Вестник транспорта Поволжья. № 3. 2010. С. 28–33.

[19] Мельников Б. Ф., Тренина М. А., Кочергин А. С. *Подход к улучшению алгоритмов расчёта расстояний между цепочками ДНК (на примере алгоритма Нидлмана – Вунша).* Известия высших учебных заведений. Поволжский регион. Физико-математические науки. № 1 (45). 2018. (https://izvuz_fmn.pnzgu.ru/fmn118)

[20] Home – Nucleotide – NCBI [Электрон. ресурс]. – Режим доступа: https://www.ncbi.nlm.nih.gov/nuccore, свободный

[21] Makarkin S., Melnikov B., Panin A. *On the metaheuristics approach to the problem of genetic sequence comparison and its parallel implementation.* Applied Mathematics (Scientific Research Publishing). Vol. 04. No. 10. P. 35–39.

Приложение. Матрицы 28×28 и дополнительная информация.

Таблица 8. Исходная матрица, заполненная в результате работы алгоритма

0	299	258	269	315	324	285	295	503	327	268	271	302	293	305	283	262	261	302	317	266	961	272	328	266	242	268	274
299	0	369	298	240	209	301	222	505	327	303	302	321	307	229	296	297	287	265	298	292	961	316	269	298	288	296	299
258	369	0	292	343	339	358	372	584	398	358	372	376	373	378	311	344	299	341	386	344	997	356	401	347	273	238	304
269	298	292	0	293	348	292	298	500	306	273	279	283	278	297	271	275	253	327	309	257	961	289	314	270	260	311	264
315	240	343	293	0	217	318	236	506	316	318	309	314	312	249	312	302	307	289	304	297	961	335	272	311	309	316	312
324	209	339	348	217	0	341	198	510	358	351	337	367	357	267	334	346	332	252	320	337	961	354	224	346	336	355	350
285	301	358	292	318	341	0	304	507	318	292	290	309	292	312	275	288	279	320	317	270	961	310	333	279	280	285	276
295	222	372	298	236	198	304	0	505	323	314	297	324	316	222	301	297	294	261	301	295	999	319	268	304	298	331	310
503	505	584	500	506	510	507	505	0	511	502	505	506	502	501	502	506	503	504	510	501	999	504	508	500	501	504	503
327	327	398	306	316	358	318	323	511	0	326	322	302	311	330	311	310	318	364	329	304	961	343	329	320	301	354	311
268	303	358	273	318	351	292	314	502	326	0	282	295	287	308	281	284	273	330	326	267	961	275	332	272	267	312	268
271	302	372	279	309	337	290	297	505	322	282	0	294	303	306	285	283	274	323	315	274	999	289	335	290	277	271	290
302	321	376	283	314	367	309	324	506	302	295	294	0	300	318	296	297	298	256	320	281	999	313	321	297	288	300	287
293	307	373	278	312	357	292	316	502	311	287	303	300	0	311	277	294	285	335	323	264	999	313	328	264	274	296	243
305	229	378	297	249	267	312	222	501	330	308	306	318	311	0	301	305	297	264	301	301	961	313	266	300	295	302	299
283	296	311	271	312	334	275	301	502	311	281	285	296	277	301	0	285	267	321	315	255	999	294	327	256	263	312	259
262	297	344	275	302	346	288	297	506	310	284	283	297	294	305	285	0	272	326	312	273	961	294	327	278	257	297	286
261	287	299	253	307	332	279	294	503	318	273	274	298	285	297	267	272	0	307	318	261	961	287	326	265	257	298	273
302	265	341	327	289	252	320	261	504	364	330	323	356	335	264	321	326	307	0	326	317	998	327	297	322	315	333	326
317	298	386	309	304	320	317	301	510	329	326	315	320	323	301	315	312	318	326	0	303	961	334	322	318	313	318	316
266	292	344	257	297	337	270	295	501	304	267	274	281	264	301	255	273	261	317	303	0	961	288	312	259	252	306	240
961	961	997	961	961	961	961	999	999	961	961	999	999	999	961	999	961	961	998	961	961	0	995	961	961	999	989	961
272	316	356	289	335	354	310	319	504	343	275	289	313	313	313	294	294	287	327	334	288	995	0	338	287	281	296	289
328	269	401	314	272	224	333	268	508	329	332	335	321	328	266	327	327	326	297	322	312	961	338	0	326	322	364	322
266	298	347	270	311	346	279	304	500	320	272	290	297	264	300	256	278	265	322	318	259	961	287	326	0	252	308	242
242	288	273	260	309	336	280	298	501	301	267	277	288	274	295	263	257	257	315	313	252	999	281	322	252	0	244	250
268	296	238	311	316	355	285	331	504	354	312	271	300	296	302	312	297	298	333	318	306	989	296	364	308	244	0	276
274	299	304	264	312	350	276	310	503	311	268	290	287	243	299	259	286	273	326	316	240	961	289	322	242	250	276	0

