ИПМ им.М.В.Келдыша РАН • Электронная библиотека

Препринты ИПМ • Препринт № 108 за 2017 г.

ISSN 2071-2898 (Print)
ISSN 2071-2901 (Online)

Назаров В.И., Клышинский Э.С.

Графовое представление сценариев сборки последовательностей иммунных рецепторов

Рекомендуемая форма библиографической ссылки: Назаров В.И., Клышинский Э.С. Графовое представление сценариев сборки последовательностей иммунных рецепторов // Препринты ИПМ им. М.В.Келдыша. 2017. № 108. 30 с. doi:10.20948/prepr-2017-108
URL: http://library.keldysh.ru/preprint.asp?id=2017-108

О р д е н а Л е н и н а
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ
имени М.В.Келдыша
Р о с с и й с к о й а к а д е м и и н а у к

В.И. Назаров, Э.С. Клышинский

Графовое представление сценариев сборки последовательностей иммунных рецепторов

Москва — 2017

В. И. Назаров, Э. С. Клышинский. *Графовое представление сценариев сборки последовательностей иммунных рецепторов.*

Аннотация. В работе представлена новая методика для анализа данных, полученных из последовательностей иммунных рецепторов. Методика основана на применении графовых алгоритмов и позволяет рассчитывать вероятности сборки всех возможных сценариев как нуклеотидных, так и аминокислотных последовательностей иммунных рецепторов, а также проводить статистический вывод параметров вероятностной модели сборки. На момент написания работы подобный подход применяется впервые в мировой практике. Разработанные алгоритмы для подсчета вероятностей и статистического вывода показывают гораздо большую скорость работы по сравнению с существующими аналогами. В работе приведены результаты численных экспериментов и исследована эффективность распараллеливания алгоритмов.

Ключевые слова: статистическая иммуноинформатика, адаптивный иммунитет, Т-клеточные рецепторы, иммуноглобулины, V(D)J рекомбинация.

V. I. Nazarov, E. S. Klyshinsky. *Graph-based data structure for enumeration of all possible generation scenarios of immune receptor sequences.*

Abstract. In this work we propose a novel graph-based approach to data analysis of immune receptor sequences. We propose algorithms for computing generation probabilities of all possible generation scenarios for both nucleotide and amino acid sequences of immune receptors, and an algorithm for statistical inference of probabilistic generation models for immune receptors. To the best of our knowledge, proposed approach is the first algorithm for computation of immune receptor amino acid sequence's generation probability. Developed algorithms demonstrated dramatically higher speed in contrast to algorithms in previous works. Additionally, we developed parallel versions of our algorithms and tested them on the experimental data.

Keywords: statistical immunoinformatics, adaptive immunity, T-cell receptors, immunoglobulins, V(D)J recombination.

1. Введение

Живые организмы ежемоментно подвергаются атаке самых разнообразных патогенов - вирусов, бактерий и других микроорганизмов. Несмотря на то, что только очень малая часть патогенов проникает непосредственно в организм, они способны нанести ему обширный урон.

Иммунитет — особое биологическое свойство многоклеточных организмов, в норме предназначенное для защиты от инфекций и иных внешних патогенов, способных при попадании во внутреннюю среду вступать в прочные связи с клетками и/или межклеточным веществом. Иммунная система млекопитающих условно делится на две ветви, которые выполняют эту задачу различными путями. Первая ветвь — это врожденный иммунитет, заложенный в геноме организма (геном возможно представить как строку над алфавитом из четырех символов-нуклеотидов — A, C, G, T), который располагает механизмами выявления и уничтожения заранее закодированных патогенов. Но поскольку геном способен хранить лишь ограниченный информационный объем, а патогены имеют способность мутировать, то в нем невозможно закодировать механизмы для распознавания и уничтожения всех возможных патогенов. Поэтому в ходе эволюции появилась вторая ветвь иммунной системы — адаптивный или лимфоцитарный иммунитет, который позволяет организму обучаться защите от наиболее распространенных патогенов в среде обитания. Носителями этой способности служат специализированные клетки — лимфоциты. Уникальным и отличительным свойством лимфоцитов как множества клеток является способность распознавать большое множество (около 10^{18}) разнообразных и эволюционно незапланированных молекулярных объектов (антигенов). После распознавания лимфоцит запускает и мобилизует как собственные, так и общевоспалительные механизмы деструкции поврежденных патогеном тканей, после чего наступает элиминация тканей из организма. [1]

Ключевой особенностью лимфоцитарного иммунитета является распознавание лимфоцитами антигенов. Антигены — это вещества или те формы вещества, которые при введении во внутреннюю среду организма способны индуцировать на себя иммунный ответ в виде выработки специальных клеток — иммунных Т-лимфоцитов или особых белковых соединений-антител из Б-лимфоцитов [1]. Белки возможно представить в виде строк над алфавитом из двадцати символов-аминокислот. Аминокислоты — это специальные органические соединения, из которых строятся белки. Гены хранят информацию о некотором белке и представляют собой нуклеотидную подстроку генома. Опуская несущественные для текущего исследования биологические процессы, можно сказать, что при транскрипции нуклеотидная строка генома превращается в молекулу мРНК (ее возможно представить в виде рибонуклеотидной (РНК) строки, то есть нуклеотидной строки, в которой символы T заменены на сим-

волы U). Также и после в процессе трансляции рибонуклеотидная строка служит "схемой" для построения аминокислотных последовательностей. Синтез белка-аминокислотной последовательности происходит с помощью специальных компонент живых клеток — рибосом. В молекуле мРНК все рибонуклеотиды формируют триграммы (разрешенные последовательности троек рибонуклеотидов), которые называются "кодоны". Рибосома "садится" на молекулу мРНК после узнавания специального стартового кодона AUG в молекуле. После этого рибосома последовательно идет по молекуле мРНК и для каждой следующей триграмы добавляют соответствующую этому кодону аминокислоту к растущему белку, который крепится на рибосоме. После того как рибосома дойдет до одного из "стоп-кодонов" (UAG, UAA, UGA), трансляция останавливается и полученный итоговый белок отделяется от рибосомы. К примеру, генная последовательность ATGCTTGGGTAA в процессе транскрипции превратится в РНК последовательность AUGCUUGGGUAA, и далее, в процессе трансляции, - в белок MLG (стоп-кодон опускается). Возможен вариант, когда рибонуклеотидная строка не имеет стоп-кодонов или ее длина имеет остаток от деления на три отличный от нуля, и тогда белок становится нефункциональным. Для сокращения в будущем будем говорить о трансляции как о транскрипции, за которой следует трансляция. Белки, в отличие от генома, только хранящего информацию, являются "строительными блоками" организма и выполняют очень широкий набор функций: структурную, защитную, сигнальную, двигательную и другие. Лимфоциты — это клетки, которые несут на своей поверхности различные антигенраспознающие белки-рецепторы, участвующие в распознавании антигенов через нековалентные взаимодействия молекул, в котором участвуют четыре типа химических связей — ионные, водородные, вандер-ваальсовы и гидрофобные. В зависимости от силы связи, образующиеся комплексы молекул могут или обратно распасться на составные части антиген/рецептор в случае слабой связи, или продолжить взаимодействовать в случае сильной связи, что потенциально и приводит к иммунному ответу [1].

Антигенраспознающие рецепторы Т- и Б-лимфоцитов называются Т-клеточными рецепторами (ТКР) и Б-клеточными рецепторами (БКР), соответственно. В дальнейшем мы сосредоточимся на природе ТКР, но БКР, за исключением ряда случаев, подчиняются тем же механизмам. ТКР является гетеродимером, то есть состоит из двух равновеликих белковых цепей разного строения. В зависимости от структуры этих белковых цепей, ТКР млекопитающих делится на два типа. В первом типе первая цепь обозначается как альфа, вторая — как бета, а соответствующие Т-лимфоциты — как Т$\alpha\beta$. Т$\alpha\beta$ встречаются среди подавляющего большинства лимфоцитов и являются основной защитой организма от внутриклеточных угроз (вирусов). Во втором типе ТКР цепи обозначаются как гамма и дельта, соответствующие Т-лимфоциты — Т$\gamma\delta$. Функции Т$\gamma\delta$ до сих пор до конца не исследованы [1].

И Т-лимфоциты, и Б-лимфоциты содержат в себе определенные части генома — гены из суперсемейства иммуноглобулинов, — которые служат своего рода "схемой" для создания геномных последовательностей соответствующих рецепторов. Эти гены специальным образом изменяются в стохастическом процессе V(D)J-рекомбинации, после чего полученные геномные последовательности транслируются в белки-рецепторы. V(D)J-рекомбинация — это процесс, в котором ряд генов из суперсемейства иммуноглобулинов вырезается и изменяются. Каждый ген состоит из набора генных сегментов-"подстрок" генома, расположенных близко друг к другу. В соответствии с особенностями своей геномной структуры и поведения в процессе V(D)J-рекомбинации, данные гены и их сегменты называются V(ariable), D(iversity), J(oining) и C(onstant). Число различных генных сегментов, исключая C-сегменты, для генов вариабельных областей семейства иммуноглобулинов человека для основных цепей ТКР и БКР приведены в Таблице 1.

