

Ускорение алгоритма 3SEQ для поиска рекомбинантных геномных последовательностей за счёт многопоточности и учета дат сбора образцов

Девятериков А.П.¹, Пальянов А.Ю.^{1,2,3}

¹Институт систем информатики им. А.П. Ершова СО РАН, Новосибирск, Россия

²НИИ вирусологии ФИЦ ФТМ, Новосибирск, Россия

³Новосибирский государственный университет, Новосибирск, Россия

Аннотация. В статье представлена эффективная многопоточная реализация современного алгоритма 3SEQ для выявления рекомбинантных генетических последовательностей, протестированная на вирусных геномах. Работа проводилась в рамках проекта по созданию отечественного программного комплекса (bioprojects.iis.nsk.su) для решения широкого спектра задач, связанных с анализом данных в области биоинформатики, вирусологии и эпидемиологии. Рекомбинантный вирусный геном получается в результате обмена частями геномов двух разных вариантов вирусов одного вида, что возможно при заражении одновременно обоими вариантами. Возникновение рекомбинантов – это редкие, но важные события в контексте изучения эволюции вируса. Одним из наиболее перспективных среди существующих алгоритмов для поиска рекомбинантов представляется 3SEQ, однако авторская версия работает только в однопоточном режиме. Мы реализовали этот алгоритм с поддержкой многопоточных вычислений и учетом дат сбора образцов, что обеспечило значительный прирост скорости вычислений. С помощью созданного программного обеспечения осуществлён поиск рекомбинантов в выборках геномов вирусов гриппа А H1N1 (анализировались только сегменты PB2 из 2174 геномов), лихорадки Денге (726 геномов), вируса Эбола (865 геномов) и в двух выборках геномов коронавируса SARS-CoV-2 (776 и 2132 генома). Для гриппа А H1N1 (сегмент PB2) и первого набора данных по SARS-CoV-2 (выборка по России) рекомбинантов найдено не было, что находится в согласии с анализом тех же данных алгоритмом RDP. Для второго набора данных по SARS-CoV-2 (выборка по Сибирскому федеральному округу) был правильно найден единственный присутствовавший рекомбинант. У вирусов лихорадки Денге найдено 725 рекомбинантов с длиной района рекомбинации в интервале от 50 до 1000 нуклеотидов. У вирусов Эбола длина района рекомбинации оказалась короче – у 572 рекомбинантов она находится в диапазоне от 50 до 100 нуклеотидов, у 249 геномов – менее 50 нуклеотидов.

Ключевые слова: вирусология, биоинформатика, поиск рекомбинантов, алгоритм 3SEQ, программа, многопоточность, ускорение вычислений.

ВВЕДЕНИЕ

На фоне пандемии SARS-CoV-2 в медицине и здравоохранении произошёл ряд изменений и нововведений. Среди них можно отметить развитие и широкое распространение как экспресс-тестирования на предмет выявления случаев заболевания распространёнными респираторными вирусами, так и внедрение практики секвенирования (прочтения нуклеотидных последовательностей) вирусных геномов в лабораториях по всему миру. Эти меры направлены, в первую очередь, на

отслеживание динамики числа случаев заражения – для своевременного реагирования в случае начала резкого роста их количества. Ещё одна цель – отслеживание изменений в вирусных геномах и выявление взаимосвязей между вирусными генотипами и фенотипами. Непрерывное накопление вариантов геномов в течение длительного времени также предоставляет ценные данные для улучшения понимания механизмов эволюции вирусов. Активность в области секвенирования образцов геномов SARS-CoV-2 и других респираторных вирусов, пополняющих специализированные базы данных, остается высокой и в нынешнее время, когда пандемия COVID-19 уже миновала, но вирус по-прежнему остается с человечеством. Рост числа вариантов геномов требует более совершенных алгоритмов и более производительного вычислительного оборудования для их исследования и анализа. Так, например, число вариантов геномов SARS-CoV-2 в крупнейших базах данных NCBI Genbank (www.ncbi.nlm.nih.gov/sars-cov-2/) и GISAID (gisaid.org) в настоящее время достигло 9 и 17 миллионов, соответственно. Общий объем данных напрямую зависит как от числа геномов в базе данных, так и от среднего размера генома.

Длины известных науке вирусных геномов лежат в пределах от 859 (*Circovirus SFBeef*) до $2.473 \cdot 10^6$ (*Pandoravirus salinus*) нуклеотидов (нукл.). Если рассматривать только РНК-вирусы – то в пределах от $1.76 \cdot 10^3$ (PCV-1) до $33.5 \cdot 10^3$ нукл. (ball python nidovirus). Последовательность длины n может быть представлена как набор символов $S = (S_1, \dots, S_n)$, где S_i принадлежит генетическому алфавиту: A(A), T(T), G(Г), C(Ц). Эти буквы кодируют азотистые основания аденин, тимин, гуанин и цитозин, соответственно. В отличие от ДНК, в РНК вместо цитозина используется урацил U(Y). При загрузке входных данных, для простоты обработки, мы всегда заменяем U на T (сохраняя информацию о том, с последовательностью РНК или ДНК мы работаем), поскольку в базах данных мировых лидеров в этой области, Genbank и GISAID, генетические последовательности представлены именно так.

Для размножения как живые организмы, так и вирусы реплицируют свой геном, однако не всегда при этом получают точные копии оригинала – в процессе возможно возникновение ошибок (мутаций). В результате генных мутаций могут происходить замены, делеции и вставки одного или нескольких нуклеотидов, транслокации, дупликации и инверсии различных частей гена. Так, например, точечные мутации возникают самопроизвольно на протяжении всей жизни организма в нормальных для него условиях окружающей среды. Для типичных клеток бактерий, дрожжей или человека это происходит с частотой около 10^{-9} – 10^{-10} на нуклеотид за клеточную генерацию организма [1]. Для вирусов частоты существенно выше – 10^{-8} – 10^{-6} для ДНК-вирусов и 10^{-6} – 10^{-4} – для РНК-вирусов [2]. Причина таких различий частот состоит в том, что РНК-полимераза, осуществляющая репликацию РНК-вирусов, лишена корректирующего ошибки механизма (за исключением коронавирусов и торовирусов, у которых она всё-таки имеется).

Помимо упомянутых механизмов, воздействующих на отдельный (одионый) геном, имеются и такие, которые позволяют создавать новые комбинации на основе генетического материала нескольких разных вариантов генома. У живых организмов это генетический материал родителей, у вирусов – геномы двух разных вариантов одного и того же вида. Когда они заражают один и тот же организм одновременно (например, Дельта и Омикрон в случае SARS-CoV-2 [3]), появляется возможность взаимодействовать во время репликации [4], в результате чего могут возникать рекомбинанты или реассортанты. Рекомбинацией называют обмен частями генома, включающий ковалентное встраивание участка (или участков) генома одного вируса в геном другого. Под реассортацией понимают обмен геномными сегментами (без ковалентного встраивания), возможный только у вирусов, чей геном представлен раздельными сегментами (например, вирус гриппа). Появление рекомбинантов или реассортантов является относительно редким, но при этом значимым событием в

эволюционной истории вируса, поскольку может приводить к появлению совершенно новых штаммов. Редким – поскольку могут быть обнаружены лишь те варианты, которые не только выжили, но и ощутимо распространились в популяции. Это могут сделать лишь те, что оказались не только жизнеспособными, но и получили некоторое преимущество перед остальными вариантами. Возможность обнаруживать рекомбинацию в геномах патогенов имеет решающее значение для точности филогенетического анализа и, следовательно, для прогнозирования распространения инфекционных заболеваний и появления опасных вариантов. Таким образом, детектирование и анализ таких событий является важным инструментом в вирусологии и эпидемиологии.