Таблица 9. Подаваемая на вход алгоритма восстановления неполностью заполненная матрица

0	—	258	—	—	—	—	—	—	—	—	—	—	—	—	—	262	—	—	—	—	—	—	—	—	—
—	0	—	—	—	—	—	222	—	—	—	—	—	—	—	—	297	—	—	—	292	—	—	269	—	—	—	—
258	—	0	—	343	339	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	273	—	—
—	—	—	0	—	—	292	—	—	—	—	—	—	—	—	—	—	—	—	—	257	—	—	—	—	—	—	—
—	—	343	—	0	—	—	236	—	—	—	—	—	—	—	—	—	—	—	304	—	—	—	—	—	—	—	—
—	—	339	—	—	0	—	—	—	—	351	—	—	—	—	—	—	—	—	—	337	—	—	—	—	—	—	—
—	—	—	292	—	—	0	—	—	318	—	—	—	—	—	—	—	279	—	—	—	—	—	—	—	—	—	—
—	222	—	—	236	—	—	0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	298	—	—
—	—	—	—	—	—	—	—	0	—	—	505	506	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
—	—	—	—	—	—	318	—	—	0	—	—	302	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
—	—	—	—	—	351	—	—	—	—	0	—	—	—	—	—	—	—	—	326	—	—	—	332	—	—	—	268
—	—	—	—	—	—	—	—	505	—	—	0	—	—	—	—	—	—	—	315	—	—	—	—	—	—	—	—
—	—	—	—	—	—	—	—	506	302	—	—	0	—	—	—	297	—	—	—	—	—	—	—	—	—	—	—
—	—	—	—	—	—	—	—	—	—	—	—	—	0	—	—	—	—	—	—	264	999	—	—	—	—	—	—
—	—	—	—	—	—	—	—	—	—	—	—	—	—	0	301	—	—	264	—	—	—	—	—	300	—	—	—
—	—	—	—	—	—	—	—	—	—	—	—	—	—	301	0	—	—	321	—	—	—	—	—	256	—	—	—
262	297	—	—	—	—	—	—	—	—	—	—	297	—	—	—	0	—	326	—	—	—	—	—	—	—	—	—
—	—	—	—	—	—	279	—	—	—	—	—	—	—	—	—	—	0	—	—	—	—	287	—	265	—	—	—
—	—	—	—	—	—	—	—	—	—	—	—	—	—	264	321	326	—	0	—	—	—	—	—	—	—	—	—
—	—	—	—	304	—	—	—	—	—	326	315	—	—	—	—	—	—	—	0	303	—	—	—	—	—	—	—
—	292	—	257	—	337	—	—	—	—	—	—	—	264	—	—	—	—	—	303	0	—	—	—	—	—	—	—
—	—	—	—	—	—	—	—	—	—	—	—	—	999	—	—	—	—	—	—	—	0	995	—	—	—	—	—
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	287	—	—	—	995	0	—	—	—	—	—
—	269	—	—	—	—	—	—	—	—	332	—	—	—	—	—	—	—	—	—	—	—	—	0	—	—	—	—
—	—	—	—	—	—	—	—	—	—	—	—	—	—	300	256	—	265	—	—	—	—	—	—	0	—	—	—
—	—	273	—	—	—	—	298	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0	244	—
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	244	0	—
—	—	—	—	—	—	—	—	—	—	268	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0