В процессе рекомбинации случайным образом выбирается ряд генных сегментов и происходят события, изменяющие их: удаление случайного числа нуклеотидов с концов сегментов и вставка случайной нуклеотидной строки произвольной длины между концами сегментов. Выбор сегментов происходит за счет активирования генов RAG-1 и RAG-2 и работы ряда ферментов. Из каждого из двух или трех множеств сегментов (из пары множеств V-J или из V-D-J в зависимости от типов цепи — для Tα и Tγ будет VJ рекомбинация, для Tβ и Tδ будет VDJ-рекомбинация) случайным образом выбирается по одному генному сегменту, т.е. для VJ-рекомбинации выбирается один V и один J сегмент, для VDJ-рекомбинации выбирается по одному V, D и J сегменту, после чего выбранные сегменты изменяются: от V-сегмента отрезается случайное число нуклеотидов с правой стороны (3' конец на рис. 1), от J-сегмента отрезается случайное число нуклеотидов с левой стороны (5' конец на рис. 1), от D-сегмента отрезается случайное число нуклеотидов с обеих сторон. После отрезания нуклеотидов между измененными сегментами вставляется случайная последовательность нуклеотидов — между V и J для VJ-рекомбинации, между V и D, а также между D и J для VDJ-рекомбинации. Получившаяся последовательность конкатенируется в итоговую нуклеотидную последовательность и, если последовательность является валидной, транслируется в белок. На рис. 1 показан процесс V(D)J рекомбинации для двух типов клеточных рецепторов — Т-клеточных рецепторов альфа-цепи (TRA или ТКРα) и Т-клеточных рецепторов бета-цепи (TRB или ТКРβ). В случае ТКР получившиеся белки проходят следующий этап — селекцию в специальном органе (тимусе), который уничтожает аутоиммунные рецепторы (способные атаковать сам организм хозяина) или рецепторы, в целом не способные к распознаванию каких-либо молекул.

Таблица 1

Число сегментов генов вариабельных областей семейства иммуноглобулинов человека [1]

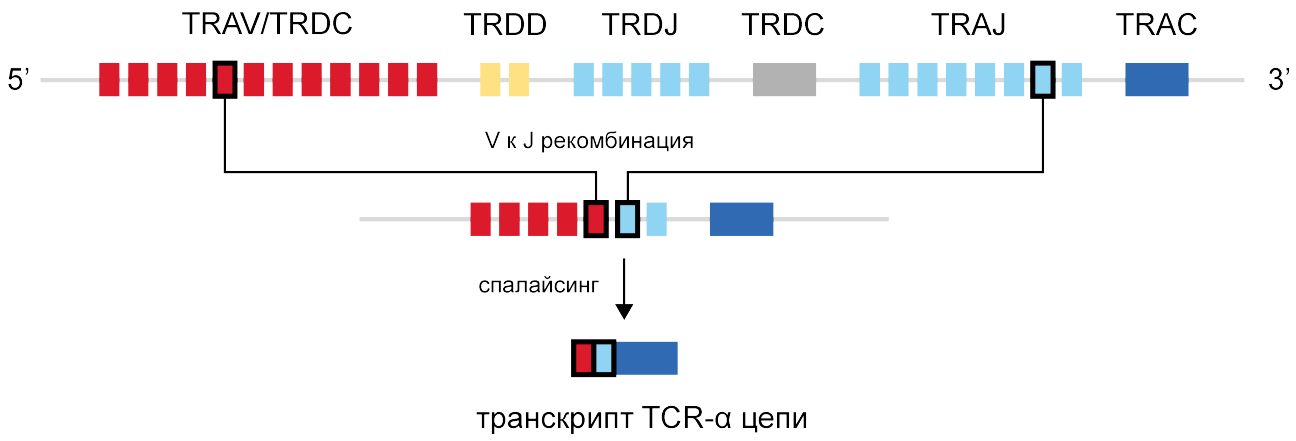
Ген	ТКРα	ТКРβ	БКРκ	БКРλ	БКР тяж.
V	70	52	40	30	50
D	Нет	2	Нет	Нет	30
J	61	13	5	4	6

Одна и та же нуклеотидная последовательность может быть создана разными способами в зависимости от выбора событий, которые формируют эту последовательность. События включают в себя выбор определенных последовательностей из генома (т.е., V, D или J сегментов, располагающихся в определенном порядке: V-D-J), количество удаленных нуклеотидов из определенных сегментов, количество нуклеотидов, вставленных между соседними сегментами (называемые N-нуклеотидами). Каждый кортеж событий, который формирует определенную последовательность, называется "сценарием".

Существующие методы анализа принципиально не могут определить вероятность появления того или иного рецептора. В связи с этим создание описываемой в данной работе вероятностной модели V(D)J-рекомбинации открыло новые возможности как для фундаментальных исследований, так и для решения практических задач. К примеру, с ее помощью стало возможным установить, что ТКР, появляющиеся в младенческом возрасте, способны жить вплоть до старости, хотя считалось, что это не так [3]. Данный факт является важным вкладом в фундаментальное понимание как структуры (другими словами, наличия или отсутствия определенных рецепторов), так и динамики (изменения численности и наличия по времени) множества ТКР в крови человека. Также модель была использована для оценки вероятностей появления рецепторов после минимальной остаточной болезни, где для выживания пациентов критически важно знание о том, собрался ли определенный рецептор случайным образом заново, или уже был в крови до проведения медицинских процедур. Данная информация является важной, так как неточное определение присутствия или отсутствия рецептора в крови может привести к повторению дорогостоющих медицинских процедур, подвергающих опасности жизнь пациента [5]. В перспективе, с помощью такой модели можно будет проверить неподтвержденную до сих пор гипотезу о том, что ТКР, реагирующие на часто встречающиеся патогены, имеют более высокую вероятность сборки по сравнению с ТКР, отвечающих на редкие патогены.

Ранее французскими исследователями предлагалась вероятностная модель процесса V(D)J-рекомбинации [2]. Опубликованная ими программная реализация вероятностной модели позволяет вычислять вероятность того или иного

a) VJ рекомбинация на TRA локусе



b) V(D)J рекомбинация на TRB локусе

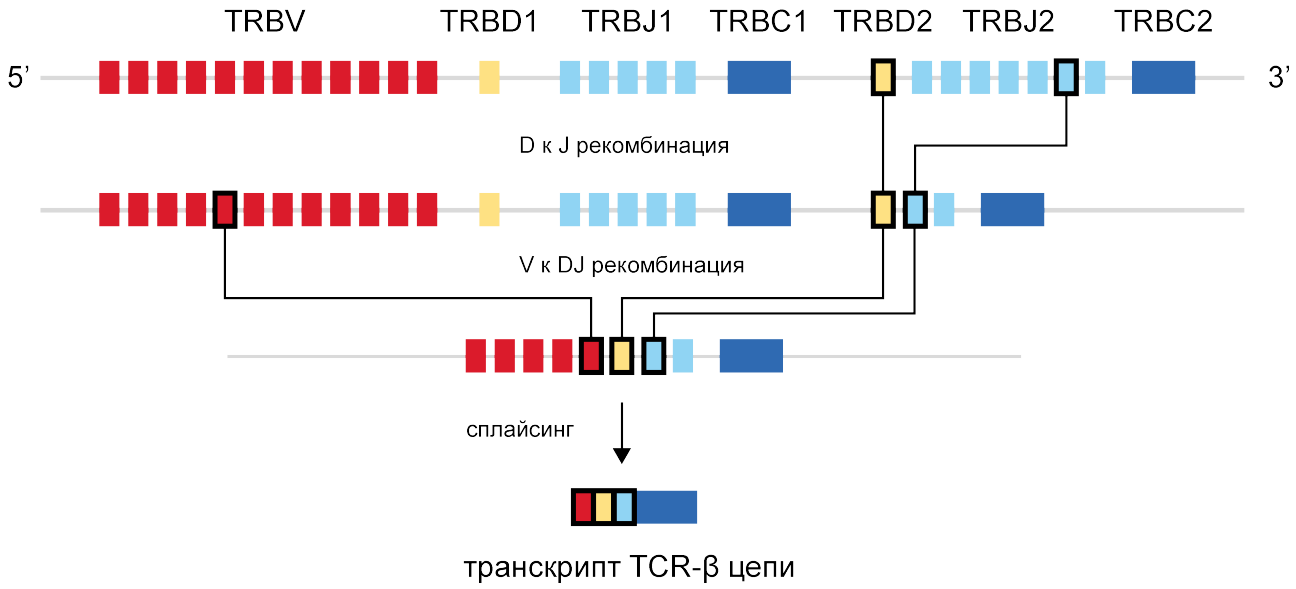


Рис. 1: V(D)J рекомбинация для Т-клеточных рецепторов для **(a)** α- и **(b)** β-цепей. Красным нарисованы сегменты V гена, желтым — D гена, голубым — J гена, синим — C гена. TRAV и TRBV — Variable ген для Tα и Tβ, TRAJ и TRBJ — Joining ген для Tα и Tβ, TRAC и TRBC — Constant ген для Tα и Tβ, соответственно. TRBD — Diversity ген для Tβ.

