(19) Евразийское патентное ведомство

(21) 201990931 (13) A1

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ЕВРАЗИЙСКОЙ ЗАЯВКЕ

(43) Дата публикации заявки
2019.11.29

(22) Дата подачи заявки
2016.10.11

(51) Int. Cl. ***G06F 19/28*** (2011.01)

(54) СПОСОБ И СИСТЕМА ДЛЯ ПЕРЕДАЧИ ДАННЫХ БИОИНФОРМАТИКИ

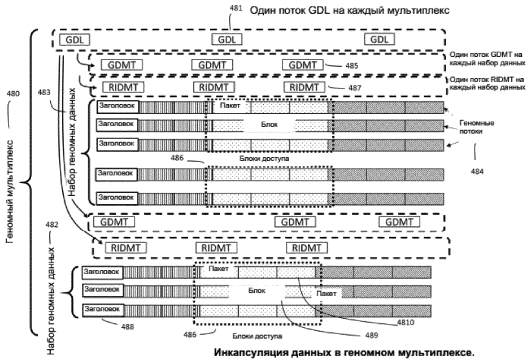
(86) **PCT/EP2016/074311**
(87) **WO 2018/068830 2018.04.19**

(71) Заявитель:
ДЖЕНОМСИС СА (CH)

(72) Изобретатель:
Зойя Джорджио, Рензи Даниэле (CH)

(74) Представитель:
Нилова М.И. (RU)

(57) Способ и система для передачи геномных данных. Передача геномных данных осуществляется путем использования мультиплексирования структурированного сжатого набора геномных данных в потоке геномных данных, разбитых на блоки доступа с произвольным доступом.



201990931 A1

201990931 A1

СПОСОБ И СИСТЕМА ДЛЯ ПЕРЕДАЧИ ДАННЫХ БИОИНФОРМАТИКИ

Область техники

Настоящая заявка относится к новым способам эффективного хранения, доступа, передачи и мультиплексирования биоинформационных данных и, в частности, данных геномного секвенирования.

Уровень техники

Надлежащее представление данных секвенирования генома имеет основополагающее значение для эффективных обработки, хранения и передачи геномных данных, чтобы сделать возможным и облегчить применения анализа, такие как определение вариантов генома и выполнение всего анализа, с различными целями, путем обработки данных и метаданных секвенирования. Сегодня информацию о секвенировании генома генерируют высокопроизводительные секвенаторы (HTS) в виде последовательностей нуклеотидов (или оснований), представленных цепочками букв из определенного словаря.

Эти секвенаторы не считывают целые геномы или гены, но они выдают короткие случайные фрагменты нуклеотидных последовательностей, называемые "риды" последовательностей. С каждым нуклеотидом в риде последовательности связан показатель качества. Это число представляет уровень достоверности, присвоенный секвенатором риду конкретного нуклеотида в конкретном месте в нуклеотидной последовательности.

Эти необработанные данные секвенирования, генерируемые секвенаторами NGS, обычно хранятся в файлах FASTQ (см. также Фигуру 1).

Наименьший словарь для представления последовательностей нуклеотидов, полученных в процессе секвенирования, состоит из пяти символов: {A, C, G, T, N}, представляющих 4 типа нуклеотидов, присутствующих в ДНК, а именно аденин, цитозин, гуанин и тимин плюс символ N для обозначения того, что секвенатор не смог определить какое-либо основание с достаточным уровнем достоверности, поэтому тип основания в таком положении остается неопределенным в процессе прочитывания. В

РНК тимин заменен на урацил (U). Последовательности нуклеотидов, полученные с помощью секвенаторов, называются "риды". В случае парных ридов термин "матрица" (шаблон) используется для обозначения исходной последовательности, из которой была извлечена эта пара ридов. Риды последовательности могут состоять из ряда нуклеотидов в диапазоне от нескольких десятков до нескольких тысяч. Некоторые технологии выдают риды последовательностей в парах, где каждый рид может происходить из одной из двух цепей ДНК.

В области секвенирования генома термин "перекрывание" используется для выражения уровня избыточности данных последовательности относительно референсного генома. Например, чтобы достичь 30-кратного перекрывания человеческого генома (длиной 3,2 млрд. оснований), секвенатор должен произвести в общей сложности около 30 x 3,2 млрд. оснований, чтобы в среднем каждое положение в референсе было "перекрыто" 30 раз.

Существующие решения

Наиболее часто используемые представления информации о геноме на основе данных секвенирования основаны на форматах FASTQ и SAM, которые обычно предоставляются в архивированном виде для уменьшения исходного размера. Традиционные форматы файлов, соответственно FASTQ и SAM для невыровненных и выровненных данных секвенирования, состоят из простых текстовых символов и, поэтому, сжимаются с использованием стандартных подходов, таких как схемы LZ (от имен Lempel и Ziv) (хорошо известные zip, gzip и т. д.). Когда используются стандартные средства сжатия файлов (компрессоры), такие как gzip, результатом сжатия обычно является один блоб-объект бинарных данных. Информацию в такой монолитной форме очень трудно архивировать, передавать и обрабатывать, особенно в случае высокопроизводительного секвенирования, когда объемы данных чрезвычайно велики.

После секвенирования каждый этап конвейера обработки геномной информации выдает данные, представленные совершенно новой структурой данных (формат файла), несмотря на тот факт, что в действительности только небольшая часть сгенерированных данных является новой по сравнению с предыдущим этапом.

На Фигуре 1 показаны основные этапы типичного конвейера обработки геномной информации с указанием представления формата ассоциированного файла.

Обычно используемые решения имеют несколько недостатков: архивирование данных неэффективно из-за того, что на каждом этапе конвейеров обработки геномной информации используется другой формат файла, что подразумевает многократную репликацию данных с последующим быстрым увеличением необходимого объёма памяти. Это неэффективно и не нужно, а также становится неприемлемым для увеличивающегося объема данных, генерируемых секвенаторами HTS. На самом деле это имеет последствия с точки зрения доступного места для хранения и генерируемых затрат, а также препятствует распространению пользы от геномного анализа в здравоохранении на большую часть населения. Влияние затрат на ИТ, вызванное экспоненциальным ростом данных о последовательностях, которые необходимо хранить и анализировать, в настоящее время является одной из основных проблем, стоящих перед научным сообществом и отраслью здравоохранения (см. Scott D. Kahn “On the future of genomic data” - Science 331, 728 (2011) и Pavlichin, DS, Weissman, T., G. Yona. 2013. “The human genome contracts again”, Bioinformatics 29 (17): 2199-2202). В то же время есть несколько инициатив, направленных на масштабирование секвенирования генома от нескольких избранных индивидов до больших групп населения (см. Josh P. Roberts “Million Veterans Sequenced” - Nature Biotechnology 31, 470 (2013))

Передача геномных данных является медленной и неэффективной, поскольку используемые в настоящее время форматы данных организованы в монолитные файлы размером до нескольких сотен гигабайт, которые должны быть полностью перенесены на принимающую сторону для обработки. Это подразумевает, что анализ небольшого сегмента данных требует передачи всего файла со значительными затратами с точки зрения потребляемой ширины пропускания канала и времени ожидания. Зачастую передача в режиме онлайн является невозможной для больших объемов передаваемых данных, и передачу данных осуществляют путем физического перемещения носителей, таких как жесткие диски или серверы хранения, из одного места в другое.

Эти ограничения, возникающие при использовании современных подходов, преодолеваются настоящим изобретением.

Обработка данных является медленной и неэффективной из-за того факта, что информация не структурирована таким образом, чтобы порции различных классов данных и метаданных, требуемые для часто используемых приложений анализа, могли быть извлечены без необходимости доступа к данным во всей их полноте. Этот факт подразумевает, что обычные конвейеры анализа могут требовать работы в течение нескольких дней или недель, тратя драгоценные и дорогостоящие ресурсы обработки из-за необходимости на каждом этапе доступа анализировать и фильтровать большие объемы данных, даже если порции данных, относящиеся к конкретной цели анализа, гораздо меньше.

Эти ограничения не позволяют медицинским работникам своевременно получать отчеты по геномному анализу и оперативно реагировать на вспышки заболеваний. Настоящее изобретение обеспечивает решение для удовлетворения этой потребности.

Существует еще одно техническое ограничение, которое преодолевает настоящее изобретение.

Фактически изобретение направлено на обеспечение надлежащих данных геномного секвенирования и представления метаданных путем организации и разделения данных таким образом, чтобы сжатие данных и метаданных было максимизировано, и были эффективно задействованы некоторые функции, такие как выборочный доступ и поддержка инкрементных обновлений.

Ключевым аспектом изобретения является точное определение классов данных и метаданных, которые должны быть представлены соответствующей моделью источника, при их кодировании (т.е. сжатии) отдельно посредством структурирования в определенных слоях. Наиболее важные преимущества этого изобретения по сравнению с существующими современными методами заключаются в следующем:

- повышение производительности сжатия из-за уменьшения энтропии источника информации, созданной путем обеспечения эффективной модели для каждого класса данных или метаданных;
- возможность осуществления выборочного доступа к частям сжатых данных и метаданных для цели любой дальнейшей обработки;

- возможность инкрементного (без необходимости перекодирования) обновления и добавления закодированных данных и метаданных с новыми данными и/или метаданными секвенирования и/или новыми результатами анализа;
- возможность эффективно обрабатывать данные, как только они выданы секвенатором или инструментами выравнивания, без необходимости ждать окончания процесса секвенирования или выравнивания.

В настоящей заявке раскрыты способ и система для решения проблемы эффективного оперирования, хранения и передачи очень больших объемов данных геномного секвенирования с использованием подхода структурированных блоков доступа в сочетании с методами мультиплексирования.

Настоящая заявка преодолевает все ограничения подходов предшествующего уровня техники, связанные с функциональными возможностями доступности геномных данных, эффективной обработкой подмножеств данных, функциональными возможностями передачи и потоковой передачи в сочетании с эффективным сжатием.

На сегодняшний день наиболее часто используемым форматом представления геномных данных является текстовый формат картирования выравнивания последовательностей (SAM) и его бинарное соответствие BAM. Файлы SAM представляют собой текстовые файлы ASCII, которые могут быть прочитаны человеком, тогда как BAM использует блочный вариант gzip. Файлы BAM могут быть проиндексированы, чтобы обеспечить ограниченный модальный режим произвольного доступа. Это поддерживается созданием отдельного индексного файла.

Формат BAM характеризуется низкой производительностью сжатия по следующим причинам:

1. Он сосредоточен на сжатии неэффективного и избыточного формата файлов SAM, а не на извлечении фактической геномной информации, передаваемой файлами SAM, и использовании соответствующих моделей для ее сжатия.

2. Он использует универсальный алгоритм сжатия текста, такой как gzip, вместо использования специфического характера каждого источника данных (самой геномной информации).
3. В нем отсутствует какая-либо концепция, связанная с классификацией данных, которая позволила бы обеспечить выборочный доступ к определенным классам геномных данных.

Более сложный подход к сжатию геномных данных, который используется не так часто, но более эффективен, чем BAM, называется CRAM (спецификация CRAM: https://samtools.github.io/hts-specs/CRAMv3.pdf). CRAM обеспечивает более эффективное сжатие для применения дифференциального кодирования по отношению к существующему референсу (он частично использует избыточность источника данных), но ему все же не хватает таких функций, как инкрементные обновления, поддержка потоковой передачи данных и выборочный доступ к определенным классам сжатых данных.

CRAM опирается на концепцию записи CRAM. Каждая запись CRAM кодирует один картированный или некартированный рид, кодируя все элементы, необходимые для его восстановления.

Основными отличиями настоящего изобретения от подхода CRAM являются следующие:

1. Для CRAM индексация данных выходит за рамки спецификации (см. раздел 12 спецификации CRAM v 3.0) и реализована в виде отдельного файла. В настоящем изобретении индексирование данных интегрировано с процессом кодирования, а индексы внедрены в кодированный битовый поток.
2. В CRAM все основные блоки данных могут содержать картированные риды любого типа (идеально совпадающие риды, риды только с заменами, риды с инделами). В настоящем изобретении отсутствует понятие классификации и группировки ридов в классах в соответствии с результатом картирования относительно референсной последовательности.
3. В описанном изобретении нет понятия записи, инкапсулирующей каждый рид потому что данные, необходимые для восстановления каждого рида, разбросаны

по нескольким контейнерам данных, называемым "слоями". Это обеспечивает более эффективный доступ к набору ридов с определенными биологическими характеристиками (например, риды с заменами, но без инделов или абсолютно картированных ридов) без необходимости декодирования каждого (блока) рида (-й) для проверки его особенностей.

4. В записи CRAM каждый тип данных обозначается специальным флагом. В отличие от CRAM в настоящем изобретении нет понятия флага, обозначающего данные, потому что тип данных по сути определяется "слоем", к которому принадлежат данные. Это подразумевает значительно уменьшенное количество используемых символов и, как следствие, уменьшение энтропии источника информации, что приводит к более эффективному сжатию. Это связано с тем, что использование разных "слоев" позволяет кодеру повторно использовать один и тот же символ в каждом слое с разными значениями. В CRAM каждый флаг должен всегда иметь одно и то же значение, поскольку отсутствует понятие контекстов, и каждая запись CRAM может содержать данные любого типа.
5. В CRAM замены, инсерции и делеции выражаются в соответствии с разными синтаксисами, тогда как настоящее изобретение использует один алфавит и кодирование для замен, инсерций и делеций. Это упрощает процесс кодирования и декодирования и создает модель источника с более низкой энтропией, кодирование которой дает потоки битов, характеризующиеся более высокой эффективностью сжатия.

Алгоритмы геномного сжатия, используемые в современном уровне техники, можно классифицировать по таким категориям:

- С преобразованием:
    - на основе LZ
    - переупорядочивание ридов
- На основе сборки
- Статистическое моделирование

Первые две категории имеют тот недостаток, что не используют специфические характеристики источника данных (риды последовательности геномов) и обрабатывают геномные данные как строку текста, подлежащего сжатию, без учета

специфических свойств такого рода информации (например, избыточность среди ридов, связь с существующим образцом). Два самых продвинутых набора инструментов для сжатия геномных данных, а именно CRAM и Goby (“Compression of structured high-throughput sequencing data”, F. Campagne, K. C. Dorff, N. Chambwe, J. T. Robinson, J. P. Mesirov, T. D. Wu), плохо используют арифметическое кодирование, поскольку они неявно моделируют данные как независимые и идентично распределенные согласно геометрическому распределению. Goby немного сложнее, поскольку он преобразует все поля в список целых чисел, и каждый список кодируется независимо с использованием арифметического кодирования без использования какого-либо контекста. В наиболее эффективном режиме работы Goby может выполнять некоторое моделирование между списками по целочисленным спискам, чтобы улучшить сжатие. Эти решения предшествующего уровня техники дают плохие коэффициенты сжатия и структуры данных, которые в лучшем случае с трудом подвергаются выборочному доступу и обработке после сжатия. Последующие этапы анализа могут оказаться неэффективными и очень медленными из-за необходимости обработки больших и жестких структур данных даже для выполнения простых операций или для доступа к выбранным областям набора геномных данных.

Упрощенное представление связи между форматами файлов, используемыми в конвейерах обработки генома, изображено на Фигуре 1. В этой диаграмме включение файла не подразумевает существование вложенной файловой структуры, а представляет только тип и объем информации, которая может быть закодирована для каждого формата (т. е. SAM содержит всю информацию в FASTQ, но организована в другой файловой структуре). CRAM содержит ту же геномную информацию, что и SAM/BAM, но обладает большей гибкостью в отношении типа сжатия, который можно использовать, поэтому он представлен как расширенный набор SAM/BAM.

Использование нескольких форматов файлов для хранения геномной информации является крайне неэффективным и дорогостоящим. Наличие разных форматов файлов на разных этапах жизненного цикла геномной информации подразумевает линейный рост используемого объёма памяти, даже если добавочная информация минимальна. Другие недостатки решений известного уровня техники перечислены ниже.

1. Доступ, анализ или добавление аннотаций (метаданных) к необработанным данным, хранящимся в сжатых файлах FastQ или любой их комбинации, требует распаковки и повторного сжатия всего файла с интенсивным использованием вычислительных ресурсов и времени.
2. Для извлечения определенных подмножеств информации, таких как положения картирования ридов, положение и тип варианта рида, положение и типы инделов, или любых других метаданных и аннотаций, содержащиеся в выровненных данных, хранящихся в файлах BAM, требуется доступ ко всему объему данных, ассоциированному с каждым ридом. Выборочный доступ к одному классу метаданных с решениями предшествующего уровня техники невозможен.
3. Форматы файлов предшествующего уровня техники требуют, чтобы весь файл был получен конечным пользователем до начала обработки. Например, выравнивание ридов может начаться до того, как процесс секвенирования будет завершен, исходя из соответствующего представления данных. Секвенирование, выравнивание и анализ могут идти и выполняться параллельно.
4. Решения предшествующего уровня техники не поддерживают структурирование и не могут различать геномные данные, полученные различными процессами секвенирования, в соответствии с их специфической семантикой генерации (например, секвенирование, полученное в разное время жизни одного и того же индивида). Такое же ограничение имеет место для секвенирования, полученного с использованием различных типов биологических образцов одного и того же индивида.
5. Шифрование всех или выбранных частей данных не поддерживается решениями предшествующего уровня техники. Например, шифрование следующих данных:
    a. выбранные области ДНК
    b. только последовательности, содержащие варианты
    c. только химерные последовательности
    d. только некартированные последовательности

e. специфические метаданные (например, происхождение секвенированного образца, идентификация секвенированного индивида, тип образца)

6. Транскодирование из данных секвенирования, выровненных с заданным референсом (т.е. файлом SAM/BAM), в новый референс требует обработки всего объема данных, даже если новый референс отличается от предыдущего референса только на одно положение нуклеотида.

Следовательно, существует необходимость в соответствующем "слое хранения геномной информации" (формат геномных файлов Genomic File Format), который бы обеспечивал эффективное сжатие, поддерживал выборочный доступ в сжатой области, поддерживал инкрементное добавление разнородных метаданных в сжатой области на всех уровнях различных этапов обработки геномных данных.

Настоящее изобретение обеспечивает решение для устранения ограничений существующего уровня техники с помощью способа, устройств и компьютерных программ, заявленных в прилагаемой формуле изобретения.

Список фигур

Фигура 1. Основные этапы типичного геномного конвейера и соответствующие форматы файлов.

Фигура 2. Взаимосвязь между наиболее часто используемыми форматами геномных файлов

Фигура 3. Схема сборки ридов геномной последовательности в весь или частичный геном посредством сборки de-novo или выравнивания с референсом.

Фигура 4. Схема расчета положений картирования ридов на референсной последовательности.

Фигура 5. Схема расчета расстояний спаривания ридов.

Фигура 6. Схема расчета ошибок спаривания.

Фигура 7. Схема кодирования расстояний спаривания, когда пара рида картируется на другой хромосоме.

Фигура 8. Риды последовательности могут быть получены из первой или второй цепи ДНК генома.

Фигура 9. Рид, картированный на цепи 2 имеет соответствующий рид-обратный комплемент на цепи 1.

Фигура 10. Четыре возможных комбинации ридов, составляющих пару ридов, и соответствующее кодирование в слое rcomp.

Фигура 11. Схема кодирования N несовпадений в слое nmis.

Фигура 12. Пример замен в картированной паре ридов.

Фигура 13. Положения замены могут быть рассчитаны как абсолютные или дифференциальные значения.

Фигура 14. Схема расчета символов, кодирующих замены, без кодов IUPAC.

Фигура 15. Схема кодирования типов замены в слое snpt.

Фигура 16. Схема расчета символов, кодирующих замены, с кодами IUPAC.

Фигура 17. Альтернативная исходная модель для замены, где кодируются только положения, но используется один слой на каждый тип замены.

Фигура 18. Кодирование замен, вставок и делеций в паре ридов класса I, когда коды IUPAC не используются.

Фигура 19. Кодирование замен, вставок и делеций в паре ридов класса I, когда используются коды IUPAC.

Фигура 20. Структура заголовка данных геномной информации.

Фигура 21. Главная индексная таблица содержит положения в референсных последовательностях первого рида в каждом блоке доступа.

Фигура 22. Пример частичной MIT (главной индексной таблицы), показывающей положения картирования первого рида в каждом AU pos класса P.

Фигура 23. Локальная индексная таблица в заголовке слоя является вектором указателей на AU (блоке доступа) в полезной нагрузке.

Фигура 24. Пример локальной индексной таблицы.

Фигура 25. функциональная связь между главной индексной таблицей и локальными индексными таблицам

Фигура 26. Блоки доступа составлены из блоков данных, принадлежащих нескольким (множеству) слоям. Слои составлены из блоков, подразделенных на пакеты.

Фигура 27. Геномный блок доступа типа 1 (содержащий информацию о положениях, спаривании, обратном комплементе и длине рида) упаковывается и инкапсулируется в мультиплекс геномных данных.

Фигура 28. Блоки доступа состоят из заголовка и мультиплексированных блоков, принадлежащих к одному или более слоям однородных данных. Каждый блок может состоять из одного или более пакетов, содержащих фактические дескрипторы геномной информации.

Фигура 29. Структура блоков доступа типа 0, которые не должны ссылаться на какую-либо информацию, поступающую из других блоков доступа для доступа или декодирования и доступа.

Фигура 30. Структура блоков доступа типа 1.

Фигура 31. Структура блоков доступа типа 2, которые содержат данные, ссылающиеся на блок доступа типа 1. Это положения N в закодированных ридах.

Фигура 32. Структура блоков доступа типа 3, которые содержат данные, ссылающиеся на блок доступа типа 1. Это положения и типы несовпадений в закодированных ридах.

Фигура 33. Структура блоков доступа типа 4, которые содержат данные, ссылающиеся на блок доступа типа 1. Это положения и типы несовпадения в закодированных ридах.

Фигура 34. Первые пять типов блоков доступа.

Фигура 35. Блоки доступа типа 1 ссылаются на блоки доступа типа 0, которые необходимо декодировать.

Фигура 36. Блоки доступа типа 2 ссылаются на блоки доступа типа 0 и 1, которые необходимо декодировать.

Фигура 37. Блоки доступа типа 3 ссылаются на блоки доступа типа 0 и 1, которые необходимо декодировать.

Фигура 38. Блоки доступа типа 4 ссылаются на блоки доступа типа 0 и 1, которые необходимо декодировать.

Фигура 39. Блоки доступа, необходимые для декодирования ридов последовательности с несовпадениями, картированными на втором сегменте референсной последовательности (AU 0-2).

Фигура 40. Необработанные данные геномной последовательности, которые становятся доступными, могут быть инкрементно добавлены к ранее закодированным геномным данным.

Фигура 41. Структура данных на основе блоков доступа позволяет начать анализ геномных данных до завершения процесса секвенирования.

Фигура 42. Новый анализ, выполненный на уже существующих данных, может означать, что риды перемещены из AU типа 4 в AU типа 3.

Фигура 43. Вновь сгенерированные данные анализа инкапсулированы в новый AU типа 6, и в MIT создается соответствующий индекс.

Фигура 44. Перекодирование данных в связи с публикацией новой референсной последовательности (генома).

Фигура 45. Как риды, картированные в новой геномной области с лучшим качеством (например, без инделов), перемещаются из AU типа 4 в AU типа 3.

Фигура 46. В случае обнаружения нового местоположения картирования (например, с меньшим количеством несовпадений) связанные риды могут быть перемещены из одного AU в другой AU того же типа.

Фигура 47. Избирательное шифрование может применяться к блокам доступа типа 4 только в том случае, если они содержат чувствительную информацию, подлежащую защите.

Фигура 48. Инкапсуляция данных в геномном мультиплексе, где один или более геномных наборов данных 482-483 содержат геномные потоки 484 и списки потоков геномных данных 481, таблицы картирования наборов геномных данных 485 и таблицы картирования референсных идентификаторов 487. Каждый геномный поток состоит из заголовка 488 и блоков доступа 486. Блоки доступа инкапсулируют блоки 489, которые состоят из пакетов 4810.

Фигура 49. Схема обработки исходных данных геномной последовательности или выровненных геномных данных для инкапсуляции в геномный мультиплекс. Для подготовки данных для кодирования могут быть необходимы этапы выравнивания, повторного выравнивания, сборки. Сгенерированные слои инкапсулируются в блоки доступа и мультиплексируются геномным мультиплексором.

Фигура 50. Геномный демультиплексор (501) извлекает слои блоков доступа из геномного мультиплекса, один декодер на каждый тип AU (502) извлекает дескрипторы генома, которые затем декодируются (503), в различные геномные форматы, такие как, например, FASTQ и SAM/BAM.

Подробное описание

Настоящее изобретение описывает формат файла с мультиплексированием и соответствующие блоки доступа, которые должны использоваться для хранения, транспортировки, доступа и обработки геномной или протеомной информации в форме последовательностей символов, представляющих молекулы.

Эти молекулы включают, например, нуклеотиды, аминокислоты и белки. Одним из наиболее важных элементов информации, представленных в виде последовательности символов, являются данные, генерируемые высокопроизводительными устройствами для секвенирования генома.

Геном любого живого организма обычно представляется в виде последовательности символов, выражающих цепочку нуклеиновых кислот (оснований), характеризующих этот организм. Современное состояние технологии секвенирования генома способно создавать только фрагментированное представление генома в виде нескольких (до миллиардов) строк нуклеиновых кислот, связанных с метаданными (идентификаторы, уровень точности и т.д.). Такие строки обычно называют "риды последовательности" или просто "риды".