Таблица 10. Восстановленная с применением первого подхода матрица

0	296	258	296	319	338	318	315	504	353	361	351	338	341	462	461	262	352	352	347	344	998	432	380	423	380	388	426
296	0	296	294	309	338	317	222	504	363	361	350	338	340	462	461	297	351	352	346	292	998	432	269	423	381	389	426
258	296	0	296	343	338	326	317	504	359	364	356	345	350	462	462	343	355	354	352	350	998	433	389	424	273	389	427
296	294	296	0	317	338	292	316	504	360	364	279	347	339	463	462	347	355	356	345	257	998	431	385	421	388	394	427
319	309	343	317	0	340	329	236	504	363	360	356	352	352	463	463	352	359	356	304	349	998	434	390	424	389	397	426
338	338	338	338	340	0	339	338	504	372	351	370	362	361	464	464	362	368	362	361	336	998	436	399	427	398	404	430
318	317	326	292	329	339	0	319	504	318	372	357	354	356	462	462	357	279	362	358	357	998	436	399	425	401	402	431
315	222	317	316	236	338	319	0	504	359	363	358	348	347	463	463	348	356	356	349	348	998	441	386	431	298	391	431
504	504	504	504	504	504	504	504	0	504	504	505	506	504	532	532	504	504	504	504	504	998	542	536	542	536	536	539
353	363	359	360	363	372	318	359	504	0	368	355	302	354	463	463	351	354	359	355	354	998	437	397	426	398	399	429
361	361	364	364	360	351	372	363	504	368	0	362	361	361	464	464	361	326	359	326	359	998	429	332	429	399	402	268
351	350	356	279	356	370	357	358	505	355	362	0	352	350	463	462	352	351	358	315	348	998	436	394	426	396	397	429
338	338	345	347	352	362	354	348	506	302	361	352	0	342	462	461	297	353	351	347	345	998	436	383	426	481	391	429
341	340	350	339	352	361	356	347	504	354	361	350	342	0	462	462	342	351	353	344	264	999	430	382	425	384	391	429
462	462	462	463	463	464	462	463	532	463	464	463	462	462	0	301	325	316	264	351	350	998	479	461	300	456	457	473
461	461	462	462	463	464	462	463	532	463	464	462	461	462	301	0	325	304	321	351	350	998	477	460	256	455	457	473
262	297	343	347	352	362	357	348	504	351	361	352	297	342	325	325	0	349	326	347	345	998	435	383	414	379	390	429
352	351	355	355	359	368	279	356	504	354	365	351	353	351	316	304	349	0	352	353	351	998	287	395	265	396	397	429
352	352	354	356	356	362	362	356	504	359	361	358	351	353	264	321	326	352	0	354	353	998	437	392	415	390	397	429
347	346	352	345	304	361	358	349	504	355	326	315	347	344	351	351	347	353	354	0	303	998	436	384	426	388	393	427
344	292	350	257	349	336	357	348	504	354	359	348	345	264	350	350	345	351	353	303	0	998	435	383	426	385	392	428
998	998	998	998	998	998	998	998	998	998	998	998	998	999	998	998	998	998	998	998	998	0	995	998	998	998	998	998
432	432	433	431	434	436	436	441	542	437	440	436	436	430	479	477	435	287	437	436	435	995	0	432	426	432	432	438
380	269	389	385	390	399	399	386	536	397	332	394	383	382	461	460	383	395	392	384	383	998	432	0	423	382	389	425
423	423	424	421	424	427	425	431	542	426	429	426	426	425	300	256	414	265	415	426	426	998	426	423	0	423	423	432
380	381	273	388	389	398	401	298	536	398	399	396	381	384	456	455	379	396	390	388	385	998	432	382	423	0	244	429
388	389	389	394	397	404	402	391	536	399	402	397	391	391	457	457	390	397	397	393	392	998	432	389	423	244	0	427
426	426	427	427	426	430	431	431	539	429	268	429	429	429	473	473	429	429	429	427	428	998	438	425	432	426	427	0

