Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

ВЫСШАЯ ШКОЛА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

Направление: 09.03.03 Прикладная информатика

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

МЕТРИЧЕСКИЕ ИНВАРИАНТЫ В ПРОСТРАНСТВАХ ДАННЫХ

Студент 4 курса

группы 11-501

«___»_____2019 г. Маликов Д.М.

Научный руководитель

Ассистент кафедры

математического анализа

«___»_____2019 г. Новиков А.А.

Директор Высшей школы ИТИС

«___»_____2019 г. Хасьянов А.Ф.

Казань – 2019 г.

ОГЛАВЛЕНИЕ

Список сокращений

- ДНК – дезоксирибонуклеиновая кислота
- A, G, T, C – аденин, гуанин, цитозин, тимин, соответственно
- П.н – пара нуклеотид
- Ori – точка начала репликации
- Ter – точка окончания репликации
- QTL – один из видов инверсии
- TE – мобильные генетические элементы
- Индел – инсерция с последующей делецией (или наоборот)
- Кб, мб – тысяча оснований и миллион оснований, соответственно
- NAHR – неаллельная гомологичная рекомбинация

Введение

Генетическая изменчивость является предпосылкой эволюционных изменений. При ее отсутствии никакое последующее видообразование не может быть достигнуто. Генетическая изменчивость в конечном итоге вся генерируется мутациями [1] .

Такие генные мутации, как инверсии, инсерции и делеции в ряду с заменой оснований являются самыми распространенными. Известно, что инверсии оказывают влияние на архитектуру генома как прокариот, так и эукариот, на поведение некоторых насекомых, половую изоляцию и в конечном счете на видообразование [2]. Многие бактерии имеют геном меньше, чем у своих предков из-за многочисленных инсерций, похожая ситуация наблюдается с беспозвоночными (насекомые) и многими позвоночными (рыбы, амфибии, птицы, млекопитающие) [3]. В процессе эволюции происходят вставки в последовательность ДНК, но чаще и намного масштабней происходят делеции. У птиц, возможно, из-за делеций было утрачено чуть менее, чем 300 генов. Таким образом, инверсии и индел вносят немалый вклад в эволюцию, создают «субстрат» для нее и в то же время являются ее инструментом [4].

На сегодняшний день существует два типа выравнивания: парное (сравниваются две последовательности) и множественное (более двух последовательностей). Алгоритмы, чаще используемого множественного выравнивания, не могут учесть инверсии и транслокации. Эти мутации считаются как сумма инсерций и делеций или же просто делеции. Схожая картина и с парным выравниванием (прогрессивное выравнивание основано на итерации алгоритмов парного выравнивания). На данный момент не существует методов, которые являлись бы достаточно чувствительными к инверсиям и транслокациям.

В данной работе описывается алгоритм, позволяющий детектировать инверсии на основании отношения вероятности их возникновения к вероятности возникновения индел. Так как детекция инверсий через перебор имеет высокую вычислительную сложность, был предложен эвристический алгоритм, который

не создает дополнительные копии последовательности и производит поиск инверсий параллельно.

Целью выпускной курсовой работы является разработка нового метода вычисления гомологии геномных последовательностей, который основан на метрике, учитывающей такие мутации ДНК, как делеция, инверсия и вставка [5].

В соответствии с поставленной целью были выдвинуты следующие задачи:

- Проанализировать существующие метрики построения филогенетических деревьев и выявить их недостатки;
- Разработать новую метрику гомологии геномных последовательностей, учитывающую выявленные на предыдущем этапе недостатки;
- Написать программу для парного выравнивания, учитывающего возможность возникновения инверсий.

Преимуществом разработанной программы выравнивания являются гибкость, так как ее основу можно легко заменить на любой из имеющихся алгоритмов выравнивания, и быстродействие, поскольку алгоритм не перебирает все возможные решения, а пользуется эвристикой.

1. Методы выравнивания нуклеотидных последовательностей

Выравнивание (сравнение) последовательностей – это представление гомологии символов (нуклеотидов или аминокислот) в двух или более последовательностях или выведение этого. Большая гомология между последовательностями обычно подразумевает структурное и функциональное сходство и позволяет экстраполировать знания от более к менее изученным последовательностям. Сравнение нескольких связанных последовательностей помогает понять ограничения, влияющие на их эволюцию, и, информативно для их структурных и функциональных исследований. Выравнивание особенно важно для эволюционного и филогенетического анализа, поскольку результаты этих анализов основаны исключительно на сходстве и различии, обнаруженных между исследуемыми последовательностями.

Выравнивания нуклеотидных последовательностей обычно визуализируются в виде вертикальных матриц, где последовательности находятся в строках, а гомологичные позиции образуют столбцы (Рисунок 1). Плоская матрица не в состоянии представить всю информацию, содержащуюся в последовательности и для определенных целей выравнивание, может быть более эффективно описано, например, как граф частичного порядка [6].

Наиболее часто используемые методы выравнивания игнорируют такие мутационные события, как дупликации, транслокации и инверсии и считают их комбинациями инсерций – делеций. С другой стороны, существуют мутационные события, которые могут привести к выравниванию последовательностей, но нарушают определение гомологии: например, после событий генной конверсии нуклеотиды, совпадающие при выравнивании, могут не происходить от общего «предка», а становятся шаблоном при помощи несвязанной последовательности. Вычислительные подходы к выравниванию обычно делятся на две категории: глобальные и локальные выравнивания, по тому, какое количество последовательностей сравнивается: парные и множественные выравнивания.

1.1. Парное выравнивание

Парное выравнивание – это описание двух последовательностей, независимо происходящих от общего предка.

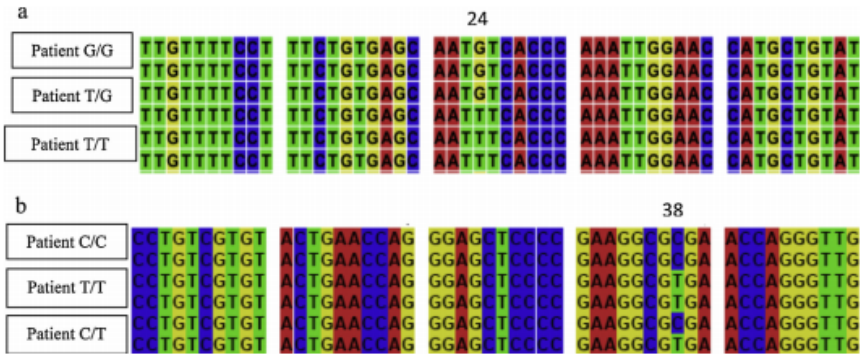

Рисунок 1. Выравнивание нуклеотидных последовательностей (выравнивание показывает полиморфизм) на примере гена IL28B rs8099917 (a) и rs12979860 (b). Произведено с помощью программного обеспечения CLC Workbench [7]

Динамическое программирование

Число расположений символов двух последовательностях длиной n и m задается как [8]

$$f\left(n, M\right) = \sum_{k=0}^{\min\{n,m\}} z^k \binom{m}{k} \binom{n}{k} \tag{1}$$

Для двух последовательностей, состоящих из 7, 8 и 10 нуклеотидов, количество возможных решений составляет 48639, 265729 и 8097453, соответственно, для двух последовательностей, состоящих из 107 символов количество решений превышает расчетное количество протонов во Вселенной [9]. С его помощью находят лучшие решения одно за другим, более поздние решения основываются на более ранних подчиненных решениях. При выравнивании последовательностей наименьшее расстояние редактирования для двух полных последовательностей является наилучшим решением для выравнивания самых последних символов; это решение зависит от решения для вторых последних символов, которое зависит от предыдущих и так далее [10].

Выравнивание двух последовательностей определяется путем заполнения матрицы расстояний редактирования для выравнивания двух подстрок соответствующей длины. На каждом шаге новое расстояние редактирования основывается на расстояниях редактирования предыдущих подрешений и стоимости операции редактирования, расширяющей выравнивание от предыдущего

подрешения до текущего. Для двух последовательностей x и y, которые состоят из символов $x_1, \ldots, x_n$ и $y_1, \ldots, y_m$, рекурсия динамического программирования может быть определена как:

Инициализация: $S(i, -1)$, $S(-1, j)$ до - ∞; $S(0, 0) = 0$

Рекурсия: для $i = 0, \ldots, n$; $j = 0, \ldots, m$

$$S(i, j) = \max \{S(i-1, j-1) + \sigma(x_i, y_i), S(i-1, j) + y, S(i, j-1) + y\} \tag{2}$$

Счет выравнивания: $S(x, y) = S(n, m)$.

Где $\sigma(x_i, y_i)$ – оценка для совпадающих символов и , а y – штраф за создание разрыва. Эти рекурсии дают только оценку $S(x, y)$ для выравнивания последовательностей и нахождение фактического сопоставления требует дополнительного алгоритма трассировки [10]. Алгоритм динамического программирования гарантирует нахождение оптимального выравнивания для двух последовательностей в рамках текущей схемы оценки, то есть затрат, определенных для различных операций редактирования. Решение имеет сложность $O(nm)$, где n и m – длины двух последовательностей для выравнивания. Это можно сделать в линейной памяти при небольшом компромиссе во время вычислений, используя алгоритм «разделяй и властвуй» [11].

Схема оценки представляет собой описание эволюционного процесса, ожидаемых частот инсерций и делеций разной длины и символов разных типов, в соответствии с которыми произошли две последовательности. Наше понимание процесса не является полным, и по практическим соображениям схема подсчета очков часто являются грубым упрощением. Также, надо помнить, что эволюция является стохастическим процессом и даже крайне маловероятные события могут произойти.