сценария сборки нуклеотидных последовательностей, а также выводить параметры вероятностей сборки, но требует для этого огромных вычислительных ресурсов. Помимо этого она не позволяет вычислять вероятности сборки аминокислотных последовательностей.

Для устранения этих и ряда других недостатков, в нашей работе предложен новый метод, основанный на применении графов и предназначенный для хранения всех возможных сценариев сборки заданной последовательности рецептора произвольной длины. Его применение позволяет существенно ускорить алгоритмы вычисления вероятностей сборки сценариев нуклеотидных последовательностей и вывода параметров вероятностной модели сборки, а также разработан алгоритм вычисления вероятностей сборки аминокислотных последовательностей.

2. Обзор

Для анализа состояния организма и иммунитета человека исследуют репертуары клеточных рецепторов — множество последовательностей рецепторов, взятых, к примеру, из крови. Анализ репертуаров позволяет понять как фундаментальные закономерности в формировании адаптивного иммунитета человека [3][4], так и закономерности в структуре и динамике репертуаров, связанные с определенными заболеваниями [5][6].

Процесс анализа репертуаров клеточных рецепторов состоит их двух шагов. На первом шаге происходит извлечение последовательностей рецепторов из необработанных данных и их аннотация. Необработанные данные представляют собой множество нуклеотидных последовательностей, для каждого символа которых известна вероятность ошибки в этом символе. Ошибки в данных возникают как из-за несовершенства биотехнологий, применяемых для извлечения данных о рецепторах из биологического материала, так и в ходе их последующей низкоуровневой обработки на секвенаторах — специальной техники для анализа нуклеотидов в биологическом материале.

В процессе извлечения последовательностей рецепторов происходит коррекция ошибок и отсеивание последовательностей с большим числом ошибок. Для каждого рецептора размечаются границы V-D-J сегментов и определяется количество N-нуклеотидов. Сложность данного шага заключается в большом объеме входных данных, а также в ошибочных символах, которых может быть очень много (вплоть до 30% ошибочных символов в последовательностях длиной от 150 до 300) и которые необходимо либо заменить на верные нуклеотиды, либо удалить.

Для обработки и аннотации последовательностей были разработаны раз-

личные методы. Одним из первых подходов было использование скрытых марковских моделей. Программа iHMMune-align [7] начинает с выравнивания V-сегментов из-за их наибольшей длины по сравнению с остальными сегментами, находит наиболее подходящих сегментов-кандидатов и с помощью алгоритма Витерби вычисляет наиболее вероятную комбинацию V-D-J сегментов для данной последовательности. Дальнейшие исследования улучшили этот метод путем анализа всех возможных вариантов аннотации для каждой последовательности и анализа вставок / удалений [8].

Иной подход был предложен в IMGT/High-V-Quest [9], который состоит из следующих шагов: выравнивание V-сегментов без учета вставок и удалений для предварительного выбора кандидатной группы сегментов; определение позиции начала V-сегмента с использованием выравнивания без вставок и удалений; выравнивание конкретных V-сегментов с учетом вставок и удалений; повтора второго и третьего шага для J и D сегментов. IMGT/High-V-Quest является на данный момент самой популярной программой для аннотирования последовательностей рецепторов, хотя многие работы, описывающие другие методы аннотации, показывают, что ее качество разметки и эффективность по времени работы уступают другим методам и программам.

Программа IgBLAST [10] основывается на популярном алгоритме BLAST [11] для выравнивания сегментов на последовательности рецепторов. Алгоритм BLAST на сегодняшний день является наиболее известным эвристическим алгоритмом выравнивания последовательностей. Он основывается на предобработке данных, которая позволяет быстро искать похожие подстроки фиксированной длины k (“k-меры”, k обычно берется равным трем) между исходной последовательностью и базой сегментов. После нахождения общих k-меров происходит отбор наиболее вероятных совпадений, которые далее выравниваются на исходную последовательность.

Оригинальный метод, основанный на графах де Брюйна и раскраске графов, был предложен в работе [12]. Граф де Брюйна строится на k-мерах из нуклеотидных последовательностей сегментов и входных неаннотированных данных. Каждый k-мер, который отвечает сегментам, красится в свой цвет, после чего происходит распространение цвета на k-меры входных данных и определение наиболее вероятного набора сегментов для каждой входной последовательности на основе этих цветов.

Разработанное исследователями из Китая программное обеспечение TCRklass [13] ищет одновременно и нуклеотидные, и аминокислотные k-меры сегментов в исходной последовательности, по которым определяются границы сегментов и происходит коррекция ошибок. TCRklass показывает очень высокую точность работы, но работает на порядок дольше своих аналогов. В отличие от TCRklass, разработчики ViDJiL [14] фокусируются не на качестве идентификации сегментов в рецепторах, а на скорости работы программы и удоб-

стве ее использования. Алгоритм ViDJiL принципиально похож на алгоритм TCRklass и состоит из двух частей. На первом этапе происходит индексация нуклеотидных последовательностей сегментов — каждая последовательность разбивается на k-меры и для каждого k-мера запоминается, какой сегмент его содержит. На втором шаге алгоритм определяет границы сегментов через выравнивание k-меров на исходную последовательность.

Наилучшие результаты показывают программы MiTCR [15] и его новая итерация [16] MiXCR. Алгоритм аннотации MiTCR / MiXCR состоит из трех этапов: выравнивание V-J сегментов, генерация рецепторов и кластеризация. На этапе выравнивания сегментов алгоритм ищет определенные подстроки длиной 5, которые являются подстроками V-J сегментов, и после их нахождения расширяет выравнивание, считая его оценку путем штрафов за несовпадающие нуклеотиды и наград за совпадающие. Все сегменты, которые имеют высокую оценку, становятся кандидатными. Границы сегментов извлекаются в зависимости от позиций выравнивания сегментов. На этапе генерации все нуклеотидные последовательности рецепторов сохраняются в префиксное дерево, причем последовательности с низким качеством либо удаляются из анализа, либо сливаются с одним из существующих рецепторов. На этапе кластеризации производится коррекция ошибок и вычисляется итоговая численность для каждого рецептора.

Заметим, что чаще всего разметка сегментов и подсчет вставленных нуклеотидов происходит путем выбора наиболее протяженного гена, т.е. выбирается только один сценарий, который, как правило, является наиболее вероятным. Для корректного подсчета полной вероятности сборки нельзя использовать только такую разметку, т.к. она является лишь одним из возможных сценариев сборки последовательности рецептора, что приведет к заниженной оценке вероятности сборки данной последовательности. Поэтому необходимо считать вероятности сборки для всех возможных разметок данного рецептора.

После того как последовательности рецепторов были аннотированы, становится возможным проводить анализ репертуаров аннотированных клеточных рецепторов. Наиболее часто в исследованиях необходимо провести сравнительный анализ двух или более репертуаров. Одним из самых популярных методов оценки похожести репертуаров является число общих рецепторов между двумя репертуарами, другими словами, мощность пересечения множеств рецепторов двух репертуаров. Однако в работе [17] было показано, что из-за отсутствия поправок на объем репертуаров, данный метод не является корректным. В связи с этим авторами была предложена нормализация на произведение объемов репертуаров. Другими часто используемыми индексами близости выступают индексы Джаккарда и Соренсена [18]. Описанные индексы не учитывают численность отдельных рецепторов, поэтому для более комплексной оценки похожести репертуаров с учетом распределения рецепторов использу-

ют индекс Мориситы-Хорна [19]. Для сравнения репертуаров не по рецепторам, а по используемым генным сегментам используют корреляцию или дивергенцию Дженсена-Шэннона в сравнении распределений V или других сегментов в репертуаре [17].

Для оценки истинной численности иммунных рецепторов в репертуаре используют различные методы оценки разнообразия. Классическим подходом к оценке разнообразия репертуаров клеточных рецепторов является подсчет индекса экологического разнообразия ${}^{\alpha}D = (\sum_{i=1}^{S} f_i^{\alpha})^{1/(1-\alpha)})$ [20], где f_i - частота i-го рецептора в репертуаре, α — весовой параметр. Данная формула в зависимости от значений параметра α порождает популярные индексы оценки разнообразия репертуаров — насыщенность, энтропию, обратный индекс Симпсона, и другие [20]. Увеличивая значение параметра α, увеличивается вес наиболее представленных рецепторов в оценке разнообразия. В работе [21] было показано, что использование сразу нескольких индексов разнообразия в сравнении разнообразий репертуаров позволяет более точно оценивать влияние различных процессов на репертуары. Альтернативой к индексу экологического разнообразия является индекс Chao1 и его различные модификации [4].

Еще одним подходом к представлению данных репертуаров и их анализу является графовый подход. Репертуары возможно представлять в виде графов, в которых каждая вершина является нуклеотидной или аминокислотной последовательностью, а ребра соединяют последовательности, между которыми определенное расстояние (Хэмминга или Левенштейна). Анализ таких графов заключается в выделении и исследовании групп вершин с высокой и низкой связностью, которые могут характеризовать, например, определенные подпопуляции рецепторов, отвечающие на один и тот же патоген [6]. Другим похожим подходом является построение деревьев эволюции рецепторов — филогенетических деревьев, которые позволяют выделять линии похожих друг на друга рецепторов, которые эволюционировали со временем для улучшения защиты организма от определенных патогенов [22].