Таблица 11. Восстановленная с применением второго подхода матрица

0	296	258	296	343	338	319	312	504	319	341	319	297	296	458	458	262	319	325	323	317	999	319	296	319	297	296	351
296	0	298	296	343	338	317	222	504	317	341	316	297	296	458	458	297	317	325	323	292	999	319	269	319	296	319	351
258	298	0	296	343	339	343	302	504	343	337	343	297	296	457	456	396	343	325	311	296	999	326	296	326	273	343	351
296	296	296	0	343	338	292	312	504	339	341	279	297	296	457	456	296	339	325	323	257	999	319	296	319	297	330	351
343	343	343	343	0	343	340	236	504	340	343	340	343	343	457	456	343	340	343	304	343	999	329	343	329	343	343	351
338	338	339	338	343	0	340	338	504	340	351	340	338	338	457	456	338	340	338	338	337	999	339	338	339	338	338	351
319	317	343	292	340	340	0	317	504	318	337	317	325	317	457	456	325	279	334	317	317	999	504	317	504	319	330	351
312	222	302	312	235	338	317	0	504	317	341	316	312	312	457	456	312	317	325	323	317	999	504	312	504	298	312	351
504	504	504	504	504	504	504	504	0	504	504	505	506	504	504	504	504	504	504	504	504	999	504	504	504	504	504	504
319	317	343	339	340	340	318	317	504	0	337	317	302	317	457	456	325	317	334	317	317	999	323	317	323	319	330	351
341	341	337	341	343	351	337	341	504	337	0	337	341	341	457	456	341	337	341	326	341	999	340	332	340	341	341	268
319	316	343	279	340	340	317	316	505	317	337	0	325	317	457	456	325	317	334	314	316	999	319	316	319	319	330	351
297	297	297	297	343	338	325	312	506	302	341	325	0	297	457	456	297	325	325	323	317	999	319	297	319	297	297	351
296	296	296	296	343	338	317	312	504	317	341	317	297	0	457	456	296	317	325	323	264	999	319	296	319	297	296	351
458	458	457	457	457	457	457	457	504	457	457	457	457	457	0	301	352	349	264	353	351	999	393	351	300	352	353	385
458	458	456	456	456	456	456	456	504	456	456	456	456	456	301	0	325	349	321	353	351	999	393	351	256	352	353	385
262	296	396	296	343	338	325	312	504	325	341	325	297	296	352	325	0	325	326	323	317	999	320	296	320	297	296	351
319	317	343	339	340	340	279	317	504	317	337	317	325	317	349	349	325	0	334	317	317	999	287	317	265	319	330	351
325	325	325	325	343	338	334	325	504	334	341	334	325	325	264	321	326	334	0	325	325	999	333	325	333	325	325	351
323	323	311	323	304	338	317	323	504	317	326	314	323	323	353	353	323	317	325	0	303	999	323	323	323	323	323	351
317	292	296	257	343	337	317	317	504	317	341	316	317	264	351	351	317	317	325	303	0	999	319	317	319	317	317	351
999	999	999	999	999	999	999	999	999	999	999	999	999	999	999	999	999	999	999	999	999	0	995	999	999	999	999	999
319	319	326	319	329	339	504	504	504	323	340	319	319	319	393	393	320	286	333	323	319	995	0	319	319	319	322	351
296	269	296	296	343	338	317	312	504	317	332	316	297	296	351	351	296	317	325	323	317	999	319	0	319	297	296	351
319	319	326	319	329	339	504	504	504	323	340	319	319	319	300	256	320	265	333	323	319	999	319	319	0	319	322	351
297	296	273	297	343	338	319	298	504	319	341	319	297	297	352	352	297	319	325	323	317	999	319	297	319	0	244	351
296	296	343	330	343	338	330	312	504	330	341	330	297	296	353	353	296	330	325	323	317	999	322	296	322	244	0	351
351	351	351	351	351	351	351	351	504	351	268	351	351	351	385	385	351	351	351	351	351	999	351	351	351	351	351	0