Типичные этапы жизненного цикла геномной информации включают извлечение, картирование и выравнивание, обнаружение вариантов, аннотацию вариантов и функционально-структурный анализ ридов последовательностей (см. Фиг 1).

Извлечение ридов последовательности - это процесс (выполняемый человеком-оператором или машиной-секвенатором) представления фрагментов генетической информации в виде последовательностей символов, представляющих молекулы, составляющие биологический образец. В случае нуклеиновых кислот такие молекулы называются "нуклеотидами". Последовательности символов, полученные в результате извлечения, обычно называют "ридами". Эта информация обычно кодируется в предшествующем уровне техники в виде файлов FASTA, включающих текстовый заголовок и последовательность символов, представляющих секвенированные молекулы.

Когда биологический образец секвенируют для извлечения из живого организма ДНК, алфавит состоит из символов (A,C,G,T,N).

Когда биологический образец секвенируют для извлечения из живого организма РНК, алфавит состоит из символов (A,C,G,U,N).

В случае расширенного набора символов IUPAC, секвенатор также генерирует так называемые "коды неоднозначности", и алфавит, используемый для символов, составляющих риды, состоит из символов (A, C, G, T, U, W, S, M, K, R, Y, B, D, H, V, N или -).

Когда коды неоднозначности IUPAC не используются, с каждой прочитанной последовательностью может быть ассоциирована последовательность показателей качества. В таком случае решения предшествующего уровня техники кодируют полученную информацию в виде файла FASTQ. Устройства секвенирования могут вносить ошибки в риде последовательности, такие как:

1. идентификация неправильного символа (то есть, представляющего другую нуклеиновую кислоту) для обозначения нуклеиновой кислоты, фактически присутствующей в секвенированном образце; обычно это называется "ошибка замены" (несовпадение);
2. инсерция дополнительных символов в одном риде последовательности, которые не относятся к какой-либо фактически присутствующей нуклеиновой кислоте; обычно это называется "ошибка инсерции";
3. делеция из одного рида последовательности символов, представляющих нуклеиновые кислоты, которые фактически присутствуют в секвенированном образце; обычно это называется "ошибка делеции";
4. рекомбинация одного или более фрагментов в один фрагмент, который не отражает реальность исходной последовательности.

Термин "перекрывание" используется в литературе для количественной оценки степени, в которой референсный геном или его часть могут быть перекрыты доступной последовательностью рида. Перекрывание называется:

- *частичным* (менее чем 1-кратным), когда некоторые части референсного генома не картированы ни одним доступным ридом последовательности
- *однократным* (1x) когда все нуклеотиды референсного генома картированы одним и только одним символом, присутствующим в риде последовательности
- *многократным* (2x, 3x, Nx) когда каждый нуклеотид референсного генома картирован несколько раз.

Термин "выравнивание последовательностей" относится к процессу упорядочения ридов последовательности путем нахождения областей сходства, которые могут быть следствием функциональных, структурных или эволюционных связей между последовательностями. Когда выравнивание выполняется со ссылкой на ранее существующую нуклеотидную последовательность, называемую "референсный геном", этот процесс называется "картирование". Выравнивание последовательностей также может быть выполнено без ранее существовавшей последовательности (то есть референсного генома), в таких случаях процесс известен в предшествующем уровне техники как выравнивание "de novo". Известные решения хранят эту информацию в файлах SAM, BAM или CRAM. Концепция выравнивания последовательностей для восстановления частичного или полного генома изображена на Фигуре 3.

Детектирование вариантов (также называемое определением вариантов) - это процесс преобразования выровненных результатов секвенаторов (риды, созданные устройствами NGS и выровненные), в сводку уникальных характеристик секвенируемого организма, которые не могут быть найдены в других ранее существующих последовательностях или может быть найдена только в нескольких ранее существующих последовательностях. Эти характеристики называются "вариантами", потому что они выражены как различия между геномом исследуемого организма и референсным геномом. В решениях предшествующего уровня техники эта информация хранится в файле определенного формата, который называется файлом VCF.

Аннотация вариантов - это процесс присвоения функциональной информации геномным вариантам, идентифицированным процессом определения вариантов. Это подразумевает классификацию вариантов по их отношению к кодирующим последовательностям в геноме и по их влиянию на кодирующую последовательность и генный продукт. В предшествующем уровне техники она обычно хранится в файле MAF.

Процесс анализа цепей ДНК (вариантов, CNV = вариаций числа копий, метилирования и т.д.) для определения их связи с функциями и структурой генов (и белков)

называется функциональным или структурным анализом. В предшествующем уровне техники существует несколько различных решений для хранения этих данных.

Формат файла геномной информации

Изобретение, раскрытое в этом документе, состоит в определении структуры сжатых данных для представления, обработки, манипулирования и передачи данных секвенирования генома, которые отличаются от решений предшествующего уровня техники по меньшей мере в следующих аспектах:

- Этот формат не зависит от каких-либо известных форматов представления геномной информации (например, FASTQ, SAM).
- Он реализует новую оригинальную классификацию геномных данных и метаданных в соответствии с их конкретными характеристиками. Риды последовательности картированы с референсной последовательностью и сгруппированы в отдельные классы по результатам процесса выравнивания. Это позволяет получить классы данных с более низкой информационной энтропией, которые можно более эффективно кодировать, применяя различные конкретные алгоритмы сжатия.
- - Он определяет синтаксические элементы и соответствующий процесс кодирования/декодирования, передающий риды последовательности и информацию выравнивания в представление, которое позволяет более эффективно обрабатывать для приложений последующего анализа.

Классификация ридов в соответствии с результатом картирования и кодирование их с использованием дескрипторов для хранения в слоях (слой положений, слой расстояния спаривания, слой типов несовпадения и т.д., и т.п.) дают следующие преимущества:

- Уменьшение информационной энтропии, когда различные синтаксические элементы моделируются конкретной моделью источника.
- Более эффективный доступ к данным, которые уже организованы в группы/слои, которые имеют особое значение для последующих этапов анализа и к которым можно получить доступ отдельно и независимо.

- Наличие модульной структуры данных, которая может обновляться инкрементно, путем доступа только к необходимой информации без необходимости декодирования всего содержимого данных.
- Геномная информация, создаваемая секвенаторами, по сути своей избыточна из-за характера самой этой информации и необходимости снижения ошибок, присущих процессу секвенирования. Это подразумевает, что релевантная генетическая информация, которая должна быть идентифицирована и проанализирована (вариации относительно референса), составляет лишь небольшую часть произведенных данных. Форматы представления геномных данных предшествующего уровня техники не предназначены для того, чтобы "изолировать" значимую информацию на данном этапе анализа от остальной информации, чтобы сделать ее быстро доступной для приложений анализа.
- Решение, предлагаемое раскрытым изобретением, состоит в том, чтобы представлять геномные данные таким образом, чтобы любая релевантная часть данных была легко доступна для приложений анализа без необходимости доступа и распаковки всей совокупности данных, и избыточность данных эффективно была уменьшена путем эффективного сжатия, чтобы минимизировать необходимый объем памяти и ширину канала передачи.

Ключевыми элементами изобретения являются:

1. Спецификация формата файла, который "содержит" структурированные и выборочно доступные элементы данных (блоки доступа (AU) в сжатом виде). Такой подход можно рассматривать как противоположность подходов предшествующего уровня техники, например SAM и BAM, в которых данные структурируют в несжатом виде, а затем весь файл сжимают. Первое очевидное преимущество этого подхода заключается в наличии возможности эффективно и естественным образом предоставлять различные формы структурированного выборочного доступа к элементам данных в сжатом домене, что невозможно или чрезвычайно неудобно в подходах предшествующего уровня техники.
2. Структурирование геномной информации в конкретные "слои" однородных данных и метаданных представляет значительное преимущество, позволяя определять различные модели источников информации, характеризующиеся низкой энтропией. Такие модели могут не только отличаться от слоя к слою, но

также могут различаться внутри каждого слоя, когда сжатые данные внутри слоев разбиты на блоки данных, включенные в блоки доступа. Такое структурирование позволяет использовать наиболее подходящее сжатие для каждого класса данных или метаданных и их части со значительным выигрышем в эффективности кодирования по сравнению с подходами предшествующего уровня техники.

3. Информация структурирована в блоках доступа (AU), поэтому любой соответствующий набор данных, используемый приложениями для геномного анализа, эффективно и избирательно доступен через соответствующие интерфейсы. Эти функции обеспечивают более быстрый доступ к данным и обеспечивают более эффективную обработку.
4. Определение главной индексной таблицы и локальных индексных таблиц, обеспечивающих избирательный доступ к информации, переносимой слоями кодированных (то есть сжатых) данных, без необходимости декодировать весь объем сжатых данных.
5. Возможность выполнения повторного выравнивания уже выровненных и сжатых геномных данных, когда их необходимо повторно выровнять по сравнению с вновь опубликованными референсными геномами, путем эффективного транскодирования выбранных порций данных в сжатой области. Частая публикация новых референсных геномов в настоящее время требует ресурсов и времени для процессов транскодирования для повторного выравнивания уже сжатых и сохраненных геномных данных относительно вновь опубликованных референсов, потому что должен быть обработан весь объем данных.

Метод, описанный в этом документе, направлен на использование имеющихся априорных знаний о геномных данных для определения алфавита элементов синтаксиса с уменьшенной энтропией. В геномике имеющиеся знания представлены существующей геномной последовательностью, обычно - но не обязательно - того же биологического вида, что и обрабатываемая. Например, человеческие геномы разных людей отличаются только на 1%. С другой стороны, этот небольшой объем данных содержит достаточно информации для ранней диагностики, персонализированной медицины, синтеза индивидуальных лекарств и т.д. Данное изобретение направлено на определение формата представления геномной информации, в котором

соответствующая информация является эффективно доступной и переносимой, а вес избыточной информация сокращен.

Технические признаки, используемые в настоящем изобретении:

1. Разложение геномной информации на однородные "слои" метаданных с целью максимально возможного уменьшения информационной энтропии;
2. Определение главной индексной таблицы и локальных индексных таблиц для обеспечения избирательного доступа к слоям кодированной информации без необходимости декодирования всей кодированной информации;
3. Внедрение различных исходных моделей и энтропийных кодеров для кодирования элементов синтаксиса, принадлежащих разным слоям, определенным в пункте 1;
4. Однозначное соответствие между зависимыми слоями для обеспечения избирательного доступа к данным без необходимости декодировать все слои, если это не требуется;
5. Дифференциальное кодирование по отношению к одной или более адаптивным референсным последовательностям, которые могут быть модифицированы для уменьшения энтропии. После первого кодирования на основании референса зарегистрированные несовпадения могут использоваться для "адаптации/модификации" референсных последовательностей для дальнейшего уменьшения информационной энтропии. Этот процесс может выполняться итеративно, пока уменьшение информационной энтропии является значимым.

Чтобы решить все вышеуказанные проблемы предшествующего уровня техники (с точки зрения эффективного доступа к случайным положениям в файле, эффективной передачи и хранения, эффективного сжатия), настоящее изобретение переупорядочивает и упаковывает вместе данные, которые являются более однородными и/или семантически значимыми для простоты обработки.

Настоящее изобретение также реализует структуру данных на основе концепции "блока доступа" и мультиплексирования релевантных данных.

Геномные данные структурированы и закодированы в различные блоки доступа. Далее следует описание геномных данных, которые содержатся в разных блоках доступа.

Классификация геномных данных

Полученные с помощью секвенаторов риды последовательности классифицируются в соответствии с раскрытым изобретением на 5 различных "классов" по результатам выравнивания относительно одной или болееи референсных последовательностей или геномов.

При выравнивании последовательности ДНК нуклеотидов относительно референсной последовательности возможны пять результатов:

1. Обнаружено, что область в референсной последовательности совпадает с ридом последовательности без каких-либо ошибок (идеальное картирование). Такая последовательность нуклеотидов будет называться "идеально совпадающий рид" или обозначаться как "класс P".
2. Обнаружено, что область в референсной последовательности совпадает с ридом последовательности с несколькими несовпадениями, состоящими из ряда положений, в которых секвенатор не смог определить ни одного основания (или нуклеотида). Такие несовпадения обозначаются буквой "N". Такие последовательности будут обозначаться как "несовпадающие N- риды " или "класс N".
3. Обнаружено, что область в референсной последовательности совпадает с ридом последовательности с несколькими несовпадениями, состоящими из ряда положений, в которых секвенатор не смог определить никакого основания (или нуклеотида), ИЛИ было определено другое основание, нежели указанное в референсной последовательности. Такой тип несовпадения называется однонуклеотидная вариация (SNV) или одиночный нуклеотидный полиморфизм (SNP). Такие последовательности будут обозначаться как "несовпадающие M-риды " или "Класс M".
4. Четвертый класс состоит из ридов, представляющих тип несовпадений, который включает в себя несовпадение класса M плюс наличие инсерции или делеции (также называемых инделы). Инсерции представлены последовательностью из

одного или более нуклеотидов, отсутствующих в референсе, но присутствующих в прочитанной последовательности. В литературе, когда вставленная последовательность находится на краях последовательности, ее называют "мягко обрезанной" (то есть нуклеотиды не соответствуют референсу, но сохраняются в выровненных ридах в противоположность "жестко обрезанным" нуклеотидам, которые отбрасываются). Сохранение или отбрасывание нуклеотидов, как правило, происходит по решению пользователя, реализованному в виде конфигурации инструмента выравнивания. Делеция - это "дыры" (недостающие нуклеотиды) в выровненном риде относительно референса. Такие последовательности будут называться "несовпадающими ридами I" или "класс I".

5. Пятый класс включает в себя все риды, которые не находят какого-либо достоверного картирования на референсной последовательности в соответствии с указанными ограничениями выравнивания. Такие последовательности называются некартированными и относятся к "классу U".

Некартированные риды можно собрать в одну последовательность, используя алгоритмы сборки de-novo. После создания новой последовательности некартированные риды могут быть дополнительно картированы по отношению к ней и классифицированы в один из четырех классов P, N, M и I.

Структура данных указанных геномных данных требует хранения глобальных параметров и метаданных, которые будут использоваться механизмом декодирования. Эти данные структурированы в главном заголовке, описанном в таблице ниже.

Элемент	**Тип**	**Описание**
Уникальный идентификатор	Байтовый массив	Уникальный идентификатор для закодированного контента
Версия	Байтовый массив	Основная + вспомогательная версия алгоритма кодирования
Размер заголовка	Целое число	Размер в байтах всего закодированного содержимого

Длина ридов	Целое число	Размер рида при постоянной длине рида. Специальное значение (например, 0) зарезервировано для переменной длины рида
Количество референсных последовательностей	Целое число	Количество использованных референсных последовательностей
Счетчики блоков доступа	Байтовый массив (например, целые числа)	Общее количество закодированных блоков доступа на каждую референсную последовательность
Идентификаторы референсных последовательностей	Байтовый массив	Уникальные идентификаторы для референсных последовательностей
Главная индексная таблица *Выравнивание положений первого рида в каждом блоке (блок доступа).* *То есть меньшее положение первого рида референсного генома на каждый блок из 4 классов* *1 на класс pos (4) на референс*	Байтовый массив (например, целые числа)	Это многомерный массив, поддерживающий произвольный доступ к блокам доступа

Таблица 1 - Структура главного заголовка

Как только классификация рида завершена с определением классов, дальнейшая обработка состоит в определении набора различных синтаксических элементов, представляющих оставшуюся информацию, позволяющую реконструировать последовательность ридов ДНК, когда она представлена в качестве картированной на

данной референсной последовательности. Сегмент ДНК, относящийся к данной референсной последовательности, может быть полностью выражен следующими параметрами:

- Начальное положение на референсной последовательности (pos) (292).
- Флаг, сигнализирующий о том, что рид должен рассматриваться как обратный комплемент к референсу rcomp (293).
- Расстояние до партнера пары в случае пары ридов pair (294).
- Значение длины рида (295) в случае технологии секвенирования дает переменную длину ридов. В случае постоянной длины рида длина рида, ассоциированная с каждым ридом, очевидно, может быть опущена и может быть сохранена в главном заголовке файла.
- Для каждого несовпадения:
    - Положение несовпадения nmis (300) для класса N, snpp (311) для класса M и indp (321) для класса I)
    - Тип несовпадения (отсутствует в классе N, snpt (312) в классе M, indt (322) в классе I)
- Флаги (296), указывающие специфические характеристики рида последовательности, такие как:
    - шаблон, имеющий несколько сегментов в секвенировании
    - каждый сегмент правильно выровнен согласно выравнивателю
    - некартированный сегмент
    - следующий сегмент в шаблоне не картирован
    - сигнализация первого или последнего сегмента
    - неудача контроля качества
    - ПЦР- или оптический дубликат
    - вторичное выравнивание
    - дополнительное выравнивание
- Строка из мягко обрезанных нуклеотидов (323), когда она присутствует в классе I

Эта классификация создает группы дескрипторов (элементов синтаксиса), которые можно использовать для однозначного представления ридов геномной последовательности. В таблице ниже приведены синтаксические элементы, необходимые для каждого класса выровненных ридов.

	P	N	M	I
pos	X	X	X	X
pair	X	X	X	X
rcomp	X	X	X	X
флаги	X	X	X	X
rlen	X	X	X	X
nmis		X		
snpp			X	
snpt			X	
indp				X
indt				X
indc				X

Таблица 2 – Определение слоев для каждого класса данных.

Риды, принадлежащие к классу P, характеризуются и могут быть полностью восстановлены только по положению, информации об обратном комплементе и смещении между членами пар в случае, если они были получены с помощью технологии секвенирования с получением пар, по некоторым флагам и длине рида.

В следующем разделе подробно описано, как определяются эти дескрипторы.

Слой дескрипторов положения

В каждом блоке доступа только положение картирования первого закодированного рида хранится в заголовке AU как абсолютное положение в референсном геноме. Все остальные положения выражаются как разность относительно предыдущего положения и хранятся в конкретном слое. Такое моделирование источника информации, определяемое последовательностью положений рида, в целом характеризуется пониженной энтропией, особенно для процессов секвенирования,

дающих результаты с высоким перекрыванием. После сохранения абсолютного положения первого выравнивания все положения другого рида выражаются как разность (расстояние) относительно первого.

Например, на Фигуре 4 показано, как после кодирования начального положения первого выравнивания в виде положения "10000" на референсной последовательности, положение второго рида, начиная с положения 10180 кодируется как "180". При данных с высоким перекрыванием (> 50x) большинство дескрипторов вектора положений будет показывать очень высокую встречаемость низких значений, таких как 0 и 1, и других небольших целых чисел. На Фигуре 4 показано, как положения трех пар ридов кодируются в слое pos.

Эта же исходная модель используется для положений ридов, принадлежащих классам N, M, P и I. Чтобы реализовать любую комбинацию избирательного доступа к данным, положения ридов, принадлежащие этим четырем классам, кодируются в отдельных слоях как изображено в таблице I.

Слой дескрипторов спаривания

Дескриптор спаривания хранится в слое "pair". Такой слой хранит дескрипторы, кодирующие информацию, необходимую для восстановления исходных пар ридов, когда используемая технология секвенирования генерирует риды по парам. Хотя на момент раскрытия изобретения подавляющее большинство данных секвенирования генерируется с использованием технологии создания парных ридов, это относится не ко всем технологиям. По этой причине присутствие этого слоя не является необходимым для восстановления всей информации данных секвенирования, если технология секвенирования рассматриваемых геномных данных не генерирует информацию по парным ридам.

Определения:

- **партнёр по паре**: рид, ассоциированный с другим ридом в паре ридов (например, рид 2 - это пара рида 1 в примере на Фигуре 4)
- **расстояние спаривания**: количество положений нуклеотидов в референсной последовательности, которые отделяют одно положение в первом риде (якорь

спаривания, например, последний нуклеотид первого рида) от одного положениы второго рида (например, первый нуклеотид второго рида)

- **наиболее вероятное расстояние спаривания** (**MPPD**): наиболее вероятное расстояние спаривания, выраженное в количестве положений нуклеотидов.
- **расстояние спаривания** в **положениях** (**PPD**): PPD - это способ выразить расстояние спаривания в числе ридов, отделяющих один рид от соответствующей пары, присутствующий в слое дескриптора конкретного положения.
- **наиболее вероятное расстояние спаривания** в **положениях** (**MPPPD**): наиболее вероятное число ридов, отделяющих один рид от его пары, присутствующее в слое дескриптора конкретного положения.
- **ошибка положений спаривания (PPE)**: определяется как разница между MPPD или MPPPD и фактического положения партнёра по паре.
- **якорь спаривания**: положение первого считанного последнего нуклеотида в паре, используемое в качестве референса для вычисления расстояния пары сопряженных элементов, выраженного в числе положений нуклеотидов или числе прочитанных положений.

На Фигуре 5 показано, как рассчитывается расстояние спаривания между парами ридов.

Слой дескрипторов пары - это вектор ошибок спаривания, рассчитанный как число ридов, которые необходимо пропустить, чтобы достичь партнёра по паре первого рида пары с учетом заданного расстояния декодирования спаривания.

На Фигуре 6 показан пример того, как рассчитываются ошибки спаривания, как в виде абсолютной величины, так и в виде дифференциального вектора (характеризуется меньшей энтропией для высоких значений перекрытия).

Для информации о спаривании ридов, принадлежащих классам N, M, P и I, используются одинаковые дескрипторы. Чтобы реализовать выборочный доступ к различным классам данных, информация о спаривании для ридов, принадлежащих четырем классам, кодируется в разных слоях, как изображено в.

Информация о спаривании в случае ридов, картированных по разным референсам

В процессе картирования рида последовательности на референсной последовательности нередко бывает, что первый рид в паре картируется на одном

референсе (например, хромосоме 1), а второе - на другом референсе (например, хромосоме 4). В этом случае описанная выше информация о спаривании должна быть объединена с дополнительной информацией, относящейся к референсной последовательности, используемой для картирования одного из ридов. Это достигается путем кодирования следующих параметров:

1. Зарезервированное значение (флаг), указывающее, что пара картируется на двух разных последовательностях (разные значения указывают, картированы ли рид 1 или рид 2 на последовательности, которая в данный момент не кодирована)

2. Уникальный референсный идентификатор, ссылающийся на идентификаторы референса, закодированные в структуре главного заголовка, как описано в таблице 1.

3. Третий элемент, содержащий информацию о картировании на референсе, идентифицированном в точке 2, и выраженный как смещение относительно последнего кодированного положения.

На Фигуре 7 приведен пример этого сценария.

На Фигуре 7, поскольку рид 4 не картируется в кодированной в данный момент референсной последовательности, геномный кодер передает эту информацию, создавая дополнительные дескрипторы в слое pair. В примере, показанном на Фигуре 7, рид 4 пары 2 картируется на референсе № 4, в то время как закодированный в данный момент референс - № 1. Эта информация кодируется с использованием 3 компонентов:

1) Одно специальное зарезервированное значение кодируется как расстояние спаривания (в этом случае - 0xffffff)

2) Второй дескриптор содержит идентификатор референса, как указано в главном заголовке (в этом случае - 4)

3) Третий элемент содержит информацию о картировании в соответствующем референсе (170).

Слой дескриптора обратного комплемента

Каждое считывание пар ридов, полученных с помощью технологий секвенирования, может происходить из любой цепи генома секвенированного органического образца. Однако только одна из двух цепей используется в качестве референсной

последовательности. На Фигуре 8 показано, как в паре ридов один рид (рид 1) может происходить из одной нити, а другой (рид 2) - из другой.

Когда в качестве референсной последовательности используется цепь 1, рид 2 может быть закодировано как обратный комплемент соответствующего фрагмента на цепи 1. Это показано на Фигуре 9.

В случае сцепленных ридов возможны четыре комбинации пар прямого и обратного комплемента. Это показано на Фигуре 10. Слой rcomp кодирует эти четыре возможных комбинации.

Такое же кодирование используется для информации по обратному комплементу для рида, принадлежащего классам P, N, M, I. Чтобы обеспечить расширенный выборочный доступ к данным, информация по обратному комплементу для рида, принадлежащего к этим четырем классам, кодируется в разных слоях, как показано в таблице 2.

Несовпадения класса N

Класс N включает все риды, которые показывают несовпадения, в которых вместо распознавания оснований присутствует "N". Все остальные основания идеально соответствуют референсной последовательности.

Положения несовпадений N в риде 1 кодируются следующим образом:

- абсолютное положение в риде 1 ИЛИ
- дифференциальное положение относительно предыдущего N в том же риде (в зависимости от того, в каком случае энтропия меньше).

Положения несовпадений N в риде 2 кодируются следующим образом:

- абсолютное положение в риде 2 + длина рида 1 ИЛИ
- дифференциальное положение относительно предыдущего N (в зависимости от того, в каком случае энтропия меньше).

В слое nmis кодирование каждой пары ридов завершается специальным символом-разделителем "S". Это показано на Фигуре 11.

Кодирование замен (несовпадения или SNP)

Замена определяется как наличие в картированном риде нуклеотида, отличного от того, который присутствует в референсной последовательности в том же положении (см. Фиг 12).

Каждая замена может быть закодирована следующим образом:

"положение" (слой snpp) и "тип" (слой snpt). См. Фигуру 13, Фигуру 14, Фигуру 16 и Фигуру 15.

ИЛИ

- только "положение", но с использованием одного слоя snpp для каждого типа несовпадения. См. Фигуру 17.

Положения замен

Положение замены рассчитывается так же, как для значений слоя nmis, т.е.:

В риде 1 замены кодируются

- как абсолютное положение в риде 1 ИЛИ
- как дифференциальное положение относительно предыдущей замены в том же риде

В риде 2 замены кодируются:

- как абсолютное положение в риде 2 + длина рида 1 ИЛИ
- как дифференциальное положение относительно предыдущей замены.