Описанные выше подходы к анализу данных репертуаров не лишены одного серьезного недостатка — отсутствию статистического описания процессов, происходящих в репертуарах клеточных рецепторов. При наличии такого описания стало бы возможным оценивать сходство репертуаров и его разнообразие используя строгие статистические выкладки с минимумом предположений о структуре такого сложного объекта, как репертуары клеточных рецепторов. Подобный подход позволит статистически проверить ряд предположений, принимаемых как аксиомы, а также в дальнейшем разработать статистические модели предсказания структуры и динамики репертуара в зависимости от влияния различных факторов, таких как заражение определенными вирусами.

В последние несколько лет появилось несколько работ, которые подходят к решению данной проблемы. В статье [2] исследователи предложили следую-

щую вероятностную модель процесса V(D)J рекомбинации (т.е., процесса сборки нуклеотидных последовательностей клеточных рецепторов) для ТКРβ:

$$P^{\beta}(\sigma) = P(V)P(J,D)P(delV|V)(delJ|J)P(delD3',delD5'|D)\times \\ P(insVD)\prod_{i=1}^{insVD} p_{VD}(x_i|x_{i-1}) \times P(insDJ)\prod_{i=1}^{insDJ} p_{DJ}(y_i|y_{i-1}), \quad (1)$$

где $P^{\beta}(\sigma)$ — вероятность сборки сценария σ ТКРβ,
$P(V)$ — вероятность выбора определенного V сегмента,
$P(J,D)$ — вероятность выбора определенной пары JD сегментов,
$P(delV|V)$ — вероятность $delV$ удалений для выбранного сегмента V,
$(delJ|J)$ — вероятность $delJ$ удалений для выбранного сегмента J,
$P(delD3',delD5'|D)$ — совместная вероятность удалений с обоих концов выбранного сегмента D,
$P(insVD)$ — вероятность вставки длиной $insVD$ между V и D сегментами,
$P(insDJ)$ — вероятность вставки длиной $insDJ$ между D и J сегментами,
$p_{VD}(x_i|x_{i-1})$ — вероятность вставки определенного нуклеотида x_i в зависимости от предыдущего x_{i-1} между V и D сегментами,
$p_{DJ}(y_i|y_{i-1})$ — вероятность вставки определенного нуклеотида y_i в зависимости от предыдущего y_{i-1} между D и J сегментами.

В статье [3] была предложена модель VJ рекомбинации для ТКРα:

$$P^{\alpha}(\sigma) = P(V,J)P(delV|V)(delJ|J)P(insVJ)\prod_{i=1}^{insVJ} p_{VJ}(x_i), \quad (2)$$

$P^{\alpha}(\sigma)$ — вероятность сборки сценария σ ТКРα,
$P(V,J)$ — вероятность выбора определенной пары V и J сегментов,
$P(delV|V)$ — вероятность $delV$ удалений для выбранного сегмента V,
$P(delJ|J)$ — вероятность $delJ$ удалений для выбранного сегмента J,
$P(insVJ)$ — вероятность вставки длиной $insVJ$ между V и J сегментами,
$p_{VJ}(x_i)$ — вероятность вставки определенного нуклеотида x_i между V и J сегментами.

С помощью EM-алгоритма на основе некодирующих последовательностей из одного репертуара выводятся параметры вероятностной модели.

В обеих рассмотренных работах используется одно и то же программное обеспечение для работы с вероятностными моделями, которое обладает двумя существенными недостатками. Первый недостаток — это высокая вычислительная сложность предложенных алгоритмов: в вычислительных экспериментах в статье [2] сообщалось о 20 часах, необходимых для статистического вы-

вода модели одного репертуара на вычислительном кластере. Второй недостаток — это принципиальная невозможность посчитать вероятности аминокислотных последовательностей эффективным путем, т.е. без подсчета вероятностей для каждой нуклеотидной последовательности, которая может кодировать данную аминокислотную последовательность. К примеру, для аминокислотной последовательности CASSLGGYETFF одного из иммунных рецепторов всего существует 573308928 нуклеотидных последовательностей, которые ее кодируют, а это число больше числа клеточных рецепторов в любом репертуаре, полученом экспериментальным путем, на сегодняшний день. В данной работе мы представляем структуру данных и алгоритмы на ее основе, которые решают обе эти проблемы. Проведенные вычислительные эксперименты показали ее высокую эффективность и применимость к существующим реальным данным.

3. Графовая структура данных для представления сборки последовательностей

Происходящую в организме V(D)J-рекомбинацию рецепторов можно промоделировать в виде случайного процесса, производящего конкатенацию модифицированных нуклеотидных строк.

На первом этапе VJ-рекомбинации в организме случайным образом выбирается нуклеотидная последовательность некоторого V-сегмента. Затем с его правого края удаляются несколько нуклеотидов. Получившаяся строка конкатенируется справа с последовательностью случайных нуклеотидов случайно выбранной длины, называемых N-нуклеотидами. Затем выбирается некоторая последовательность J-сегмента, у которой с левого края удаляется несколько нуклеотидов, и получившаяся строка конкатенируется с предыдущей с правой стороны. Получившаяся последовательность $V - N - J$ и будет последовательностью клеточного рецептора. В случае VDJ-рекомбинации происходит похожий процесс, но только для D-сегмента удаления происходит с двух сторон, и есть две последовательности N-нуклеотидов: $V - N_1 - D - N_2 - J$.

Описанная выше программная реализация [2] использовала следующий метод для подсчета вероятностей сборки. Для определения всех возможных сценариев, которыми могла быть создана определенная последовательность, перебирались все возможные подстроки, соответствующие событиям, и для каждой комбинации подстрок подсчитывалась вероятность сборки. Подобный подход не учитывает тот факт, что одни и те же события встречаются в большом числе сценариев. Например, если последовательность рецептора состоит из нуклеотидов TTTTGGGCCCC, то возможно несколько сценариев сборки. В первом из

сценариев V соответствует TTTT, вставленные N-нуклеотиды соответствуют GGG, а J — CCCC; во втором сценарии возможен вариант, когда V соответствует TTT, а N-нуклеотиды - TGGG. Легко видеть, что оба описанных сценария включают в себя событие существования последовательности CCCC для J сегмента. В соответствии с методом, описанным выше, приходится всякий раз заново вычислять вероятность CCCC для J сегмента несмотря на то, что его вероятность для двух сценариев остается одной и той же.

Разработанный нами подход решает данную проблему необходимости пересчета вероятностей событий, которые встречались ранее. Ключевой идеей описываемого в данной работе метода является представление каждой последовательности в виде ациклического взвешенного графа, в котором каждая вершина соответствует определенной позиции в исходной последовательности, а каждая дуга — определенной подстроке-событию. Дуги соединяют вершины таким образом, что прохождение по последовательности дуг дает полную последовательность. Веса на дугах равны вероятностям событий, расположенных на этих дугах. Таким образом, произведение весов на последовательности дуг дает вероятность сборки сценария. Например, для описанной выше последовательности TTTTGGGCCCC описываемый метод построит граф, в котором из вершины, соответствующей последнему нуклеотиду G на седьмой позиции, будет идти дуга с последовательностью CCCC. В эту же вершину будут входить дуги для последовательностей GGG и TGGG. Дуга TTTT будет приходить в вершину, из которой начинается дуга GGG, а дуга TTT — в вершину, из которой начинается TGGG. Построив один раз последовательность дуг TTTT-GGG-CCCC, нет необходимости перестраивать и пересчитывать вероятность сборки CCCC для последовательности TTT-TGGG-CCCC, поскольку дуга CCCC уже имеется в графе. Такой подход позволяет избежать перечисления одних и тех же событий, которые встречаются среди большого числа сценариев, таким образом минимизируя общее число вычислений. Эффект оптимизации тем сильнее, чем больше повторяющихся событий.

Назовем такой граф "графом сборки". Поскольку известны границы подстрок, из которых состоит исходная последовательность рецептора, то возможно поставить крайние позиции подстрок в соответствии с вершинами графа сборки. Соединив дугами вершины, соответствующие началу и концу одной подстроки (например, началу и концу D-сегмента), или вершины между соседними событиями (например, между V-сегментом и N-нуклеотидами), получим, что каждая дуга условно соответствует определенной подстроке, которая в свою очередь соответствует определенному событию. Поставив на дугах веса, равные вероятности соответствующему событию, получим, что каждый сценарий сборки кодируется одним путем в таком графе.

Для каждой последовательности рецепторов заранее заданы возможные сегменты V, J (в случае VJ-рекомбинации) и V, D и J (в случае VDJ-рекомбинации)

и их расположение на последовательности, т.е. в каких позициях эти сегменты начинаются и заканчиваются. Для данной последовательности s будем обозначать сегменты, которые для нее заданы, как V_i^s, J_j^s и D_k^s, где i, j и k принимают значения от единицы до числа известных сегментов для каждого типа сегмента. Некоторые сегменты могут пересекаться, т.е. располагаться таким образом на последовательности, что имеют общие нуклеотидные позиции.