Методы выравнивания обычно представляют оценки для правок замещения с использованием логарифмических коэффициентов $\log(p_{ab}) / (f_a / f_b)$, где p_{ab} – это вероятность наблюдения символов a и b в гомологичных положениях в двух последовательностях через определенное количество времени после их расхождения, f_a и f_b фоновые частоты этих символов, дающие возможность

наблюдать такую пару только случайно [10]. Эволюционное расстояние оценивается одновременно с выравниванием, и оценка корректируется соответствующим образом; грубое приближение состоит в том, чтобы заранее определенные эволюционные расстояния от направляющего древа, чтобы пересчитать оценки для каждого выравнивания [12]. Делеции и инсерции могут быть смоделированы как вероятностные процессы, а их вероятности затем преобразованы в логарифмические значения, аналогичные лог.коэффициентам.

В ранних алгоритмах выравнивания использовалась стоимость линейного промежутка (т.е. стоимость операций вставки или удаления символов редактирования), что означало, что все события вставки и удаления предполагались длиной в один символ, а более длинные события считались продуктами нескольких односимвольных событий. Это было явно нереально и элегантное и эффективное с точки зрения вычислений решение для его исправления было предложено Гото [13] путем разделения открытия и расширения зазора на два параметра.

Этот ныне широко использующийся подход называется стоимостью аффинного разрыва. Это позволяет улучшить моделирование длин инсерций и делеций, стоимость аффиного промежутка все еще является упрощенным описанием очень разных процессов мутаций, вызывающих изменение длины последовательности, варьирующееся от коротких проскальзываний ДНК-полимеразы до длинных вставок транспозона [14]. Поскольку основное внимание уделялось распределению событий по длине и способам его моделирования, методы в значительной степени игнорировали тот факт, что вставки и удаления, как и замены, являются процессами мутации, зависящими от времени, а число событий зависит от эволюционного расстояния между последовательностями. Можно ожидать гораздо большего количества вставок-удалений между двумя отдаленно связанными последовательностями, чем между двумя тесно связанными, и, таким образом, стоимость открытия пробела не должна фиксироваться. Более правильный подход состоит в том, чтобы определить зависящую от времени скорость для вставки-удаления и отрегулировать стоимость открытия промежутка в соответствии с эволюционным расстоянием [15]. Фиксированная

стоимость расширения кажется более оправданной, и, когда событие вставки-удаления действительно происходит, его длина не должна зависеть от расстояния между последовательностями, которые выбраны для выравнивания.

Альтернативы динамического программирования

Алгоритм динамического программирования гарантирует поиск оптимального решения. Однако, это означает не оптимальное выравнивание, а одно из многих возможных комбинаций совпадения символов в двух последовательностях, имеющих одинаковое расстояние редактирования. Зачастую выбор метода выравнивания зависит от выбора программы. Основные методы выравнивания просто игнорируют альтернативные решения и «гарантируют» производить одинаковое выравнивание при каждом запуске. Практика, очевидно, выбрана для того, чтобы не вводить пользователя в заблуждение и позволить ему воспроизвести анализ и выводы, основанные на согласовании. И некоторым разработчиками была введена пост-обработка выравниваний. Наиболее широко известен метод «голова или хвост» (HoT), в котором используется расхождение между прямым и обратным выравниванием [16]. При использовании подхода HoT предполагается, что области выравнивания, которые являются последовательными в прямом и обратном решениях надежны, тогда как выводы о гомологии, которые различаются между двумя выравниваниями, считаются менее надежными. Ожидается, что число различий между прямым и обратным выравниванием будет коррелировать с расхождением последовательностей (идентичные последовательности правильно выровнены в любом направлении) и, следовательно, с достижимой точностью выравнивания. Однако высокий показатель HoT не гарантирует точного выравнивания, только два последовательных.

Метод HoT выдвигает на первый план очевидную проблему во многих программах выравнивания, но мера согласованности, которую он обеспечивает, является слишком грубой, чтобы эффективно использоваться, например, взвешивать выравнивающие столбцы в последующих анализах. Альтернативой сравнению только прямого и обратного решений, представляющих два из множества возможных различных выравниваний, является создание вариации выравнивания с использованием либо разных значений параметра промежутка

[17], либо разных топологий направляющего дерева [18]. Конечно, большинство различных топологий и значений параметров неверны (обычно не известно какие из них правильные) и, следовательно, могут привести к ошибке, которую можно избежать. Более общее решение будет состоять в том, чтобы позволить алгоритмам выравнивания случайным образом разорвать связи и предоставить пользователю возможность решать, сколько копий выравнивания, созданных с использованием топологий дерева и значений параметров рассмотреть по своему выбору. Реализация этого в рамках стандартного алгоритма будет генерировать вариации и обнаруживать некоторые неопределенные области, но не приведет к выравниванию, которое правильно представляет распределение возможных решений. Это может быть исправлено путем выборки пути выравнивания из апостериорного распределения [19], среднее значение по многим отобранным выравниваниям, тогда представляющее вероятности того, что сайты последовательности действительно гомологичны. Альтернативно, апостериорные вычисления могут использоваться для вычисления оценок надежности для решения с фиксированным выравниванием [20], и эти оценки затем используются в качестве объективной меры для удаления наиболее подозрительно выровненных столбцов или областей выравнивания.

Статистическое выравнивание

Эволюционный процесс может быть описан с помощью модели и решения, которое максимизирует связанные параметры, такие как скорость для различных событий мутаций или эволюционные расстояния между последовательностями, выведенные с использованием статистических методов. В дополнение к заменам символов, модель для выравнивания последовательностей также должна включать инсерции и делеции. Наиболее успешными из таких моделей были модели Торна, Кишино и Фельзенштейна, известные как модель TKF [21].

Выравнивание последовательностей в рамках эволюционной модели подчеркивает ограничения во многих традиционных методах анализа. В качестве примера, большинство основных методов не допускают смежные пропуски в двух противоположных последовательностях и, следовательно, не могут произ-

водить выравнивания, такие как $\begin{matrix} TCGAG \\ CCTAG \end{matrix}$.Хотя шансы на правильное восстановление таких решений могут быть небольшими, комбинации событий - две инсерции, две делеции или инсерции и делеция на одном и том же сайте или двух соседних сайтах - тем не менее, должны быть разрешены. Фактически, следует также рассмотреть возможность возникновения многих других событий, таких как вставка символа, за которым сразу следует удаление того же символа, хотя прямых доказательств этого не имеется. Однако некоторые ограничения носят чисто технический характер, например, эволюционно правильные схемы разрыва с соседними разрывами в двух последовательностях также могут быть разрешены в рамках детерминистских подходов. Большим преимуществом статистического моделирования проблемы является то, что неопределенность решения может быть количественно определена, правильно принимая во внимание, как возможность альтернативных решений, так и недостаток информации. Кроме того, методы выравнивания, основанные на правильной эволюционной модели непрерывного времени, хорошо масштабируются от сравнения двух последовательностей до совместного вывода филогении и множественного выравнивания последовательностей и описания более сложных биологических процессов [22]. Однако это относится только к теоретической основе модели, и анализ быстро становится чрезвычайно сложным в вычислительном отношении. К счастью, эти проблемы могут быть решены с помощью стандартных инструментов из статистики, при этом методы Марковской цепочки Монте-Карло (MCMC) используются наиболее успешно [23].

1.2. Множественное выравнивание

Учитывая, что алгоритм, находящий оптимальное решение для выравнивания двух последовательностей, имеет сложность $O(nm)$, было бы неправдоподобным предположить, что выравнивание k последовательностей длины l имеет сложность $O\left(i^k\right)$. Это так, но множественное выравнивание вводит новый фактор: филогения последовательности. Для более чем трех последовательностей филогенетическое дерево содержит внутренние ветви, и мутационные события, которые произошли в этих ветвях, совместно используются

последовательностями-потомками. Таким образом, выравнивание нескольких последовательностей требует не только выполнения выравнивания, но также знания филогении и восстановления наследственных последовательностей во внутренних узлах дерева. Однако, добавленная вычислительная нагрузка не слишком серьезна, поскольку даже сложность $O\left(i^k\right)$ невыполнима для более чем нескольких последовательностей.

Прогрессивное выравнивание

Практические методы множественного выравнивания основаны на эвристике и, следовательно, не гарантируют, что решение является глобально оптимальным. Некоторые предлагаемые эвристические методы, такие как оптимальная согласованная последовательность или минимальные попарные расстояния, могут быть привлекательны в качестве проблем информатики, но есть один подход, который сочетает вычислительные характеристики с биологическим реализмом: прогрессивное выравнивание.