На Фигуре 13 показано, как положения замен кодируются в слое snpp. Положения замен могут быть рассчитаны либо как абсолютные, либо как дифференциальные значения.

В слое snpp кодирование каждой пары ридов завершается специальным символом-разделителем.

Дескрипторы типов замены

Для класса M (и I, как описано в следующих разделах), несовпадения кодируются индексом (с перемещением справа налево) от фактического символа,

присутствующего в референсе, до соответствующего символа замены, присутствующей в риде {A, C, G , T, N, Z}. Например, если выровненный рид показывает C вместо T, который присутствует в том же положении в референсе, индекс несовпадения будет обозначен как "4". Процесс декодирования считывает закодированный синтаксический элемент, нуклеотид в заданном положении на референсе и перемещается слева направо с извлечением (возвращением) декодированного символа. Например, "2", полученное для положения, где в референсе присутствует G, будет декодировано как "N". На Фигуре 14 показаны все возможные замены и соответствующие символы кодирования, когда коды неоднозначности IUPAC не используются, а на Фигуре 15 представлен пример кодирования типов замены в слое snpt.

В случае применения кодов неоднозначности IUPAC индексы замен изменяются, как показано на Фигуре 16.

В случае, когда кодирование типов замен, описанных выше, имеет высокую информационную энтропию, альтернативный способ кодирования замен состоит в хранении только положения несовпадений в отдельных слоях, по одному на нуклеотид, как изображено на Фигуре 17.

Кодирование инсерций и делеций

Для класса I, несовпадения и делеции кодируются с помощью индексов (с перемещением при кодировании справа налево) замены с фактического символа, присутствующего в референсе, на соответствующий символ замены, присутствующий в риде: {A, C, G, T, N, Z}. Например, если выровненный рид показывает C вместо T, присутствующего в том же положении в референсе, индекс несовпадения будет равен "4". В случае, если рид показывает делецию, где в референсе присутствует A, кодированный символ будет "5". Процесс декодирования считывает закодированный синтаксический элемент, нуклеотид в заданном положении на референсе и перемещается слева направо для извлечения декодированного символа. Например, "3", полученное для положения, где в референсе присутствует G, будет декодировано как "Z", что указывает на наличие делеции в риде последовательности.

Инсерции кодируются как 6, 7, 8, 9, 10 соответственно для вставленных A, C, G, T, N.

В случае принятия кодов неоднозначности IUPAC механизм замены оказывается точно таким же, однако вектор замены расширяется следующим образом: S = {A, C, G, T, N, Z, M, R, W, S, Y, K, V, H, D, B}.

На Фигуре 18 и Фигуре 19 показаны примеры того, как кодируются замены, инсерции и делеции в паре ридов класса I.

Следующие структуры формата файла, блоков доступа и мультиплексирования описаны со ссылкой на элементы кодирования, раскрытые в настоящем документе выше. Однако блоки доступа, формат файла и мультиплексирование дают то же техническое преимущество также и с другими и разными алгоритмами моделирования источников и сжатия геномных данных.

Формат файла: выборочный доступ к областям геномных данных

Главная индексная таблица

Для поддержки выборочного доступа к определенным областям выровненных данных, структура данных, описанная в этом документе, реализует инструмент индексирования, называемый Главной индексной таблицей (MIT). Это многомерный массив, содержащий локусы, по которым конкретные риды картируются на используемых референсных последовательностях. Значения, содержащиеся в MIT, представляют собой положения картирования первого рида на каждом слое pos, так что поддерживается непоследовательный доступ к каждому блоку доступа. MIT содержит один раздел для каждого класса данных (P, N, M и I) и для каждой референсной последовательности. MIT содержится в главном заголовке закодированных данных. На Фигуре 20 показана общая структура главного заголовка, на Фигуре 21 показано общее визуальное представление MIT, а на Фигуре 22 показан пример MIT для класса P кодированных ридов.

Значения, содержащиеся в MIT, изображенной на Фигуре 22, используются для прямого доступа к интересующей области (и соответствующему AU) в сжатом домене. Например, со ссылкой на Фиг. 22, если требуется обратиться к области, находящейся между положениями 150 000 и 250 000 в референсе 2, декодирующее приложение будет переходить сразу ко второму референсу в MIT и будет искать два значения k1 и k2, так что что к1 <150 000 и к2> 250 000. Где k1 и k2 - два индекса, считанные из MIT. В

примере на Фигуре 22 это приведет к положениям 3 и 4 второго вектора MIT. Эти возвращенные значения будут затем использоваться декодирующим приложением для извлечения положений соответствующих данных из локальной индексной таблицы слоя pos, как описано в следующем разделе.

Вместе с указателями на слой, содержащий данные, относящиеся к четырем классам геномных данных, описанным выше, MIT можно использовать в качестве реестра дополнительных метаданных и/или аннотаций, добавляемых к геномным данным в течение их жизненного цикла.

Локальная индексная таблица

Каждый слой данных, описанный выше, имеет префикс структуры данных, называемый *локальным заголовком*. Локальный заголовок содержит уникальный идентификатор слоя, вектор счетчиков блоков доступа на каждую референсную последовательность, локальную индексную таблицу (LIT) и, возможно, некоторые специфические метаданные слоя. LIT - это вектор указателей на физическое положение данных, принадлежащих каждому AU в полезной нагрузке слоя. На Фигуре 23 изображены обобщённый заголовок и полезная нагрузка слоя, где LIT используется для доступа к определенным областям кодированных данных непоследовательным образом.

В предыдущем примере для доступа к региону от 150 000 до 250 000 ридов, выровненных по референсной последовательности №. 2, декодирующее приложение извлекло из MIT положения 3 и 4. Эти значения должны использоваться процессом декодирования для доступа к 3-му и 4-му элементам соответствующего раздела LIT. В примере, показанном на Фигуре 24, счетчики "общее число блоков доступа", содержащиеся в заголовке слоя, используются для пропуска индексов LIT, связанных с AU, относящимися к референсу 1 (в примере 5). Индексы, содержащие физические положения запрошенных AU в кодированном потоке, таким образом рассчитываются как:

положение блоков данных, принадлежащих запрошенному AU = блоки данных, принадлежащие AU референса 1, которые должны быть пропущены + положение, извлеченная с использованием MIT, т.е.

положение первого блока: 5 + 3 = 8

положение последнего блока: 5 + 4 = 9

Блоки данных, полученные с использованием механизма индексации, называемого "Локальная индексная таблица", являются частью запрошенных блоков доступа.

На Фигуре 26 показано, как блоки данных, полученные с использованием MIT и LIT, составляют один или более блоков доступа.

Блоки доступа

Геномные данные, классифицированные по классам данных и структурированные в сжатые или несжатые слои, организованы в разные блоки доступа.

Геномные блоки доступа (AU) определяются как участки данных генома (в сжатом или несжатом виде), которые восстанавливают нуклеотидные последовательности и/или релевантные метаданные, и/или последовательности ДНК/РНК (например, виртуальный референс) и/или данные аннотаций, полученные с помощью секвенатора генома и/или устройства для обработки генома или приложения для анализа.

Блок доступа представляет собой блок данных, которые можно декодировать независимо от других блоков доступа, используя только глобально доступные данные (например, конфигурацию декодера), или используя информацию, содержащуюся в других блоках доступа.

Блоки доступа содержат информацию о данных, связанную с геномными данными, в форме информации о местоположении (абсолютной и/или относительной), информацию, связанную с обратным комплементом и, возможно, спариванием, и дополнительные данные. Можно выделить несколько типов блоков доступа.

Блоки доступа различаются по:

- типу, характеризующему природу геномных данных и наборов данных, которые они несут, и то, как к ним можно получить доступ,
- порядку, предоставляющему уникальный порядок доступа к блокам, принадлежащим к одному типу.

Блоки доступа любого типа могут быть далее классифицированы на различные "категории".

Далее приведен неисчерпывающий список определения различных типов геномных блоков доступа:

1) Блоки доступа типа 0 не должны ссылаться на какую-либо информацию, поступающую от других блоков доступа, для доступа к ним или декодирования и доступа (см. Фигуру 29). Вся информация, переносимая данными или наборами данных, которые они содержат, может независимо считываться и обрабатываться декодирующим устройством или обрабатывающим приложением.

2) Блоки доступа типа 1 содержат данные, которые ссылаются на данные, переносимые блоками доступа типа 0 (см. Фигуру 30). Для считывания или декодирования и обработки данных, содержащихся в блоках доступа типа 1, требуется доступ к одному или более блокам доступа типа 0.

   Блоки доступа этого типа могут содержать информацию о несовпадении, несходстве или несоответствии с информацией, содержащейся в блоке доступа типа 0.

3) Блоки доступа типа 2, 3 и 4 содержат данные, которые ссылаются на блок доступа типа 1 (см. Фигуру 31, Фигуру 32 и Фигуру 33). Для считывания или декодирования и обработки данных или наборов данных, содержащихся в блоках доступа типов 2, 3 и 4, требуется информация, переносимая данными или наборами данных, содержащимися в блоках доступа типов 0 и 1. Разница между блоками доступа типа 2, 3, и 4 определяется характером информации, которую они содержат.

4) Блоки доступа типа 5 содержат метаданные (например, показатели качества) и/или данные аннотации, ассоциированные с данными или наборами данных, содержащимися в блоке доступа типа 1. Блоки доступа типа 5 могут классифицироваться и маркироваться в разных слоях.

5) Блоки доступа типа 6 содержат данные или наборы данных, классифицированные как данные аннотаций. Блоки доступа типа 6 могут быть классифицироваться и маркироваться в слоях.

6) Блоки доступа дополнительных типов могут расширять структуру и механизмы, описанные в настоящем документе. В качестве примера, но не в качестве ограничения, результаты определения геномных вариантов, структурного и функционального анализа могут быть закодированы в блоки доступа новых типов. Организация данных в блоках доступа, описанная в данном документе, не

препятствует инкапсуляции данных любого типа в блоках доступа, являясь механизмом, полностью прозрачным по отношению к природе кодируемых данных.

Блоки доступа этого типа могут содержать информацию о несовпадении, несходстве или несоответствии с информацией, содержащейся в блоке доступа типа 0.

На Фигуре 28 показано, что блоки доступа состоят из заголовка и одного или более слоев однородных данных. Каждый слой может состоять из одного или более блоков. Каждый блок содержит несколько (множество) пакетов, а пакеты представляют собой структурированную последовательность дескрипторов, введенных выше для представления, например, положений ридов, информации о спаривании, информации об обратном комплементе, положений и типов несовпадений и т.д.

Каждый блок доступа может иметь различное количество пакетов в каждом блоке, но в пределах блока доступа все блоки имеют одинаковое количество пакетов.

Каждый пакет данных может быть идентифицирован комбинацией трех идентификаторов X Y Z, где:

- X идентифицирует блок доступа, к которому принадлежит пакет
- Y идентифицирует блок, которому принадлежит пакет (то есть тип данных, который он инкапсулирует)
- Z является идентификатором, выражающим порядок пакетов относительно других пакетов в том же блоке

На Фигуре 28 показан пример блоков доступа и маркировки пакетов.

На Фигурах с 34 по 38 показаны блоки доступа нескольких типов, причем общий синтаксис для их обозначения следующий:

AU_T_N - блок доступа типа T с идентификатором N, который может подразумевать или не подразумевать понятие порядка в соответствии с типом блока доступа. Идентификаторы используются для уникальной связи блоков доступа одного типа с идентификаторами других типов, необходимых для полного декодирования переносимых геномных данных.

Блоки доступа любого типа могут быть классифицированы и маркированы в разных "категориях" в соответствии с различными процессами секвенирования. Например, но не в качестве ограничения, классификация и маркировка могут иметь место при:

- секвенировании одного и того же организма в разное время (блоки доступа содержат геномную информацию с "временной" коннотацией),
- секвенировании органических образцов различной природы одних и тех же организмов (например, кожи, крови, волос для человеческих образцов). Это блоки доступа с "биологической" коннотацией.

Блоки доступа типа 1, 2, 3 и 4 построены в соответствии с результатом функции сопоставления, примененной к фрагментам последовательности генома (т.е. ридами) относительно референсной последовательности, закодированной в блоках доступа типа 0, на которую они ссылаются.

Например, блоки доступа (AU) типа 1 (см. Фигуру 30) могут содержать положения и флаги обратного комплемента тех ридов, которые приводят к идеальному совпадению (или максимально возможному баллу, соответствующему выбранной функции сопоставления), когда функция сопоставления применяется к конкретным областям референсной последовательности, кодируемой в AU типа 0. Вместе с данными, содержащимися в AU типа 0, такой информации функции согласования достаточно для полного восстановления всех ридов, представленных набором данных, переносимым блоками доступа типа 1.

Со ссылкой на классификацию геномных данных, ранее описанную в этом документе, блоки доступа типа 1, описанные выше, будут содержать информацию, относящуюся к ридам геномной последовательности класса P (идеальные совпадения).

В случае ридов переменной длины и парных ридов данные, содержащиеся в AU типа 1, указанных в предыдущем примере, должны быть интегрированы с данными, представляющими информацию о спаривании ридов и длины ридов, чтобы иметь возможность полностью восстанавливать геномные данные включая ассоциацию пар ридов. Что касается классификации данных, ранее представленной в настоящем документе, слои pair и rlen будут закодированы в AU типа 1.

Функции сопоставления, применяемые в отношении блоков доступа типа 1 для классификации содержимого AU для типов 2, 3 и 4, могут обеспечивать такие результаты, как:

- каждая последовательность, содержащаяся в AU типа 1, полностью совпадает с последовательностями, содержащимися в AU типа 0, в соответствии с указанным положением;
- каждая последовательность, содержащаяся в AU типа 2, совпадает с последовательностью, содержащейся в AU типа 0, в соответствии с указанным положением, за исключением присутствующих символов "N" (основание не распознается секвенатором) в последовательности в AC типа 2;
- каждая последовательность, содержащаяся в AU типа 3, включает варианты в виде замещенных символов (вариантов) относительно последовательности, содержащейся в AU типа 0 в соответствии с указанным положением;
- каждая последовательность, содержащаяся в AU типа 4, включает варианты в виде замещенных символов (вариантов), инсерции и/или делеции относительно последовательности, содержащейся в AU типа 0 в соответствии с указанным положением.

Блоки доступа типа 0 упорядочены (например, пронумерованы), но их не нужно хранить и/или передавать упорядоченным образом (техническое преимущество: параллельная обработка/параллельная потоковая передача, мультиплексирование). Блоки доступа типа 1, 2, 3 и 4 не нужно упорядочивать и не нужно хранить и/или передавать упорядоченным способом (техническое преимущество: параллельная обработка/параллельная потоковая передача).

Технические результаты

Технический результат структурирования геномной информации в блоки доступа, как описано в настоящем документе, заключается в том, что геномные данные:

1. могут быть выборочно запрошены для доступа:
   - специфические "категории" данных (например, с особой временной или биологической коннотацией) без необходимости распаковывать полные геномные данные или наборы данных и/или связанные метаданные.

- специфические области генома для всех "категорий", подмножества "категорий", одной "категории" (с ассоциированными метаданными или без них) без необходимости распаковки других областей генома

2. могут инкрементно обновляться новыми данными, которые могут стать доступны, когда:
   - выполнен новый анализ геномных данных или наборов данных
   - получены новые геномные данные или наборы данных путем секвенирования тех же организмов (разных биологических образцов, разных биологических образцов одного типа, например образца крови, но полученных в другое время и т.д.)
3. могут быть эффективно перекодированы в новый формат данных в случае
   - новых геномных данных или наборов данных, которые будут использоваться в качестве нового референса (например, новый референсный геном, переносимый AU типа 0)
   - обновления спецификации формата кодирования

По сравнению с решениями предшествующего уровня техники, таких как SAM/BAM, вышеуказанные технические особенности решают проблемы, связанные с необходимостью фильтрации данных на уровне приложений, когда все данные извлекаются и распаковываются из кодированного формата.

Далее следуют примеры сценария применения, в котором такая структура блока доступа становится полезной для технологического преимущества.

Выборочный доступ

В частности, раскрытая структура данных на основе блоков доступа различных типов позволяет

- извлекать только информацию о ридах (данные или наборы данных) из всей последовательности всех "категорий" или подмножества (то есть одного или более слоев) или одной "категории" без необходимости распаковывать также информацию ассоциированных метаданных (ограничение нынешнего уровня техники: SAM/BAM, который не может даже поддерживать различие между различными категориями или слоями)

- извлекать все риды, выровненные по конкретным областям предполагаемой референсной последовательности для всех категорий, подмножеств категорий, одной категории (с ассоциированными метаданными или без них) без необходимости распаковки также других областей генома (ограничение нынешнего уровня техники: SAM/BAM) ;

На Фигуре 39 показано, как для доступа к геномной информации, картированной на втором сегменте референсной последовательности (AU 0-2) с несовпадениями, требуется только декодирование AU 0-2, 1-2 и 3-2. Это пример селективного доступа в соответствии как с критериями, связанными с картированием региона (т.е. положение в референсной последовательности), так и с критериями, связанными с функцией согласования, приложенной к кодированным ридам последовательности относительно референсной последовательности (например, несовпадения только в этом примере).

Еще одним техническим преимуществом является то, что запросы к данным намного более эффективны с точки зрения доступности данных и скорости выполнения, поскольку они могут основываться на доступе и декодировании только выбранных "категорий", определенных областей более длинных геномных последовательностей и только конкретных слоев для блоков доступа типа 1, 2, 3, 4, которые соответствуют критериям применяемых запросов и любой их комбинации.

Организация блоков доступа типа 1, 2, 3, 4 в слои позволяет эффективно извлекать нуклеотидные последовательности

- имеющие конкретные вариации (например, несовпадения, инсерции, делеции) относительно одного или более референсных геномов;
- которые не картируются ни на один из рассматриваемых референсных геномов;
- которые идеально картируются на один или более референсных геномов;
- которые картируются с одним или более уровнями точности

Инкрементное обновление

Блоки доступа типа 5 и 6 позволяют легко вводить аннотации без необходимости распаковывать/декодировать/распаковывать весь файл, тем самым повышая эффективность обработки файла, что является ограничением подходов предшествующего уровня техники. Существующим решениям по сжатию может

потребоваться доступ и обработка большого количества сжатых данных, прежде чем будут доступны желаемые геномные данные. Это приводит к неэффективному использованию ширины пропускания канала ОЗУ и большему энергопотреблению также в аппаратных реализациях. Проблемы энергопотребления и доступа к памяти могут быть устранены с помощью подхода, основанного на описанных в настоящем документе блоках доступа.

Механизм индексации данных, описанный в главной индексной таблице (см. Фигуру 21), вместе с использованием блоков доступа позволяет реализовать инкрементное обновление закодированного содержимого, как описано ниже.

Добавление дополнительных данных

Новая геномная информация может периодически добавляться к существующим геномным данным по нескольким причинам. Например, когда:

- Организм секвенируют в разные моменты времени;
- Одновременно секвенируют несколько разных образцов одного и того же индивидуума;
- Процесс секвенирования сгенерировал новые данные (потоковая передача).

В вышеуказанных ситуациях структурирование данных с использованием описанных в настоящем документе блоков доступа и структуры данных, описанной в разделе формата файла, позволяет инкрементную интеграцию вновь генерируемых данных без необходимости перекодирования существующих данных. Процесс инкрементного обновления может быть реализован следующим образом:

1. Вновь сгенерированные AU могут быть просто конкатенированы в файле с уже существующими AU и
2. индексирование вновь сгенерированных данных или наборов данных включено в главную индексную таблицу, описанную в разделе формата файла этого документа. Один индекс должен позиционировать вновь сгенерированный AU на существующей референсной последовательности, другие индексы состоят из указателей на вновь сгенерированных AU в физическом файле для прямого и выборочного доступа к ним.

Этот механизм проиллюстрирован на Фигуре 40, где ранее существующие данные, закодированные в трех AU типа 1 и четырех AU для каждого типа от 2 до 4,

обновляются тремя AU для каждого типа, причем данные кодирования поступают, например, из новой последовательности, полученной для того же самого индивида.

В конкретном случае использования потоковой передачи геномных данных и наборов данных в сжатой форме инкрементное обновление ранее существующего набора данных может быть полезно при анализе данных, как только они генерируются секвенатором и до того, как фактическое секвенирование будет завершено. Механизм кодирования (компрессор) может собирать несколько (множество) AU параллельно путем "кластеризации" ридов последовательности, которые картируются на той же области выбранной референсной последовательности. Как только первый AU будет содержать количество ридов выше предварительно сконфигурированного порога/параметра, этот AU готов к отправке в анализирующее приложение. Вместе с вновь закодированным блоком доступа механизм кодирования (компрессор) должен убедиться, что все блоки доступа, от которых зависит новый AU, уже отправлены принимающей стороне или отправляются вместе с ним. Например, AU типа 3 потребует наличия соответствующего AU типа 0 и типа 1 на принимающей стороне для правильного декодирования.

Посредством описанного механизма принимающее приложение определения вариантов сможет начать определение вариантов в полученном AU, до того, как процесс секвенирования будет завершен на передающей стороне. Схема этого процесса изображена на Фигуре 41.

Новый анализ результатов.

В течение жизненного цикла обработки генома к одним и тем же данным могут применяться несколько итераций анализа генома (например, различные варианты определения с использованием другого алгоритма обработки). Использование AU, описанных в этом документе, и структура данных, описанная в разделе о формате файлов этого документа, позволяют инкрементное обновление существующих сжатых данных результатами нового анализа.

Например, новый анализ, выполненный на существующих сжатых данных, может произвести новые данные в следующих случаях:

1. Новый анализ может модифицировать существующие результаты, уже ассоциированные с закодированными данными. Этот вариант использования изображен на Фигуре 42 и реализуется путем полного или частичного перемещения содержимого одного блока доступа из одного типа в другой. В случае, если необходимо создать новые AU (из-за предварительно определенного максимального размера на каждый AU), должны быть созданы связанные индексы в главной индексной таблице, и соответствующий вектор должен быть отсортирован при необходимости.
2. Новые данные получены из нового анализа и должны быть ассоциированы с существующими закодированными данными. В этом случае новые AU типа 5 могут быть созданы и объединены с существующим вектором AU того же типа. Эта ситуация и соответствующее обновление главной индексной таблицы изображены на Фигуре 43.

Варианты использования, описанные выше и изображенные на Фиг 42 и 43, реализуются благодаря:

1. Возможности иметь прямой доступ только к данным с низким качеством картирования (например, AU типа 4);
2. Возможности повторного картирования рида в новый геномный регион путем простого создания нового блока доступа, возможно, принадлежащего к новому типу (например, риды, включенные в AU типа 4, могут быть рекартированы в новый регион с меньшим количеством (типа 2-3) несовпадения и включены во вновь созданный AU);
3. Возможности создания AU типа 6, содержащего только недавно полученные результаты анализа и/или связанные аннотации. В этом случае вновь созданные AU должны содержать "указатели" на существующие AU, на которые они ссылаются.

Транскодирование

Сжатые геномные данные могут потребовать транскодирования, например, в следующих ситуациях:

- Публикация новых референсных последовательностей;
- Использование другого алгоритма картирования (рекартирование).

Когда геномные данные картированы в существующем общедоступном референсном геноме, публикация новой версии указанной референсной последовательности или желание картировать данные с использованием другого алгоритма обработки, сегодня требует процесса рекартирования. При рекартировании сжатых данных с использованием форматов файлов предшествующего уровня техники, таких как SAM или CRAM, все сжатые данные должны быть распакованы в их "необработанную" форму для повторного картирования со ссылкой на вновь доступную референсную последовательность или с использованием другого алгоритма картирования. Это верно, даже если недавно опубликованный референс лишь незначительно отличается от предыдущего, или другой используемый алгоритм картирования производит картирование, которое очень близко (или идентично) к предыдущему картированию.

Преимущество транскодирования геномных данных, структурированных с использованием описанных в настоящем документе блоков доступа, заключается в следующем:

1. Картирование относительно нового референсного генома требует только перекодирования (декомпрессии и сжатия) данных из AU, которые картируются в областях генома, имеющих изменения. Кроме того, пользователь может выбрать те сжатые риды, которые по любой причине, возможно, потребуется повторно картировать, даже если они изначально не картируются в измененной области (это может произойти, если пользователь считает, что предыдущее картирование имеет плохое качество). Этот вариант использования изображен на Фигуре 44.

2. Если недавно опубликованный референсный геном отличается от предыдущего только тем, что целые области смещены в другие места генома ("локусы"), операция транскодирования получается особенно простой и эффективной. Фактически, чтобы переместить все картированные риды в "сдвинутую" область, достаточно изменить только значение абсолютного положения, содержащееся в связанном заголовке (набора) AU. Каждый заголовок AU содержит абсолютную положение, в котором картируется первый рид, содержащийся в AU, в референсной последовательности, в то время как все остальные положения рида кодируются дифференциально по отношению к первой. Поэтому, просто обновляя значение абсолютного положения первого рида, все риды в AU перемещаются

соответственно. Этот механизм не может быть реализован современными подходами, такими как CRAM и BAM, поскольку положения данных генома кодируются в сжатой полезной нагрузке, что требует полной распаковки и повторного сжатия всех наборов данных генома.