Обозначим через $pos(v_\omega)$ позицию на последовательности, которая соответствует вершине v, которая, в свою очередь, соответствует некоторому событию ω, например, пяти вставленным нуклеотидам начиная с четвертой позиции, или трем удалениям V сегмента. В случае события трех удалений из определенного V-сегмента, именованного как V_4, для обозначения такой вершины мы будем писать $v_{d_3|V_4}$, и $pos(v_{d_3|V_4})$ для обозначения позиции в последовательности, начиная с которой данный сегмент имеет три удаления.

Введем вектор Z вероятностей для всех событий; Z_i — i-й элемент вектора. Каждое событие - выбор определенного сегмента, определенное количество удалений для конкретных сегментов, число вставленных нуклеотидов и т.д. - будет иметь определенный индекс, и Z_i будет соответствовать вероятности наступления такого события. Пусть есть некоторое событие ω. Обозначим через $event(\omega)$ индекс события ω в Z. Каждой дуге (v_i, v_j) мы ставим в соответствие вероятность $P(\omega) \equiv Z_{event(\omega)}$ события ω и индекс этого события $event(\omega)$. Обозначим вероятность на такой дуге из v_i в v_j как $prob(v_i, v_j) \equiv P(\omega)$.

Назовем графом сборки для последовательности s граф с вершинами, которые отвечают всем тем событиям, которые могли произойти при сборке такой последовательности, и дугами, соединяющим вершины.

Графы сборки с фиксированными сегментами $V - D - J$ возможно закодировать в виде матрицы смежности, где каждые строка и столбец соответствуют некоторой вершине, а каждый элемент на позиции i, j матрицы содержит вес дуги, которая проведена из вершины с номером i в вершину с номером j.

Поскольку каждый класс событий (выбор сегментов, удаления, и т.д.) изолирован друг от друга, т.е. сборка идет последовательно, это означает, что вершины одного класса событий соединены только с соседними классами событий (например, вершины удалений V-сегмента соединены только с выбором V-сегмента (входящие дуги) и с вершинами, соответствующими N-нуклеотидам (выходящие дуги)).

Таким образом, в матрице смежности будет много нулей и ее можно представить как последовательность отдельных матриц, каждая из которых соответствует определенному классу событий. Назовем эти подматрицы матрицами классов событий M_Ω, где Ω соответствует классу событий — выбор сегмента, удаления и т.д.

В случае, если для представления нуклеотидной последовательности возможно использовать несколько сегментов одного типа, то структура данных

строится таким образом, что для классов событий строятся матрицы одинаковой размерности для каждого возможного сегмента. К примеру, если данная последовательность ТКРα кодируется двумя V-сегментами и тремя J-сегментами, то для класса событий "удаления из последовательности сегмента V" будут сохранены две матрицы, а для класса событий "удаления из последовательности сегмента J" будут сохранены три матрицы. Для представления в памяти компьютера используется один вектор со всеми значениями элементов матрицы, в котором известно, какие значения относятся к какой матрице и какому классу событий. В следующих подразделах мы подробнее рассмотрим реализации такой графовой структуры данных для представления сценариев сборки для обоих видов V(D)J рекомбинации и для нуклеотидных / аминокислотных последовательностей.

3.1. Представление сборки нуклеотидных последовательностей и вычисление вероятностей сборки

Алгоритм построения графа сборки G^{nuc}_{VJ} для нуклеотидных последовательностей иммунных рецепторов VJ рекомбинации.

1. Инициализировать G^{nuc}_{VJ} двумя вершинами (начальная v_S и конечная v_F) с нулевыми степенями.

2. Для выравненных V^s_i и всех возможных чисел удалений j создать вершину $v_{d_j|V^s_i}$ с позицией $pos(v_{d_j|V^s_i})$ и построить дугу $(v_S, v_{d_j|V^s_i})$ с весом $P(d_j|V^s_i)$ и индексом события $event(d_j|V^s_i)$.

3. Для выравненных J^s_i и всех возможных чисел удалений j создать вершину $v_{d_j|J^s_i}$ с позицией $pos(v_{d_j|J^s_i})$ и построить дугу $(v_{d_j|J^s_i}, v_F)$ с весом $P(d_j|J^s_i)$ и индексом события $event(d_j|J^s_i)$.

4. Построить дуги $(v_{d_j|V^s_i}, v_{d_m|J^s_k}) : pos(v_{d_j|V^s_i}) < pos(v_{d_m|J^s_k})$ с весами $P_{ins}(pos(v_{d_m|J^s_k}) - pos(v_{d_j|V^s_i}) - 1) \times \prod_{n=pos(v_{d_j|V^s_i})}^{pos(v_{d_m|J^s_k})} P_{nuc}(s[n])$. Каждой дуге ставится в соответствие индекс события вставки N-нуклеотидов длиной $pos(v_{d_m|J^s_k}) - pos(v_{d_j|V^s_i}) - 1$.

Алгоритм построения графа сборки G^{nuc}_{VDJ} для нуклеотидных последовательностей иммунных рецепторов V(D)J рекомбинации.

1. Инициализировать G^{nuc}_{VDJ} двумя вершинами (начальная v_S и конечная v_F) с нулевыми степенями.

2. Для выравненных V_i^s построить вершины $v_{V_i^s}$ и соединить их дугами $(v_S, v_{V_i^s})$ с весами $P(V_i^s)$ и индексами событий $event(V_i^s)$.

3. Для выравненных V_i^s и всех возможных чисел удалений j для этого сегмента создать вершину $v_{d_j|V_i^s}$ с позицией $pos(v_{d_j|V_i^s})$ и построить дугу $(v_{V_i^s}, v_{d_j|V_i^s})$ с весом $P(d_j|V_i^s)$ и индексом события $event(d_j|V_i^s)$.

4. Для выравненных J_i^s и всех возможных чисел удалений j создать вершину $v_{d_j|J_i^s}$ с позицией $pos(v_{d_j|J_i^s})$ и построить дугу $(v_{d_j|J_i^s}, v_F)$ с весом $P(d_j|J_i^s)$ и индексом события $event(d_j|J_i^s)$.

5. Для выравненных D_i^s и всех возможных пар чисел удалений j^{left}, j^{right} создать вершины $v_{d_j^{left}|D_i^s}$ и $v_{d_j^{right}|D_i^s}$ с позициями $pos(v_{d_j^{left}|D_i^s})$ и $pos(v_{d_j^{right}|D_i^s})$, соответственно, и построить дугу $(v_{d_j^{left}|D_i^s}, v_{d_j^{right}|D_i^s})$ с весом $P(d_j^{left}, d_j^{right}|D_i^s)$ и индексом события $event(d_j^{left}, d_j^{right}|D_i^s)$.

6. Построить дуги $(v_{V_i^s,d_j}, v_{D_k^s,d_m^{left}}) : pos(v_{d_j|V_i^s}) < pos(v_{d_m^{left}|D_k^s})$ с весами $P_{ins}(pos(v_{d_m^{left}|D_k^s}) - pos(v_{d_j|V_i^s}) - 1) \times \prod_{n=pos(v_{d_j|V_i^s})+1}^{pos(v_{d_m^{left}|D_k^s})} P_{nuc}(s[n]|s[n-1])$ и индексами событий вставок N-нуклеотидов между сегментами V и D длиной $pos(v_{d_m^{left}|D_k^s}) - pos(v_{d_j|V_i^s}) - 1$.

7. Построить дуги $(v_{D_k^s,d_m^{right}}, v_{J_i^s,d_j}) : pos(v_{d_m^{right}|D_k^s}) < pos(v_{d_j|J_i^s})$ с весами $P_{ins}(pos(v_{d_j|J_i^s}) - pos(v_{d_m^{right}|D_k^s}) - 1) \times \prod_{n=pos(v_{d_m^{right}|D_k^s})+1}^{pos(v_{d_j|J_i^s})} P_{nuc}(s[n]|s[n-1])$ и индексами событий вставок N-нуклеотидов между сегментами D и J длиной $pos(v_{d_j|J_i^s}) - pos(v_{d_m^{right}|D_k^s}) - 1$.