Прогрессивные алгоритмы используют тот факт, что гомологичные последовательности создаются эволюционным процессом, который имеет древовидную иерархическую структуру родства (Рисунок 2a). Естественно попытаться отследить этот процесс и сначала выровнять наиболее тесно связанные и, следовательно, самые простые для выравнивания последовательности и отложить более сложные выравнивания до более поздней точки, надеясь получить некоторую дополнительную информацию из более ранних выравниваний. Кроме того, кластеризация последовательностей, основанная на их родстве, автоматически учитывает общую эволюционную историю некоторых последовательностей и должна правильно обрабатывать мутации, которые они унаследовали от своих общих предков. На практике прогрессивные методы выполняют итерацию парных выравниваний так, что каждое выравнивание кластеризует два узла, представляющих либо отдельные последовательности, либо выравнивания, и создает новый узел, представляющий это парное решение (Рисунок 2b). Многочисленные варианты этого подхода в основном отличаются только деталями того, как выровнять два узла и как преобразовать полученное выравнивание, чтобы представить новый узел [10; 24]. В матрице множественного выравнива-

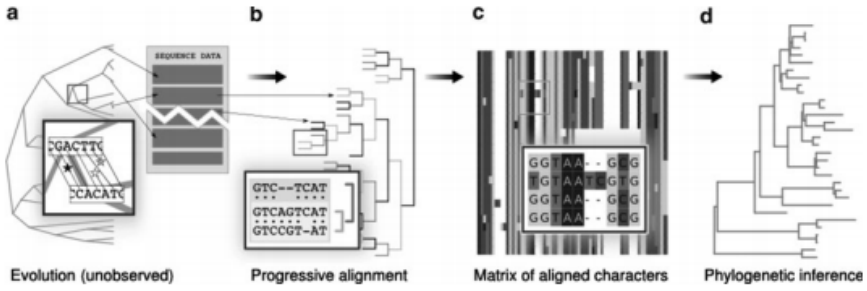

Рисунок 2. Схематическое представление всех процессов, нужных для построения филогенетического древа с использованием прогрессивного выравниванием. Эволюция последовательностей представляет собой древовидный процесс ветвления (a). (b), выравнивание последовательностей направлено на реконструкцию процесса, в ходе которого появились эти последовательности, и нахождение того, какие символы в последовательностях - потомках связаны через общих предков (указано линиями на (a)). Эти эволюционно гомологичные символы помещены в столбцы матрицы выравнивания (c). Прогрессивные алгоритмы (b) не пытаются выравнивать все последовательности одновременно, а разбивают множественное выравнивание на несколько попарных выравниваний: каждое попарное выравнивание объединяет два дочерних узла и создает новый наследственный узел, представляющий решение. Как это ни парадоксально, выравнивание последовательностей (b) требует филогении последовательности (d) [12]

ния инсерции и делеции выглядят очень по-разному. Делеция в наследственной последовательности вызывает разрыв выравнивания, что указывает на отсутствие гомологичных символов в этих положениях во всех последовательностях-потомках. Напротив, инсерция добавляет новые символы, которые передаются потомкам (и которые впоследствии могут быть удалены), и для правильного представления гомологии требует размещения пробела выравнивания во всех последовательностях, не являющихся потомками. Если событие не произошло глубоко в филогении, столбец выравнивания с правильно указанной вставкой имеет больше знаков пробела, чем реальные символы; обратное верно для делеции. Если мы предполагаем, что инсерции действительно происходят, мы должны принять - и даже ожидать - что наши выравнивания последовательностей содержат столбцы, в основном состоящие из знаков пробелов.

Выравнивания, правильно представляющие события инсерции, могут иметь фрагментированный и эстетически неприятный вид. Прогрессивное выравнивание основано на итерации попарного выравнивания. Однако сравнение двух последовательностей может только указывать на различия и не указывать направление изменения. Если две последовательности различаются по длине, мы знаем, что произошла либо инсерция, либо делеция, но мы не можем их различить (Рисунок 3a,b). Алгоритмы, реализованные в широко используемых программах прогрессивного выравнивания, игнорируют эту неопределенность и рассматривают все различия в длине как делеции. Когда разница в длине вызвана инсерцией, они заканчивают тем, что штрафуют одно событие несколько раз, и, учитывая высокую стоимость этого и низкую вероятность размещения множества пропусков в правильном положении, редко удается создать столбцы выравнивания, правильно представляющие событие инсерции [25] . Искусственные исправления, такие как штрафы за пробелы в конкретных сайтах, которые снижают стоимость многократного штрафования за перекрывающиеся пробелы, не могут исправить алгоритмический недостаток и вызвать дополнительную ошибку при размещении делеций. Последовательности в множественном выравнивании, однако, связаны, и можно разрешить эту неопределенность в типе события мутации, используя соседние последовательности. Алгоритм «с пониманием филогении» [25] помечает вновь созданные разрывы как неопределенные, а затем использует информацию внешней группы из последующих выравниваний, чтобы сделать вывод, была ли разница в длине вызвана вставкой или удалением (Рисунок 3). На отмеченных позициях новые промежутки для инсерций могут быть созданы без дополнительного штрафа; после повторного использования пробела флаг может быть сохранен для дополнительных свободных пробелов, или событие подтверждается как инсерция, а флаг преобразуется в постоянный, что предотвращает любое последующее сопоставление этого сайта. Если наилучшее выравнивание соответствует отмеченным символам, событие выводится как делеция, а флаги удаляются.

Естественным ограничением алгоритма, учитывающего филогению, является то, что он требует от филогенетической информации выводить тип собы-

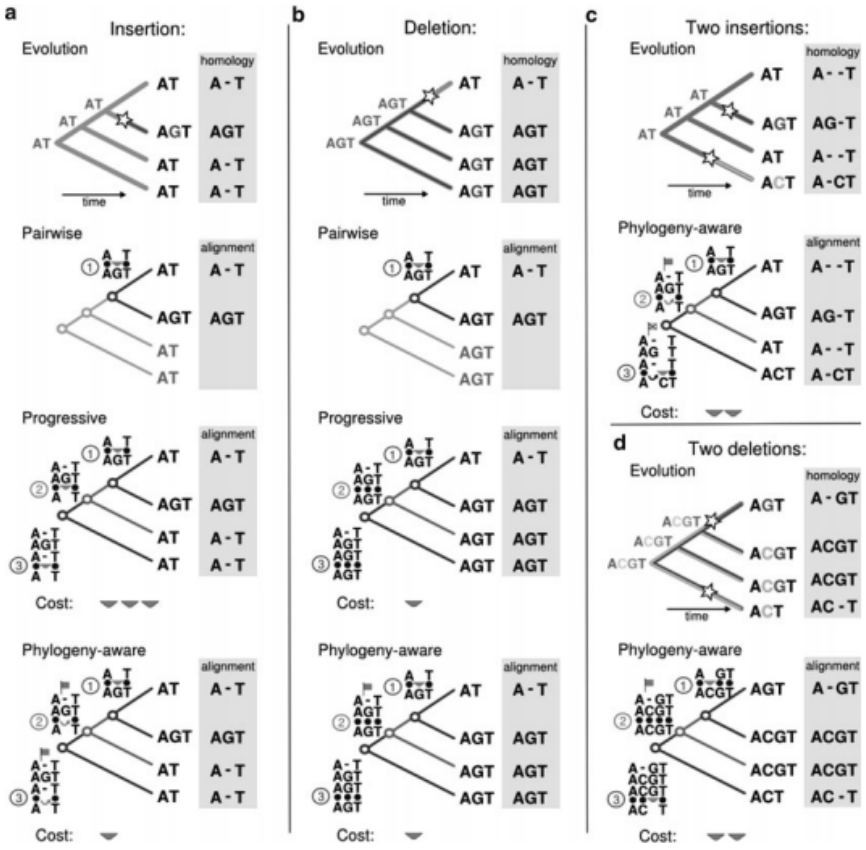

Рисунок 3. Простые филогении с одноосновной инсерцией и делецией (a, b). При парном выравнивании два события выглядят одинаково. Прогрессивный алгоритм повторяет парные выравнивания и должен создавать новый пробел для вставленного символа на каждом этапе. Это наказывает одно эволюционное событие несколько раз (обозначено треугольниками). Алгоритм, учитывающий филогению, помечает позиции с пропусками, а затем позволяет размещать новые пропуски в помеченных позициях без дополнительного штрафа. В случае удаления лучшее выравнивание найдено путем сопоставления символов, и помеченные удалены. Повторное использование помеченного пробела может рассматриваться как подтверждение вставки, а сайт, помеченный постоянным флагом (пересекается) (c). Это предотвращает последующее сопоставление сайта и гарантирует, что независимые вставки в гомологичных положениях хранятся отдельно. Это не влияет на выравнивание соседних событий удаления (d) [12]

тия инсерция-делеция и допускает ошибки, когда дерево направляющего выравнивания неверно. Это также предполагает, что последовательности довольно тесно связаны, и в дочерних ветвях не произошло никаких перекрывающихся событий, ведущих к двум последовательностям (или узлам), которые выровне-

ны. Когда эти условия выполняются, алгоритм хорошо масштабируется, и его производительность не страдает даже при выравнивании большого количества последовательностей; алгоритм корректно обрабатывает несколько инсерций в гомологичных позициях, сохраняя при этом независимые делеции в соседних позициях отдельно (Рисунок 3c, d).

Основным недостатком прогрессивных алгоритмов является их «жадность»: каждое попарное выравнивание фиксирует часть общего решения выравнивания, которую более поздние выравнивания не могут изменить. Это проблематично, поскольку промежуточные этапы процесса выравнивания могут иметь несколько локальных оптимумов, из которых только один так же является глобально оптимальным, или для достижения глобального оптимума может потребоваться выбор локально неоптимальных решений. Существуют эвристические расширения для прогрессивного выравнивания, которые в первую очередь пытаются избежать таких «жадных» ошибок или исправить их после первоначального выравнивания.