3. Когда используется другой алгоритм картирования, его можно применять только к части сжатого рида, которая считается картированной с низким качеством. Например, может быть целесообразно применять новый алгоритм картирования только для рида, которое не полностью совпадало референсным геномом. С существующими форматами сегодня невозможно (или только частично возможно с некоторыми ограничениями) извлекать риды в соответствии с их качеством картирования (т.е. наличием и количеством несовпадений). Если новые результаты картирования возвращаются новыми инструментами картирования, связанные риды могут быть перекодированы из одного AU в другой того же типа (фигура 46) или из одного AU одного типа в AU другого типа (фигура 45).

Кроме того, существующим решениям для сжатия необходим доступ к большому количеству сжатых данных и обрабатывать их до того, как будут получены желаемые геномные данные. Это приводит к неэффективному использованию пропускной способности ОЗУ и большему энергопотреблению в аппаратных реализациях. Проблемы энергопотребления и доступа к памяти могут быть устранены с помощью подхода, основанного на описанных в настоящем документе блоках доступа.

Еще одним преимуществом реализации геномных блоков доступа, описанных в настоящем документе, является облегчение параллельной обработки и пригодность для аппаратных исполнений. Современные решения, такие как SAM/BAM и CRAM, предназначены для реализации однопоточного программного обеспечения.

Выборочное шифрование

Подход, основанный на блоках доступа, организованных в нескольких типах слоев, как описано в этом документе, позволяет реализовать механизмы защиты содержимого, в противном случае невозможные с современными монолитными решениями.

Специалистам в данной области известно, что большая часть геномной информации, относящейся к генетическому профилю организма, основана на различиях (вариантах) относительно известной последовательности (например, референсного генома или популяции геномов). Индивидуальный генетический профиль, который должен быть защищен от несанкционированного доступа, поэтому будет закодирован в Блоки доступа типа 3 и 4, как описано в этом документе. Поэтому реализация контролируемого доступа к наиболее чувствительной геномной информации, получаемой в процессе секвенирования и анализа, может быть реализована путем шифрования только полезной нагрузки блоков AU типа 3 и 4 (см. пример на Фигуре 47). Это обеспечит значительную экономию с точки зрения как вычислительной мощности, так и пропускной способности, поскольку ресурсы, потребляющие процесс шифрования, должны применяться только к подмножеству данных.

Транспортировка геномных блоков доступа

Мультиплекс геномных данных

Геномные блоки доступа могут передаваться по сети связи в рамках мультиплекса геномных данных. Мультиплекс геномных данных определяется как последовательность пакетированных геномных данных и метаданных, представленных в соответствии с классификацией данных, раскрытой в рамках этого изобретения, передаваемых в сетевых средах, где могут возникать ошибки, такие как потери пакетов.

Мультиплекс геномных данных призван упростить и сделать более эффективной транспортировку геномных кодированных данных в различных средах (обычно сетевых средах) и обладает следующими преимуществами, которых нет в современных решениях:

1. Он позволяет инкапсулировать поток или последовательность геномных данных (описанных ниже) или формат геномного файла, сгенерированный инструментом кодирования, в один или более мультиплексных геномных данных, чтобы перенести их в сетевую среду, а затем восстановить достоверный и идентичный поток или формат файла, чтобы сделать передачу и доступ к информации более эффективной

2. Он позволяет выборочно извлекать закодированные геномные данные из инкапсулированных потоков данных для декодирования и представления.
3. Он позволяет мультиплексировать несколько (множество) наборов геномных данных в один контейнер информации для транспортировки и демультиплексировать подмножество передаваемой информации в новый мультиплекс **геномных данных**.
4. Он позволяет мультиплексировать данные и метаданные, созданные различными источниками (с последующим раздельным доступом), и/или процессы секвенирования/анализа и передавать полученный мультиплекс геномных данных через сетевую среду.
5. Он поддерживает обнаружение ошибок, таких как потери пакетов.
6. Он поддерживает правильные данные переупорядочения, которые могут поступать не по порядку из-за сетевых задержек, что делает передачу геномных данных более эффективной по сравнению с современными решениями.

Пример мультиплексирования геномных данных показан на Фигуре 49.

Геномный набор данных

В контексте настоящего изобретения набор геномных данных определяется как структурированный набор геномных данных, включающий, например, геномные данные живого организма, одну или более последовательностей и метаданные, сгенерированные несколькими этапами обработки геномных данных, или результат секвенирования генома живого организма. Один мультиплекс геномных данных может включать несколько наборов геномных данных (как в многоканальном сценарии), где каждый набор данных относится к разным организмам. Механизм мультиплексирования нескольких наборов данных в один мультиплекс геномных данных регулируется информацией, содержащейся в структурах данных, называемых списком наборов геномных данных (GDL) и таблицей картирования геномных данных (GDMT).

Список наборов **геномных данных**

Список наборов геномных данных (GDL) определяется как структура данных, в которой перечислены все наборы геномных данных, доступные в мультиплексе геномных

данных. Каждый из перечисленных наборов геномных данных идентифицируется уникальным значением, называемым идентификатором набора геномных данных (GID).

Каждый набор геномных данных, перечисленный в GDL, ассоциирован с:

- одним потоком геномных данных, несущим одну таблицу картирования наборов геномных данных (GDMT) и идентифицируемым конкретным значением идентификатора потока (genomic_dataset_map_SID);
- одним потоком геномных данных, несущим одну таблицу картирования референсных идентификаторов (RIDMT) и идентифицируемым конкретным значением идентификатора потока (reference_id_map_SID).

GDL отправляется как полезная нагрузка одного транспортного пакета в начале передачи потока геномных данных; затем его можно периодически повторно передавать для обеспечения произвольного доступа к потоку.

Синтаксис структуры данных GDL представлен в таблице ниже с указанием типа данных, ассоциированного с каждым синтаксическим элементом.

Синтаксис	**Тип данных**
genomic_dataset_list() {	
list_length (длина списка)	строка битов
multiplex_id (идентификатор мультиплекса)	строка битов
version_number (номер версии)	строка битов
applicable_section_flag (применимый флаг раздела)	бит
list_ID (идентификатор списка)	строка битов
for (i = 0; i < N; i++) {	N = количество геномных наборов данных в этом геномном мультиплексе
genomic_dataset_ID	строка битов

(идентификатор геномного набора данных)	
genomic_dataset_map_SID (SID карты набора геномных данных)	строка битов
reference_id_map_SID (SID карты референсных идентификаторов)	строка битов
}	
CRC_32	строка битов
}	

Синтаксические элементы, составляющие описанный выше GDL, имеют следующее значение и функцию.

section_length	поле строки битов, указывающее количество байтов, составляющих раздел, начинающееся сразу после поля section_length и включающее CRC.
multiplex_id	поле строки битов, служащее меткой для идентификации этого мультиплексированного потока из любого другого мультиплекса внутри сети.
version_number	строка битов, указывающая номер версии всего раздела списка геномных данных. Номер версии должен увеличиваться на 1 всякий раз, когда изменяется определение таблицы картирования геномных данных. Достигнув значения 127, оно возвращается на 0. Когда значение apply_section_flag установлено на "1", то номер версии должен соответствовать номеру применимого в настоящее время списка наборов геномных данных. Когда applicable_section_flag установлен на "0", то version_number должен соответствовать номеру следующего применимого списка наборов геномных данных.
applicable_section_flag	1-битный индикатор, который при значении "1" указывает,

	что отправленная таблица картирования набора геномных данных применима в настоящее время. Когда бит установлен на "0", это означает, что отправленная таблица еще не применима и должна стать следующей таблицей, которая станет действительной.
list_ID	Это поле строки битов, идентифицирующее текущий список геномных данных.
genomic_dataset_ID	genomic_dataset_ID - поле строки битов, которое задает геномный набор данных, к которому применяется genomic_dataset_map_SID. Это поле не должно принимать одно значение более одного раза в пределах одной версии таблицы картирования набора геномных данных.
genomic_dataset_map_SID	genomic_dataset_map_SID - поле строки битов, идентифицирующее поток геномных данных, содержащий таблицу картирования набора геномных данных (GDMT), ассоциированную с этим набором геномных данных. Ни один genomic_dataset_ID не должен иметь более одного ассоциированного genomic_dataset_map_SID. Значение genomic_dataset_map_SID определяется пользователем.
reference_id_map_SID	reference_id_map_SID - поле строки битов, идентифицирующее поток геномных данных, содержащий таблицу картирования референсных идентификаторов (RIDMT), ассоциированную с этим набором геномных данных. Ни один genomic_dataset_ID не должен иметь более одного ассоциированного reference_id_map_SID. Значение reference_id_map_SID определяется пользователем.
CRC_32	Это поле строки битов, которое содержит значение проверки целостности для всего GDL. Типичным алгоритмом, используемым для этой функции, является алгоритм CRC32, выдающий 32-битное значение.

Таблица картирования наборов геномных данных

Таблица картирования наборов геномных данных (GDMT) создается и передается в начале процесса потоковой передачи (и, возможно, периодически передается повторно, обновляется или идентифицируется для обеспечения возможности обновления точек совпадения (соответствия) и релевантных зависимостей в потоковых данных). GDMT переносится одним пакетом, следующим за списком наборов геномных данных, и содержит списки SID, идентифицирующие потоки геномных данных, составляющие один набор геномных данных. GDMT является полной коллекцией всех идентификаторов потоков геномных данных (например, геномной последовательности, референсного генома, метаданных и т.д.), составляющих один набор геномных данных, переносимый геномным мультиплексом. Таблица картирования набора геномных данных способствует произвольному доступу к геномным последовательностям предоставляя идентификатор потока геномных данных, ассоциированный с каждым набором геномных данных.

Синтаксис структуры данных GDMT представлен в таблице ниже с указанием типа данных, ассоциированного с каждым элементом синтаксиса.

Syntax	Тип данных
genomic_dataset_mapping _table() { (Таблица картирования наборов геномных данных)	
table_length	строка битов
genomic_dataset_ID	строка битов
version_number	строка битов
applicable_section_flag	бит
mapping_table_ID (ID таблицы картирования)	строка битов
genomic_dataset_ef_length (длина поля расширения набора	строка битов

геномных данных)	
for (i=0; i<N; i++) {	N = количество полей расширения, ассоциированных с этим набором геномных данных
extension_field() (поле расширения)	структура данных
}	
for (i = 0;i < M ; i++) {	M = количество потоков геномных данных, ассоциированных с этим конкретным набором данных
data_type (тип данных)	строка битов
genomic_data_SID	строка битов
gd_component_ef_length (длина поля расширения компонента геномных данных)	строка битов
for (I = 0; I < K; i++) {	K = количество полей расширения, ассоциированных с каждым потоком геномных данных
extension_field ()	структура данных
}	
}	
CRC_32	строка битов
}	

Синтаксические элементы, составляющие GDMT, описанные выше, имеют следующее значение и функцию.

version_number, applicable_section_flag	Эти элементы имеют то же значение, что и для GDL
table_length,	поле строки битов, указывающая количество байтов, составляющих таблицу, начиная с поля table_length и включая поле CRC_32.

genomic_dataset_ID	поле строки битов, идентифицирующее геномный набор данных
mapping_table_ID	поле строки битов, идентифицирующее текущую таблицу картирования набора геномных данных
genomic_dataset_ef_length	поле строки битов, указывающее число байтов необязательного поля extension_field, ассоциированного с этим набором геномных данных
data_type	поле строки битов, указывающее тип геномных данных, переносимых пакетами, идентифицированными параметром genomic_data_SID.
genomic_data_SID	поле строки битов, указывающее идентификатор потока пакетов, несущих закодированные геномные данные, ассоциированные с одним компонентом этого набора геномных данных (например, положения ридов read p, информация о спаривании read p и т.д., как определено в этом изобретении)
gd_component_ef_length	поле строки битов, указывающее число байтов необязательного поля extension_field, ассоциированного с геномным потоком, идентифицированным параметром genomic_data_SID.
CRC_32	Это поле строки битов, которое содержит значение проверки целостности для всего GDMT. Типичным алгоритмом, используемым для этой функции, является алгоритм CRC32, выдающий 32-битное значение.

Поля расширения *extension_fields* являются необязательными дескрипторами, которые могут использоваться для дальнейшего описания либо набора геномных данных, либо одного компонента набора геномных данных.

Таблица картирования референсных идентификаторов

Таблица картирования референсных идентификаторов (RIDMT) создается и передается в начале процесса потоковой передачи. RIDMT переносится одним пакетом, следующим за списком наборов геномных данных. RIDMT определяет картирование между числовыми идентификаторами референсных последовательностей (REFID), содержащимися в заголовке блока доступа, и (обычно литеральными) референсными идентификаторами, содержащимися в главном заголовке, указанном в таблице 1.

RIDMT может периодически повторно передаваться для:

- реализации обновления точек соответствия и релевантных зависимостей в потоковых данных,
- поддержки интеграции новых референсных последовательностей, добавленных к ранее существующим (например, синтетических референсов, созданных в процессах сборки de-novo)

Синтаксис структуры данных RIDMT представлен в таблице ниже с указанием типа данных, ассоциированного с каждым элементом синтаксиса.

Синтаксис	**Тип данных**
reference_id_картирование_table() { (таблица картирования референсных идентификаторов)	
table_length	строка битов
genomic_dataset_ID	строка битов
version_number	строка битов
applicable_section_flag	бит
reference_id_mapping _table_ID	строка битов
for (i = 0; i < N; i++) {	N = количество референсных последовательностей, ассоциированных с набором геномных данных, идентифицированных параметром genomic_dataset_ID

ref_string_length (длина строки референса)	строка битов
for (i=0;i<ref_string_length;i++){	
ref_string[i]	байт
}	
REFID	строка битов
}	
CRC_32	строка битов
}	

Синтаксические элементы, составляющие RIDMT, описанные выше, имеют следующее значение и функцию.

table_length, genomic_dataset_ID, version_number, applicable_section_flag	Эти элементы имеют то же значение, что и для GDMT
reference_id_mapping _table_ID	поле строки битов, идентифицирующее текущую таблицу картирования референсных идентификаторов
ref_string_length	поле строки битов, указывающее количество символов (байтов), составляющих ref_string, за исключением символа конца строки (‘\ 0’).
ref_string[i]	поле байта, кодирующее каждый символ строкового представления референсной последовательности (например, "chr1" для хромосомы 1). Символ конца строки (‘\ 0’) необязателен, так как он неявно выводится из поля ref_string_length
REFID	Поле строки битов, однозначно идентифицирующее референсную последовательность. Оно закодировано в

	заголовке блока данных как поле REFID.
CRC_32	Поле строки битов, которое содержит значение проверки целостности для всего RIDMT. Типичным алгоритмом, используемым для этой функции, является алгоритм CRC32, выдающий 32-битное значение.

Поток геномных данных

Мультиплекс геномных данных содержит один или более потоков геномных данных, причем каждый поток может транспортировать

- структуры данных, содержащие транспортную информацию (например, список наборов геномных данных, таблицу картирования набора геномных данных и т.д.)
- данные, принадлежащие одному из слоев геномных данных, описанному в данном изобретении.
- метаданные, относящиеся к геномным данным
- любые другие данные

Поток геномных данных, содержащий геномные данные, по сути представляет собой пакетированную версию Слоя геномных данных, где каждый пакет предваряется заголовком, описывающим содержимое пакета и его связь с другими элементами мультиплекса.

Формат потока геномных данных, описанный в этом документе, и **формат файла**, определенный в этом изобретении, являются взаимно конвертируемыми. Хотя полный формат файла может быть восстановлен полностью только после того, как все данные были получены, в случае потоковой передачи инструмент декодирования может восстановить и получить доступ, а также начать обработку частичных данных в любое время.

Поток геномных данных состоит из нескольких блоков геномных данных, каждый из которых содержит один или более пакетов геномных данных. Блоки геномных данных

(GDB) - это контейнеры геномной информации, составляющие один геномный AU. GDB может быть разделен на несколько пакетов геномных данных в соответствии с требованиями канала связи.

Геномные блоки доступа состоят из одного или более блоков геномных данных, принадлежащих разным потокам геномных данных.

Пакеты геномных данных (GDP) - это единицы передачи, составляющие один GDB. Размер пакета обычно устанавливается в соответствии с требованиями канала связи.

На Фигуре 27 показана взаимосвязь между геномным мультиплексом, потоками, блоками доступа AU, блоками и пакетами при кодировании данных, принадлежащих классу P, как определено в этом изобретении. В этом примере три геномных потока инкапсулируют информацию о положении, спаривании и обратном комплементе ридов последовательности.

Блоки геномных данных состоят из заголовка, полезной нагрузки сжатых данных и заполнения дополнительной информацией.

В таблице ниже приведен пример реализации заголовка GDB с описанием каждого поля и типичным типом данных.

Тип данных	**Описание**	**Тип данных**
Префикс стартового кода блока (BSCP)	Зарезервированное значение, используемое для однозначной идентификации начала блока геномных данных.	строка битов
Идентификатор формата (FI)	Однозначно идентифицирует слой геномных данных, к которому принадлежит блок.	строка битов
Флаг POS (PSF)	Если установлен флаг POS, блок содержит 40-битное поле POS в конце заголовка блока и перед необязательными полями.	бит
Флаг заполнения (PDF)	Если установлен флаг заполнения, блок содержит дополнительные байты заполнения после полезной	бит

	нагрузки, которые не являются частью полезной нагрузки.	
Размер блока (BS)	Количество байтов, составляющих блок, включая этот заголовок и полезную нагрузку и исключая заполнение (общий размер блока будет равен BS + размер заполнения).	строка битов
ID блока доступа (AUID)	Однозначный идентификатор, линейно увеличивающийся (не обязательно на 1, хотя рекомендуется). Необходим для реализации правильного произвольного доступа, как описано в главной индексной таблице, определенной в этом изобретении.	строка битов
(Необязательно) ID референса (REFID)	Однозначный идентификатор, идентифицирующий референсную последовательность, к которой относится AU, содержащий этот блок. Это необходимо, наряду с полем POS, для обеспечения надлежащего произвольного доступа, как описано в главной индексной таблице.	строка битов
(Необязательно) POS (POS)	Присутствует, если PSF равен 1. Положение на референсной последовательности первого рида в блоке.	строка битов
(Дополнительные необязательные поля)	Дополнительные необязательные поля, присутствие сообщается параметром BS.	строка байтов
Полезная нагрузка	Блок кодированной геномной информации (синтаксические элементы описаны в данном изобретении	строка байтов

(Необязательно) заполнение	(Необязательно, присутствие сигнализируется PDF) Фиксированное значение строки битов, которое можно вставить для достижения соответствия требованиям канала. Если присутствует, первый байт указывает, сколько байтов составляет заполнение. Отбрасывается декодером.	строка битов

Использование AUID, POS и BS позволяет декодеру восстанавливать механизмы индексации данных, называемые главной индексной таблицей (MIT) и локальной индексной таблицей (LIT) в этом изобретении. В сценарии потоковой передачи данных использование AUID и BS позволяет принимающей стороне динамически повторно воссоздавать LIT локально, без необходимости отправлять дополнительные данные. Использование AUID, BS и POS позволит воссоздать MIT локально без необходимости отправки дополнительных данных.

Это имеет техническое преимущество для

- снижения издержек кодирования, которые могут быть большими, если передается вся LIT;
- устранения необходимости полного картирования между геномными положениями и блоками доступа, которое обычно недоступно в сценарии потоковой передачи

Блок геномных данных может быть разделен на один или более пакетов геномных данных, в зависимости от ограничений сетевого уровня, таких как максимальный размер пакета, коэффициент потери пакетов и т.д. Пакет геномных данных состоит из заголовка и полезной нагрузки кодированных или зашифрованных геномных данных. как описано в таблице ниже.

Тип данных	**Описание**	**Размер данных**

ID потока (SID)	Однозначно идентифицирует тип данных, переносимых этим пакетом. Чтобы картировать идентификаторы потока с типами данных, в начале потока необходима таблица картирования набора геномных данных. Используется также для обновления точек соответствия и релевантных зависимостей.	строка битов
Маркерный бит блока доступа (MB)	Установлен для последнего пакета блока доступа. Позволяет идентифицировать последний пакет AU.	бит
Номер счетчика пакетов (SN)	Счетчик, ассоциированный с каждым идентификатором потока, линейно увеличивающимся на 1. Необходим для идентификации разрывов/потерь пакетов. Обнуляется после 255.	строка битов
Размер пакета (PS)	Количество байтов, составляющих пакет, включая заголовок, необязательные поля и полезную нагрузку.	строка битов
Флаг расширения (EF)	Установлен, если присутствуют поля расширения.	бит
Поля расширения	Необязательные поля, присутствие сигнализируется параметром PS.	строка байтов
Полезная нагрузка	Данные блока (весь блок или фрагмент)	строка байтов

Геномный мультиплекс может быть декодирован должным образом только в том случае, если получен хотя бы один список наборов геномных данных, одна таблица картирования набора геномных данных и одна таблица картирования референсных

идентификаторов, позволяющая картировать каждый пакет с конкретным компонентом набора геномных данных.

Процесс мультиплексного кодирования

На Фигуре 49 показано, как перед преобразованием в структуры данных, представленные в этом изобретении, необработанные данные геномной последовательности необходимо картировать на одной или более референсных последовательностях, известных априори (493). Если референсная последовательность недоступна, может быть построен синтетический референс из необработанных данных последовательности (490). Этот процесс называется сборкой de-novo. Уже выровненные данные могут быть повторно выровнены для уменьшения информационной энтропии (492). После выравнивания геномный классификатор (494) создает классы данных в соответствии с функцией сопоставления ридов с одной или болееи референсными последовательностями и отделяет метаданные (432) (например, показатели качества) и данные аннотаций (431) от геномных последовательностей. Затем анализатор данных (495) генерирует блоки доступа, описанные в этом изобретении, и отправляет их в геномный мультиплексор (496), который генерирует геномный мультиплекс.

ФОРМУЛА ИЗОБРЕТЕНИЯ

1. Способ передачи геномных данных с использованием мультиплексирования, в котором мультиплексированные потоки данных (480) содержат:

структуру данных списка набора геномных данных (481) для предоставления списка всех наборов геномных данных (482-483), причем указанные наборы геномных данных содержат геномные данные, доступные в геномных потоках (484);

структуру данных таблицы картирования набора геномных данных (485) для предоставления идентификатора каждого потока указанных геномных данных, связанных с каждым набором геномных данных (482 – 483);

и наборы геномных данных, разбитые на блоки доступа с произвольным доступом (486)

2. Способ по п.1, дополнительно содержащий таблицу картирования референсных идентификаторов (487) для обеспечения картирования между числовыми идентификаторами референсных последовательностей, содержащихся в заголовке блока (291) указанных блоков доступа (486), и референсными идентификаторами, содержащимися в главном заголовке (488) потока.

3. Способ по п.2, характеризующийся тем, что указанный геномный набор данных разделен на блоки доступа

4. Способ по п.3, характеризующийся тем, что указанные блоки доступа разбиты на блоки (489)

5. Способ по п.4, характеризующийся тем, что указанные блоки разбиты на пакеты (4810)

6. Способ по любому из предыдущих пунктов, характеризующийся тем, что указанный список наборов геномных данных содержит информацию для идентификации потока, ассоциированного с каждым набором геномных данных, предназначенного для мультиплексирования в мультиплексированном потоке.

7. Способ по любому из пп.1-5, характеризующийся тем, что указанная таблица картирования набора геномных данных содержит информацию для идентификации точек соответствия и соответствующих зависимостей между различными мультиплексированными потоками.

8. Способ по п.7, характеризующийся тем, что указанные различные мультиплексированные потоки содержат: геномную последовательность, референсную геномную последовательность, метаданные.

9. Способ по п.1, характеризующийся тем, что указанную таблицу картирования набора геномных данных передают в одном пакете после списка наборов геномных данных.

10. Способ по п.9, характеризующийся тем, что указанную таблицу картирования набора геномных данных периодически передают повторно или обновляют для обновления точек соответствия и соответствующих зависимостей в потоковых данных.

11. Способ по п.1, характеризующийся тем, что указанный список (481) геномных данных отправляют в качестве полезной нагрузки одного транспортного пакета

12. Способ по п.12, характеризующийся тем, что указанный список геномных данных периодически передают повторно для обеспечения произвольного доступа к потоку.

13. Устройство для передачи геномных данных, использующее мультиплексирование, содержащее средства, подходящие для осуществления способа по пунктам 1-12

14. Вспомогательные данные, сохраняющие геномные данные, мультиплексированные согласно способу по пунктам 1-12

15. Машиночитаемый носитель информации, на котором записана программа, содержащая наборы команд для выполнения способа по пунктам 1-12

16. Формат файла хранения геномных данных, мультиплексированных согласно способу по пунктам 1-12

17. Устройство для приема геномных данных, содержащее средство для демультиплексирования потока геномных данных, причем указанный поток сформирован способом по пунктам 1-12

18. Система для передачи геномных данных, содержащая устройство для мультиплексирования и устройство для демультиплексирования, указанные в пунктах 13 и 17.

ФОРМУЛА ИЗОБРЕТЕНИЯ

(по ст. 34PCT, для рассмотрения на рег. фазе в РФ)

1. Способ передачи геномных данных в виде мультиплексированных потоков данных (480), содержащих:

структуру данных списка набора геномных данных (481) для предоставления списка всех наборов геномных данных (482-483), причем указанные наборы геномных данных содержат геномные данные, доступные в геномных потоках (484);

структуру данных таблицы картирования набора геномных данных (485) для предоставления идентификатора каждого потока указанных геномных данных, связанных с каждым набором геномных данных (482 – 483);

и наборы геномных данных, разбитые на блоки доступа с произвольным доступом (486), причем указанные потоки геномных данных (484) содержит кодированные выровненные риды, организованные в множество слоев дескрипторов гомогенных данных для однозначного представления ридов последовательности, причем в одном слой (pos) хранится положение картирование первого рида относительно референсного генома, а все другие положения выражены как разность относительно предыдущего положения и хранятся в специальном слое, причем указанный способ дополнительно включает

сжатие указанных слоев гомогенных дескрипторов данных и передачу указанных потоков данных.