Полная вероятность сборки $P(s)$ нуклеотидной последовательности s некоторого клеточного рецептора - это сумма вероятностей сборки всех сценариев $P(\sigma)$, которые приводят к данной последовательности. Поскольку в графах сборки G_{VJ}^{nuc} не запоминается вероятность $P(V, J)$, а в G_{VDJ}^{nuc} не запоминается вероятность $P(J, D)$, то их необходимо запомнить отдельно и умножать при подсчете вероятностей сценариев с соответствующими парами сегментов. В силу того, что каждый путь в графах сборки является определенным сценарием, то полная вероятность сборки будет равна сумме произведений всех событий на каждом пути в графах сборки:

$$P^{VJ}(s) = \sum_{\sigma} P(\sigma) = \sum_{\sigma} \Big[P_\sigma(V,J) P_\sigma(delV|V)(delJ|J) \times \\ P_\sigma(insVJ) \prod_{i=1}^{insVJ} p_{VJ}(x_i|x_{i-1}) \Big], \tag{3}$$

$$P^{VDJ}(s) = P_\sigma(V) P_\sigma(J,D) P_\sigma(delV|V)(delJ|J) P_\sigma(delD3', delD5'|D) \times \\ P_\sigma(insVD) \prod_{i=1}^{insVD} p_{VD}(x_i|x_{i-1}) \times P_\sigma(insDJ) \prod_{i=1}^{insDJ} p_{DJ}(y_i|y_{i-1}), \tag{4}$$

где $P_\sigma(\omega)$ обозначает конкретную вероятность события ω в сценарии σ. В случае множественных V_i^s, D_j^s и J_k^s для эффективного подсчета полной вероятности сборки для каждой вершины v_i каждого класса, начиная с выбора v_S, считается значение $f(v_i)$:

$$f(v_S) = 1, \\ f(v_i) = \sum_{\exists(v_j, v_i)} f(v_j) prob(v_j, v_i). \tag{5}$$

После выполнения таких подсчетов, значение $f(v_F)$ будет равно полной вероятности сборки. В случае одиночных V_i^s, D^s и J_k^s полная вероятность сборки рецептора вычисляется как произведение матриц классов событий, что позволяет использовать программные библиотеки для линейной алгебры (например, BLAS или LAPACK), в которых реализовано оптимизированное матричное произведение.

3.2. Представление сборки аминокислотных последовательностей и вычисление вероятностей сборки

Аминокислотные последовательности получаются из нуклеотидных путем преобразования каждого следующего друг за другом нуклеотидного триплета-кодона в определенную аминокислоту, и каждая аминокислота может кодироваться цепочками от одного до шести кодонов.

Как и для нуклеотидных последовательностей, для каждой аминокислотной последовательности рецепторов заранее заданы возможные сегменты V, J

(в случае VJ рекомбинации) и D (в случае V(D)J рекомбинации) и их расположение на последовательности.

Будем обозначать $codon(v_i)$ для некоторой вершины v_i битовый вектор длиной шесть, который показывает, какие кодоны были использованы для кодирования аминокислоты, которой соответствует нуклеотидная позиция $pos(v_i)$. Например, если вершина v_i соответствует аминокислоте L, которая может кодироваться шестью кодонами - TTA, TTG, CTT, CTC, CTA, CTG, и, исходя из входных данных, в позиции v_i находится только первый и второй кодон, то $codon(v_i) = [1, 1, 0, 0, 0, 0]$. Битовое представление позволяет эффективно хранить информацию о кодонах в памяти компьютера.

Построение таких битовых векторов для позиций происходит следующим образом. Предположим, что мы рассматриваем определенный V-сегмент, который выровнялся на двенадцать нуклеотидных позиций. Он занимает четыре кодона со следующими граничными позициями на последовательности: $[1, 3]$, $[4, 6]$, $[7, 9]$, $[10, 12]$. Предположим, что последние три нуклеотида (на десятой, одиннадцатой и двенадцатой позициях) этого сегмента это CTA, и предположим, что эти три позиции соответствуют аминокислоте L (другими словами, в исходной аминокислотной последовательности рецептора, на который выравнивается этот V-сегмент, на четвертой позиции стоит L). Известно, что аминокислота L может задаваться последовательностью нуклеотидов CT*, TTA или TTG. Если мы допускаем, что было произведено удаление в одиннадцатой и двенадцатой позициях, то, начиная с десятой позиции, могут быть только кодоны CTT, CTC, CTA и CTG, поскольку в десятой позиции в V-сегменте стоит нуклеотид C. Тогда соответствующий битовый вектор будет равен $[0, 0, 1, 1, 1, 1]$. На одиннадцатой позиции в случае одного удаления тоже могут быть только такие же кодоны, поскольку в одиннадцатой позиции стоит нуклеотид T. Но нуклеотид на двенадцатой позиции может соответствовать только кодону CTA, и тогда битовый вектор на двенадцатой позиции будет равен $[0, 0, 0, 0, 1, 0]$. Результирующие битовые векторы для позиций следующие: для десятой — $[0, 0, 1, 1, 1, 1]$, для одиннадцатой — $[0, 0, 1, 1, 1, 1]$, для двенадцатой — $[0, 0, 0, 0, 1, 0]$. Данный пример приведен для ситуации, когда удаления идут справа налево (т.е. для V-сегментов и правой части D-сегментов). Если удаления идут слева направо (т.е. для J-сегментов и левой части D-сегментов), то происходит та же операция, но уточнение битового вектора идет в обратную сторону, справа налево.

Таким образом, ключевым отличием графов сборки для аминокислотных последовательностей является то, что для каждой позиции в аминокислотной последовательности известны кодоны, которые могут соответствовать этой позиции, и, поскольку каждая аминокислота кодируется набором триплетов из нуклеотидов, каждой аминокислотной позиции в последовательности будут соответствовать по три вершины графа сборки, каждая из которых будет содер-

жать индекс события, вероятность события и возможные кодоны.

Алгоритм построения графа сборки G^{aa}_{VJ} для аминокислотных последовательностей иммунных рецепторов VJ рекомбинации:

1. Инициализировать G^{aa}_{VJ} двумя вершинами (начальная v_S и конечная v_F) с нулевыми степенями.

2. Для выровненных V^s_i и всех возможных чисел удалений j создать вершину $v_{V^s_i,d_j}$ с позицией $pos(v_{d_j|V^s_i})$ и возможными кодонами $codon(v_{d_j|V^s_i})$ и построить дугу $(v_S, v_{V^s_i,d_j})$ с весом $P(d_j|V^s_i)$ и индексом события $event(d_j|V^s_i)$.

3. Для выровненных J^s_i и всех возможных чисел удалений j создать вершину $v_{d_j|J^s_i}$ с позицией $pos(v_{d_j|J^s_i})$ и возможными кодонами $codon(v_{d_j|J^s_i})$ и построить дугу $(v_{d_j|J^s_i}, v_F)$ с весом $P(d_j|J^s_i)$ и индексом события $event(d_j|J^s_i)$.

Алгоритм построения графа сборки G^{aa}_{VDJ} для аминокислотных последовательностей иммунных рецепторов VDJ рекомбинации.

1. Инициализировать G^{aa}_{VDJ} двумя вершинами (начальная v_S и конечная v_F) с нулевыми степенями.

2. Для выравненных V^s_i построить вершины $v_{V^s_i}$ и соединить их дугами $(v_S, v_{V^s_i})$ с весами $P(V^s_i)$ и индексами событий $event(v_{V^s_i})$.

3. Для выравненных V^s_i и всех возможных чисел удалений j создать вершину $v_{d_j|V^s_i}$ с позицией $pos(v_{d_j|V^s_i})$ и возможными кодонами $codon(v_{d_j|V^s_i})$ и построить дугу $(v_S, v_{d_j|V^s_i})$ с весом $P(d_j|V^s_i)$ и индексом события $event(v_{d_j|V^s_i})$.

4. Для выравнненых J^s_i и всех возможных чисел удалений j создать вершину $v_{d_j|J^s_i}$ с позицией $pos(v_{d_j|J^s_i})$ и возможными кодонами $codon(v_{d_j|J^s_i})$ и построить дугу $(v_{d_j|J^s_i}, v_F)$ с весом $P(d_j|J^s_i)$ с индексом события $event(d_j|J^s_i)$.

5. Для выравненных D^s_i и всех возможных пар чисел удалений j^{left}, j^{right} создать вершины $v_{D^s_i,d^{left}_j}$ и $v_{D^s_i,d^{right}_j}$ с позициями $pos(v_{d^{left}_j|D^s_i})$ и $pos(v_{d^{right}_j|D^s_i})$, соответственно, и построить дугу $(v_{D^s_i,d^{left}_j}, v_{D^s_i,d^{right}_j})$ с весом $P(d^{left}_j, d^{right}_j|D^s_i)$ и индексом события $event(d^{left}_j, d^{right}_j|D^s_i)$. Присвоить вершинам $v_{d^{left}_j|D^s_i}$ возможные кодоны $codon(v_{d^{left}_j|D^s_i})$, вершинам $v_{d^{right}_j|D^s_i}$ — возможные кодоны $codon(v_{d^{right}_j|D^s_i})$.

В приведенных алгоритмах построения графа сборки для аминокислотных последовательностей, в отличие от алгоритмов построения G^{nuc}_{VJ} и G^{nuc}_{VDJ}, не строятся ребра, соответствующие вставкам между сегментами, поскольку из-за того, что невозможно однозначно задать битовые векторы для вставленных нуклеотидов для всех пар соседних сегментов, на это потребуются очень большие затраты по памяти. Поэтому для подсчета полной вероятности сборки для определенной пары или тройки сегментов сначала рассчитывается вероятность вставки N-нуклеотидов для этих сегментов путем перебора всех возможных нуклеотидных последовательностей, которые могли бы вставить между сегментами в исходной аминокислотной строке, а потом считается полная вероятность сборки по формулам (3) и (4). Для повышения скорости подсчета вероятности вставки N-нуклеотидов для G^{aa}_{VJ} возможно предварительно рассчитать всевозможные вставки длиной в один, два или три нуклеотида, и при подсчете вероятности перемножать их. Для этого необходимо построить хэш-таблицу, в которой ключами будут служить кортежи $\langle i, \alpha, \phi \rangle$, где i - позиция в кодоне, т.е. либо первая, либо вторая, либо третья позиция, α — аминокислота, ϕ — направление удалений (либо слева направо, либо справа налево), а значениями будет вероятность вставки такого вида. Для повышения скорости подсчета вероятности вставки N-нуклеотидов для G^{aa}_{VDJ} предварительный расчет всех возможных ставок является слишком затратным по памяти, поскольку вставки для VDJ рекомбинации описываются марковской цепью первого порядка. Возможно предварительно рассчитать подмножество всех возможных вставок и использовать их при встрече вставок из такого подмножества.