Простое прогрессивное выравнивание учитывает только два узла, которые выровнены, и не видит, как различные способы выравнивания этих последовательностей будут соответствовать последовательностям, которые будут добавлены позже. Этот тип внешней информации из связанных последовательностей может быть включен в решение о парном выравнивании с использованием согласованных объективных функций. Эти функции ищут попарное выравнивание между последовательностями A и B, максимально совместимое с независимым выравниванием A и B с последовательностями внешней группы C, D, E и т.д. Библиотека парных выравниваний, используемая для вычисления согласованности, может быть любой, что позволяет методу использоваться в качестве «мета-выравнивателя», объединяющего решения многих других методов выравнивания [24]. В первоначальном подходе согласованности используется набор дискретных выравниваний, которые взвешиваются в соответствии с их сходством [26], но их можно обобщить, чтобы рассмотреть все возможные выравнивания и использовать их апостериорные вероятности в качестве весов. Если нельзя избежать ошибок на этапе прогрессивного выравнивания,

можно попытаться исправить их позже. Варианты методов исправления после выравнивания обычно разбивают выравнивание на два подмножества, а затем повторно выравнивают подмножества обратно в полное выравнивание; если это выравнивание лучше предыдущего, оно принимается и процесс продолжается. Это имеет две основные проблемы: для эволюционно правильного решения два подмножества должны быть объединены в филогенетическом контексте представляемых ими поддеревьев, то есть, выровнены по наследственным узлам, которые значимо представляют состояния символов, а также наличие и отсутствие символов в этих точках. ; и предлагаемое решение выравнивания должно быть оценено таким образом, чтобы инсерция-делеция и замены были правильно учтены с учетом филогенетических связей между последовательностями. К сожалению, это не относится к основным методам, поддерживающим исправление выравнивания. Вторая проблема, сложность измерения качества выравнивания, не только влияет на методы исправления, но и является более общей проблемой: любая стратегия поиска выравнивания бесполезна, если нельзя правильно оценить решения, которые она предлагает.

Многократное выравнивание последовательностей требует филогении последовательности, но это не может быть точно оценено до выравнивания последовательностей (Рисунок 2). Методы выравнивания способны генерировать выравнивания, поэтому должен быть способ разорвать эту циклическую зависимость. Наиболее широко используемый способ состоит в том, чтобы сделать парные выравнивания «все против всех», а затем построить дерево направляющих из парных генетических расстояний, оцененных по ним, с использованием метода кластеризации, такого как соединение соседей [27]. К сожалению, парные выравнивания последовательностей редко согласуются с выравниванием тех же самых последовательностей в множественном выравнивании, и начальные оценки расстояния могут быть ошибочными. Ошибки могут быть критичными, так как направляющее дерево, основанное на попарных расстояниях, определяет порядок выравнивания последовательностей: промежуточные выравнивания предназначены для представления наследственных узлов, а неправильная направляющая филогения может указывать на наследственные узлы,

которые на самом деле не существовали. Как правило, методы выравнивания пытаются устранить эту ошибку, оценивая новое направляющее дерево из первых множественных последовательностей выравнивания и повторного выравнивания в соответствии с этим, возможно, повторяя, пока результат не сойдется. Однако существует риск, что на первое выравнивание влияет филогения, использованная для его генерации [28], и оценки расстояния отражают эти начальные ошибки, и никакая итерация не может избежать локального максимума. Учитывая, что влияние неправильной филогении направляющей выравнивания на выравнивание не было изучено и маловероятно, что это случайный шум. Поскольку выравнивание зависит от филогении последовательности, а вывод филогении основан на выводе гомологии из выравнивания (Рисунок 2), очевидным решением является вывод выравнивания и филогении одновременно. Дополнительным преимуществом такой совместной оценки является то, что инсерции и делеции могут быть правильно учтены и точно внесли вклад в оценки эволюционной связанности [23]. Основным недостатком является то, что это требует больших вычислительных ресурсов: поиск топологии дерева уже отнимает много времени, и, когда каждая операция, влияющая на топологию дерева, также требует повторного выравнивания некоторых последовательностей, текущие методы для совместного вывода выравнивания и филогении ограничены относительно небольшими наборами данных [29]. С другой стороны, совместная оценка с байесовскими методами обеспечивает огромное преимущество перед любым другим существующим методом, объединяя результат со всеми филогениями и всеми выравниваниями. Для больших наборов данных совместная оценка выравнивания и филогении все еще может быть слишком сложной в вычислительном отношении, но есть место для улучшений в более традиционных подходах. Одним из наиболее очевидных недостатков методов прогрессивного выравнивания является упрощенный алгоритм на основе расстояний, который используется для построения направляющей филогении. Можно вручную повторить выравнивание последовательности с помощью древа, выведенного с использованием более продвинутого филогенетического метода, но эти два шага также можно легко объединить в одном приложении, которое применяет ин-

струменты итеративным образом. Использование более сложных алгоритмов логического вывода древа, по-видимому, уменьшает ошибки, вызванные древом направляющим, и такая «совместная оценка» выравнивания и древа должна обеспечить значительные улучшения для анализа очень больших наборов данных [30]. Однако методы «совместной оценки» все еще страдают от той же проблемы, что и другие методы выборки, которые не основаны на эволюционной модели: как оценить решения и определить хорошие? Максимальная оценка правдоподобия, которая явно игнорирует все инсерции и делеции, не является идеальным выбором для сравнения выравниваний последовательностей, которые отличаются только расстановкой пробелов [31].

Непрогрессивные методы

Алгоритм Нидлмана-Вунша [32] производит глобальное выравнивание между последовательностями. Однако глобальное выравнивание может быть нежелательным, если последовательности не являются полностью коллинеарными или они настолько расходятся, что значимо точное выравнивание не достижимо по всем сайтам последовательностей. Хотя «чрезмерное выравнивание» несвязанных областей последовательности может быть уменьшено путем корректировки оценки замещения символов и вставки-удаления и правильной обработки событий инсерций в множественных выравниваниях последовательности, прогрессивного выравнивания, и программы выравнивания имеют тенденцию сделать вывод о многих ложных гомологиях [33]. Проблема создания множественных выравниваний только частично гомологичных последовательностей породила категорию методов непрогрессивного выравнивания, и некоторые из них были позже обобщены для генерации глобальных выравниваний, подобных тем, которые были получены прогрессивными методами [24] . Одним из наиболее широко используемых непрогрессивных методов является подход выравнивания сегментов [34], который строит множественное выравнивание из диагоналей с высокими показателями - коротких очень похожих сегментов последовательности - которые согласуются между различными парами последовательностей. Диагонали сортируются в соответствии с их показателями веса и перекрываются с диагоналями, найденными в других парах последователь-

ностей; начиная с самых высоких баллов, диагонали затем объединяются во множественное выравнивание, заполняя отверстия в окончательном выравнивании знаками разрыва. Преимущество этого подхода состоит в том, что сегменты последовательности рассматривают набор последовательных сайтов и, таким образом, содержат больше информации, чем отдельные сайты. Кроме того, если область последовательности не показывает сходства с какой-либо другой последовательностью, она не будет считаться гомологичной и будет оставлена невыровненной (однако последовательность будет включена в матрицу выравнивания в ее контексте, и ее можно случайно считать выровненной и использовать для последующего анализа). Недостатком сегментного подхода является то, что идея может не соответствовать биологическим фактам большинства задач выравнивания последовательностей, и он не использует всю доступную информацию: сегменты обнаруживаются в парных выравниваниях последовательностей, которые, как известно, иерархически связаны, и процесс построения множественного выравнивания не использует филогению последовательностей или пытается реконструировать наследственные последовательности.

2. Бактериальные геномы

Бактерии обладают компактной геномной архитектурой, у них нет сплайсосомных интронов, но есть интроны первой группы. Геном большинства бактерий представлен одной кольцевой хромосомой, но есть и бактерии с линейной хромосомой. Бактериальные геномы имеют размер от 130 000 до примерно 14 000 000 п.н [35] . Размер бактериального генома сильно коррелирует с числом генов (около 90% всей ДНК кодирует РНК или белки), большинство генов уникальны и число семейств близко по значению к общему числу генов . В среднем, размер генома около 4 мпн, а длина гена – 1000 п.н. Большинство генов организованы в опероны.

Инсерции, вызванные горизонтальным или латеральным переносом генов и дублированием генов, имеют тенденцию включать перенос больших количеств генетического материала. Предполагая отсутствие этих процессов, геномы будут иметь тенденцию к уменьшению в размере в отсутствие селективного ограничения. Доказательства смещения делеции присутствуют в соответствующих размерах геномов свободноживущих бактерий, факультативных и недавно ставшими патогенами и облигатных патогенов и симбионтов. Свободноживущих бактерий имеют больший размер популяции и подвергаются более сильному селективному давлению.

2.1. Мутации и отбор

Мутации являются неоднозначными во всех организмах, являясь основным источником вариаций, используемых для эволюционной адаптации, и в то же время являются преимущественно вредными и источником генетических нарушений. Следовательно, исследователи долго искали основные факторы, влияющие на эволюцию скорости мутаций. Некоторые утверждают, что уровень мутаций в организме отражает баланс между вредным эффектом мутаций и физиологическими ограничениями, при этом дальнейшее увеличение точности репликации ограничивает скорость синтеза ДНК, необходимую для эффективного производства дочерних клеток. Однако точность воспроизведения может быть улучшена без значительного сокращения времени удвоения, и прокариоты подвергаются высоким скоростям деления клеток и имеют низкую частоту

мутаций [36], предполагая, что точность репликации не ограничивает скорость производства дочерних клеток.

Общая взаимосвязь, описывающая изменение скорости мутаций, была предложена Drake et al. [37], который предположил, что частота мутаций на нуклеотидный сайт обратно пропорциональна размеру генома у бактерий и одноклеточных эукариот, так что существует постоянная мутация 0,003 на гаплоидный геном на клеточное деление. Однако, когда стали доступны прямые оценки частоты мутаций для дополнительных организмов, общая взаимосвязь между размером генома и частотой мутаций стала менее очевидной, даже при масштабировании на количество клеточных делений на поколение у многоклеточных видов [36]. Поскольку мутации, как правило, вредны, отбор позволяет снизить частоту мутаций в геноме путем увеличения точности репликации и репарации ДНК, пока дальнейшие улучшения не станут слишком незначительными, чтобы преодолеть силу случайного генетического дрейфа.