2. Способ по п.1, дополнительно содержащий таблицу картирования референсных идентификаторов (487) для обеспечения картирования между числовыми идентификаторами референсных последовательностей, содержащихся в заголовке блока (291) указанных блоков доступа (486), и референсными идентификаторами, содержащимися в главном заголовке (488) потока.

3. Способ по п.2, характеризующийся тем, что указанный геномный набор данных разделен на блоки доступа.

4. Способ по п.3, характеризующийся тем, что указанные блоки доступа разбиты на блоки (489).

5. Способ по п.4, характеризующийся тем, что указанные блоки разбиты на пакеты (4810).

6. Способ по любому из предыдущих пунктов, характеризующийся тем, что указанный список наборов геномных данных содержит информацию для идентификации потока, ассоциированного с каждым набором геномных данных, предназначенного для мультиплексирования в мультиплексированном потоке.

7. Способ по любому из пп.1-5, характеризующийся тем, что указанная таблица картирования набора геномных данных содержит информацию для идентификации точек соответствия и соответствующих зависимостей между различными мультиплексированными потоками.

8. Способ по п.7, характеризующийся тем, что указанные различные мультиплексированные потоки содержат: геномную последовательность, референсную геномную последовательность, метаданные.

9. Способ по п.1, характеризующийся тем, что указанную таблицу картирования набора геномных данных передают в одном пакете после списка наборов геномных данных.

10. Способ по п.9, характеризующийся тем, что указанную таблицу картирования набора геномных данных периодически передают повторно или обновляют для обновления точек соответствия и соответствующих зависимостей в потоковых данных.

11. Способ по п.1, характеризующийся тем, что указанный список (481) геномных данных отправляют в качестве полезной нагрузки одного транспортного пакета.

12. Способ по п.12, характеризующийся тем, что указанный список геномных данных периодически передают повторно для обеспечения произвольного доступа к потоку.

13. Устройство для передачи мультиплексированных геномных данных, содержащее средства, подходящие для осуществления способа по пунктам 1-12.

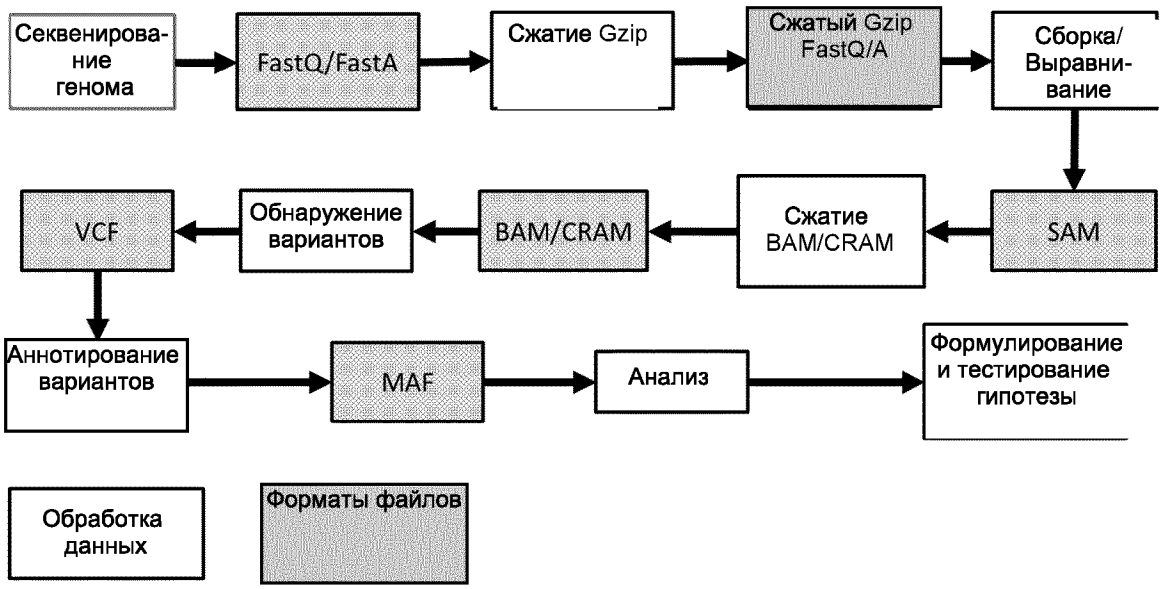
14. Устройство для хранения, сохраняющее геномные данные, сжатые согласно способу по пунктам 1-12

15. Машиночитаемый носитель информации, на котором записана программа, содержащая наборы команд для выполнения способа по пунктам 1-12.

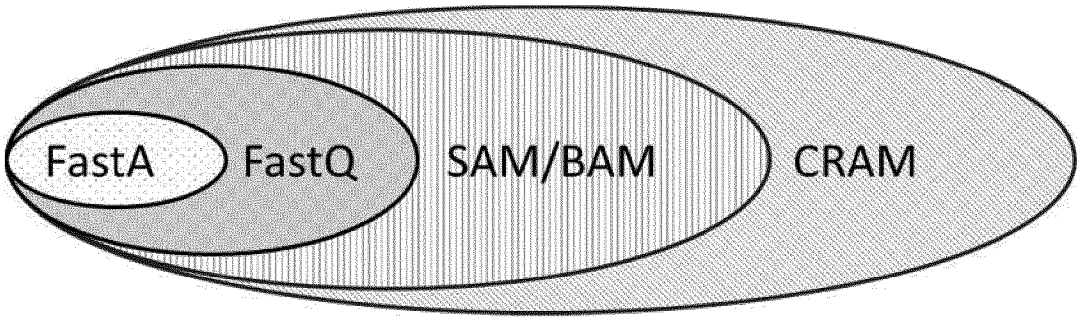
16. Способ по пп. 1-12, в котором данные организованы с получением формата файла.

17. Устройство для приема геномных данных, содержащее средство для демультиплексирования потока геномных данных, причем указанный поток сформирован способом по пунктам 1-12.

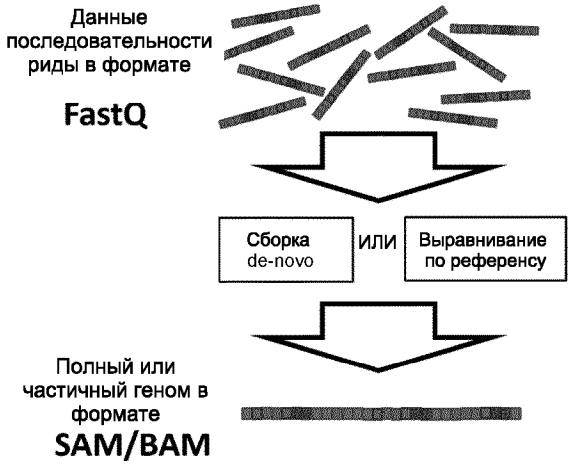
18. Система для передачи мультиплексированных геномных данных, содержащая устройство для передачи и устройство для приема, указанные в пунктах 13 и 17.



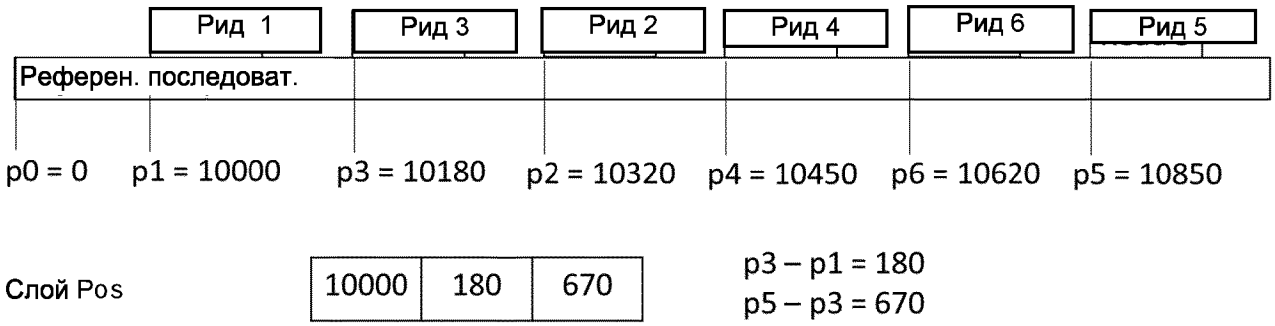
Фигура 1 - Типичный конвейер обработки генома и форматы файлов.



Фигура 2 - Форматы геномных файлов и их связь.

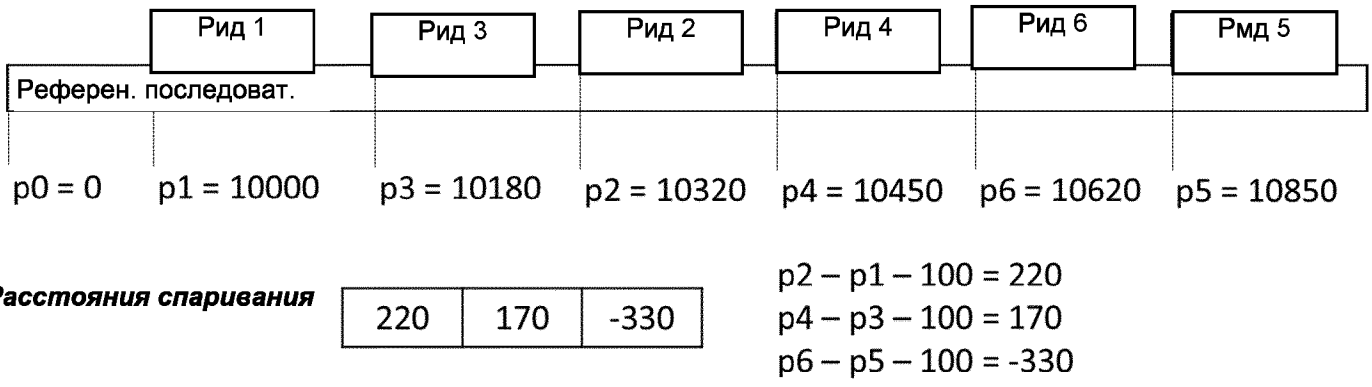


Фигура 3 - Выравнивание ридов геномной последовательности.



Фигура 4 - Как кодируется положение первого рида трех картированных пар ридов в слое pos.

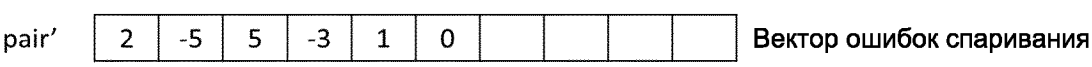
ПРИМЕР
Постоянная длина прочтений = 100



Фигура 5 - Расчет расстояния спаривания для трех пар ридов

ПРИМЕР
Наиболее вероятное расстояние спаривания положений (MPPPD) = 2

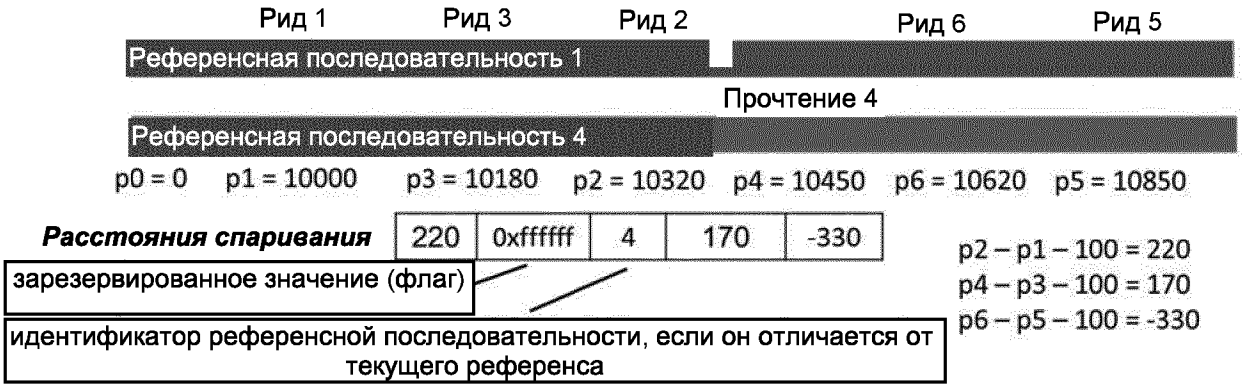
Опционально, слой pair может быть дифференциально закодирован в pair'.



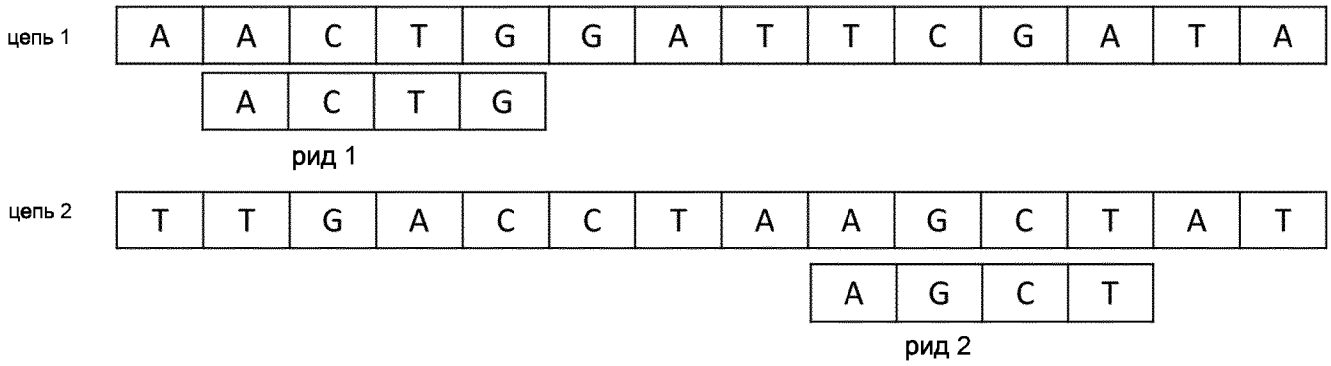
Фигура 6 - Расчет ошибок спаривания.

ПРИМЕР
Постоянная длина ридов = 100

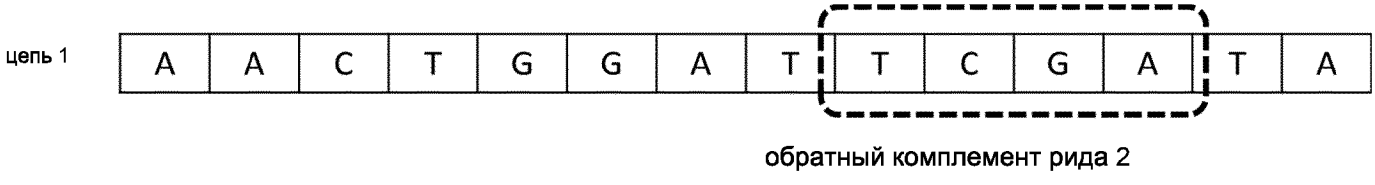
Пара 1 = Рид 1 + Рид 2
Пара 2 = Рид 3 + Рид 4
Пара 3 = Рид 5 + Рид 6



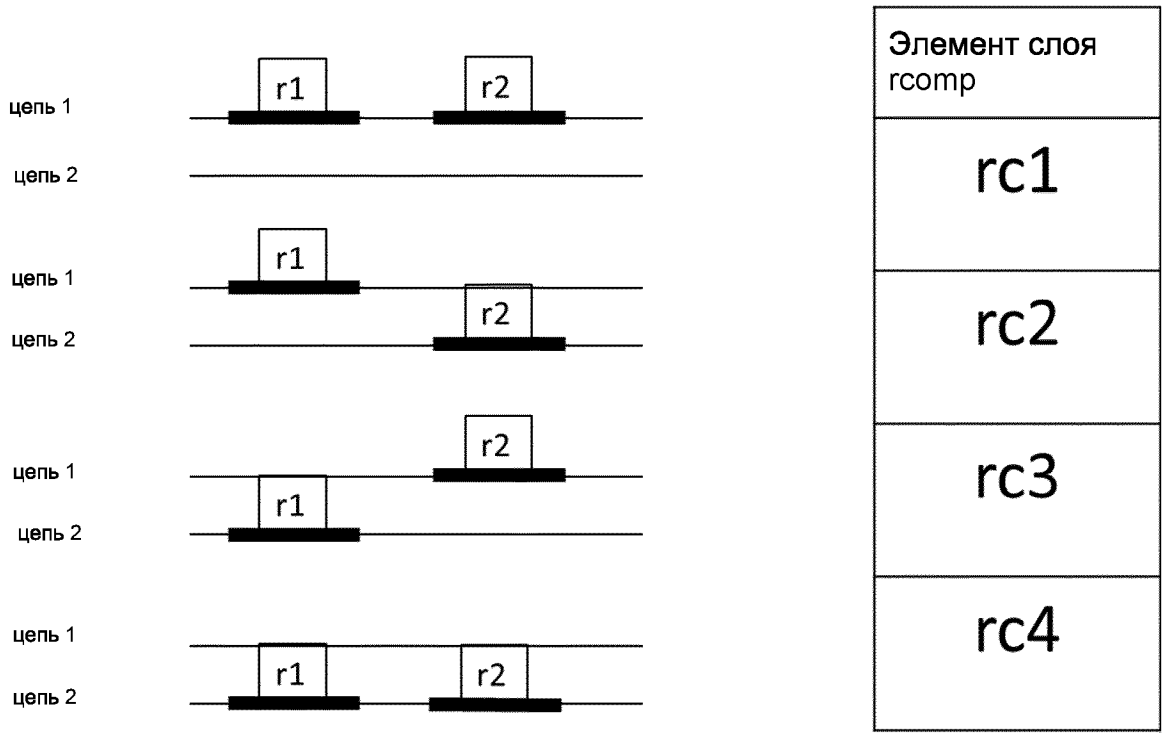
Фигура 7 - Когда Рид картируется на другой референс, нежели его пара Рид 4), к расстоянию спаривания добавляют дополнительные дескрипторы. Один дескриптор - это сигнальный флаг, второй - идентификатор референса, а затем - расстояние спаривания.



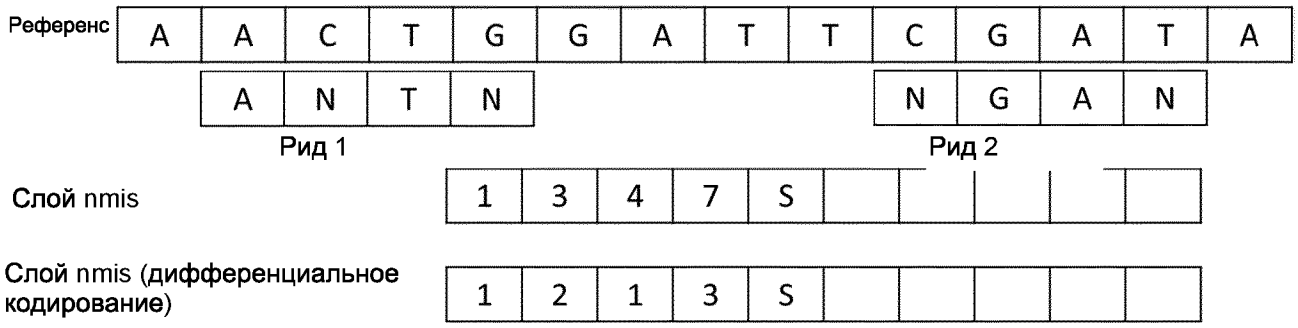
Фигура 8 - В этой паре прочтений рид 1 происходит из цепи 1, а рид 2 - из цепи 2.



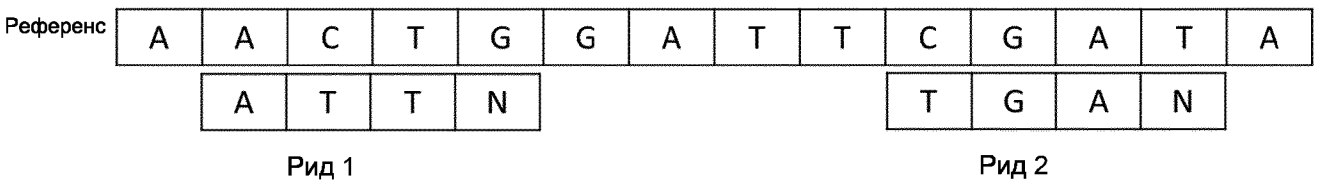
Фигура 9 - Обратный комплемент рида 2 будет закодирован, если в качестве референса используется цепь 1.



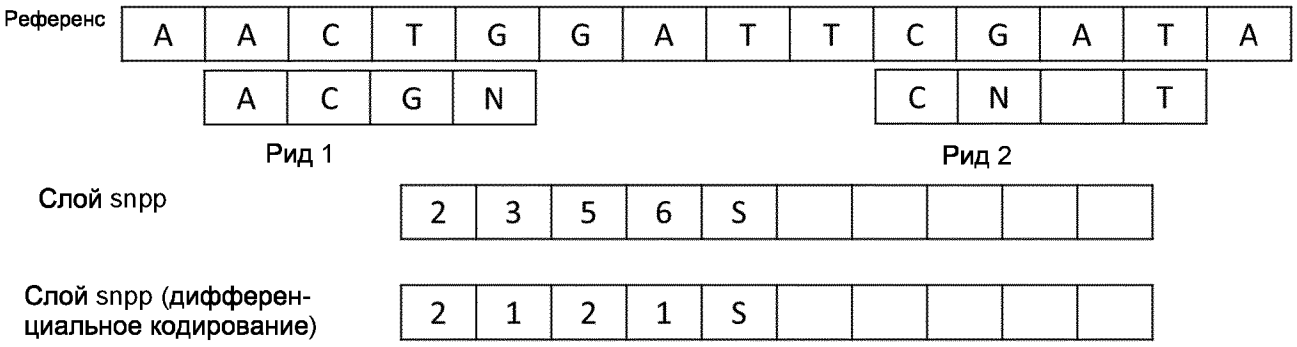
Фигура 10 - Четыре возможных комбинации ридов составляющих пару прочтений, и соответствующее кодирование в слое rcomp.



Фигура 11 - Расчет N-несоответствий в слое nmis.



Фигура 12 - Замены в картированной паре ридов



Фигура 13- Расчет положений замен в виде абсолютных и дифференциальных значений.

Слой snpt (без кодов IUPAC)

Тип замен рассчитывается как индекс вектора замен, составленного из всех возможных символов. Например:
S = [A, C, G, T, N, Z], **где Z = делеция**
Направление индекса

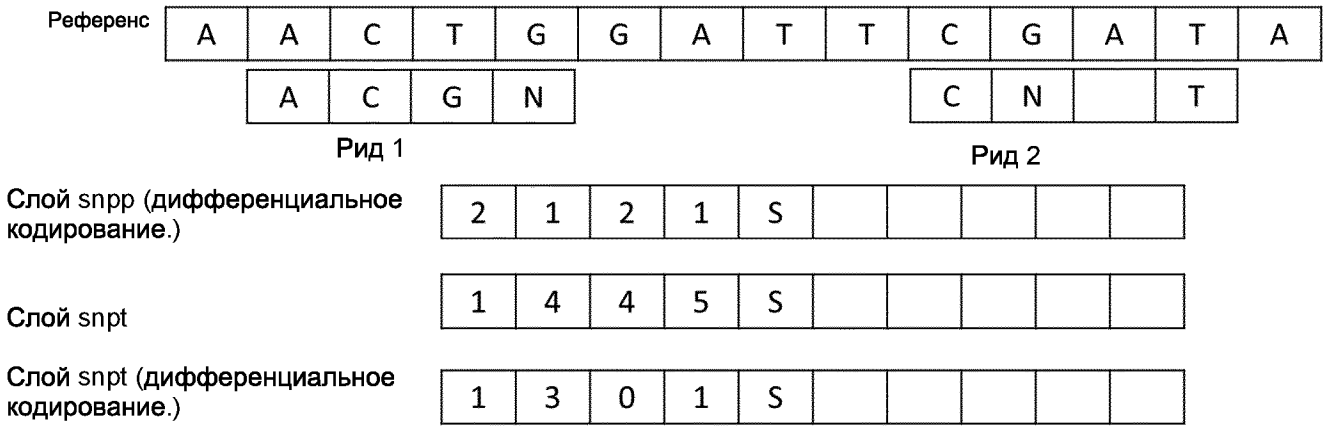
- КОДИРОВАНИЕ справа налево
- ДЕКОДИРОВАНИЕ слева направо

Референс	Рид	Закодированный символ
A	C	idx(A,C) = 1
A	G	idx(A,G) = 2
A	T	idx(A,T) = 3
A	N	idx(A,N) = 4
C	A	idx(C,A) = 5
C	G	idx(C,G) = 1
C	T	idx(C,T) = 2
C	N	idx(C,N) = 3
G	A	idx(G,A) = 4
G	C	idx(G,C) = 5
G	T	idx(G,T) = 1
G	N	idx(G,N) = 2
T	A	idx(T,A) = 3
T	C	idx(T,C) = 4
T	G	idx(T,G) = 5
T	N	idx(T,N) = 1

Референс	Рид	Закодированный символ
A	del.	idx(A,Z) = 5
C	del.	idx(C,Z) = 4
G	del.	idx(G,Z) = 3
T	del.	idx(T,Z) = 2

Референс	Рид	Закодированный символ
N	A	idx(N,A) = 2
N	C	idx(N,C) = 3
N	G	idx(N,G) = 4
N	T	idx(N,T) = 5

Фигура 14 - Расчеты символов, кодирующих замены без кодов IUPAC.