В случае, если в исходной аминокислотной последовательности есть только по одному V, D и J, то можно хранить всю информацию в памяти, что позволит существенно ускорить процесс подсчета вероятности сборки.

4. Статистический вывод параметров модели сборки клеточных рецепторов

Для статистического вывода параметров модели сборки клеточных рецепторов используется алгоритм, подобный алгоритму Баума-Велша для скрытых марковских моделей.

1. На вход алгоритм получает $\mathcal{R}$ (репертуар последовательностей s_i с известными сегментами) и два параметра: ϵ (значение разницы между правдоподобиями новой и старой модели, при котором процесс обучения останавливается) и n_{max} (максимальное число итераций).

2. Инициализировать параметры Z вероятностной модели сборки равномерным распределением внутри классов событий. Инициализировать $\mathcal{L}' = 0, \mathcal{L} = 0$.

3. Построить $\mathcal{G} : G_i = G(s_i; Z)$ — множество графов сборки G для каждой последовательности $s_i \in \mathcal{R}$ с использованием параметров Z.

4. Инициализировать новые параметры Z' модели сборки нулями.

5. Для каждого $G_i \in \mathcal{G}$:

   5.1. Для всех выравненных V, J в G_i инициализировать $f(v_S, v_V) = 1; b(v_J, v_F) = 1$.

   5.2. Для каждого ребра $(v_i, v_j) \in G_i$ вычислить значение $f(v_i, v_j) = \sum_{\exists (v_k, v_i) \in G_i} f(v_k, v_i) prob(v_i, v_j)$.

   5.3. Для каждого ребра $(v_i, v_j) \in G_i$ вычислить значение $b(v_i, v_j) = \sum_{\exists (v_j, v_k) \in G_i} b(v_j, v_k) prob(v_j, v_k)$.

   5.4. Для каждого события ω на всех дугах $(v_i, v_j) \in G_i$ с событиями ω вычислить $Z'_{event(\omega)} = Z'_{event(\omega)} + \frac{f(v_i, v_j) b(v_i, v_j)}{P(s_i)}$.

6. Нормализовать события в Z'.

7. $Z = Z'$.

8. $\mathcal{L}' = \mathcal{L}; \mathcal{L} = \prod_{i=1}^{|\mathcal{R}|} P(s_i)$.

9. Повторять шаги $3 - 7$, пока $\mathcal{L} - \mathcal{L}' > \epsilon$ или пока не будет совершено n_{max} итераций.

Для оптимизации вычислений на шаге 3 возможно не строить заново все графы сборки. Зная индексы событий для каждого ребра $event(\omega)$, можно присвоить новые вероятности событий $Z_{event(\omega)}$ соответствующим ребрам.

5. Результаты вычислительных экспериментов

Поскольку не всегда возможно определить, какие сегменты V, D и J участвовали в сборке последовательности, то необходимо провести вычислительные эксперименты для двух случаев — когда для каждой последовательности точно известны единственные V, D и J ("ед." в таблицах 1 и 2), и когда для каждой

последовательности есть набор возможных V, D и J ("мн." (от "множественные")). Для каждого случая были реализованы алгоритмы параллельного построения графов сборки и опробованы на четырех- и восьмиядерных процессорах (помеченные как "(x4)" и "(x8)", соответственно). Эксперименты по построению графов и вычислению вероятностей сборки были проведены на объеме данных в 500000 нуклеотидных (таблица 1) и аминокислотных (таблица 2) последовательностей.

Таблица 2

Время построения графов и вычисления вероятностей сборки нуклеотидных последовательностей (500000 последовательностей)

	VJ (ед.)	VDJ(ед.)	VJ (мн.)	VDJ (мн.)
Построение графов	18 сек	53 сек	72 сек	90 сек
Построение графов (x4)	6 сек	16 сек	28 сек	28 сек
Построение графов (x8)	5 сек	14 сек	23 сек	25 сек
Вычисление вероятностей	13 сек	52 сек	75 сек	9 мин
Вычисление вероятностей (x4)	4 сек	16 сек	35 сек	2 мин
Вычисление вероятностей (x8)	3 сек	13 сек	31 сек	75 сек

Таблица 3

Время построения графов и вычисления вероятностей сборки аминокислотных последовательностей (500000 последовательностей)

	VJ (ед.)	VDJ(ед.)	VJ (мн.)	VDJ (мн.)
Построение графов	10 сек	1 мин	56 сек	2 мин
Построение графов (x4)	4 сек	32 сек	20 сек	46 сек
Построение графов (x8)	3 сек	22 сек	16 сек	34 сек
Вычисление вероятностей	24 сек	8 час.	30 мин	9 час.
Вычисление вероятностей (x4)	9 сек	3 часа	12 мин	5 час.
Вычисление вероятностей (x8)	6 сек	2 часа	6 мин	4 часа

Вычисление аминокислотных вероятностей занимает гораздо большее время, нежели нуклеотидных, но даже в данном случае время расчета существенно сокращается в сравнении со способом, когда создаются все возможные нуклеотидные последовательности для определенной аминокислотной последовательности. Поскольку одна аминокислотная последовательность ТКРβ может быть закодирована в среднем $10{\cdot}10^{10}$ нуклеотидными последовательностями, то для подсчета по последнему способу необходимо будет $10{\cdot}10^{10}{\cdot}75/60/60/24/365 = 23782$ года в сравнении с девятью часами с использованием разработанных алгоритмов.

Эксперименты по статистическому выводу параметров модели сборки были проведены на 100000 последовательностях на 30 итерациях.

Таблица 4

Статистический вывод параметров вероятностной модели (100000 последовательностей, 30 итераций)

	VJ (ед.)	VDJ(ед.)	VJ (мн.)	VDJ (мн.)
Вывод параметров	3 мин	50 мин	49 мин	3 часа
Вывод параметров (x4)	42 сек	16 мин	13 мин	48 мин
Вывод параметров (x8)	34 сек	14 мин	11 мин	42 мин

Предыдущие исследования [2] докладывали о работе на 8-ядерном вычислительном кластере в течение 20 часов для вывода параметров вероятностной модели сборки ТКРβ на примерно таком же объеме данных. Настоящие вычислительные эксперименты показывают, что в случае работы на одноядерном процессоре время вывода параметров модели сборки уменьшается примерно в 6 раз, в случае работы на четырехъядерном процессоре — в 25 раз, а в случае восьмиядерного процессора — примерно в 28 раз. По таблицам видно, что разница между непараллельными алгоритмами и параллельными алгоритмами в четыре процесса различается примерно в четыре раза, тогда как разница между параллельными алгоритмами (в четыре и в восемь процессов) гораздо меньше. Это связано с особенностями аппаратного обеспечения: эксперименты проводились на четырех физических процессорах, каждый из которых с помощью технологии Intel© Hyperthreading© виртуально представляется для операционной системы как два разных процессора. Использование данной технологии несколько ускоряет переключение задач, но не позволяет кардинально ускорить процесс вычислений. Как видно из таблицы, на оборудовании с более чем четырьмя физическими процессорами увеличение скорости возможно практически линейно от числа процессоров в силу естественного параллелизма алгоритмов построения графов, вычисления вероятностей сборки и статистического вывода параметров вероятностной модели. Высокая скорость работы разработанных алгоритмов позволяет быстро и эффективно работать с данными репертуаров клеточных рецепторов на пользовательских компьютерах без необходимости использовать вычислительные кластеры, что открывает доступ к использованию разработанного инструмента обычным пользователям.

Задача предсказания вероятностей сборки аминокислотных последовательностей клеточных рецепторов возникает при оценке разнообразия репертуаров антигенраспознающих рецепторов и построения моделей тимической и клональной селекции. Приведенные алгоритмы позволяют считать вероятности сборки аминокислотных последовательностей клеточных рецепторов без подсчета всех возможных кодирующих нуклеотидных последовательностей. Данные алгоритмы не имеют аналогов по вычислительной эффективности при сходной точности проводимых вычислений. Существующие решения имеют принципиальные ограничения в своей реализации, заставляющие считать вероятно-

сти аминокислотных последовательностей только путем перебора всех возможных кодирующих ее нуклеотидных последовательностей.