2.2. Инверсии

Инверсия – это хромосомная перестройка, при которой участок хромосомы поворачивается на 180 градусов. Инверсии делятся на два типа: парацентрические (не включают центромеры, и оба перелома происходят в одном плече хромосомы) и перицентрические (включают центромеру и в каждом плече есть точка разрыва). Как у эукариот, так и у прокариот архитектура генома и его эволюция часто не случайны. Фундаментальный вопрос в этом отношении заключается в том, вызывается ли неслучайная организация генома смещенными мутационными процессами, например, связанными с динамикой рекомбинации или с помощью отбора. У прокариот многие аспекты неслучайной организации генома были правдоподобно отнесены к последним. Это включает кластеризацию функционально родственных генов в опероны и обогащение необходимых генов на ведущей цепи репликации, где они избегают встречных столкновений между активными ДНК- и РНК-полимеразами[38]. Кроме того, гены с высокой экспрессией (рРНК, тРНК, гены рибосомных белков) обычно обнаруживаются вблизи ori как у бактерий, так и у архей, согласуется с отбором по повышенной

дозировке: последовательности, близкие к ori, реплицируются раньше и поэтому временно присутствуют в большем количестве копий.

Что касается других аспектов эволюции прокариотического генома, то тяжело определить, вызвана ли неслучайная структура генома отбором, смещенными мутационными процессами или их комбинацией. В частности, это включает частоту встречаемости крупномасштабных инверсий, которые составляют основной источник структурного разнообразия в прокариотических геномах [39] . Любопытно, что инверсии у прокариот выглядят преимущественно симметричными; то есть их конечные точки примерно равноудалены от начала репликации, генерируя заметные X-паттерны при выравнивании целого генома Инверсии, симметричные оси начала координат (ori-ter), первоначально наблюдались в нескольких близкородственных бактериальных геномах, включая пары Chlamydia, Mycobacterium и Helicobacter spp., и впоследствии были выделены во множестве других сравнений генома, особенно с участием γ-протеобактерий (Yersinia, Blochmannia, Buchnera) и Bacilli (Lactobacillus (Bacillus, Streptococcus), но также и археон одного происхождения Pyrococcus [40]. Эти наблюдения породили представление о том, что предвзятые инверсионные ландшафты являются преобладающей особенностью эволюции прокариотического генома.

Генезис инверсий является приблизительно случайным, но симметричные инверсии с большей вероятностью сохранятся после действия очищающего отбора, поскольку в среднем они менее разрушительны для адаптивной архитектуры генома. Примечательно, что, хотя потенциально большое количество локусов транслоцируется в противоположный реплихор, они сохранят свою первоначальную ориентацию ведущей / отстающей цепи. Это может быть важно не только для того, чтобы избежать конфликтов репликации и транскрипции, но также для связывающих мотивов, которые функционируют поляризованным образом, таких как FtsK-ориентирующие полярные последовательности, которые облегчают перемещение FtsK к концу. Напротив, инверсии в пределах одного и того же реплихора неизбежно приводят к переключению ведущих / запаздывающих цепей. Симметричные инверсии также не изменяют расстояние

определенного геномного элемента до ori или ter, таким образом избегая потенциально вредных изменений в гене и смещения мотивов, функция которых зависит от их близости к ori или ter (например, блоки DnaA, parS) [41].

Как и у прокариот, большие инверсии у эукариот более распространены. Более длинным хромосомным инверсиям способствует отбор относительно более коротких инверсий, поскольку они подавляют рекомбинацию между большим количеством генетически удаленных локусов. Большие инверсии могут составлять значительную долю всего генома. Например, восемь инверсий в D. melanogaster представляют 71,25 мб или 43% генома. Пять инверсий в G. morhua представляют 51 мб или более 6% генома. Одиночные инверсии, такие как инверсия в 100 мб у белогорлого воробья (Zonotrichia albicollis), представляют приблизительно 10% генома этого вида. Таким образом, неудивительно, что число генов в любой данной инверсии может быть большим, составляя в среднем 418 генов [42].

Длительное удержание внутривидовых полиморфизмов инверсии будет облегчено, если будет задействована некоторая форма балансирующего отбора. Для некоторых животных в качестве вероятного эволюционного процесса была предложена одна из нескольких форм балансирующего отбора (например, частотно-зависимый отбор, антагонистическая плейотропия, диссортативное спаривание, избыточное преобладание или пространственно-временной выбор), способствующая инверсии [42]. Например, у насекомых Timema cristinae, загадочные цветовые фенотипы, по-видимому, связаны с инверсиями, которые разошлись миллионы поколений назад [43]. Этот пример подтверждает новое мнение о том, что сбалансированный отбор играет важную роль в поддержании генетической изменчивости даже в течение продолжительных периодов времени.

У североамериканских D. melanogaster 3RP-клин оставался стабильным в течение > 40 лет, и частоты сильно коррелируют с климатическими факторами, независимо от структуры популяции [Kapun et al., 2016]. Работая над одним и тем же видом, Rane et al. [44], обнаружили, что тот же регион показал сильную дифференциацию между тропической и умеренной зоной, и идентифицировал

связанные с инверсией полиморфизмы одиночных нуклеотидов (SNP), расположенные в генах, связанных с признаками физического развития, которые демонстрируют параллельную дифференциацию вдоль североамериканской линии.

У воробьиных птиц родственные симпатрические виды значительно чаще отличаются по инверсии, чем родственные аллопатрические виды, причем число различий инверсии лучше всего объясняется уровнем географического перекрытия. Другая работа, сфокусированная на видах, населяющих неоднородные среды, имеет документально подтвержденные связи между адаптацией к окружающей среде и накоплением инверсий, связанных с репродуктивной изоляцией. Одним из примеров является сравнительное сопоставление связей в сестринских обезьяньих цветах Mimulus lewisii и Mimulus cardinalis, в примере с экологическим видообразованием. Картирование выявило две инверсии, специфичные для M. cardinalis [45] , и выявило, что как цветочные QTL, так и QTL, связанные с адаптацией к окружающей среде, кластеризованы в предполагаемых перестроенных регионах, и что все QTL для мужской стерильности, включая два преобладающих локуса, картированы в областях с подавленной рекомбинацией[45]. Это является убедительным доказательством роли инверсий в создании и укреплении экологических барьеров для потока генов между этими двумя таксонами.

2.3. Инсерции и делеции

Инсерция – это мутация, которая представляет собой добавление одной или нескольких нуклеотидных пар оснований в последовательность ДНК. Вставки могут быть любого размера от одной пары оснований, неправильно вставленной в последовательность ДНК, до участка одной хромосомы, вставленного в другую. Делеция представляет собой мутацию, при которой часть хромосомы или последовательность ДНК удаляется. Любое количество нуклеотидов может быть удалено от одного основания до целого фрагмента хромосомы. И в том и в другом случае если количество вставленных/вырезанных нуклеотидов не кратно 3 происходит сдвиг рамки считывания.

Природа и относительная важность молекулярных механизмов и эволюционных сил, лежащих в основе изменения размера генома, была предметом

интенсивных исследований и дискуссий. Изменение размеров генома не всегда может происходить на уровне, когда естественный отбор достаточно силен, чтобы предотвратить генетический дрейф [46] . Кроме того, вероятность фиксации слегка вредных делеций или вставок была бы выше у видов с меньшими эффективными размерами популяции, где естественный отбор действует менее эффективно (раздел 2.4). С другой стороны, ряд корреляционных связей между размером генома и фенотипическими признаками, такими как размер клеток позволяет предположить, что естественный отбор и адаптивные процессы также формируют эволюцию размера генома. Разделяя относительную важность этих двух сил (дрейфа и отбора), требуется лучшее понимание способа и процессов, с помощью которых ДНК вставляется и удаляется в течение длительных эволюционных периодов в разных таксонах.

Исследование геномной ДНК, полученной и потерянной в ходе эволюции эйтериев и птиц показало то, два таксона показывают относительно небольшие межвидовые различия в размерах генома по сравнению с другими (такими, как растения или насекомые)[47]. Одна из интерпретаций этого очевидного «застоя» в размере генома может заключаться в том, что эти линии просто испытывали относительно небольшие количества прироста и потери ДНК в процессе эволюции. Анализ Kapusta et al. показывает, что это явно не тот случай: на протяжении всей эволюции эйтериев и птиц происходило значительное увеличение и потеря ДНК. Например, количество ДНК, полученное с помощью специфической для линии передачи транспозиции в линии мыши, способствовало чистому приросту ДНК, эквивалентному 33% текущего содержания генома, тогда как эквивалент 44% содержания генома был потерян за тот же период времени [47]. Родословная дятла является еще одним ярким примером. Среди линии птиц этот вид испытал наибольший прирост ДНК (255 Мб, преимущественно через транспозицию CR1 LINE [48]), но также и наибольший объем потери ДНК (424 Мб, что эквивалентно примерно одной трети генома) по сравнению с прошедшими 70 млн. лет, в результате чего текущий размер генома сопоставим с размером других современных видов птиц [47]. Таким образом,

наши данные показывают ранее недооцененный уровень эластичности в геномах eutherian и avian.

Эти результаты позволяют выявить общую закономерность эволюции генома по основным линиям птичьего и эвтериального происхождения, при которой (часто большое) количество ДНК, полученное в результате специфической для линии передачи транспозиции, по существу сбалансировано количеством ДНК, потерянным за тот же период времени. Этот аккордеонный процесс помогает объяснить относительное поддержание размера генома в филогенезе эвтериев и птиц. Это особенно очевидно у птиц, которые демонстрируют положительную корреляцию между усилением ДНК и потерей ДНК. Таким образом, эти результаты показывают, что относительно небольшой размер генома птиц связан не только с недостатком транспозиции в этих линиях, как предполагалось ранее [49], но скорее с результатом динамического взаимодействия между ТЕ-опосредованное приобретение ДНК и последующая потеря ДНК (как предполагается в [50]).