Задача определения рекомбинантов носит вероятностный характер, поэтому для её решения, как правило, используются статистические методы. К примеру, предположение о том, что последовательность *C* возникла в результате рекомбинации двух родительских последовательностей *A* и *B*, может быть охарактеризовано некоторым значением вероятности. Для многих алгоритмов поиска рекомбинантов требуется предварительно произвести множественное выравнивание набора последовательностей. Согласно одному из определений, выравнивание последовательностей – это биоинформатический метод, основанный на размещении двух или более последовательностей мономеров ДНК, РНК или белков друг под другом таким образом, чтобы оптимальным образом расположить их сходные участки, обеспечив минимально возможное различие [5]. Сходство последовательностей мономеров может отражать их функциональные, структурные или эволюционные взаимосвязи. В процессе построения выравнивания в строки-последовательности могут быть добавлены разрывы между основаниями (англ. “gaps”), обозначаемые символом “–”, таким образом, чтобы одинаковые или похожие элементы были расположены в следующих друг за другом столбцах матрицы (рис. 1):

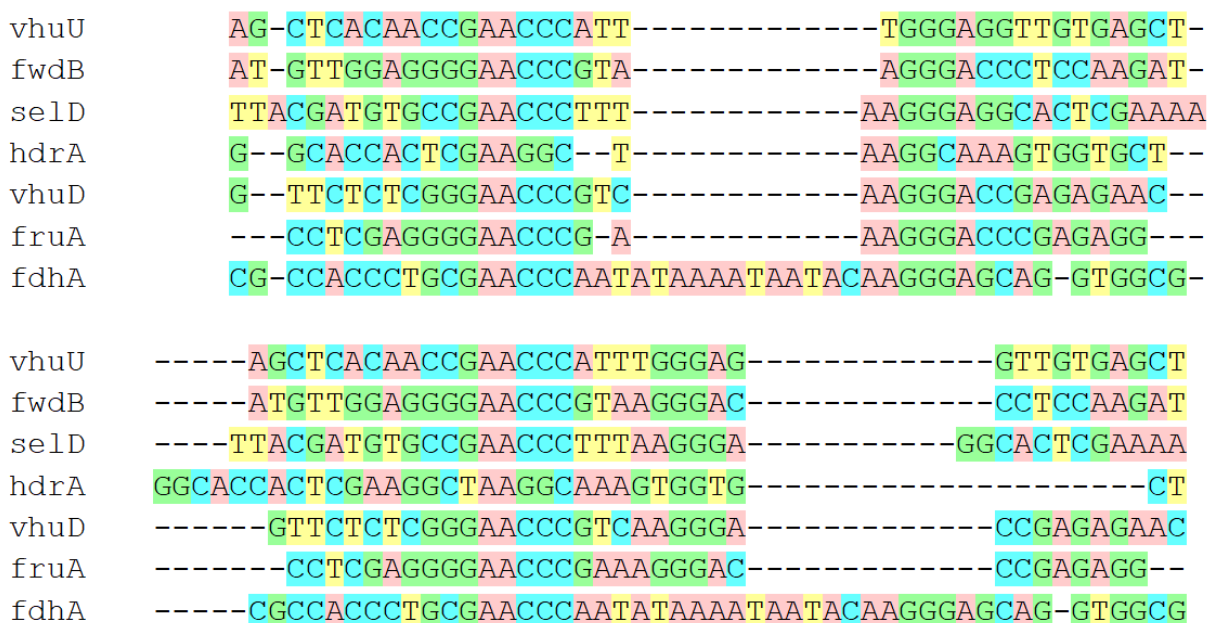


Рис. 1. Примеры множественных выравниваний, полученные на одних и тех же исходных данных – выборке коротких фрагментов РНК из генома архей (из репозитория программы LocARNA, <https://github.com/s-will/LocARNA> [6]). Результаты расчётов могут заметно отличаться для разных программ (выравнивания, сделанные при помощи LocARNA – сверху, при помощи MAFFT v7.511 (<https://mafft.cbrc.jp/alignment/server/index.html>) [7] – снизу).

Ввиду большей (по сравнению с парным выравниванием) вычислительной сложности при множественном выравнивании используются более сложные алгоритмы, как правило – эвристические, поскольку поиск глобального оптимального

выравнивания может занимать весьма значительное время, заметно возрастающее в зависимости от числа последовательностей и их длины. Далее, если не сказано обратное, будем считать, что алгоритм работает с уже выровненными последовательностями.

В настоящее время существует значительное число компьютерных методов выявления рекомбинантов. Одной из первых таких программ, получившей относительно широкое распространение, стала RDP v.1 (под Windows95/98/NT) [8]. К настоящему времени создана уже пятая версия этой программы [9].

Алгоритмы, работающие с тройками последовательностей и использующие скользящее окно, представляют одну из основных групп среди всего множества алгоритмов поиска рекомбинантов. RDP v.1 и 3SEQ также входят в эту группу. Для множества поданных на вход генетических последовательностей программа перебирает все возможные тройки и проверяет каждую тройку на наличие признаков рекомбинации. Процедура проверки состоит из трёх шагов. На первом шаге в тройке выбираются последовательности P и Q , расстояние между которыми является меньшим, чем между каждой из них и третьей последовательностью, C . Под расстоянием между двумя последовательностями в данном случае имеется в виду количество несовпадающих элементов (расстояние Хэмминга). Поскольку после выравнивания длины последовательностей одинаковы, такой подход будет корректен. Далее, анализируя последовательности поэлементно, из каждой из них «выбрасываются» так называемые «неинформативные черты», а именно:

- Элементы, идентичные для каждой из трёх последовательностей.
- Элементы, различные во всех трёх последовательностях.
- Элементы, уникальные для P и Q и не представленные ни в одном члене группы референсных последовательностей (группа референсных последовательностей может быть выбрана пользователем).

Таким образом из тройки последовательно удаляются нуклеотиды, не подходящие по описанным выше критериям, в результате чего происходит сокращение длины последовательностей для упрощения дальнейшего анализа. На втором и третьем шаге происходит движение скользящего окна (ширина которого задаётся пользователем) и вычисление значения некоторой функции *Prob* на основе анализа сокращенных подпоследовательностей. Её значение сравнивается с порогом, определенным пользователем. В случае попадания *Prob* в указанный интервал программа сохраняет анализируемую тройку и помечает последовательность C как рекомбинантную.

За прошедшие 20 с лишним лет объемы генетических данных, требующих анализа, многократно возросли. При этом возросли и вычислительные мощности компьютеров, а также появились новые алгоритмы, более быстрые и эффективные. Один из них – программный комплекс Bolotie [10], представляющий собой более быструю альтернативу алгоритмам, осуществляющим перебор всех троек геномных последовательностей из заданного набора. Авторы нацеливались на работу с коронавирусом SARS-CoV-2, для которого на момент публикации статьи в базе данных GISAID было более 2 млн. геномов, а в настоящее время – уже 17 млн. Первый рекомбинантный геном SARS-CoV-2 был выявлен в 2021 г., а к 2024 г. уже стало известно более 19 генетических линий этого вируса, идентифицируемых как рекомбинантные [11].