Таблица 12. Восстановленная с применением статической функции риска матрица

0	296	258	262	294	298	275	258	398	294	296	297	287	266	285	288	262	271	269	282	280	996	330	294	303	293	288	307
296	0	267	266	290	309	277	222	398	295	300	298	286	268	289	292	297	274	271	287	292	995	335	269	308	297	292	311
258	267	0	263	343	339	324	291	398	320	308	324	307	293	298	300	295	293	293	294	290	993	337	313	312	273	308	310
262	266	263	0	303	310	292	274	408	299	303	279	296	276	291	293	282	277	279	287	257	996	335	301	308	299	296	309
294	290	343	303	0	307	284	236	395	302	308	305	301	281	295	297	288	275	283	303	286	995	334	306	308	303	301	310
298	309	339	310	307	0	284	290	401	304	351	317	314	300	303	305	298	290	295	296	337	994	345	317	320	315	314	317
275	277	324	292	284	284	0	274	404	318	311	310	307	288	294	297	289	279	288	292	289	994	341	313	314	309	306	313
258	222	291	274	236	290	274	0	406	301	301	305	291	272	287	290	278	273	274	285	282	994	339	298	309	298	294	311
398	398	398	408	395	401	404	406	0	416	394	505	506	414	381	379	410	406	403	400	397	993	435	417	414	412	410	406
294	295	320	299	302	304	318	301	416	0	312	310	302	292	295	297	290	284	290	293	289	994	344	314	317	310	307	313
296	300	308	303	308	351	311	301	394	312	0	320	312	294	300	302	294	289	292	326	288	994	344	332	321	315	313	268
297	298	324	279	305	317	310	305	505	310	320	0	301	295	299	302	295	290	292	315	295	993	352	320	324	317	313	322
287	286	307	296	301	314	307	291	506	302	312	301	0	283	297	300	297	293	287	296	295	994	351	310	323	307	306	321
266	268	293	276	281	300	288	272	414	292	294	295	283	0	297	300	276	272	272	285	264	999	331	307	322	306	302	322
285	289	298	291	295	303	294	287	381	295	300	299	297	297	0	301	313	294	264	270	278	940	391	350	299	341	337	335
288	292	300	293	297	305	297	290	379	297	302	302	300	300	301	0	313	282	320	287	290	987	401	359	256	346	342	341
262	296	295	282	288	298	289	278	410	290	294	295	297	276	313	313	0	279	326	291	289	980	344	306	312	302	299	315
271	274	293	277	275	290	279	273	406	284	289	290	293	272	294	282	279	0	279	287	282	899	287	305	265	301	297	311
269	271	293	279	283	295	288	274	403	290	292	292	287	272	264	320	325	279	0	289	286	873	340	302	312	300	296	314
282	287	294	287	303	296	292	285	400	293	326	315	296	285	270	287	291	287	289	0	303	878	349	312	322	312	310	319
280	292	290	257	286	337	289	282	397	289	288	295	295	264	278	290	289	282	286	303	0	991	32	12	315	308	305	314
996	995	993	996	995	994	994	994	993	994	995	993	994	999	940	987	980	899	873	878	991	0	995	884	873	864	855	847
330	335	337	335	334	345	341	339	435	344	344	352	351	331	391	401	344	287	340	349	342	995	0	335	326	330	325	327
294	269	313	301	306	317	313	298	417	314	332	320	310	307	350	359	306	305	302	312	312	884	335	0	309	297	292	312
303	308	312	308	308	320	314	309	414	317	321	324	323	322	299	256	312	264	312	322	315	873	326	309	0	307	302	308
293	297	273	299	303	315	309	298	412	310	315	317	307	306	341	346	302	301	300	312	308	864	330	297	307	0	244	310
288	292	308	296	301	314	306	294	410	307	313	313	306	302	337	342	299	297	296	310	305	855	325	292	302	244	0	309
307	311	310	309	310	317	313	311	406	313	268	322	321	322	335	341	315	311	314	319	314	847	327	312	308	310	309	0