Предыдущие исследования по оценке потери ДНК были в основном сосредоточены на делециях в ТЕ-последовательностях, которые устанавливают относительно небольшой верхний предел для размера наблюдаемых событий (потому что ТЕ-копии редко превышают 10 кб). Показано, что частота делеций, оцененная с помощью этого подхода, является основным предиктором эволюции размера генома у насекомых, растений и нескольких позвоночных [51; 52]. Тем не менее, вопрос о том, может ли вариация в скорости небольших делеций действительно отражать вариации размера генома, наблюдаемые между таксонами, был поставлен под сомнение [53]. Действительно, количественные оценки из ограниченных сравнительных наборов данных показали, что одни только микроделеции не могут объяснить степень сокращения генома, наблюдаемого в некоторых линиях позвоночных [54]. Оценки скорости микроделеции (1–30 п.н.) показывают, что этот тип события может объяснить только незначительную долю содержания ДНК, потерянного во время эволюции птиц и эвтериев, и как таковые не делают по-видимому, основной вклад в эволюцию размера генома в этих таксонах.

Результаты, полученные Kapusta et al. показывают, что делеции среднего размера (от 31 до 10 т.п.н.) играют большую роль, чем микроделеции, в объяснении наблюдаемого межвидового изменения потери ДНК. В целом, однако, делеции микро- и среднего размера, по-прежнему составляют ограниченную долю (в среднем 9,5–40% и 20%) от общей потери ДНК в эволюции эвтериев и птиц. Эти данные свидетельствуют о том, что подавляющее большинство потерь ДНК у амниот обусловлено относительно большими делециями (> 10 т.п.н.). Такие большие делеции сложно систематически обнаруживать с помощью доступных в настоящее время сборок генома, что не позволяет измерить скорость этих событий по линиям птиц и млекопитающих. Точно так же состоялись большие сегментальные делеции у общего предка птиц (118 событий на общую сумму 58 МБ и до 2,1 МБ на событие) [48]. Такие большие делеции в сочетании с явным количеством потери ДНК в некоторых исследованных линиях млекопитающих и птиц (до 37,9% и 22,6% от содержания ядерной ДНК, соответственно) подчеркивают необязательность большой доли геномной ДНК у этих животных [55], однако это не исключает того, что процесс потери сегментарной ДНК сыграл важную роль в развитии фенотипической эволюции. На самом деле, есть серьезные намеки на то, что большие делеции вызывали значительную потерю генов у птиц (274 кодирующих белок гена) с потенциально глубокими фенотипическими последствиями [56] . Предсказуемое улучшение сборки генома с помощью секвенирования третьего поколения предоставит способ более непосредственно проверить гипотезу о том, что крупные события делеции играют заметную роль в эволюции генома амниот. Кроме того, разрешение тандемных повторов, которые обычно отсутствуют в текущих сборках, улучшится, что позволит количественно оценить их вклад в динамику генома.

3. Описание алгоритма

3.1. Структура алгоритма

Работу алгоритма можно поделить на 3 стадии: выделение ”зон несоответствий”, поиск и замена в данных зонах инверсий, классическое выравнивание последовательностей.

Выделение зон несоответствий

Так как при хромосомных перестройках происходит сильное изменение генома, данные участки характеризуются большим количеством не совпадающих нуклеотидов. Таким образом при выравнивании в данных участках можно наблюдать большое количество разрывов. Исходя из того, что инверсии находятся в этих зонах (если при инверсии нуклеотиды в большем количестве совпадают, то такая инверсия не сильно влияет на результат выравнивания), первым этапом выполнения программы является выделение зон несоответствий, в которых вероятнее всего произошла либо вставка/делеция, либо инверсия. Для оптимизации поиска данных зон алгоритм проходит по первой последовательности 1 раз, не создавая в памяти ее копию:

1. Из первой последовательности, начиная с индекса курсора первой последовательности (при первой итерации курсор инициализирован нулем), извлекаются n нуклеотидов, данный кусок объявляется референсной подпоследовательностью.
2. Во второй последовательности, начиная с индекса конца прошлой найденной схожей подпоследовательности (для первой итерации данный индекс инициализирован нулем) производится поиск похожей на референсную подпоследовательность участок с шагом в stride. Похожим считается участок, между которым и референсной подпоследовательностью расстояние Левенштейна меньше коэффициента r. Параметр r является порогом толерантности при определения зон неточностей. При большом его значении, мы имеем риск относить гомологичные зоны к зонам неточности, при маленьком - зоны неточности к гомологичным зонам.

3. Если в радиусе m от начала поиска схожая подпоследовательность не была найдена, то референсная подпоследовательность помечается как зона несоответствия.
4. При нахождении такого участка найденная область и референсная подпоследовательность помечаются как гомологичные зоны.
5. Курсор поиска первой последовательности сдвигается на n.
6. Шаги 1,2,3,4 повторяются до окончания первой последовательности

Поиск и замена инверсий в найденных зонах неточностей

После выявления гомологичных зон происходит анализ зон неточностей для поиска внутри них инверсий. В качестве зон неточностей берутся участки, находящиеся между одинаковыми гомологичными зонами.

В найденных зонах неточностей могут присутствовать как индел, так и инверсии. Для ускорения работы программы анализ каждой отдельной зоны неточности происходит параллельно. Разберем анализ одной зоны неточности:она представляет из себя зону неточности первой последовательности (назовем ее верхняя подпоследовательность) и зону неточности второй последовательности (назовем ее нижняя подпоследовательность)

1. На нижней подпоследовательности применяется операция комплементарной инверсии (симуляция настоящей инверсии).
2. К перевернутой нижней подпоследовательности и верхней подпоследовательности применяется классический алгоритм выравнивания.
3. Происходит поиск зон, в которых длина участков с совпадающими нуклеотидами больше минимально значимой длины инверсии h. Данные зоны помечаются как инверсии.
4. В первой последовательности на зоны инверсий применяется операция комплементарной инверсии.

Классическое выравнивание

Завершающим этапом работы алгоритма является классическое выравнивание измененной первой последовательности и второй последовательности. Если в первой последовательности произошла значимая инверсия, то зона, в которой она произошла на 1 этапе будет отнесена к зоне неточности. После это-

го при перевороте нижней подпоследовательности инверсия будет выровнена с идентичным участком в верхней подпоследовательности. После этого так как ее длина выше минимально значимой длины инверсии, то в верхней последовательности данные участки будут заменены на инвертированные и выравнены со второй последовательностью на этапе классического выравнивания, увеличив оценку схожести последовательностей. Итоговая метрика схожести последовательностей должна быть скорректирована с учетом количества возникших инверсий.

Так как вероятность инверсии ниже вероятности индел примерно в 10 раз, мною была предложена следующая итоговая формула оценки схожести последовательностей:

$$p = \frac{2 * t - 0.1 \cdot g - i}{n + m} \tag{3}$$

Где t - число совпадений, g - число разрывов, i - число инверсий, n и m - длины первой и второй последовательности, соответственно.

3.2. Интерфейс программы

Программа представляет из себя скрипт, написанный на языке программирования Python 3. Для запуска в командной строке необходимо указать путь к файлам, в которых хранятся сравниваемые геномные последовательности. После выполнения программы в командную строку выводятся число найденных инверсий и итоговая оценка схожести последовательностей.

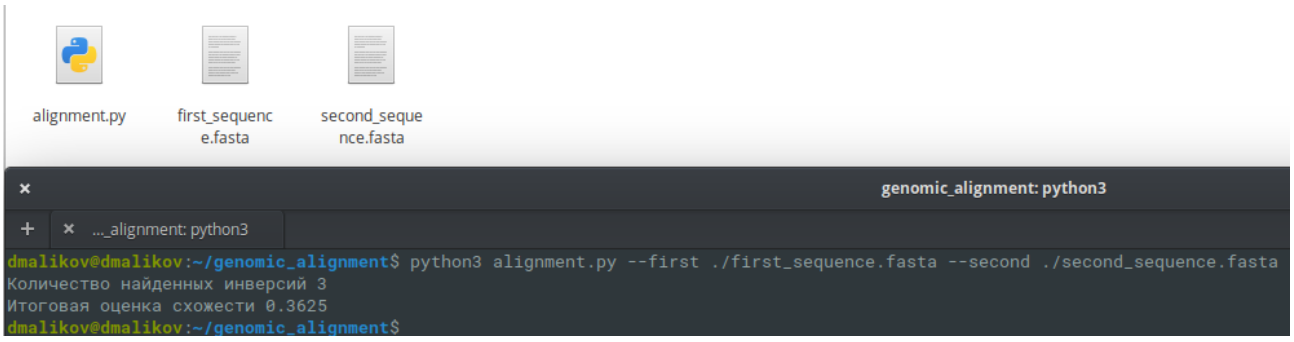

Рисунок 4. Пример работы алгоритма: на вход через командную строку подается путь к файлам, в которых хранятся геномные последовательности, на выходе выводится результат работы алгоритма

Результаты

Для отладки программа выдает также индексы найденных инверсий и оценку классического алгоритма выравнивания. Для тестирования работы алгоритма мы сгенерировали следующие последовательности:

1: ACTGTCACCCCACCACTGTCACCCCACCACTGTCACCCCACCACTGTCACCCCACC

2: TGACAGTCCCCCCCCTGACAGTCCCCCCCCTGACAGTCCCCCCCCTGACAGTCCCCCCCC

Рисунок 5. Сгенерированные последовательности 1 и 2. Красным отмечена инверсия, зеленым - гомологичная зона, синим - индел

В данном примере в последовательности 1 каждые 7 нуклеотидов с шагом в 14 были заменены на инверсию. Рисунок 6 является результатом работы программы на последовательностях 1 и 2.

```
Оценка классического алгоритма 0.6470588235294118
Индексы найденных инверсий:
[[[0, 7]], [[14, 21]], [[28, 35]], [[42, 49]]]
Количество найденных инверсий 4
Итоговая оценка схожести 0.8857142857142858
```

Рисунок 6. Результат работы алгоритма на сгенерированных последовательностях 1 и 2

Рассмотрим пример, когда инверсия составляет значительную часть последовательности, для этого сгенерируем последовательности 3 и 4:

3:TTTTTTTCCCCCCCACTGACTGAACTGAACTGAACTGACTGGACTGACTGAACTGAACTGAACTGACTGG
4:CCCCCCCAAAACAGTCAGTTCAGTTCAGTTCAGTCAGTCCAGTCAGTTCAGTTCAGTTCAGTCAGT

Рисунок 7. Сгенерированные последовательности 3 и 4. Красным отмечена инверсия, зеленым - гомологичная зона, синим - индел

Результат работы алгоритма на последовательностях 3 и 4 представлен на рисунке 8 .