Для работы алгоритма необходимо иметь некоторый готовый набор генетических вариантов геномов, представляющих наиболее значимые кластера. Каждая из входящих последовательностей запроса рассматривается отдельно. Для каждого из элементов каждой из последовательностей считается некоторая взвешенная условная вероятность того, что последовательность принадлежит определенному кластеру на основании текущей рассматриваемой позиции. Далее используется СММ (скрытая марковская

модель, или Hidden Markov Model) для моделирования и алгоритм Витерби [12], относящийся к алгоритмам динамического программирования. Он необходим для поиска наиболее подходящего списка состояний, который в контексте цепей Маркова получает наиболее вероятную последовательность произошедших событий. За $O(nc^2)$, где n – длина последовательности, а c – число кластеров, можно получить наиболее вероятные последовательности кластеров, к которым можно отнести геном на основании конкретной рассматриваемой позиции.

В период с 2021 по 2024 г. были предложены и реализованы ещё несколько алгоритмов, рассмотренных во введении статьи [11], авторы которой предложили и реализовали свой автоматический метод поиска рекомбинантов, названный RecombinHunt. В отличие от RDP и 3SEQ, он не реализует подход на основе рассмотрения троек геномов, где каждая кандидатная рекомбинантная последовательность оценивается путем обширных сравнений со всеми потенциальными парами родителей. Вместо этого он абстрагирует независимые кластеры геномов, как определено референтной системой классификации/номенклатурой в форме списка характеризующих мутаций. Впоследствии каждая кандидатная рекомбинантная последовательность оценивается путем вычисления ее сходства/различия со многими существующими линиями/группами схожих геномных последовательностей. RecombinHunt не реконструирует филогении, а вычисляет вероятность набора предопределенных обозначений/линий (pre-defined designations/lineages) и их комбинаций (рекомбинантов) на основе мутаций в целевой последовательности. Наиболее вероятных кандидатов-родителей для рекомбинантной последовательности RecombinHunt идентифицирует, используя алгоритм, концептуально похожий на гипергеометрическое случайное блуждание, описанное в работах [13, 14]. Однако, в отличие от RecombinHunt, эти методы не учитывают частоту каждой отдельной точечной мутации явным образом и, таким образом, привязаны к совершенно иной статистической структуре.

По каждой генетической последовательности, проанализированной на предмет того, является ли она потенциально рекомбинантной, как RecombinHunt, так и 3SEQ могут выдать один из трех возможных результатов: (а) нет рекомбинации, (б) имеется 1 точка разрыва (обмена генетическим материалом) или (в) имеется 2 точки разрыва (рис. 2):

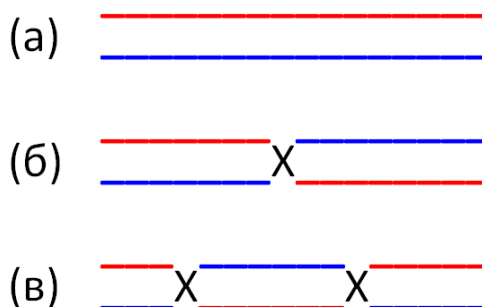


Рис. 2. Перераспределение генетического материала родительских вирусных геномов в геномах их потомков в случае, если (а) рекомбинации не произошло, (б) если она произошла с одной точкой разрыва и (в) – если с двумя.

Узнать больше о молекулярных механизмах, лежащих в основе изменчивости вирусов (включая механизмы рекомбинации), можно, например, в работе [2].

МЕТОДЫ И АЛГОРИТМЫ

Составив представление о современных подходах к проблеме выявления рекомбинантных вирусных последовательностей, мы пришли к выводу, что подходы на основе анализа троек представляются наиболее надежными. Причина в том, что, хотя

они и являются более вычислительно затратными, но зато исключают возможность упустить то или иное рекомбинационное событие из-за неправильно подобранной или недостаточно точной эвристики. Как оказалось, 3SEQ – достаточно широко используемый, высокоцитируемый и всё ещё довольно современный алгоритм, реализован авторами лишь в однопоточной версии, что не позволяет использовать преимущества современных многоядерных процессоров. Мы предположили, что он может быть существенно ускорен благодаря двум усовершенствованиям – (а) многопоточной реализации и (б) учету даты сбора образца для каждой генетической последовательности, что позволяет исключить существенное число “некорректных” троек – таких, в которых “потомки” старше “родителей”.

Так же, как и в ранее описанном методе, лежащем в основе алгоритма RDP, в случае алгоритма 3SEQ сначала необходимо избавиться от неинформативных черт в каждой из рассматриваемых троек последовательностей. В данном методе информативными считаются те черты (нуклеотид или гэп на конкретной позиции), которые совпадают и у потомка, и у родителей. И, наоборот, неинформативные черты – это такие, которые

1. совпадают во всех трех последовательностях, или
2. отличаются во всех трех последовательностях, или
3. совпадают у родителей, но отличаются у потомка.

Можно заметить, что третий пункт здесь отличается от аналогичного условия для метода, используемого в RDP v.1. Эти пункты необходимы для того, чтобы каждый из элементов последовательности потомка можно было отнести к первой или второй родительской последовательности (пункты 2 и 3). При этом пункт 1 гарантирует отсутствие какой-либо двойственности, когда элемент можно отнести к одной или другой из родительских последовательностей. Рассмотрим пример тройки последовательностей, в которых красным цветом выделены нуклеотиды, которые мы можем пометить как неинформативные:

p: **ATGCCTA**
q: **TACCGAA**
c: **AGCCGTC**

Тройка последовательностей (*p*, *q*, *c*) имеет *m* информативных черт для родителя *p* и *n* информативных черт – для родителя *q*. Теперь каждому из элементов последовательности *c* мы можем назначить один из символов набора {P, Q}. Символ P на позиции *i* означает, что последовательность *c* унаследовала *i*-й нуклеотид от родителя *p*. То же для символа Q и последовательности *q*. Например, для некоторой последовательности длиной 21 мы можем получить следующее:

PQRRRRRRRRQQQRRQQQRRQQ

При нулевой гипотезе (последовательность не является результатом рекомбинации) предполагается, что расположение P и Q полностью случайно. То есть можно получить каждый из $\frac{(m+n)!}{m!n!}$ вариантов (используя формулу числа сочетаний из *L* по *n* или *m*) с равной вероятностью. В приведенном примере видно, что символы P как будто смещены влево, а символы Q – вправо, что может свидетельствовать в пользу того, что имела место рекомбинация. Основная задача – выразить на языке статистики то, что смещение символов Q и P является значимым для того, чтобы считать последовательность рекомбинантной. Здесь используется процесс, похожий на случайное блуждание, чтобы визуализировать получившуюся последовательность символов. Пусть имеется вертикальная ось, где P – шаг вверх, а Q – шаг вниз. Стоит отметить, что это не в точности случайное блуждание, так как заранее известно, сколько шагов будет сделано вверх и сколько вниз, а вся случайность здесь

закljučается именно в порядке их осуществления. Такой процесс называется гипергеометрическим случайным блужданием. Он хорошо описан в книге [15], в которой исследуются выборы, где играет роль порядок голосования за кандидата. Для каждой тройки вычисляется значение максимального спуска: $\Delta_{m,n,2} = \max_{0 \leq s \leq t \leq m \leq m+n} (X_s - X_t)$, где X_s, X_t – высоты на шагах s и t .