Таблица 13. Восстановленная с применением динамической функции риска матрица

0	296	258	296	319	338	318	315	504	353	361	351	338	341	462	461	262	352	352	347	344	998	432	380	423	380	388	426
296	0	296	294	309	338	317	222	504	353	361	350	338	340	462	461	297	351	352	346	292	998	432	269	423	381	389	426
258	296	0	296	343	339	326	317	504	359	364	356	345	350	462	462	343	355	354	352	350	998	433	389	424	273	389	427
296	294	296	0	217	338	292	316	504	360	364	279	347	339	463	462	347	355	356	345	257	998	431	385	421	388	394	427
319	309	343	217	0	340	329	236	504	363	360	356	352	352	463	463	352	359	356	304	349	998	434	390	424	389	397	426
338	338	339	338	340	0	339	338	504	372	351	370	362	361	464	464	362	368	362	361	337	998	436	399	427	398	404	430
318	317	326	292	329	339	0	319	504	318	372	357	354	356	462	462	357	279	362	358	357	998	436	399	425	401	402	431
315	222	317	316	236	338	319	0	504	359	363	358	348	347	463	463	348	356	356	349	348	998	441	386	431	298	391	431
504	504	504	504	504	504	504	504	0	504	504	505	506	504	532	532	504	504	504	504	504	998	542	536	542	536	536	539
353	353	359	360	363	372	318	359	504	0	368	355	302	354	463	463	351	354	359	355	354	998	437	397	326	398	399	429
361	361	364	364	360	351	372	363	504	368	0	362	361	361	464	464	361	365	361	326	359	998	440	332	429	399	402	268
351	350	356	279	356	370	357	358	505	355	362	0	352	350	463	462	352	351	358	315	348	998	436	394	426	396	397	429
338	338	345	347	352	362	354	348	506	302	361	352	0	342	462	463	297	353	351	347	345	998	436	383	426	381	391	429
341	340	350	339	352	361	356	347	504	354	361	350	342	0	462	462	342	351	353	344	264	999	430	382	425	384	391	429
462	462	462	463	463	464	462	463	532	463	464	463	462	462	0	301	325	316	264	351	350	998	479	461	300	456	457	473
461	461	462	462	463	464	462	463	532	463	464	462	461	462	301	0	325	304	321	351	350	998	477	460	256	455	457	473
262	297	343	347	352	362	357	348	504	351	361	352	297	342	325	325	0	349	326	347	345	998	435	383	414	379	390	429
352	351	355	355	359	368	279	356	504	354	365	351	353	351	316	304	349	0	352	353	351	998	287	395	265	396	397	429
352	352	354	356	356	362	362	356	504	359	361	358	351	353	264	321	325	352	0	305	353	998	437	392	415	390	397	429
347	346	352	345	303	361	358	349	504	355	326	315	347	344	351	351	347	353	305	0	303	998	436	384	426	388	393	427
344	292	350	257	349	337	357	348	504	354	359	348	345	264	350	350	345	351	353	303	0	998	435	383	426	385	392	428
998	998	998	998	998	998	998	998	998	998	998	998	998	999	998	998	998	998	998	998	998	0	995	998	998	998	998	998
432	432	433	431	434	436	436	441	542	437	440	436	436	430	479	477	435	286	437	436	435	995	0	432	426	432	432	438
380	269	389	385	390	399	399	386	536	397	332	394	383	382	461	460	383	395	392	384	383	998	432	0	423	382	389	425
423	423	424	421	424	427	425	431	542	426	429	426	426	425	300	256	414	265	415	426	426	998	426	423	0	423	423	432
380	381	273	388	389	398	401	298	536	398	399	396	381	384	456	455	379	396	390	388	385	998	432	382	423	0	244	426
388	389	389	394	397	404	402	391	536	399	402	397	391	391	457	457	390	397	397	393	392	998	432	389	423	244	0	427
426	426	427	427	426	430	431	431	539	429	268	429	429	429	473	473	429	429	429	427	428	998	438	425	432	426	427	0