Фигура 15 - Кодирование замен в слой snpt.

Слой snpt (с кодами IUPAC)

Тип замен рассчитывается как индекс вектора замен, составленного из всех возможных символов. Например:
S = [A, C, G, T, N, Z, M, R, W, S, Y, K, V, H, D, B]
Направление индекса

- КОДИРОВАНИЕ справа налево
- ДЕКОДИРОВАНИЕ слева направо

Референс	Рид	Закодированный символ
N	M	idx(N,M) = 2
N	W	idx(N,W) = 4
N	S	idx(N,S) = 5
N	B	idx(N,B) = 11

Референс	Рид	Закодированный символ
D	M	idx(D,M) = 8
A	Y	idx(A,Y) = 10
A	T	idx(A,T) = 3
A	N	idx(A,N) = 4
C	R	idx(C,R) = 6
C	G	idx(C,G) = 1
C	T	idx(C,T) = 2
C	W	idx(C,W) = 7
G	H	idx(G,H) = 11
G	C	idx(G,C) = 15
G	B	idx(G,B) = 13
G	N	idx(G,N) = 2
T	A	idx(T,A) = 13
T	M	idx(T,M) = 4
T	K	idx(T,K) = 8
T	V	idx(T,V) = 9

Фигура 16 - Расчеты символов, кодирующих замены с кодами IUPAC.

Фигура 17 – Альтернативная исходная модель для замен, где кодируются только положения но используется один слой на каждый тип замены.

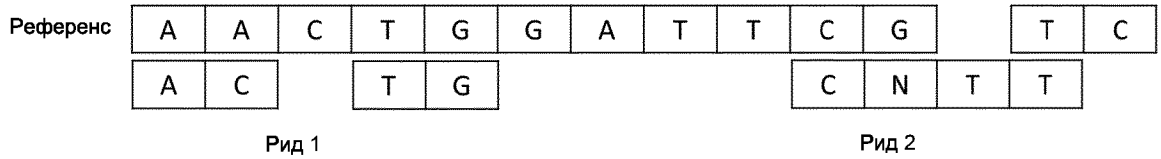
Слой indt (без кодов IUPAC)

S = [A, C, G, T, N, Z]

Направление

- КОДИРОВАНИЕ справа налево
- ДЕКОДИРОВАНИЕ слева направо

Инсерция	Закодированный символ
A	6
C	7
G	8
T	9
N	10



Слой indp (дифференциальное кодирование)	1	1	4	1	S					
Слой indt	5	2	4	9	S					
Слой indt (дифференциальное кодирование)	5	-3	2	5	S					

Фигура 18 - Кодирование замен, инсерций и делеций в паре ридов класса I, когда коды IUPAC не используются.

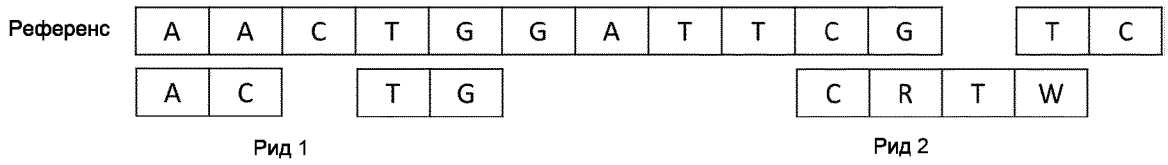
Слой indt (с кодами IUPAC)

S = [A, C, G, T, N, Z, M, R, W, S, Y, K, V, H, D, B]

Направление

- КОДИРОВАНИЕ справа налево
- ДЕКОДИРОВАНИЕ слева направо

Инсерция	Закодированный символ
A	16
C	17
G	18
T	19
N	20

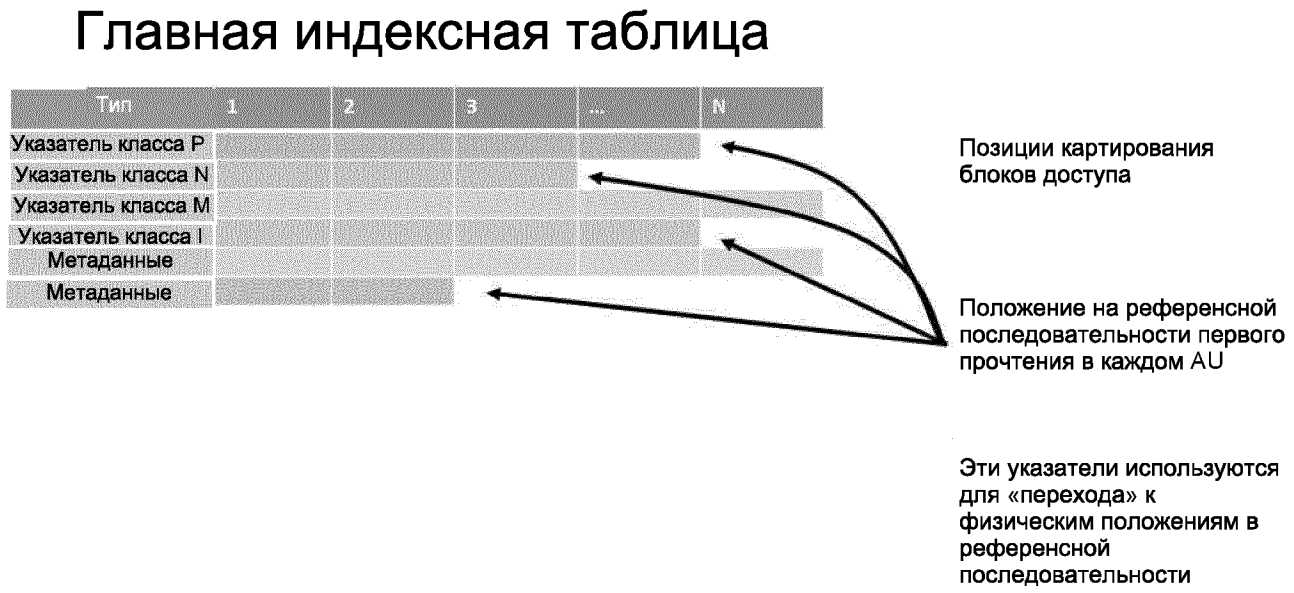


Слой snpp (дифференциальное кодирование.)	1	1	4	1	1	S				
Слой snpt	15	4	5	19	5	S				
Слой snpt (дифференциальное кодирование.)	15	-11	1	18	-14	S				

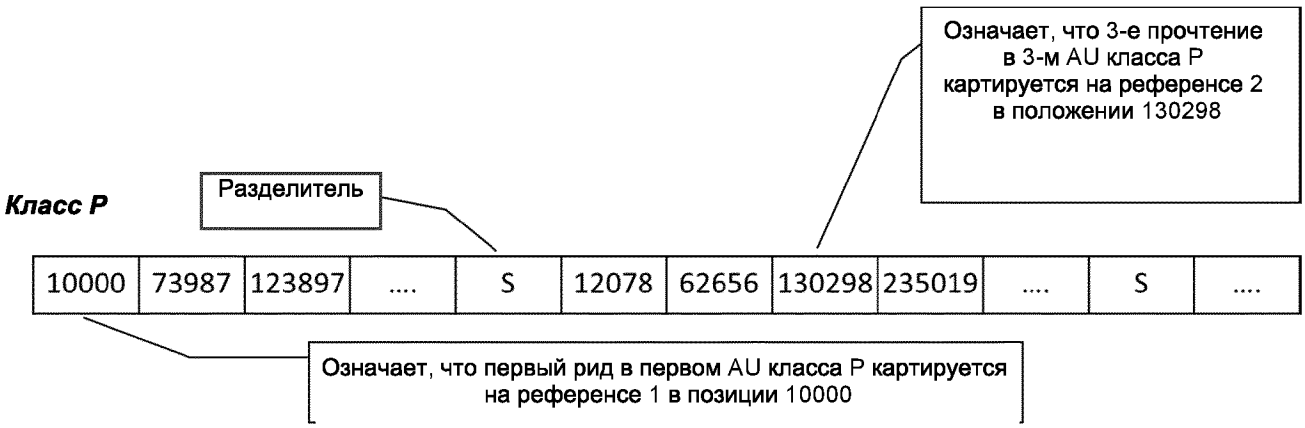
Фигура 19 - Кодирование замен, инсерций и делеций в паре ридов класса I, когда используются коды IUPAC.

Уникальный ID	Версия	Размер заголовка	Отсчёт референса	N блоки	Идентификаторы референса	Референсы	M.I.T.

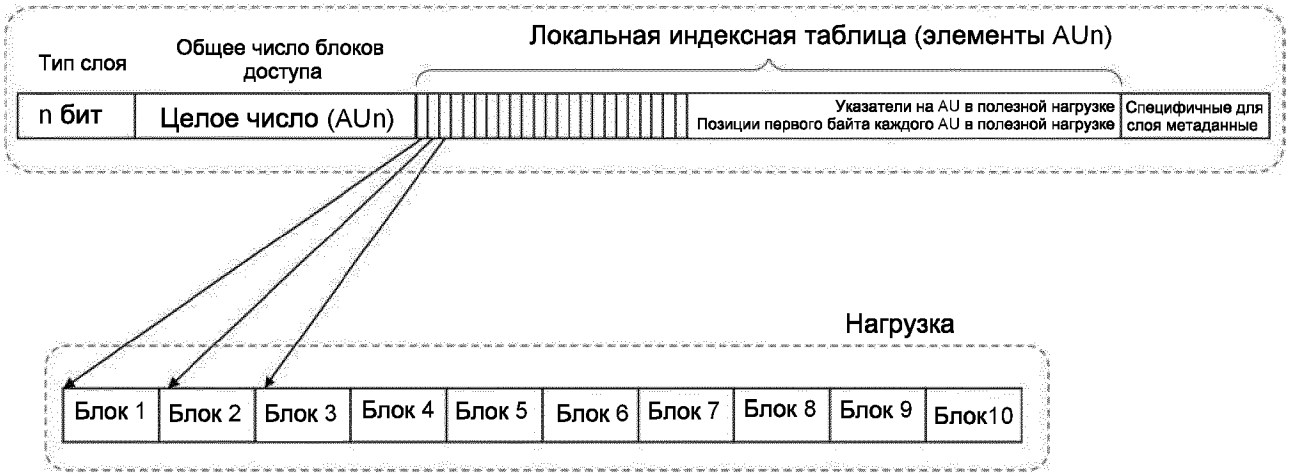
Фигура 20 – Заголовок закодированной геномной информации.



Фигура 21 –Главная индексная таблица содержит положения в референсных последовательностях первого прочтения в каждом AU.

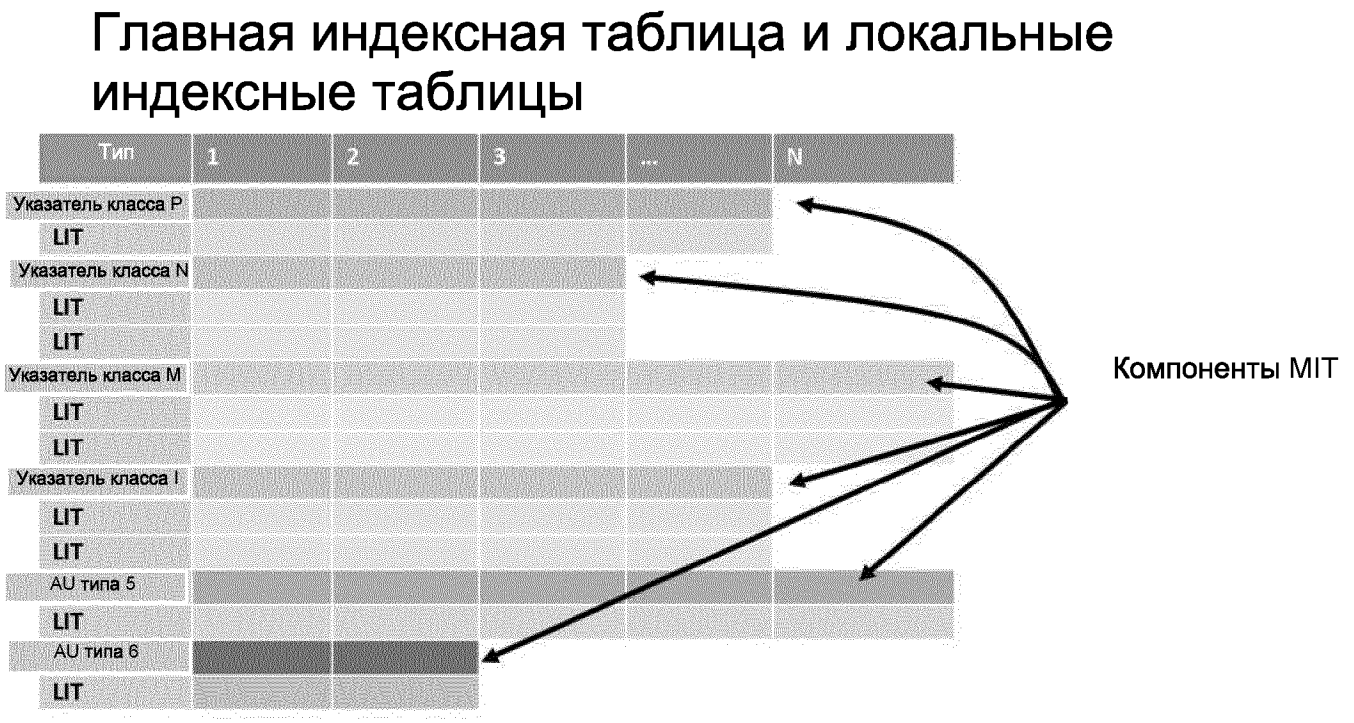


Фигура 22 – Пример частичной MIT, показывающий положения картирования первого прочтения в каждом AU положения класса P.

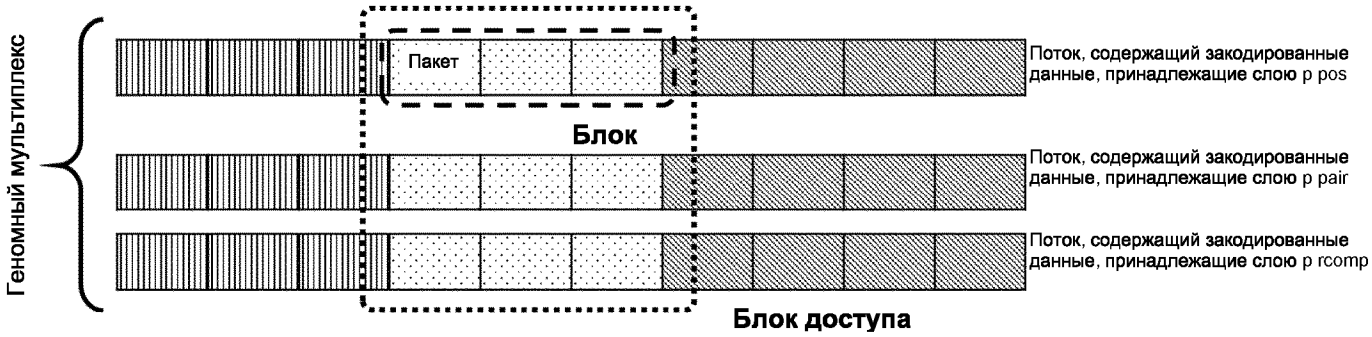


Фигура 23 – Локальная индексная таблица в заголовке слоя представляет собой вектор указателей на AUn в полезной нагрузке.

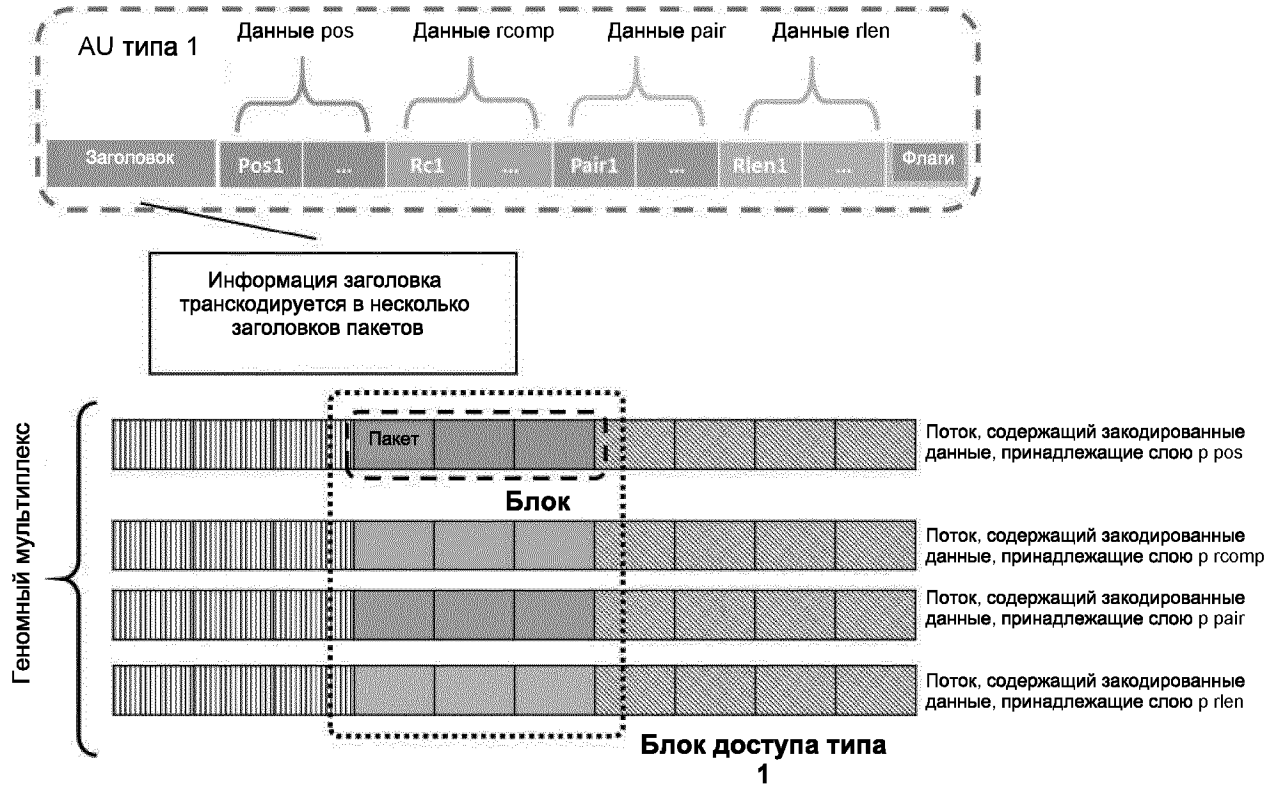
Фигура 24 – Пример локальной индексной таблицы.



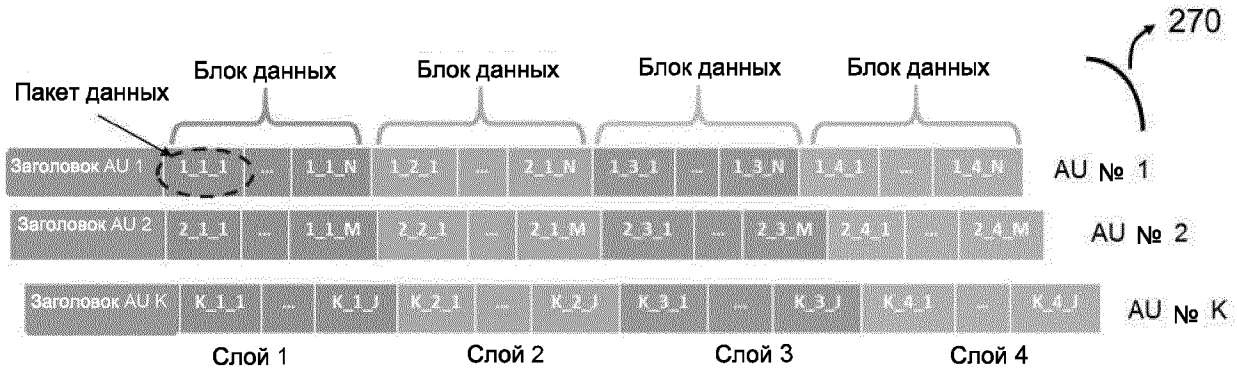
Фигура 25 – Функциональная связь между главной индексной таблицей и локальными индексными таблицами.



Фигура 26 – Блоки доступа состоят из блоков данных, принадлежащих нескольким уровням и проиндексированных с использованием механизмов MIT и LIT.

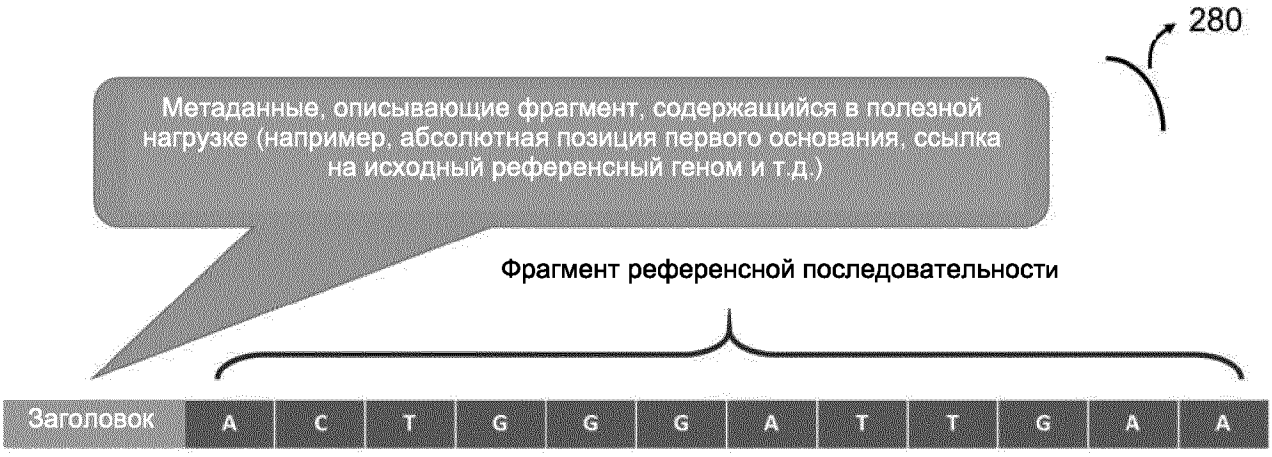


Фигура 27 - Блок доступа типа 1 упаковывается и транспортируется в мультиплексе геномных данных.

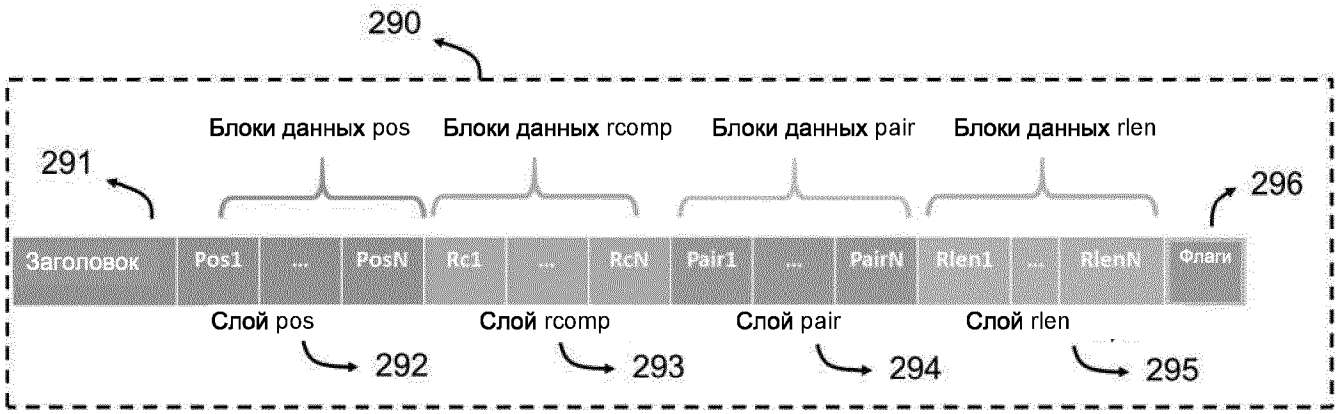


Фигура 28 - Блоки доступа состоят из заголовка и мультиплексированных блоков, принадлежащих одному или нескольким слоям однородных данных. Каждый блок может состоять из одного или нескольких пакетов, содержащих фактические дескрипторы геномной информации.

Тип 0
Блок доступа типа 0 состоит из заголовка и части референсной последовательности, используемой для выравнивания кодированных данных.
Он используется для кодирования прочтений как позиций и несовпадений по отношению к референсу



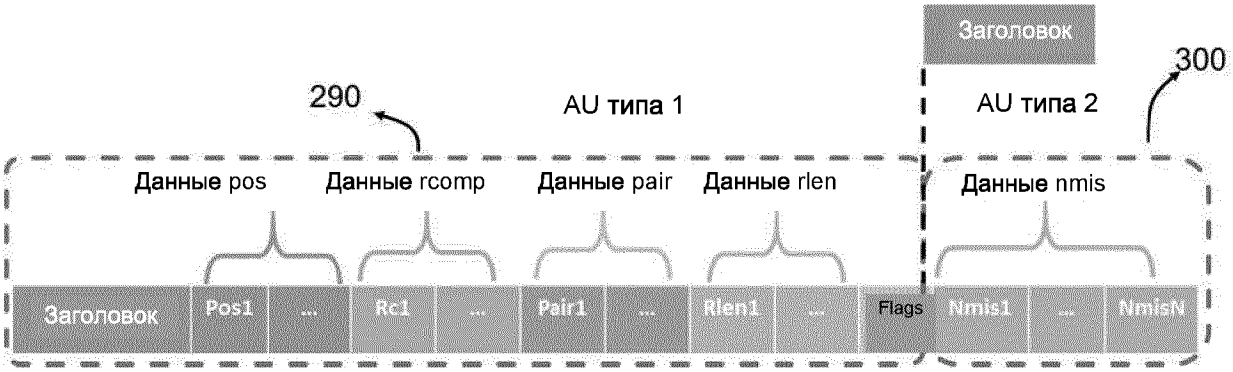
Фигура 29 - **Блоки доступа типа** 0 **не должны ссылаться на какую-либо информацию**, **поступающую от других блоков доступа**, **для доступа или декодирования и доступа**.