Обозначим $k = \max_{0 \leq s \leq t \leq m \leq m+n} (X_s - X_t)$. Сложность расчёта p -значения (p -value) для наблюдаемой тройки, $p_{m,n,k}$, оцениваемая как $O(mn^2)$, обусловлена необходимостью решения набора рекурсивных уравнений. В статье [14] подробно приведены обоснования и формулы для вычисления p -значений для троек. Один из эффективных способов решения поставленной задачи представлен алгоритмом 3SEQ.

РЕЗУЛЬТАТЫ

Программная система

Алгоритм 3SEQ с предложенными нами усовершенствованиями был реализован на языке C++ в виде консольного приложения. Программный код может быть скомпилирован как под Linux, так и под Windows. Помимо базовой функциональности, связанной с обработкой геномных последовательностей и их анализом, также имеется возможность строить филогенетические деревья (используя метрику из clades.nextstrain.org [16, 17]) и назначать последовательностям, образующим множество входных данных программы, соответствующие им кластера из числа уже обработанных последовательностей. Исходный код программы и использованные тестовые данные (выборки геномов нескольких различных вирусов) доступны в репозитории github: <https://github.com/NotNa19/RecombinantDetector>. Программа снабжена подробной инструкцией по использованию, также размещенной в репозитории. Имеется возможность задавать пользовательские настройки, изменяя параметры в файле [UserSettings.rec](#), среди которых – число потоков, используемых при вычислениях, путь к директории с выходными данными, пороговое значение для срабатывания детектора рекомбинантов и др., всего – 19 параметров.

Как и в оригинальной реализации, на вход подаются выровненные последовательности в виде fasta-файлов (содержащих уже выровненные геномы), которые проверяются на наличие рекомбинантных троек. Для проверки используется упомянутый выше статистический тест. Таблица p -значений должна быть посчитана заранее, что тоже реализовано в программе в виде отдельной функции.

Основными преимуществами нашей реализации являются поддержка многопоточности и возможность использования заголовочной информации из fasta-файлов для извлечения даты получения образца. Также реализованы различные крупные и мелкие оптимизации для повышения производительности. Программная система включает 38 файлов с исходным кодом (> 5300 строк, 138.5 кб), файл настроек, файл документации и ряд файлов с входными тестовыми данными.

Поддержка параллельных вычислений на CPU осуществляется посредством использования паттерна ThreadPool в реализации, представленной в репозитории <https://github.com/progschj/ThreadPool/tree/master>. Мы разбиваем данные на блоки и, используя ThreadPool, создаем очередь из задач, которые выполняются на имеющихся потоках (всех доступных в системе или с учетом ограничений, заданных пользователем в настройках). Согласно закону Амдала [18], ускорение выполнения программы за счёт распараллеливания её инструкций на множестве вычислителей ограничено временем, необходимым для выполнения её последовательных инструкций. Так, например, если можно распараллелить 95 % программы, то теоретически максимальное ускорение будет 20-кратным, невзирая на то, сколько процессоров или ядер используется

параллельно. В нашем случае мы рассчитали эту зависимость для двух наборов данных, описанных в разделах “Коронавирус SARS-CoV-2” и “Вирус гриппа А (серотип H1N1, белок PB2)”, для которых работа однопоточной версии заняла около 47 минут и более 8.5 часов, соответственно (рис. 3):

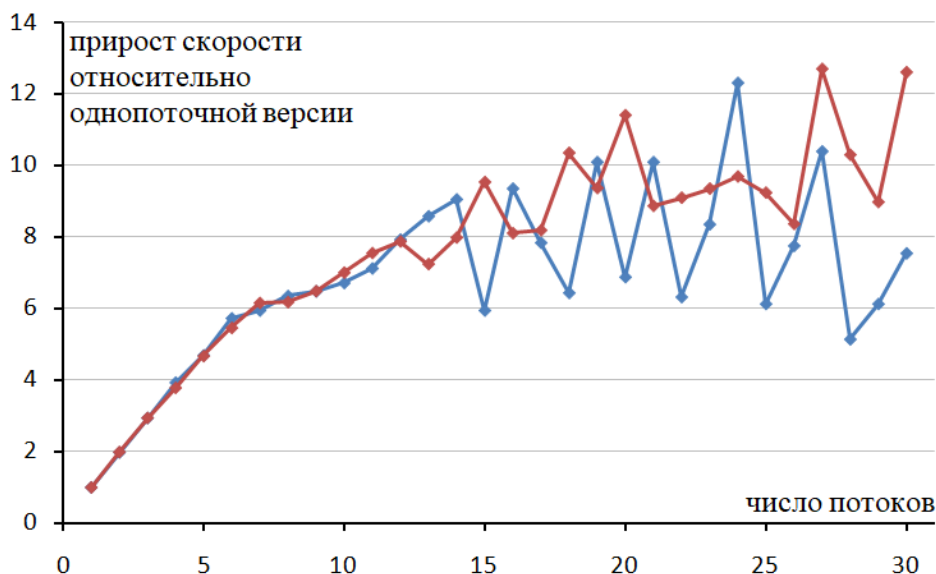


Рис. 3. Отношение времени работы многопоточной версии к времени работы однопоточной на одних и тех же входных данных в зависимости от числа задействованных потоков. Синяя кривая – расчеты для данных по SARS-CoV-2, красная – для данных по вирусу гриппа А.

Получается, что наиболее выгодно использовать 6 потоков – в этом случае получается ускорение в 5.7 раз. Впрочем, если задействовать большее их число, то можно добиться ускорения и до 12 раз, но ценой значительно больших вычислительных затрат.

Осуществив сравнение производительности нашей реализации программы с оригиналом (на одном и том же вычислительном оборудовании и исходных данных), мы обнаружили, что действительно получили существенный прирост скорости работы. Проиллюстрируем это на ряде примеров, с которыми мы работали.

Тестирование качества и производительности

При осуществлении тестирования качества работы программы и проверке правильности выдаваемых ею результатов желательно иметь набор данных, для которых известны “правильные ответы” – какие из геномов в наборе являются родительскими, а какие – результатами их рекомбинации.

1. Искусственные данные

Одним из подходов к созданию таких наборов данных является их генерация искусственным образом. Для этого хорошо подходят и могут быть применены две библиотеки языка Python, *msprime* [19] и *pyvolve* [20], достаточно широко используемые исследователями. Использование модуля *msprime* позволяет сгенерировать структуру дерева, указав желаемую длину последовательностей, размер набора и, что самое главное, указать вероятность появления рекомбинантов (включая вариант, когда она равна нулю). Модуль *pyvolve* позволяет сгенерировать последовательности нуклеотидов, начиная с корня и с учетом всей информации о структуре дерева. Создав набор последовательностей, в котором гарантированно нет рекомбинантов, мы можем затем их сгенерировать, выбирая родительские последовательности и позиции точек разрыва (одной или двух). Соответствующий

скрипт был написан нами на языке Python. Информация о всех полученных рекомбинантах и их родительских последовательностях сохраняется для дальнейшего использования при проверке результатов работы алгоритма.