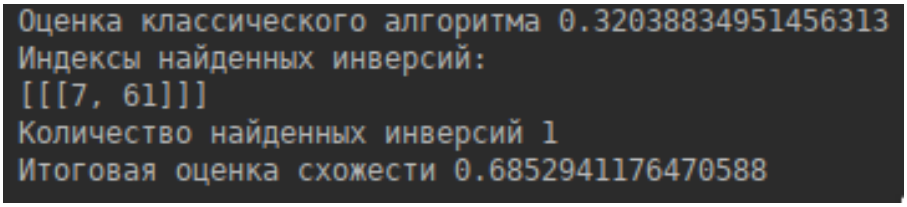

Рисунок 8. Результат работы алгоритма на сгенерированных последовательностях 3 и 4

Выводы

При сравнении последовательностей, в которых присутствует несколько маленьких инверсий и индел, разработанный алгоритм находит все инверсии с идеальной точностью. Так как они являются значимыми, алгоритм производит аналогичные инверсии первой последовательности, тем самым увеличивая итоговую оценку схожести.

В случае большой значимой инверсии классический алгоритм сильно проигрывает алгоритму, описываемому в данной работе, так как пытается объяснить инверсию через комбинации индел, тем самым делая оценку схожести данных последовательностей очень низкой. Разработанный алгоритм в свою очередь объясняет весь неточный участок инверсии 1 хромосомной перестройкой, что делает его оценку более точной.

Заключение

Разработанный алгоритм позволяет проводить парное выравнивание нуклеотидных последовательностей с учетом возможности возникновения инверсии.

Следующим этапом развития проекта является его валидация на реальных геномных последовательностях, взятых из базы данных NCBI.

Данная работа вместе с кодом доступна в репозитории: ’http://gititis.kpfu.ru/Malikov/diploma-work-malikov.git’

СПИСОК ЛИТЕРАТУРЫ

1. *Hershberg R.* Mutation—the engine of evolution: studying mutation and its role in the evolution of bacteria // Cold Spring Harbor perspectives in biology. — 2015. — Т. 7, № 9. — a018077.
2. *Repar J.*, *Warnecke T.* Non-random inversion landscapes in prokaryotic genomes are shaped by heterogeneous selection pressures // Molecular biology and evolution. — 2017. — Т. 34, № 8. — С. 1902—1911.
3. *Sehn J. K.* Insertions and deletions (Indels) // Clinical Genomics. — Elsevier, 2015. — С. 129—150.
4. Evolution of the insertion-deletion mutation rate across the tree of life / W. Sung [и др.] // G3: Genes, Genomes, Genetics. — 2016. — Т. 6, № 8. — С. 2583—2591.
5. An alignment-free method to find and visualise rearrangements between pairs of DNA sequences / D. Pratas [и др.] // Scientific reports. — 2015. — Т. 5. — С. 10203.
6. *Lee C.* Generating consensus sequences from partial order multiple sequence alignment graphs // Bioinformatics. — 2003. — Т. 19, № 8. — С. 999—1008.
7. Polymorphism of the IL28B gene (rs8099917, rs12979860) and virological response of Pakistani hepatitis C virus genotype 3 patients to pegylated interferon therapy / H. Aziz [и др.] // International Journal of Infectious Diseases. — 2015. — Т. 30. — С. 91—97.
8. *Covington M. A.* The number of distinct alignments of two strings // Journal of Quantitative Linguistics. — 2004. — Т. 11, № 3. — С. 173—182.
9. *Torres A.*, *Cabada A.*, *Nieto J. J.* An exact formula for the number of alignments between two DNA sequences // DNA Sequence. — 2003. — Т. 14, № 6. — С. 427—430.
10. *Eddy S. R.* Where did the BLOSUM62 alignment score matrix come from? // Nature biotechnology. — 2004. — Т. 22, № 8. — С. 1035.
11. *Myers E. W.*, *Miller W.* Optimal alignments in linear space // Bioinformatics. — 1988. — Т. 4, № 1. — С. 11—17.

12. *Löytynoja A.* Alignment methods: strategies, challenges, benchmarking, and comparative overview // Evolutionary Genomics. — Springer, 2012. — С. 203—235.
13. *Gotoh O.* An improved algorithm for matching biological sequences // Journal of molecular biology. — 1982. — Т. 162, № 3. — С. 705—708.
14. *Cartwright R. A.* Logarithmic gap costs decrease alignment accuracy // BMC bioinformatics. — 2006. — Т. 7, № 1. — С. 527.
15. *Löytynoja A.*, *Goldman N.* An algorithm for progressive multiple alignment of sequences with insertions // Proceedings of the National Academy of Sciences. — 2005. — Т. 102, № 30. — С. 10557—10562.
16. *Landan G.*, *Graur D.* Heads or tails: a simple reliability check for multiple sequence alignments // Molecular biology and evolution. — 2007. — Т. 24, № 6. — С. 1380—1383.
17. *Löytynoja A.*, *Milinkovitch M. C.* SOAP, cleaning multiple alignments from unstable blocks // Bioinformatics. — 2001. — Т. 17, № 6. — С. 573—574.
18. An alignment confidence score capturing robustness to guide tree uncertainty / O. Penn [и др.] // Molecular biology and evolution. — 2010. — Т. 27, № 8. — С. 1759—1767.
19. Biological sequence analysis: probabilistic models of proteins and nucleic acids / R. Durbin [и др.]. — Cambridge university press, 1998.
20. *Löytynoja A.*, *Goldman N.* webPRANK: a phylogeny-aware multiple sequence aligner with interactive alignment browser // BMC bioinformatics. — 2010. — Т. 11, № 1. — С. 579.
21. *Miklós I.*, *Lunter G.*, *Holmes I.* A “long indel” model for evolutionary sequence alignment // Molecular Biology and Evolution. — 2004. — Т. 21, № 3. — С. 529—540.
22. *Satija R.*, *Pachter L.*, *Hein J.* Combining statistical alignment and phylogenetic footprinting to detect regulatory elements // Bioinformatics. — 2008. — Т. 24, № 10. — С. 1236—1242.
23. *Redelings B. D.*, *Suchard M. A.* Joint Bayesian estimation of alignment and phylogeny // Systematic biology. — 2005. — Т. 54, № 3. — С. 401—418.

24. M-Coffee: combining multiple sequence alignment methods with T-Coffee / I. M. Wallace [и др.] // Nucleic acids research. — 2006. — Т. 34, № 6. — С. 1692—1699.
25. *Löytynoja A.*, *Goldman N.* Phylogeny-aware gap placement prevents errors in sequence alignment and evolutionary analysis // Science. — 2008. — Т. 320, № 5883. — С. 1632—1635.
26. *Notredame C.*, *Higgins D. G.*, *Heringa J.* T-Coffee: A novel method for fast and accurate multiple sequence alignment // Journal of molecular biology. — 2000. — Т. 302, № 1. — С. 205—217.
27. *Saitou N.*, *Nei M.* The neighbor-joining method: a new method for reconstructing phylogenetic trees. // Molecular biology and evolution. — 1987. — Т. 4, № 4. — С. 406—425.
28. *Kumar S.*, *Filipski A.* Multiple sequence alignment: in pursuit of homologous DNA positions // Genome research. — 2007. — Т. 17, № 2. — С. 127—135.
29. StatAlign: an extendable software package for joint Bayesian estimation of alignments and evolutionary trees / Á. Novák [и др.] // Bioinformatics. — 2008. — Т. 24, № 20. — С. 2403—2404.
30. Rapid and accurate large-scale coestimation of sequence alignments and phylogenetic trees / K. Liu [и др.] // Science. — 2009. — Т. 324, № 5934. — С. 1561—1564.
31. *Löytynoja A.*, *Goldman N.* Uniting alignments and trees // Science. — 2009. — Т. 324, № 5934. — С. 1528—1529.
32. *Needleman S. B.*, *Wunsch C. D.* A general method applicable to the search for similarities in the amino acid sequence of two proteins // Journal of molecular biology. — 1970. — Т. 48, № 3. — С. 443—453.
33. *Fletcher W.*, *Yang Z.* The effect of insertions, deletions, and alignment errors on the branch-site test of positive selection // Molecular biology and evolution. — 2010. — Т. 27, № 10. — С. 2257—2267.
34. DIALIGN: finding local similarities by multiple sequence alignment. / B. Morgenstern [и др.] // Bioinformatics (Oxford, England). — 1998. — Т. 14, № 3. — С. 290—294.