Предложенные в данной работе алгоритмы были применены для анализа клеточных рецепторов после пересадки костного мозга [5]. Разработанный метод позволил выявить недостатки существующих методов для диагностики минимальной остаточной болезни на основе присутствия определенных последовательностей Т-клеточных рецепторов в крови пациента. Минимальная остаточная болезнь — это небольшое число опухолевых клеток, оставшихся в организме после ремиссии, в том числе — больных Т-клеток при лейкозе. Метод лечения лейкоза включает в себя полное уничтожение больных клеток. Однако остается вероятность того, что часть клеток останется после терапии и вызовет минимальную остаточную болезнь, которая может снова привести к лейкозу. В то же время, из-за стохастической природы создания Т-клеточных рецепторов, часть рецепторов Т-клеток, которые создаются в организме человека уже после терапии, могут иметь такую же последовательность, как и выявленные ранее больные клетки. Если такие рецепторы были найдены, то необходимо повторение терапии, что само по себе является огромным стрессом для организма и может привести к летальному исходу, вероятность которого увеличивается с числом пройденных терапий. В нашей предыдущей работе мы показали, что использовавшиеся ранее методы дают слишком много ложноположительных срабатываний из-за высокой вероятности сборки ряда рецепторов больных Т-клеток, что позволяет не проводить заново терапию в ряде случаев, когда вероятность сборки Т-клеточных рецепторов очень высока, таким образом, снижая риски для пациентов.

Приведенные алгоритмы также возможно использовать для анализа данных БКР при построении филогенетических деревьев по нуклеотидным последовательностям БКР, в которых близких по последовательности БКР выстраивают в дерево наследования из-за мутаций, при которых одни нуклеотиды заменяются на другие. Основная сложность в построении филогенетических деревьев для БКР в том, что не всегда ясна последовательность мутаций, и известная вероятность сборки позволяет предположить, какая из последовательностей была предком, а какая является ее мутировавшим наследником.

В случае уточнения V(D)J рекомбинации или открытия других видов рекомбинации можно будет легко изменить алгоритм для подсчета вероятностей сборки и вывода параметров для новых моделей.

6. Заключение

В данной работе предложен новый метод представления всех возможных способов сборки нуклеотидных и аминокислотных последовательностей иммунных рецепторов. Разработанные алгоритмы для подсчета вероятностей сборки рецепторов и статистического вывода параметров вероятностной модели сборки рецепторов работают как минимум на порядок быстрее существующих аналогов. Подобное ускорение позволяет использовать разработанное программное обеспечение обычным пользователям, которые не имеют доступа к вычислительным кластерам. Естественный параллелизм алгоритмов позволяет обрабатывать большие массивы данных и за счет этого увеличивать точность вероятностных моделей. Предложенный быстрый способ подсчета вероятностей сборки аминокислотных последовательностей не имеет аналогов, так как предыдущие решения требовали слишком больших ресурсов (до трех лет вычислений на доступном кластере взамен нескольких часов на одном вычислительном ядре для предложенного метода). Разработанные методы послужат базой для дальнейшего развития методики анализа иммунитета на основе вероятностных моделей создания иммунных рецепторов.

Список литературы

[1] Хаитов Р.М., Игнатьева Г.А., Сидорович И.Г. Иммунология: учебник // М.: "Медицина" — 2000. — 432 с. — ISBN 5-225-04543-X.

[2] Murugan A., Mora T., Walczak A.M. et al. Statistical inference of the generation probability of T-cell receptors from sequence repertoires // Proceedings of National Academics of Sciences. — 2012. — Vol. 109(40). — P. 16161-66. — URL: `http://www.pnas.org/content/109/40/16161.full.pdf`

[3] Pogorelyy M.V., Elhanati Y., Marcou Q. et al. Persisting fetal clonotypes influence the structure and overlap of adult human T cell receptor repertoires // http://biorxiv.org/content/early/2016/02/09/039297 — 2016. — URL: `http://biorxiv.org/content/early/2016/02/09/039297`

[4] Britanova OV, Putintseva EV, Shugay M et al. Age-related decrease in TCR repertoire diversity measured with deep and normalized sequence profiling // Journal of Immunology. — 2015. — Vol. 192(6). — P. 2689-98. — URL: `https://www.ncbi.nlm.nih.gov/pubmed/24510963`

[5] Nazarov V.I., Minervina A.A., Komkov A.Y. et al. Reliability of immune receptor rearrangements as genetic markers for minimal residual disease monitoring //

Bone Marrow Transplantation. — 2015. — Vol. 51(10). — P. 1408-10. — URL: https://www.ncbi.nlm.nih.gov/pubmed/27214078

[6] Hoehn K.B., Gall A., Bashford-Rogers R. et al. Dynamics of immunoglobulin sequence diversity in HIV-1 infected individuals // Philosophical transactions of the Royal Society of London Biological sciences. — 2015. — Vol. 370. — P. 1676. — URL: https://www.ncbi.nlm.nih.gov/pubmed/26194755

[7] Gaeta B.A., Malming H.R., Jackson K.J. et al. iHMMune-align: hidden Markov model-based alignment and identification of germline genes in rearranged immunoglobulin gene sequences // Bioinformatics. — 2007. — Vol. 23(13). — P. 1580-7. — URL: https://www.ncbi.nlm.nih.gov/pubmed/17463026

[8] Munshaw S., Kepler T.B. SoDA2: a Hidden Markov Model approach for identification of immunoglobulin rearrangements // Bioinformatics. — 2010. — Vol. 26(7). — P. 867-872. — URL: www.ncbi.nlm.nih.gov/pubmed/20147303

[9] Li S., Lefranc M.P., Miles J.J. et al. IMGT/HighV QUEST paradigm for T cell receptor IMGT clonotype diversity and next generation repertoire immunoprofiling // Nature Communications. — 2013. — Vol. 4. — P. 2333. — URL: https://www.ncbi.nlm.nih.gov/pubmed/12520010

[10] Ye J., Ma N., Madden T.L. et al. IgBLAST: an immunoglobulin variable domain sequence analysis tool // Nucleic Acids Research. — 2013. — Vol. 41. — P. 34-40. — URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3692102/

[11] Altschul S.F., Gish W., Miller W. et al. Basic local alignment search tool // Journal of Molecular Biology. — 1990. — Vol. 215(3). — P. 403-10. — URL: https://www.ncbi.nlm.nih.gov/pubmed/2231712

[12] Bonissone S.R., Pevzner P.A. Immunoglobulin Classification Using the Colored Antibody Graph // Journal of Computational Biology. — 2016. — Vol. 23(6). — P. 483-494. — URL: www.ncbi.nlm.nih.gov/pubmed/27149636

[13] Yang X., Liu D., Lv N. et al. TCRklass: a new K-string-based algorithm for human and mouse TCR repertoire characterization // Journal of Immunology. — 2015. — Vol. 194(1). — P. 446-454. — URL: https://www.ncbi.nlm.nih.gov/pubmed/25404364

[14] Duez M., Giraud M., Herbert R. et al. Vidjil: A Web Platform for Analysis of High-Throughput Repertoire Sequencing // PLoS ONE. — 2016. — Vol. 11(11). — P. e0166126. — URL: https://www.ncbi.nlm.nih.gov/pubmed/27835690

[15] Bolotin D.A., Shugay M., Mamedov I.Z. et al. MiTCR: software for T-cell receptor sequencing data analysis // Nature Methods. — 2013. — Vol. 10. — P. 813-4. — URL: https://www.ncbi.nlm.nih.gov/pubmed/23892897

[16] Bolotin D.A., Poslavsky S., Mitrophanov I. et al. MiXCR: software for comprehensive adaptive immunity profiling // Nature Methods. — 2015. — Vol. 12. — P. 380-1. — URL: https://www.ncbi.nlm.nih.gov/pubmed/25924071

[17] Zvyagin I.V., Pogorelyy M.V., Ivanova M.E. et al. Distinctive properties of identical twins' TCR repertoires revealed by high-throughput sequencing // Proceedings of National Academics of Sciences. — 2014. — Vol. 111(16). — P. 5980-5. — URL: https://www.ncbi.nlm.nih.gov/pubmed/24711416

[18] Rempala G.A., Seweryn M. Methods for diversity and overlap analysis in T-cell receptor populations // Journal of Mathematical Biology. — 2013. — Vol. 67. — P. 6-7. — URL: https://www.ncbi.nlm.nih.gov/pubmed/23007599

[19] Horn H.S. Measurement of overlap in comparative ecological studies // The American Naturalist. — 1966. — Vol. 100. — P. 419-24. — URL: https://www.jstor.org/stable/2459242

[20] Greiff V., Miho E., Menzel U., Reddy S.T. Bioinformatic and Statistical Analysis of Adaptive Immune Repertoires // Trends in Immunology. — 2015. — Vol. 36. — P. 11. — URL: http://www.cell.com/trends/immunology/abstract/S1471-4906(15)00223-9

[21] Greiff V., Bhat P., Cook S.C., Menzel U., Kang W., Reddy S.T. A bioinformatic framework for immune repertoire diversity profiling enables detection of immunological status // Genome Medicine. — 2015. — Vol. 7. — P. 49. — URL: https://genomemedicine.biomedcentral.com/articles/10.1186/s13073-015-0169-8

[22] Tipton C.M., Fucile C.F., Darce J. et al. Diversity, cellular origin and autoreactivity of antibody-secreting cell population expansions in acute systemic lupus erythematosus // Nature Immunology. — 2015. — Vol. 16(7). — P. 755-65. — URL: https://www.ncbi.nlm.nih.gov/pubmed/26006014

Содержание