Для тестирования мы использовали как «короткие» геномы – по 3,000 нукл. на каждую последовательность, так и «длинные» – по 12,000 нукл. Тестирование осуществлялось на компьютере с процессором Intel® Core™ i9-13900KF (24 ядра, 32 потока, тактовая частота до 5.7 ГГц) с оперативной памятью 32 Гб DDR4 3200 МГц. Перед запуском основных расчётов предварительно была вычислена таблица p_val размером $1000 \times 1000 \times 1000$, используемая здесь и далее. Время расчета таблицы – 4 мин. 40 сек., размер – 1,3 Гб.

На первом шаге тестирования был использован набор из 500 геномов, каждый длиной 3,000 нукл., без рекомбинантов. Получилось, что длина последовательности с учетом полиморфных черт (после исключения “неинформативных” позиций) составила 2,922 (т.е., получился достаточно разнообразный набор). Поиск рекомбинантов на данном наборе показал их отсутствие, как и предполагалось.

На следующем шаге тестирования мы добавили в набор 50 рекомбинантов, сгенерированных с использованием некоторых из исходных 500 в качестве их предков. В результате тестирования были выявлены все 50 рекомбинантов, лишних обнаружено не было, а также были верно определены точки разрывов при рекомбинации.

В набор «длинных» геномов (по 12,000 нукл.) также вошли 500 последовательностей. Длина последовательностей с учетом полиморфных черт составила 11,723. Поиск рекомбинантов (при их фактическом отсутствии) с помощью нашей программы, как и ожидалось, не выявил их наличия. Тестирование на наборе данных, в который были добавлены 50 рекомбинантов, показало, что все они были обнаружены, а найденные точки разрыва совпадают с фактическими. На этом тестирование на искусственных данных было завершено и мы перешли к испытаниям с использованием геномов реальных вирусов.

2. Вирус гриппа А (серотип H1N1, белок PB2)

Интервал времени – 2009–2010 годы (время вспышки вируса), регион – Северная Америка, носитель – человек. Набор состоит из 2,174 последовательностей, длина выровненной последовательности – 2,361, укороченная длина с учетом только полиморфных черт – 1,075. В результате обработки данного набора и поиска в нём мы ожидали увидеть отсутствие рекомбинантов, в соответствии выводами, сделанными в работе [21]. Это ожидание подтвердилось как при использовании RDP, так и при использовании нашей программы – в результате их работы не было найдено ни одного рекомбинанта.

Время работы оригинальной реализации алгоритма 3SEQ, установленного на нашу рабочую станцию, на 1 потоке составило 8 часов 18 минут. Время работы нашей программы на ней же на 1 потоке составило 8 часов 35 минут, на 10 потоках – 1 час 13 минут. Время работы на 10 потоках с использованием информации о дате получения (collection date) каждого из образцов вируса составило 51 минуту. Благодаря использованию этой информации из всех возможных 6,913,290,720 троек более половины из них, 3,714,953,484 (53.7 %), удалось исключить как некорректные и тем самым сократить время вычислений.

3. Вирус Эбола, заирский штамм

Выбранный интервал времени, с 2015 по 2019 год, был взят за пределами периода основной вспышки, что позволило охватить пятилетний период эволюции вируса при выборке умеренного размера (чтобы провести расчёты за разумное время). Набор состоит из 865 последовательностей, длина выровненной последовательности – 19,029,

укороченная длина с учетом только полиморфных черт – 2,156. Анализ с помощью RDP показал наличие 56 рекомбинационных сигналов, однако с пометкой, что эти последовательности также могли быть получены в результате накопления эволюционных изменений без участия механизма рекомбинации. Мы обнаружили 821 рекомбинант, однако все они имеют достаточно малую длину рекомбинантного участка (табл. 1).

Таблица 1. Результаты поиска рекомбинантов для набора данных по вирусу Эбола: зависимость числа обнаруженных рекомбинантов от длины рекомбинационного участка

длина рекомбинантного участка, нукл.	число обнаруженных рекомбинантов
менее 50	249
от 50 до 100	572
более 100	0

Для оригинальной реализации время работы на 1 потоке составило 1 час 5 минут. Для нашей программы: на 1 потоке – 1 час 7 минут, на 10 потоках – 11 минут, на 10 потоках с использованием информации о дате получения образцов вирусов – 4 минуты (в результате удалось “отсеять” 250,669,904 троек из 504,357,540 – 49.7%). По результатам поиска рекомбинантов в выборке геномов вируса Эбола построена зависимость числа геномов и обнаруженных в ней рекомбинантов от времени (рис. 4) и распределение позиций точек рекомбинации вдоль генома (рис. 5).

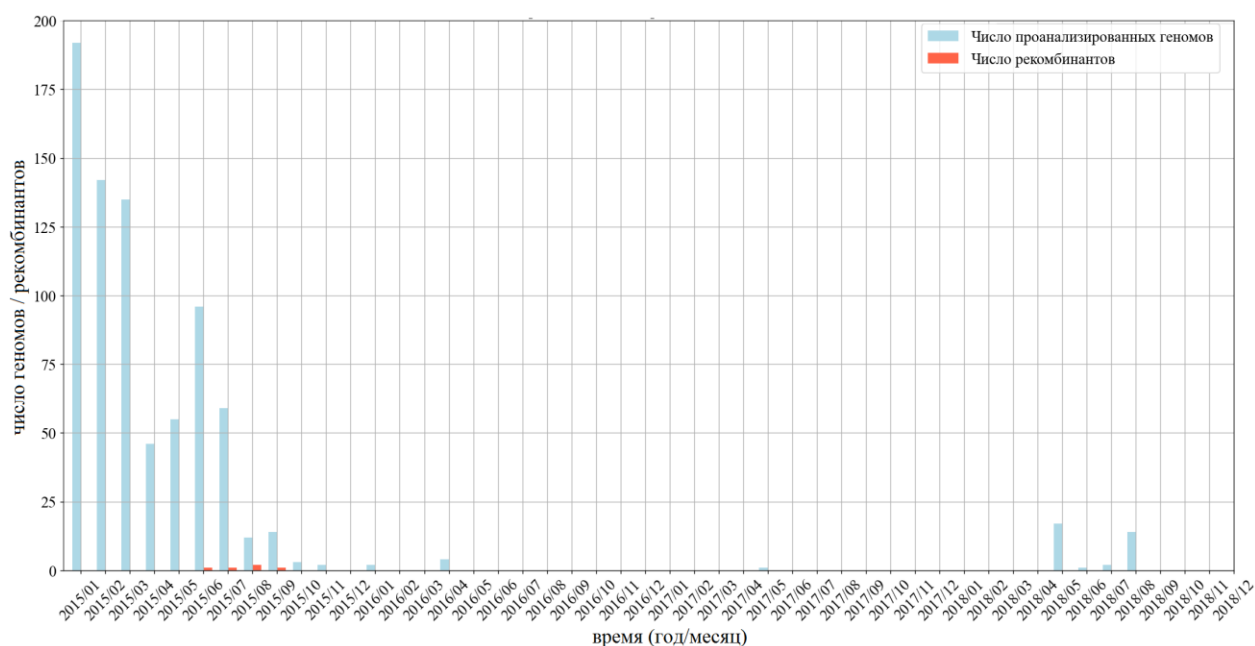


Рис. 4. Зависимость числа геномов в выборке и обнаруженных в ней рекомбинантов от времени (от даты получения образцов) для выборки вирусов Эбола.