35. Sympatric speciation in a bacterial endosymbiont results in two genomes with the functionality of one / J. T. Van Leuven [и др.] // Cell. — 2014. — Т. 158, № 6. — С. 1270—1280.
36. *Lynch M.* Evolution of the mutation rate // TRENDS in Genetics. — 2010. — Т. 26, № 8. — С. 345—352.
37. Rates of spontaneous mutation / J. W. Drake [и др.] // Genetics. — 1998. — Т. 148, № 4. — С. 1667—1686.
38. Evolutionary rates and gene dispensability associate with replication timing in the archaeon Sulfolobus islandicus / K. M. Flynn [и др.] // Genome biology and evolution. — 2010. — Т. 2. — С. 859—869.
39. *Hughes D.* Evaluating genome dynamics: the constraints on rearrangements within bacterial genomes // Genome biology. — 2000. — Т. 1, № 6. — reviews0006—1.
40. *Darling A. E.*, *Miklós I.*, *Ragan M. A.* Dynamics of genome rearrangement in bacterial populations // PLoS genetics. — 2008. — Т. 4, № 7. — e1000128.
41. DNA motifs that sculpt the bacterial chromosome / F. Touzain [и др.] // Nature Reviews Microbiology. — 2011. — Т. 9, № 1. — С. 15.
42. *Wellenreuther M.*, *Bernatchez L.* Eco-evolutionary genomics of chromosomal inversions // Trends in ecology & evolution. — 2018. — Т. 33, № 6. — С. 427—440.
43. Long-term balancing selection on chromosomal variants associated with crypsis in a stick insect / D. Lindtke [и др.] // Molecular ecology. — 2017. — Т. 26, № 22. — С. 6189—6205.
44. Genomic evidence for role of inversion 3 RP of Drosophila melanogaster in facilitating climate change adaptation / R. V. Rane [и др.] // Molecular Ecology. — 2015. — Т. 24, № 10. — С. 2423—2432.
45. Chromosomal rearrangements and the genetics of reproductive barriers in Mimulus (monkey flowers) / L. Fishman [и др.] // Evolution. — 2013. — Т. 67, № 9. — С. 2547—2560.
46. *Lynch M.*, *Walsh B.* The origins of genome architecture. Т. 98. — Sinauer Associates Sunderland, MA, 2007.

47. *Kapusta A.*, *Suh A.*, *Feschotte C.* Dynamics of genome size evolution in birds and mammals // Proceedings of the National Academy of Sciences. — 2017. — T. 114, № 8. — E1460—E1469.
48. Comparative genomics reveals insights into avian genome evolution and adaptation / G. Zhang [и др.] // Science. — 2014. — T. 346, № 6215. — C. 1311—1320.
49. Phylogenomics of nonavian reptiles and the structure of the ancestral amniote genome / A. M. Shedlock [и др.] // Proceedings of the National Academy of Sciences. — 2007. — T. 104, № 8. — C. 2767—2772.
50. *Ji Y.*, *DeWoody J. A.* Genomic landscape of long terminal repeat retrotransposons (LTR-RTs) and solo LTRs as shaped by ectopic recombination in chicken and zebra finch // Journal of molecular evolution. — 2016. — T. 82, № 6. — C. 251—263.
51. *Bennetzen J. L.*, *Ma J.*, *Devos K. M.* Mechanisms of recent genome size variation in flowering plants // Annals of botany. — 2005. — T. 95, № 1. — C. 127—132.
52. *Nam K.*, *Ellegren H.* Recombination drives vertebrate genome contraction // PLoS genetics. — 2012. — T. 8, № 5. — e1002680.
53. *Gregory T. R.* Is small indel bias a determinant of genome size? // TRENDS in Genetics. — 2003. — T. 19, № 9. — C. 485—488.
54. Sequence shortening in the rodent ancestor / S. Laurie [и др.] // Genome research. — 2012. — T. 22, № 3. — C. 478—485.
55. 8.2% of the human genome is constrained: variation in rates of turnover across functional element classes in the human lineage / C. M. Rands [и др.] // PLoS genetics. — 2014. — T. 10, № 7. — e1004525.
56. Conserved syntenic clusters of protein coding genes are missing in birds / P. V. Lovell [и др.] // Genome biology. — 2014. — T. 15, № 12. — C. 565.

Приложения

Приложение 1. Главное тело функции

```
# Размер референсной подпоследовательности
size_of_pasage = 5
# Размер шага поиска второй последовательности ищется схожая на первую подпоследовательности
stride = 1
small_tolerance_value = 0.8
# Максимальная зона поиска, определяет максимально возможную длину вставки
maximum_allowable_searching_area = 10
first_sequence_size = len(first_sequence)
second_sequence_size = len(second_sequence)
# Получение индексов гомологичных зон
zones = list(get_homology_zones(first_sequence, second_sequence, size_of_pasage,
             stride, small_tolerance_value, maximum_allowable_searching_area))
if zones:
    # получение индексов зон неточностей из индексов зон гомологий
    dissimilar_zones = cast_from_homology_zones_to_dissimilar_zones(
    first_sequence_size, second_sequence_size, zones)

    # Распараллеленный проход по зонам неточностей в поисках инверсий
    inversion_list = list(all_inversions_mp(dissimilar_zones))
    print(inversion_list)
    number_of_inversions = len(inversion_list)
    inversion_list = list(filter(lambda x: len(x) != 0, inversion_list))
    inversion_list = [e for sl in inversion_list for e in sl]
    # Замена найденных инверсий
    first_sequence = replace_inversions(first_sequence, inversion_list)
    # классическое выравнивание
    alignments = pairwise2.align.globalxx(first_sequence, second_sequence,
    one_alignment_only=True)
    # Итоговая оценка схожести
    similarity = (2 * alignments[0][2] - 0.1 * (alignments[0][4] - alignments
    [0][2]) - number_of_inversions) / \
                 (first_sequence_size + second_sequence_size)
    print('Количество найденных инверсий ' + str(number_of_inversions))
    print('Итоговая оценка схожести ' + str(similarity))

else:
    print("Sorry. We didn't founded homology zones. Please check sequences or
    change settings ")
```

Приложение 2. Функция поиска гомологичных зон

```
# Размер референсной подпоследовательности
size_of_pasage = 5
# Размер шага поиска второй последовательности ищется схожая на первую подпоследовательности
stride = 1
small_tolerance_value = 0.8
# Максимальная зона поиска, определяет максимально возможную длину вставки
maximum_allowable_searching_area = 10
first_sequence_size = len(first_sequence)
second_sequence_size = len(second_sequence)
# Получение индексов гомологичных зон
zones = list(get_homology_zones(first_sequence, second_sequence, size_of_pasage,
             stride, small_tolerance_value, maximum_allowable_searching_area))
if zones:
    # получение индексов зон неточностей из индексов зон гомологий
    dissimilar_zones = cast_from_homology_zones_to_dissimilar_zones(
    first_sequence_size, second_sequence_size, zones)

    # Распараллеленный проход по зонам неточностей в поисках инверсий
    inversion_list = list(all_inversions_mp(dissimilar_zones))
    print(inversion_list)
    number_of_inversions = len(inversion_list)
    inversion_list = list(filter(lambda x: len(x) != 0, inversion_list))
    inversion_list = [e for sl in inversion_list for e in sl]
    # Замена найденных инверсий
    first_sequence = replace_inversions(first_sequence, inversion_list)
    # классическое выравнивание
    alignments = pairwise2.align.globalxx(first_sequence, second_sequence,
    one_alignment_only=True)
    # Итоговая оценка схожести
    similarity = (2 * alignments[0][2] - 0.1 * (alignments[0][4] - alignments
    [0][2]) - number_of_inversions) / \
                 (first_sequence_size + second_sequence_size)
    print('Количество найденных инверсий ' + str(number_of_inversions))
    print('Итоговая оценка схожести ' + str(similarity))

else:
    print("Sorry. We didn't founded homology zones. Please check sequences or
    change settings ")
```

Приложение 3. Функция поиска инверсий внутри зон неточностей

```
def get_inversion_indexes(alighnment, minimum_allowable_inversion_length):
    number_of_first_sequence_gaps = 0
    number_of_second_sequence_gaps = 0
    number_of_current_matches_strike = 0
    inversion_indexies_first_sequence = []
    for i, letter in enumerate(alighnment[0][0]):
        if letter == '-':
            if number_of_current_matches_strike > minimum_allowable_inversion_length:
                inversion_indexies_first_sequence.append(
                    [i - number_of_current_matches_strike - number_of_first_sequence_gaps,
                     i - number_of_first_sequence_gaps])
            number_of_current_matches_strike = 0
            number_of_first_sequence_gaps += 1
        if alighnment[0][1][i] == '-':
            if number_of_current_matches_strike > minimum_allowable_inversion_length:
                inversion_indexies_first_sequence.append(
                    [i - number_of_current_matches_strike - number_of_first_sequence_gaps,
                     i - number_of_first_sequence_gaps])
            number_of_current_matches_strike = 0
            number_of_second_sequence_gaps += 1
        elif letter == alighnment[0][1][i]:
            number_of_current_matches_strike += 1
        elif letter != alighnment[0][1][i]:
            if number_of_current_matches_strike > minimum_allowable_inversion_length:
                inversion_indexies_first_sequence.append(
                    [i - number_of_current_matches_strike - number_of_first_sequence_gaps,
                     i - number_of_first_sequence_gaps])
            number_of_current_matches_strike = 0
    if number_of_current_matches_strike > minimum_allowable_inversion_length:
        inversion_indexies_first_sequence.append(
            [len(alighnment[0][1]) - number_of_current_matches_strike - number_of_first_sequence_gaps,
             len(alighnment[0][1]) - number_of_first_sequence_gaps])
    return inversion_indexies_first_sequence
```