Фигура 30 – Блоки доступа типа 1 содержат данные, которые относятся к данным, переносимым блоками доступа типа 0.

Тип 2
Блок доступа типа 2 состоит из заголовка и мультиплексирования блоков данных типов pos, rcomp, pair (необязательный), rlen (необязательный) и nmis
Используется для кодирования ридов класса N

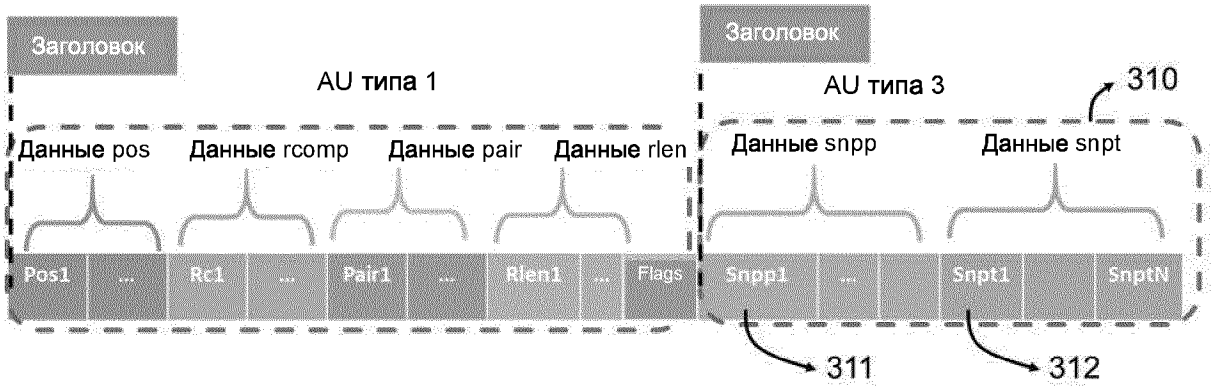
- Pos: положение рида на референсе (1 на каждый класс)
- Rcomp: флаг обратного комплемента (1 на каждый класс)
- Pair: (необязательный) информация о спаривании (1 на каждый класс)
- Rlen: (необязательный) длина рида в случае вариабельной длины рида



Фигура 31 - Блоки доступа типа 2 содержат данные, которые ссылаются на блок доступа типа 1. Это позиции N в закодированных ридах

Тип 3
Блок доступа типа 3 состоит из заголовка и мультиплексирования блоков данных типов **pos, rcomp, pair** (необязательный), **rlen** (необязательный), **snpp, snpt**
Используется для кодирования hbljd класса M

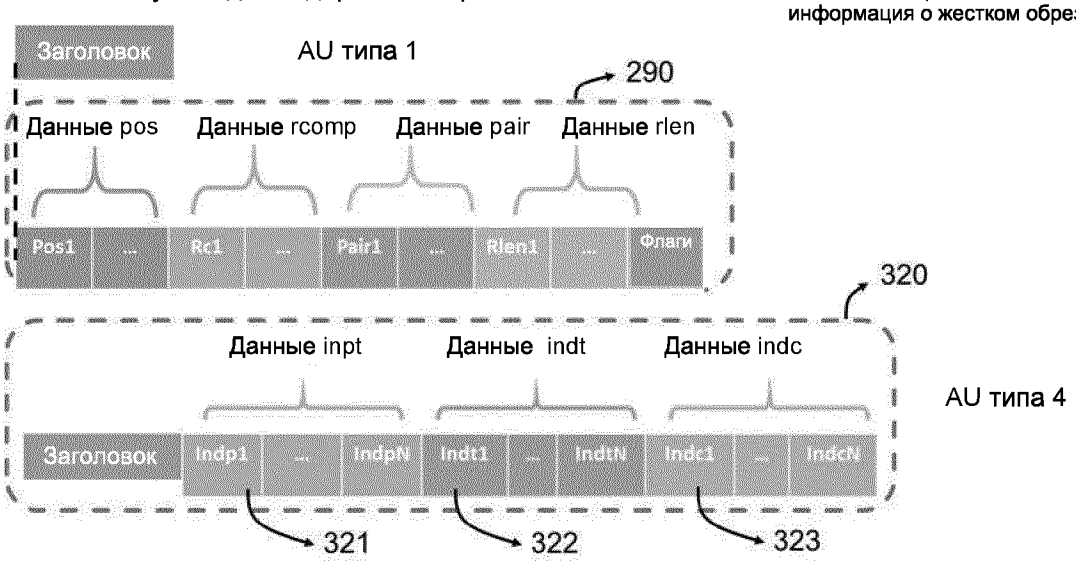
- Pos: позиция прочтения на референсе (1 на каждый класс)
- Rcomp: флаг обратного комплемента (1 на каждый класс)
- Pair: (необязательный) информация о спаривании (1 на каждый класс)
- Rlen: (необязательный) длина прочтения в случае вариабельной длины прочтения
- Snpp: позиции несовпадений
- Snpt: типа несовпадений



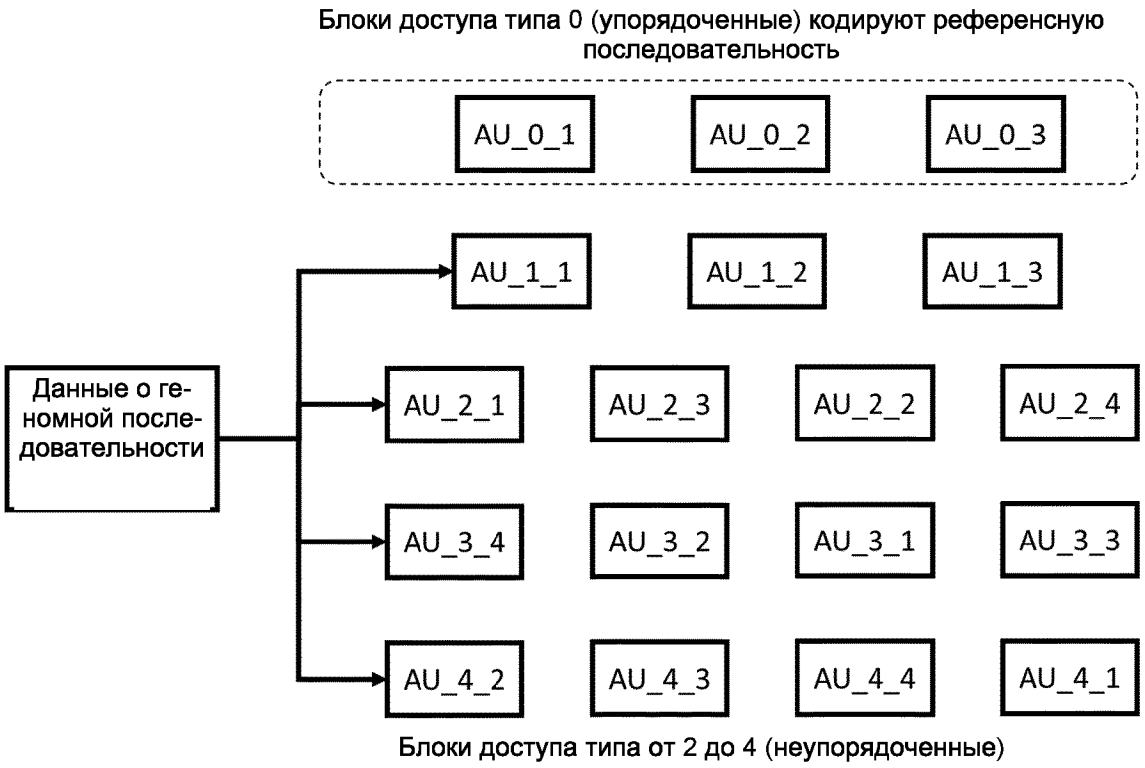
Фигура 32 - Блоки доступа типа 3 содержат данные, которые ссылаются на блок доступа типа 1. Это позиции и типы несовпадений в закодированных ридах

Тип 4
Блок доступа типа 4 состоит из заголовка и мультиплексирования блоков данных типа **ipos, ircomp, ipair** (необязательный), **irlen** (необязательный), **indp, inpt, indc**
Он используется для кодирования прочтений класса I

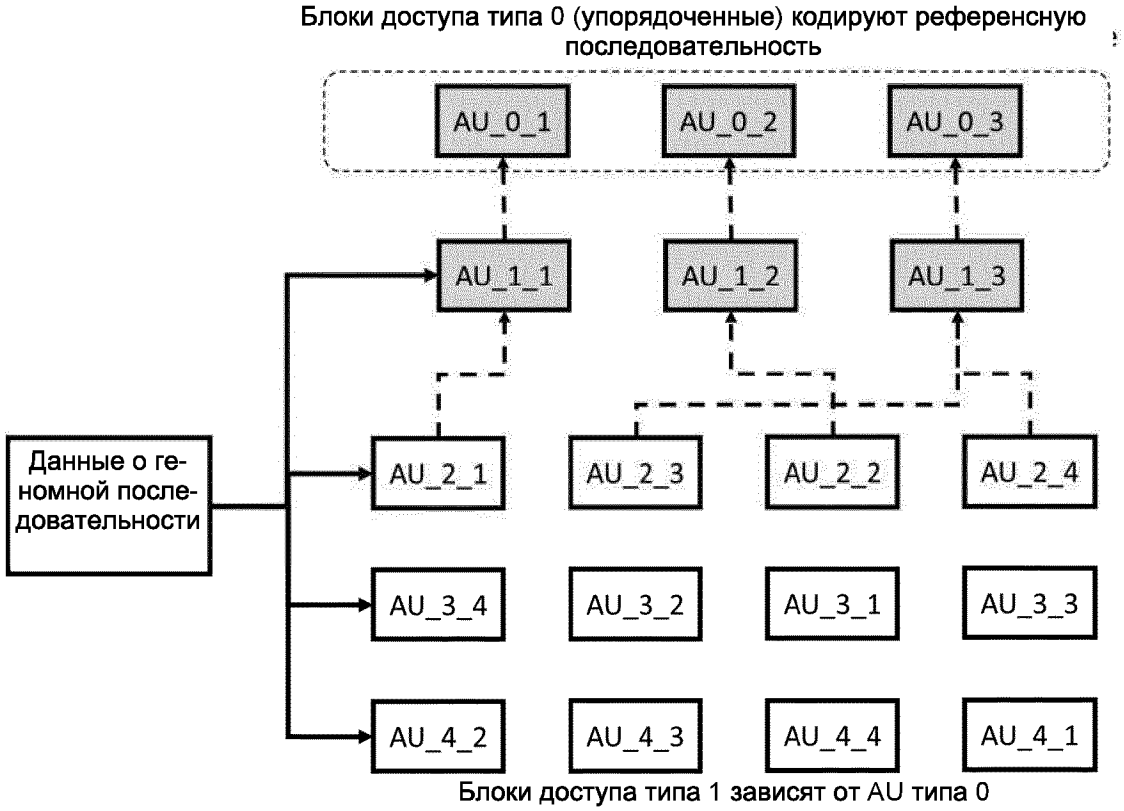
- Pos: положение рижа на референсе (1 на каждый класс)
- Rcomp: флаг обратного комплемента (1 на каждый класс)
- Pair: (необязательный) информация о спаривании (1 на каждый класс)
- Rlen: (необязательный) длина рида в случае вариабельной длины проридачтения
- Indp: позиции инделов и несовпадений
- Indt: инделы и несовпадения
- Indc: мягко обрезанные основания и информация о жестком обрезании



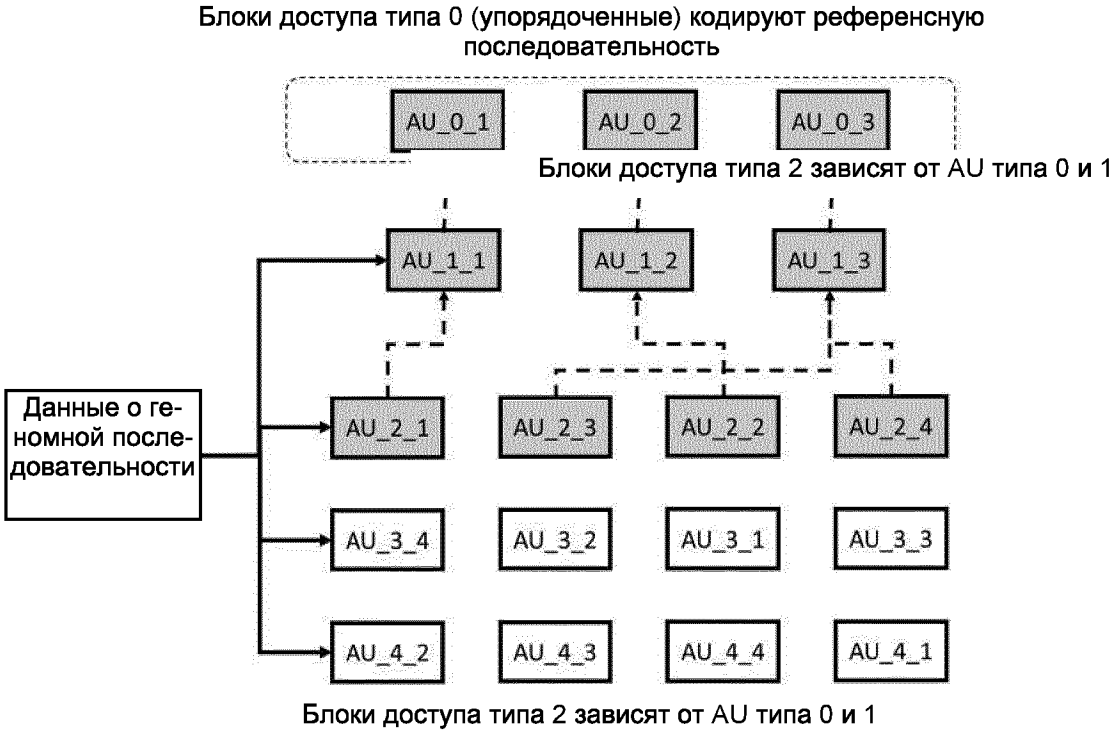
Фигура 33 - Блоки доступа типа 4 содержат данные, которые ссылаются на блок доступа типа 1. Это позиции и типы несовпадений в закодированных ридах



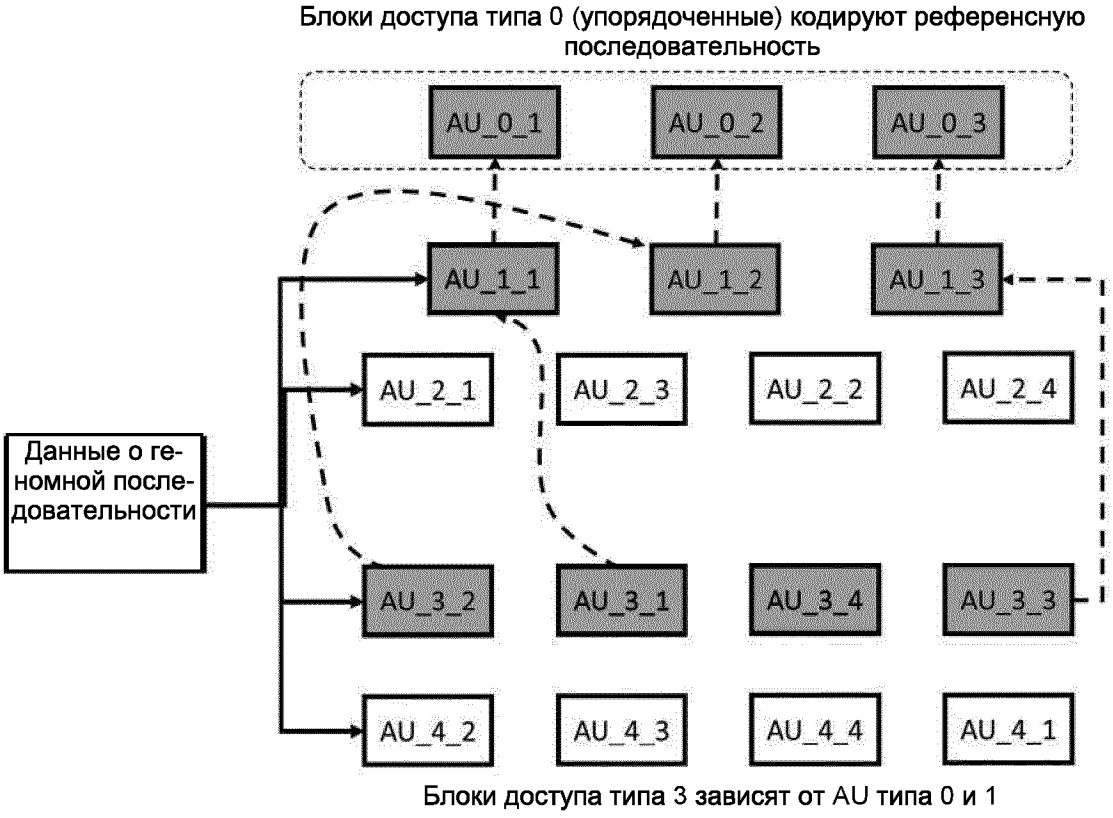
Фигура 34 - Первые пять типов блоков доступа.



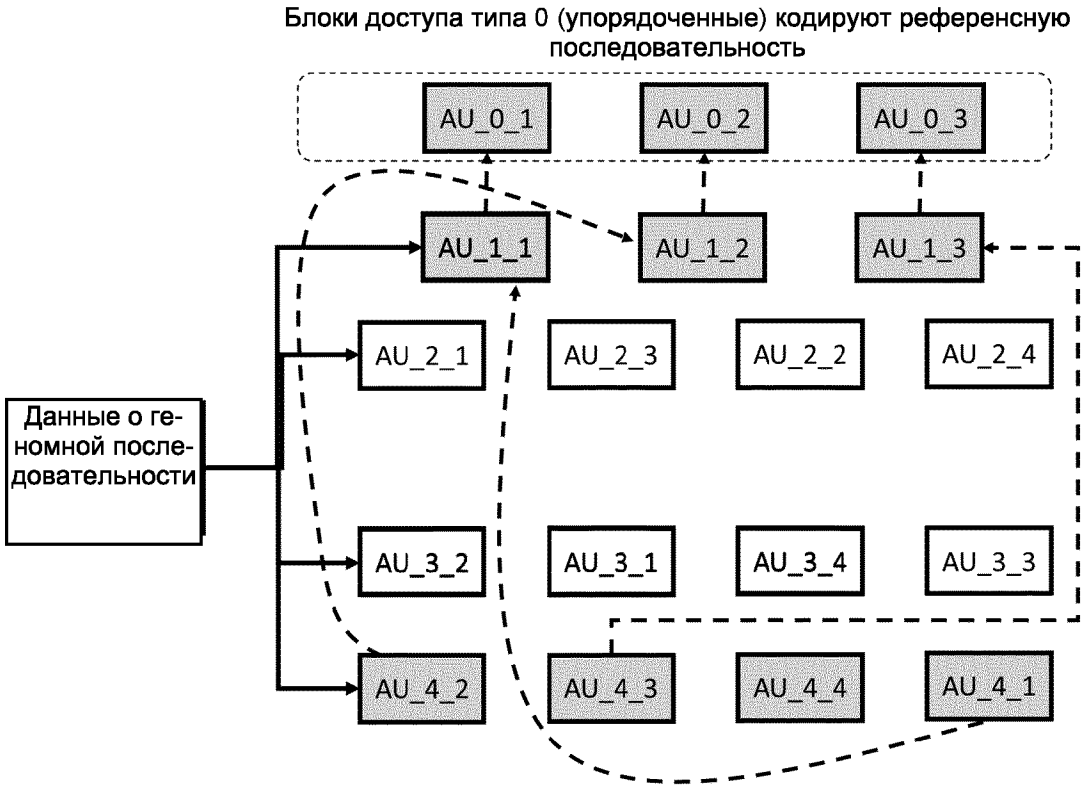
Фигура 35 - Блоки доступа типа 1 ссылаются на блоки доступа типа 0 для декодирования



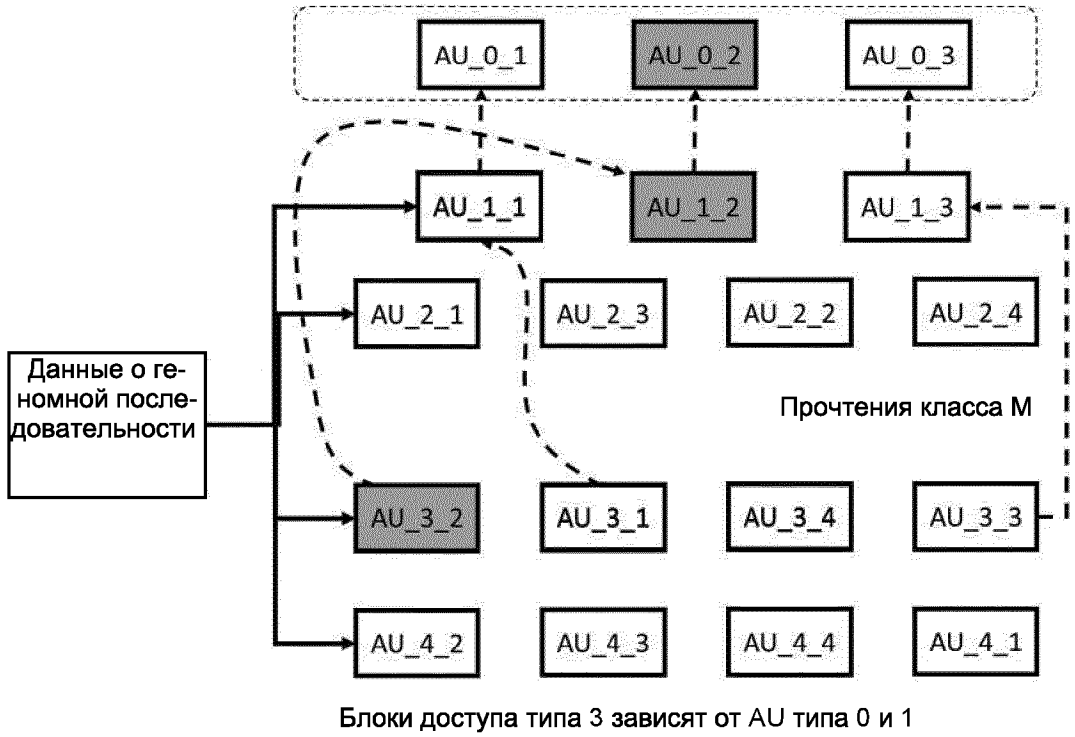
Фигура 36 - Блоки доступа типа 2 ссылаются на блоки доступа типа 0 и 1, для декодирования



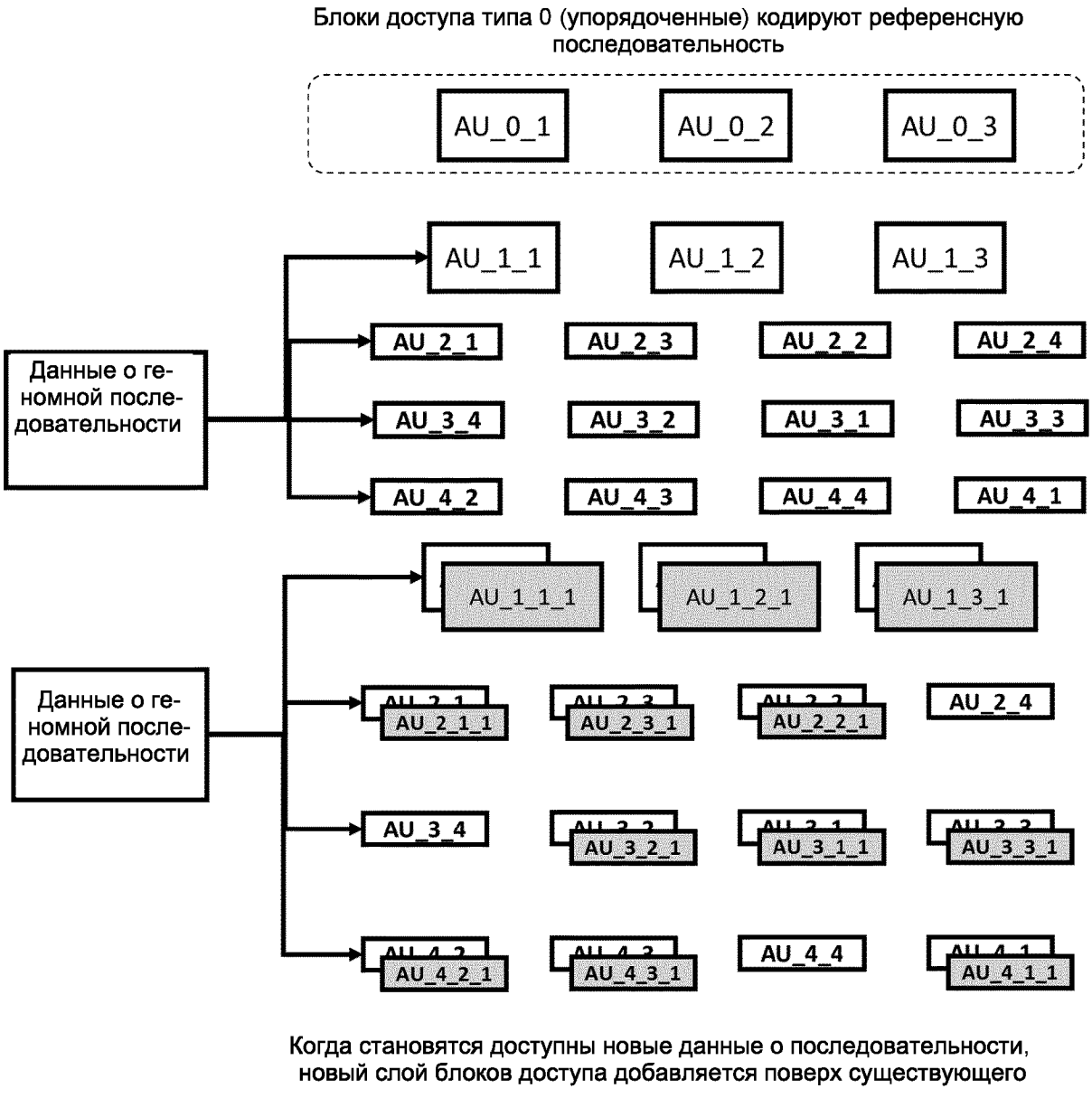
Фигура 37 - Блоки доступа типа 3 ссылаются на блоки доступа типа 0 и 1 для декодирования