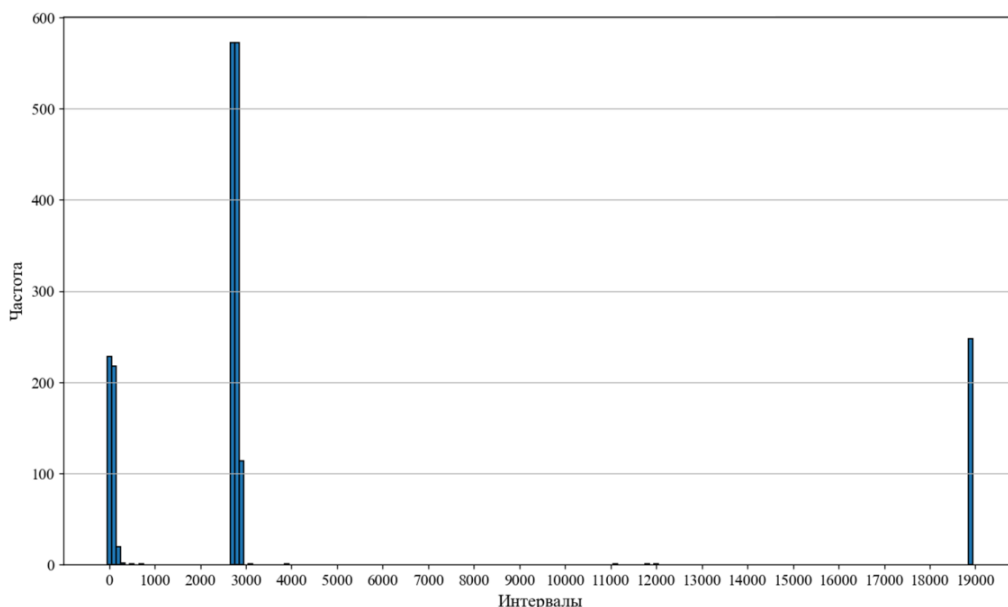


Рис. 5. Распределение позиций точек разрыва при рекомбинации вдоль генома вирусов Эбола.

Наиболее значительный пик обнаружился в районе 3000. Это можно объяснить тем, что участок с ~2200 до ~3400 занимает белок VP35, достаточно сильно подверженный мутациям. Возможно, это дало некоторый процент ложных результатов, поскольку, как было указано ранее, найденные для этого набора рекомбинационные области являются достаточно короткими.

4. Коронавирус SARS-CoV-2

Данные по России за короткий промежуток с сентября по ноябрь 2023 года. В наборе имеется 776 последовательностей. Длина выровненной последовательности — 29,903, укороченная длина с учетом только полиморфных черт — 2,250. Здесь мы не ожидаем увидеть каких-либо рекомбинационных событий, поскольку с помощью программы RDP они не были обнаружены. Наша программа также не выявила здесь каких-либо признаков рекомбинации.

Время работы оригинальной версии 3SEQ на одном потоке составило 1 час 34 минуты. Время работы нашей программы на одном потоке — 1 час 20 минут. Время работы на 10 потоках, с использованием информации о дате получения образцов, составило 7 минут (отфильтровано 1,990,288 троек из 465,483,600 — 42.7 %). Более низкий процент связан с тем, что интервал времени довольно узкий, и, соответственно, варианты вируса менее разнообразные (тем более что коронавирус вообще эволюционирует заметно медленнее большинства других РНК вирусов, поскольку обладает механизмом коррекции ошибок, возникающих в процессе репликации).

Также для SARS-CoV-2 мы проанализировали ещё один набор данных — все последовательности из Сибирского федерального округа, полученные от начала пандемии в декабре 2019 г. до середины 2023 г., загруженные из базы данных GISAID (gisaid.org), 2132 шт. Запуск на 25 потоках занял 2 часа 3 минуты, на 25 потоках с учетом дат получения образцов — 1 час 5 минут. В обоих режимах удалось найти 1 рекомбинант (один и тот же) — и это правильный результат, поскольку в использованном наборе данных действительно имеется лишь единственный рекомбинант, и именно он был обнаружен нашей программой. Имя найденного рекомбинанта — “hCoV-19/Russia/KYA-R11-MH128738S/2023”, идентификатор в БД GISAID — EPI_ISL_17317906, дата получения образца — 2023-01-11. Его предками, предположительно, является следующая пара:

- 1) hCoV-19/Russia/NVS-R11-MH79179S/2022, EPI_ISL_15069502, 2022-08-16

2) hCoV-19/Russia/OMS-RII-MH130450S/2022, EPI_ISL_17318554, 2022-12-16

Левая точка разрыва при рекомбинации, предположительно, лежит в интервале 19326–21129, правая – в интервале 28560–29903.

5. Лихорадка Денге, серотип 2

Регион – Азия. Всего в наборе имеется 726 последовательностей. Интервал времени – с 1995 по 2019 г. Длина выровненной последовательности – 11,185, укороченная длина с учетом только полиморфных черт – 5,523 (достаточно разнообразный набор). Данные по лихорадке Денге содержат более высокий процент рекомбинантов по сравнению с предыдущими примерами. Этот набор данных мы взяли, чтобы проверить, насколько хорошо наша программа может обнаруживать наличие рекомбинационных событий в течение значительного интервала времени. Результаты представлены в таблице 2 и на рисунках 6 и 7.

Таблица 2. Зависимость числа обнаруженных рекомбинантов от длины рекомбинационного участка

длина рекомбинантного участка, нукл.	число обнаруженных рекомбинантов
< 50	1
50...100	34
100...500	374
500...1000	146
> 1000	170



Рис. 6. Распределение количества найденных рекомбинантов по годам в интервале 1995–2019 (длина рекомбинантного участка с учетом только полиморфных черт – не менее 50% от всей её длины, равной 5523 нукл.)

Время работы оригинальной версии 3SEQ на одном потоке составило 1 час 40 минут, время работы нашей программы в однопоточном режиме – 1 час 46 минут, на 10 потоках – 16 минут. Большинство последовательностей (77 %) из использованного набора данных не содержали точную информацию о датах получения

образцов. Как видно из таблицы, здесь мы обнаружили большое число рекомбинантов, причём с достаточно большими длинами (> 1000 нукл.) участков обмена.