Таблица 14. Виды млекопитающих, чьи геномы были использованы для вычислений

	Вид	Семейство
1	Южный малый полосатик (лат. Balaenoptera bonaerensis)	Китообразные (лат. Cetacea)
2	Эквадорский ценолест (лат. Caenolestes fuliginosus)	Ценолесты (лат. Paucituberculata)
3	Домашняя коза (лат. Capra hircus)	Парнокопытные (лат. Artiodactyla)
4	Девятипоясный броненосец (лат. Dasypus novemcinctus)	Броненосцы (лат. Cingulata)
5	Виргинский опоссум (лат. Didelphis virginiana)	Опоссумы (лат. Didelphimorphia)
6	Соневидный опоссум (лат. Dromiciops gliroides)	Микробиотерии (лат. Microbiotheria)
7	Малый ежовый тенрек (лат. Echinops telfairi)	Афросорициды (лат. Afrosoricida)
8	Толстоголовый бандикут (лат. Echymipera rufescens)	Бандикуты (лат. Peramelemorphia)
9	Дикий осёл (лат. Equus asinus)	Непарнокопытные (лат. Perissodactyla)
10	Европейский ёж (лат. Erinaceus europaeus)	Насекомоядные (лат. Eulipotyphla)
11	Малайский шерстокрыл (лат. Galeopterus variegates)	Шерстокрылы (лат. Dermoptera)
12	Соня-полчок (лат. Glis glis)	Грызуны (лат. Rodentia)
13	Заяц-беляк (лат. Lepus timidus)	Зайцеобразные (лат. Lagomorpha)
14	Саванный слон (лат. Loxodonta africana)	Хоботные (лат. Proboscidea)
15	Горный кенгуру (лат. Macropus robustus)	Двурезцовые (лат. Diprotodontia)
16	Короткоухий прыгунчик (лат. Macroscelides proboscideus)	Прыгунчики (лат. Macroscelidea)
17	Длиннохвостый ящер (лат. Manis tetradactyla)	Панголины (лат. Pholidota)
18	Гигантский муравьед (лат. Myrmecophaga tridactyla)	Неполнозубые (лат. Pilosa)
19	Сумчатый крот (лат. Notoryctes typhlops)	Сумчатые кроты (лат. Notoryctemorphia)
20	Утконос (лат. Ornithorhynchus anatinus)	Однопроходные (лат. Monotremata)
21	Трубкозуб (лат. Orycteropus afer)	Трубкозубые (лат. Tubulidentata)
22	Обыкновенный шимпанзе (лат. Pan troglodytes)	Приматы (лат. Primates)
23	Тафа (Phascogale tapoatafa)	Хищные сумчатые (лат. Dasyuromorphia)
24	Капский даман (лат. Procavia capensis)	Дамановые (лат. Dasyuromorphia)
25	Австралийская летучая лисица (лат. Pteropus scapulatus)	Рукокрылые (лат. Chiroptera)
26	Пума (лат. Puma concolor)	Хищные (лат. Carnivora)
27	Американский ламантин (лат. Trichechus manatus)	Сирены (лат. Sirenia)
28	Малайская тупайя (Tupaia belangeri)	Тупайи (лат. Scandentia)

On a problem of the reconstruction of distance matrices between DNA sequences

Boris Melnikov, Marina Trenina

Abstract—**In practice, quite often there is a need to calculate in a special way certain distances between sequences of different nature. Similar algorithms are used in bioinformatics to compare sequenced genetic chains. Due to the large dimension of such chains, it is necessary to use heuristic algorithms that give approximate results.**

There are various heuristic algorithms for determining the distance between genomes, but the obvious disadvantage in calculating the distance between the same pair of DNA strings is to obtain several different results when using different algorithms for calculating metrics. Therefore, there is a problem of assessing the quality of the used metrics (distances), the results of which can be concluded about the applicability of the algorithm to various studies.

In addition, one of the problems considered in biocybernetics is the problem of recovering the matrix of distances between DNA sequences, when not all elements of the considered matrix are known at the input of the algorithm. In this regard, a problem of developing method for comparative evaluation of algorithms calculating distances between sequences is used for another problem, i.e., the problem of restoring the matrix of distances between DNA sequences.

In this article, we consider the possibility of using the developed and studied by us earlier method of comparative evaluation of algorithms for calculating distances between a pair of DNA strings to restore the partially filled matrix of distances. Matrix recovery occurs as a result of several computational passes. Estimation of unknown matrix elements are averaged in a special way with the use of so-called risk function, and the result of this averaging is considered as the resulting value of the unknown element.

Keywords—**DNA sequences, metric, distance matrix, partially filled matrix, recovery, risk functions.**