Может показаться, что представленное на рисунке 6 число рекомбинантов довольно велико по сравнению с числом геномов в выборке, однако причина здесь – в определении рекомбинанта. Как в RDP, так и в оригинальной 3SEQ, найденным рекомбинантом считается любая тройка нуклеотидных последовательностей, для которой вероятность того, что одна из входящих в неё последовательностей образовалась в результате рекомбинации остальных двух, превышает заданное пороговое значение. Представим теперь, что спустя некоторое время этот рекомбинант был размножен в результате репликации, и некоторые его потомки со временем накопили некоторое небольшое количество изменений – точечных мутаций. Получается, что и для них программа найдёт подходящие тройки, которые тоже будут идентифицированы как рекомбинанты, что и объясняет их количество. На наш взгляд, более правильным представляется подход, при котором «настоящим рекомбинантом» (т.е. геномом, возникшим именно в результате рекомбинации, а не репликации уже имеющегося рекомбинанта) среди множества близких друг к другу геномов, обнаруживаемых программой как рекомбинатные, является только самый ранний из них. Остальные, возникшие позднее в результате его репликации с возможным возникновением точечных мутаций, являются его потомками, но не рекомбинантами. В дальнейшем мы планируем расширить возможности программы, предоставив пользователю выбирать, какой именно критерий рекомбинантного генома использовать при расчетах, добавив этот параметр в конфигурационный файл программы.

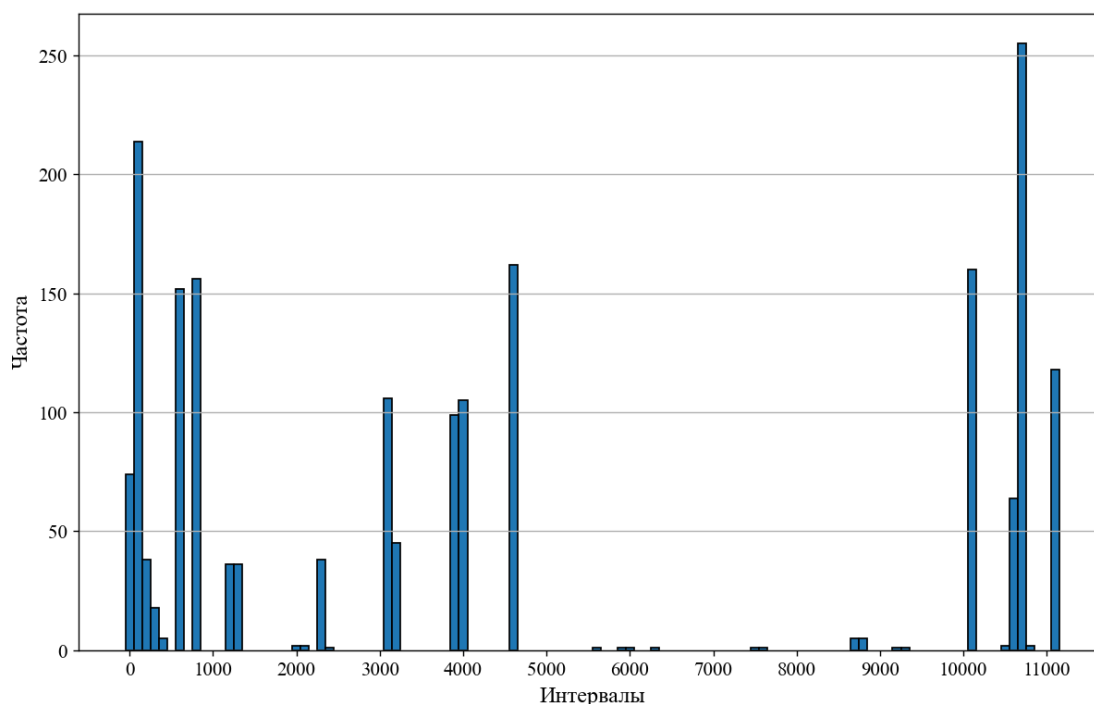


Рис. 7. Распределение позиций точек разрыва при рекомбинации в геноме лихорадки Денге.

На данном этапе работы основной задачей являлось создание надежной, быстрой и удобной для использования усовершенствованной реализации алгоритма 3SEQ, и она была успешно решена. Разнообразные наборы данных на основе геномов ряда вирусов были использованы, чтобы провести тестирование на коротких, средних и длинных геномных последовательностях с различными уровнями их разнообразия в

выборках. Для контроля результаты, рассчитанные с использованием нашей реализации алгоритма 3SEQ, сравнивались с результатами, полученными на тех же данных при расчётах на оригинальной версии 3SEQ, а также на RDP.

ЗАКЛЮЧЕНИЕ

Основным результатом проделанной работы является создание программной системы с открытым исходным кодом для поиска рекомбинантов в заданном наборе геномов. В её основе лежит алгоритм 3SEQ, однако, в отличие от оригинала – с поддержкой многопоточности и возможностью учета дат получения образцов вирусов. Проведено тестирование качества работы созданной программы и её производительности. Для этого были использованы как искусственно сгенерированные данные, так и реальные выборки геномов ряда различных вирусов. Тестирование не выявило случаев ложных предсказаний или получения результатов с неполным множеством рекомбинантов (т.е. такого, в котором некоторые результаты были бы пропущены). Производительность нашей реализации в однопоточном режиме оказалась практически такой же, как и у оригинального алгоритма, тогда как вычисления в многопоточном режиме привели к существенному увеличению скорости – в 5–7 раз при использовании 6–10 ядер, и ещё в 2–4 раза – за счёт исключения “некорректных” троек, в которых “потомки” старше “родителей”. В дальнейшем планируется существенное расширение возможностей программы – добавление веб-интерфейса и встраивание в систему bioprojects.iis.nsk.su, дальнейшая оптимизация, добавление возможности поиска реассортантов и другие усовершенствования.

Исследование выполнено за счет гранта Российского научного фонда № 23-64-00005, <https://rscf.ru/project/23-64-00005/>.

СПИСОК ЛИТЕРАТУРЫ

1. Drake J.W., Charlesworth B., Charlesworth D., Crow J.F. Rates of spontaneous mutation. *Genetics*. 1998. V. 148. No. 4. P. 1667–1686. doi: [10.1093/genetics/148.4.1667](https://doi.org/10.1093/genetics/148.4.1667)
2. Sanjuán R., Domingo-Calap P. Mechanisms of viral mutation. *Cell Mol. Life Sci.* 2016. V. 73. No. 23. P. 4433–4448. doi: [10.1007/s00018-016-2299-6](https://doi.org/10.1007/s00018-016-2299-6)
3. Bolze A., Basler S., White S., Rossi A.D., Wyman D., Dai H., Roychoudhury P., Greninger A.L., Hayashibara K., Beatty M. et al. Evidence for SARS-CoV-2 Delta and Omicron co-infections and recombination. *Med.* 2022. V. 3. No. 12. P. 848–859. doi: [10.1016/j.medj.2022.10.002](https://doi.org/10.1016/j.medj.2022.10.002)
4. Simon-Loriere E., Holmes E.C. Why do RNA viruses recombine? *Nat. Rev. Microbiol.* 2011. V. 9. No. 8. P. 617–626. doi: [10.1038/nrmicro2614](https://doi.org/10.1038/nrmicro2614)
5. Mount D.M. *Bioinformatics: Sequence and Genome Analysis*. Cold Spring Harbor, NY: Cold Spring Harbor Laboratory Press, 2004.
6. Will S., Joshi T., Hofacker I.L., Stadler P.F., Backofen R. LocARNA-P: Accurate boundary prediction and improved detection of structural RNAs. *RNA*. 2012. V. 18. No. 5. P. 900–914. doi: [10.1261/rna.029041.111](https://doi.org/10.1261/rna.029041.111)
7. Katoh K., Rozewicki J., Yamada K.D. MAFFT online service: multiple sequence alignment, interactive sequence choice and visualization. *Briefings in Bioinformatics*. 2019. V. 20. No. 4. P. 1160–1166. doi: [10.1093/bib/bbx108](https://doi.org/10.1093/bib/bbx108)
8. Martin D., Rybicki E. RDP: detection of recombination amongst aligned sequences. *Bioinformatics*. 2000. V. 16. No. 6. P. 562–563. doi: [10.1093/bioinformatics/16.6.562](https://doi.org/10.1093/bioinformatics/16.6.562)
9. Martin D.P., Varsani A., Roumagnac P., Botha G., Maslamoney S., Schwab T., Kelz Z., Kumar V., Murrell B. RDP5: a computer program for analyzing recombination in, and

- removing signals of recombination from, nucleotide sequence datasets. *Virus Evolution*. 2020. V. 7. No. 1. Article No. veaa087. doi: [10.1093/ve/veaa087](https://doi.org/10.1093/ve/veaa087)
10. Varabyou A., Pockrandt C., Salzberg S.L., Pertea M. Rapid detection of inter-clade recombination in SARS-CoV-2 with Bolotie. *Genetics*. 2021. V. 218. No. 3. Article No. iyab074. doi: [10.1093/genetics/iyab074](https://doi.org/10.1093/genetics/iyab074)
 11. Alfonsi T., Bernasconi A., Chiara M., Ceri S. Data-driven recombination detection in viral genomes. *Nat. Commun.* 2024. V. 15. Article No. 3313. doi: [10.1038/s41467-024-47464-5](https://doi.org/10.1038/s41467-024-47464-5)
 12. Forney G.D. The Viterbi algorithm. *Proceedings of the IEEE*. 1973. V. 61. No. 3. P. 268–278. doi: [10.1109/proc.1973.9030](https://doi.org/10.1109/proc.1973.9030)
 13. Boni M.F., Posada D., Feldman M.W. An exact nonparametric method for inferring mosaic structure in sequence triplets. *Genetics*. 2007. V. 176. No. 2. P. 1035–1047. doi: [10.1534/genetics.106.068874](https://doi.org/10.1534/genetics.106.068874)
 14. Lam H.M., Ratmann O., Boni M.F. Improved algorithmic complexity for the 3SEQ recombination detection algorithm. *Mol. Biol. Evol.* 2018. V. 35. No. 1. P. 247–251. doi: [10.1093/molbev/msx263](https://doi.org/10.1093/molbev/msx263)
 15. Feller W. *An Introduction to Probability Theory and Its Applications*, Vol. I. New York: John Wiley & Sons, 1957.
 16. Hadfield J., Megill C., Bell S.M., Huddleston J., Potter B., Callender C., Sagulenko P., Bedford T., Neher R.A. Nextstrain: real-time tracking of pathogen evolution. *Bioinformatics*. 2018. V. 34. No. 23. P. 4121–4123. doi: [10.1093/bioinformatics/bty407](https://doi.org/10.1093/bioinformatics/bty407)
 17. Aksamentov I., Roemer C., Hodcroft E.B., Neher R.A. Nextclade: clade assignment, mutation calling and quality control for viral genomes. *Journal of Open Source Software*. 2021. V. 6. No. 67. Article No. 3773. doi: [10.21105/joss.03773](https://doi.org/10.21105/joss.03773)
 18. Amdahl G.M. Validity of the Single Processor Approach to Achieving Large-Scale Computing Capabilities. *AFIPS Conference Proceedings*. 1967. V. 30. P. 483–485. doi: [10.1145/1465482](https://doi.org/10.1145/1465482)
 19. Baumdicker F., Bisschop G., Goldstein D., Gower G., Ragsdale A.P., Tsambos G., Zhu S., Eldon B., Ellerman E.C., Galloway J.G. et al. Efficient ancestry and mutation simulation with msprime 1.0. *Genetics*. 2022. V. 220. No. 3. Article No. iyab229. doi: [10.1093/genetics/iyab229](https://doi.org/10.1093/genetics/iyab229)
 20. Spielman S.J., Wilke C.O. Pyvolve: a flexible Python module for simulating sequences along phylogenies. *PloS One*. 2015. V. 10. No. 9. Article No. e0139047. doi: [10.1371/journal.pone.0139047](https://doi.org/10.1371/journal.pone.0139047)
 21. Boni M.F., Smith G.J.D., Holmes C.E., Vijaykrishna D. No evidence for intra-segment recombination of 2009 H1N1 influenza virus in swine. *Gene*. 2012. V. 494. No. 2. P. 242–245. doi: [10.1016/j.gene.2011.10.041](https://doi.org/10.1016/j.gene.2011.10.041)

Рукопись поступила в редакцию 30.09.2024, переработанный вариант поступил 10.10.2024.
Дата опубликования 05.11.2024.

Acceleration Of Recombinant Viral Sequences Search By 3SEQ Algorithm Via Adding Support Of Multi-Threaded Calculations And Considering Sample Collection Dates

Devyaterikov A.P.¹, Palyanov A.Yu.^{1,2,3}

¹A.P. Ershov Institute of Informatics, Siberian Branch of the Russian Academy of Sciences, Novosibirsk, Russia

²Research Institute of Virology, Federal Research Center of Fundamental and Translational Medicine, Novosibirsk, Russia

³Novosibirsk State University, Novosibirsk, Russia

Abstract. The article presents an efficient multithreaded implementation of the modern 3SEQ algorithm for detecting recombinant genetic sequences, tested on viral genomes. The work was carried out within the framework of the project to create a domestic (Russian) web-platform (bioprojects.iis.nsk.su) for solving a wide range of problems related to data analysis in the field of bioinformatics, virology and epidemiology. A recombinant viral genome emerges when two different variants of virus genomes of the same species exchange their parts, which is possible in case of infection with both variants simultaneously. The emergence of recombinants is rare but important events in the context of virus evolution research. One of the most promising among the existing algorithms for searching for recombinants is 3SEQ, but the author's version works only in single-threaded mode. We implemented this algorithm with support for multithreaded computing and taking into account the dates of sample collection, which provided a significant increase in the computing speed. The developed software was used to search for recombinants in the samples of influenza A H1N1 (only PB2 segments from 2174 genomes were analyzed), Dengue fever (726 genomes), Ebola virus (865 genomes) and in two samples of SARS-CoV-2 coronavirus (776 and 2132 genomes). No recombinants were found for influenza A H1N1 (PB2 segment) and the first dataset on SARS-CoV-2 (variant from Russia), which is in agreement with the analysis of the same data by the RDP algorithm. For the second SARS-CoV-2 dataset (variants from the Siberian Federal District), the only recombinant present in the dataset was correctly found. 725 recombinants were found in Dengue fever viruses, with a recombination region length in the range from 50 to 1000 nucleotides. In Ebola viruses, the length of the recombination region was shorter – in 572 recombinants it was in the range of 50 to 100 nucleotides, and in 249 genomes – was less than 50 nucleotides.

Key words: virology, bioinformatics, recombinants detection, 3SEQ algorithm, multithreaded, computational performance, acceleration, software.