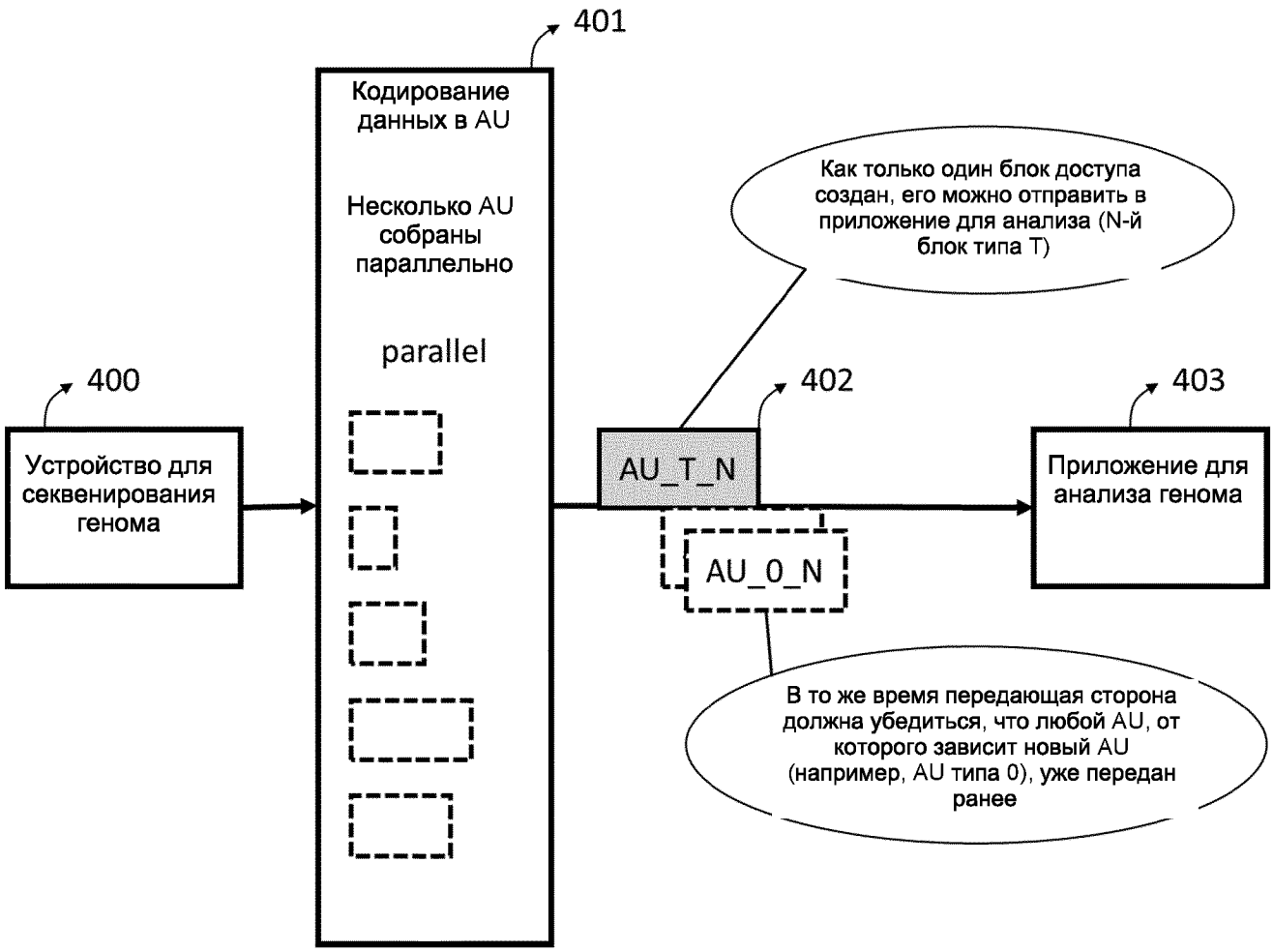
Фигура 38 - Блоки доступа типа 4 ссылаются на AU типа 1 для кодирования ридов класса I.



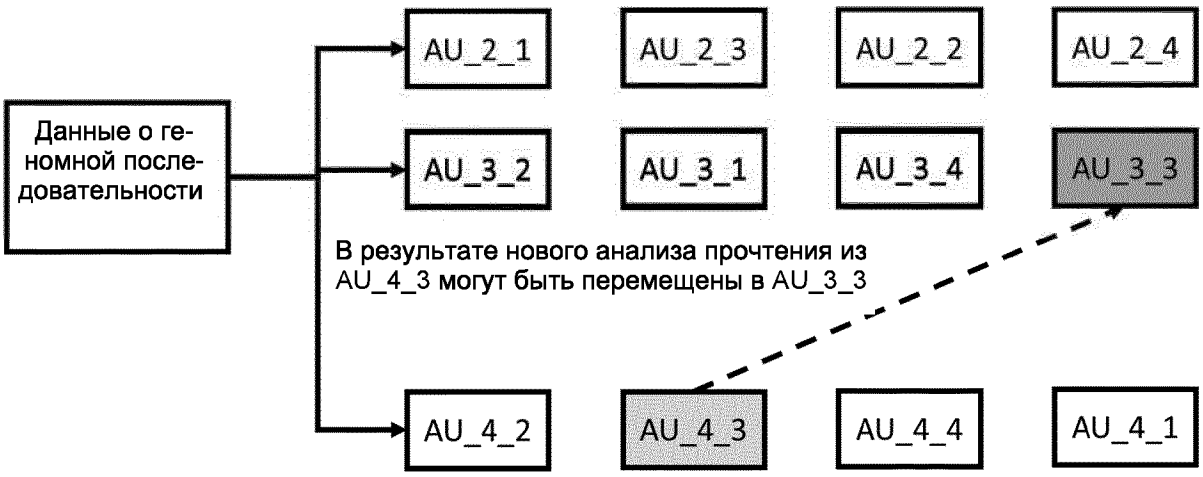
Фигура 39 – Блоки доступа, необходимые для декодирования ридов последовательностей с несовпадениями, картированных во втором сегменте референсной последовательности (AU 0-2).



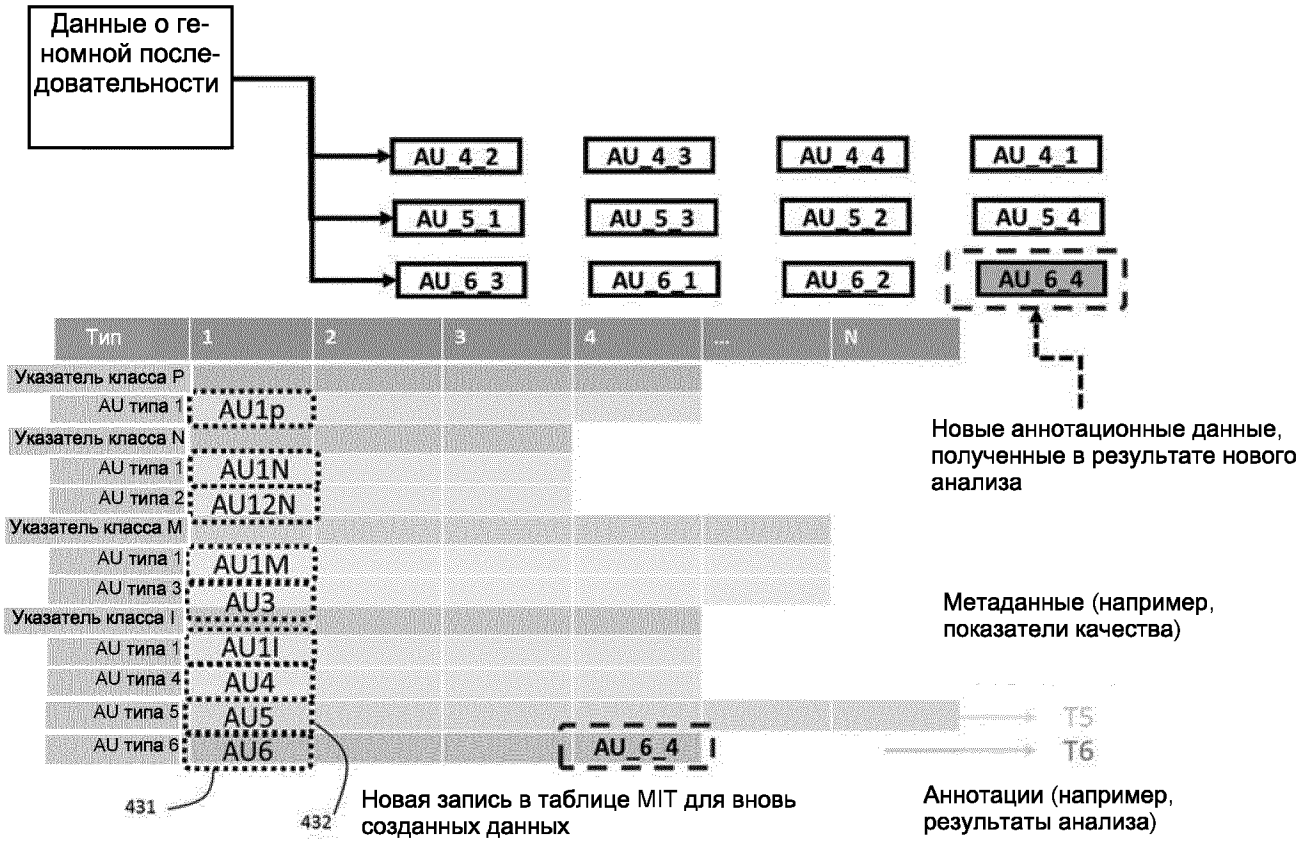
Фигура 40 - Когда становятся доступны новые данные, поверх существующих данных строится новый слой блоков доступа.



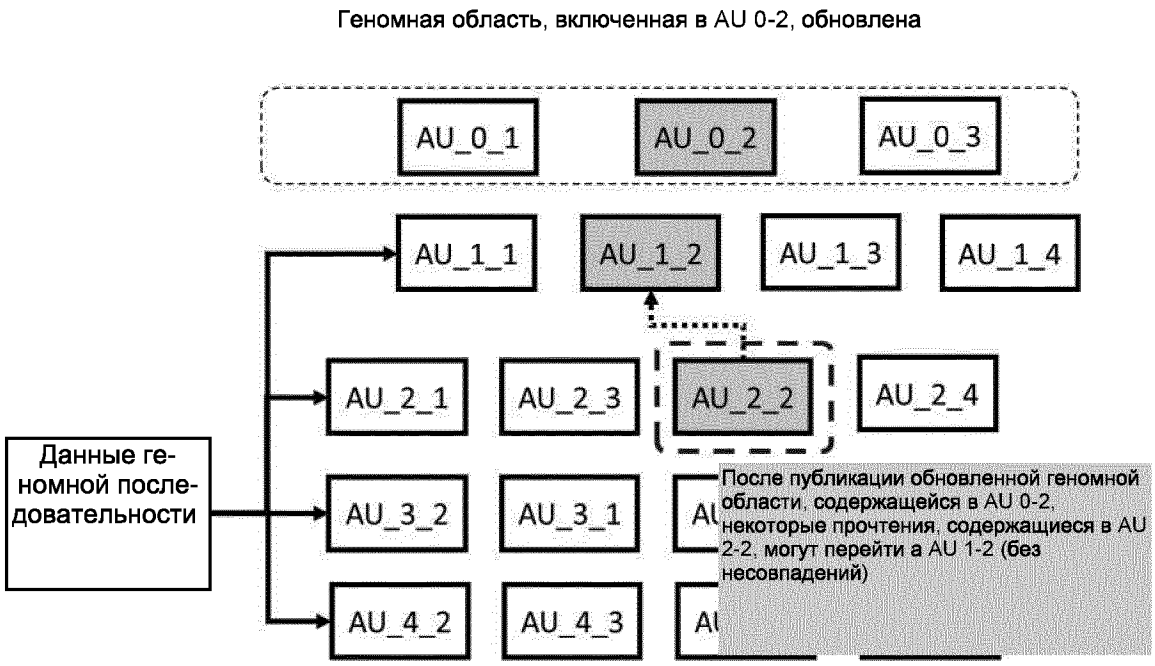
Фигура 41 - Структура данных, основанная на е блок ах доступа, позволяет начать анализ геномных данных до завершения всего процесса секвенирования.



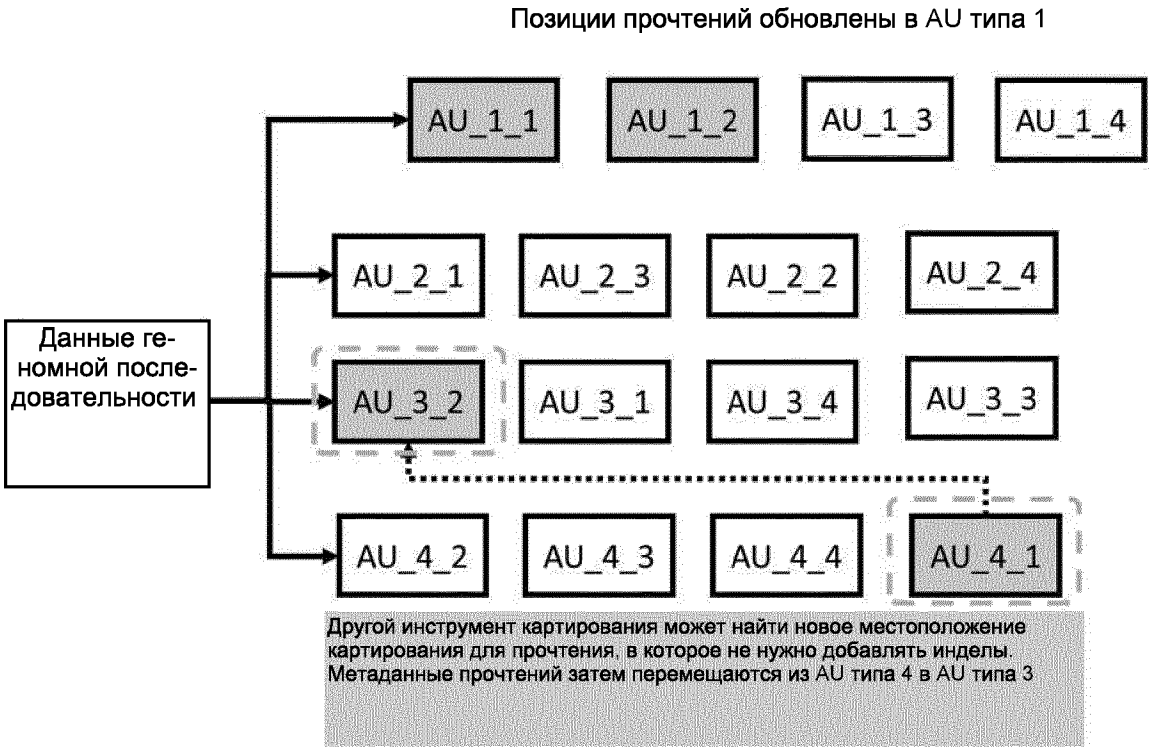
Фигура 42 - Новый анализ, выполненный на уже существующих данных, может потребовать перемещения прочтений из AU типа 4 в AU типа 3



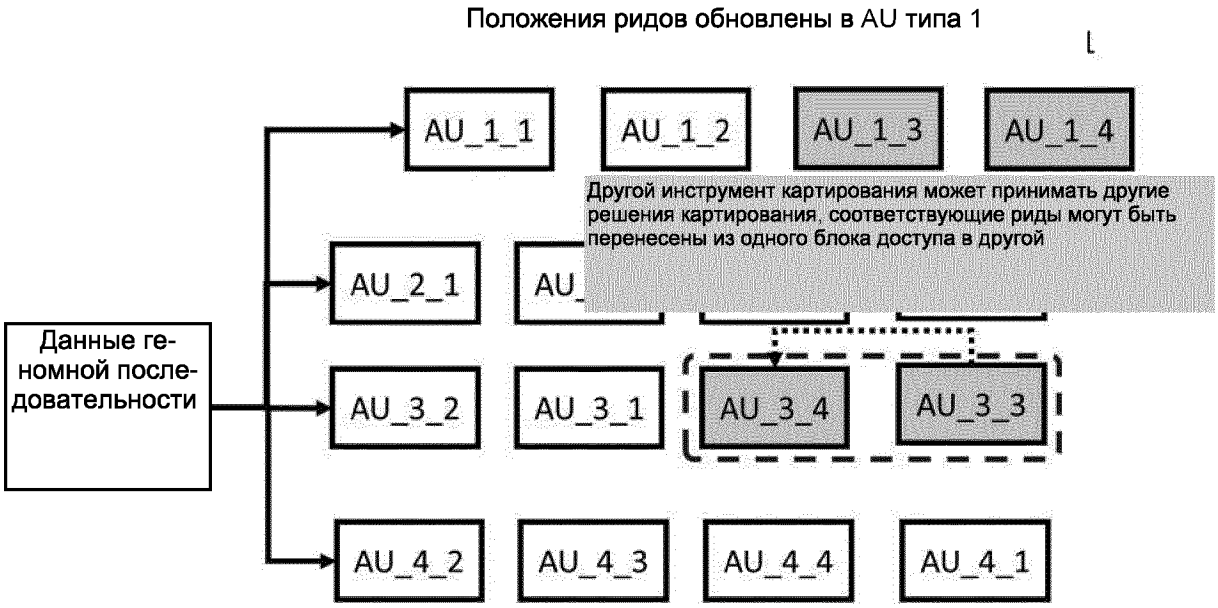
Фигура 43 - Вновь сгенерированные данные анализа инкапсулируются в новый AU типа 6, и в MIT создается соответствующий индекс



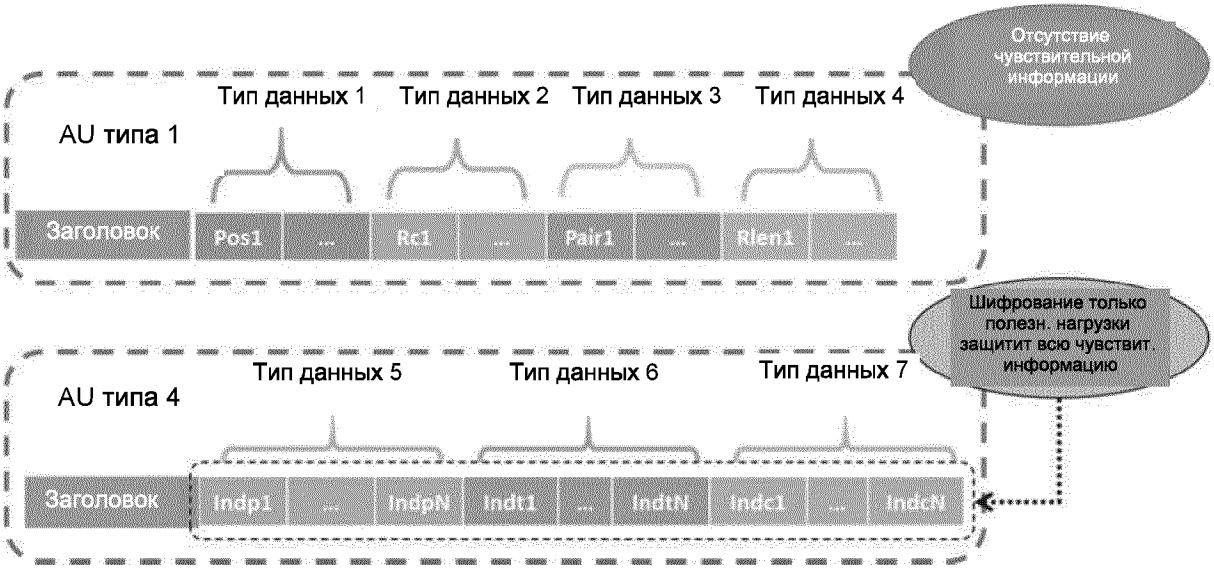
Фигура 44 – Транскодирование, вызванное публикацией новой референсной последовательности (генома)



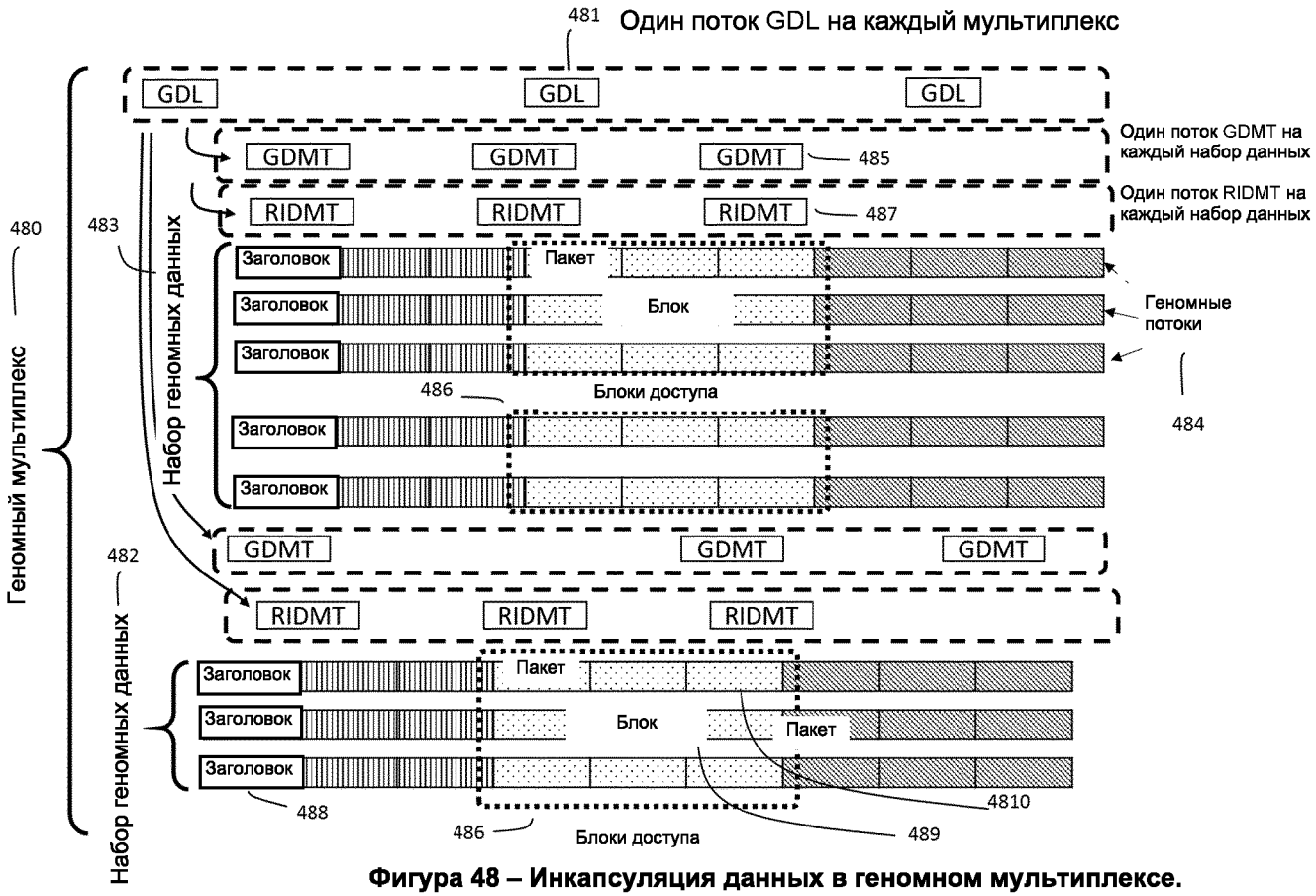
Фигура 45 - Риды картированные с новым геномным регионом с лучшим качеством (например, без инделов), перемещаются из AU типа 4 в AU типа 3



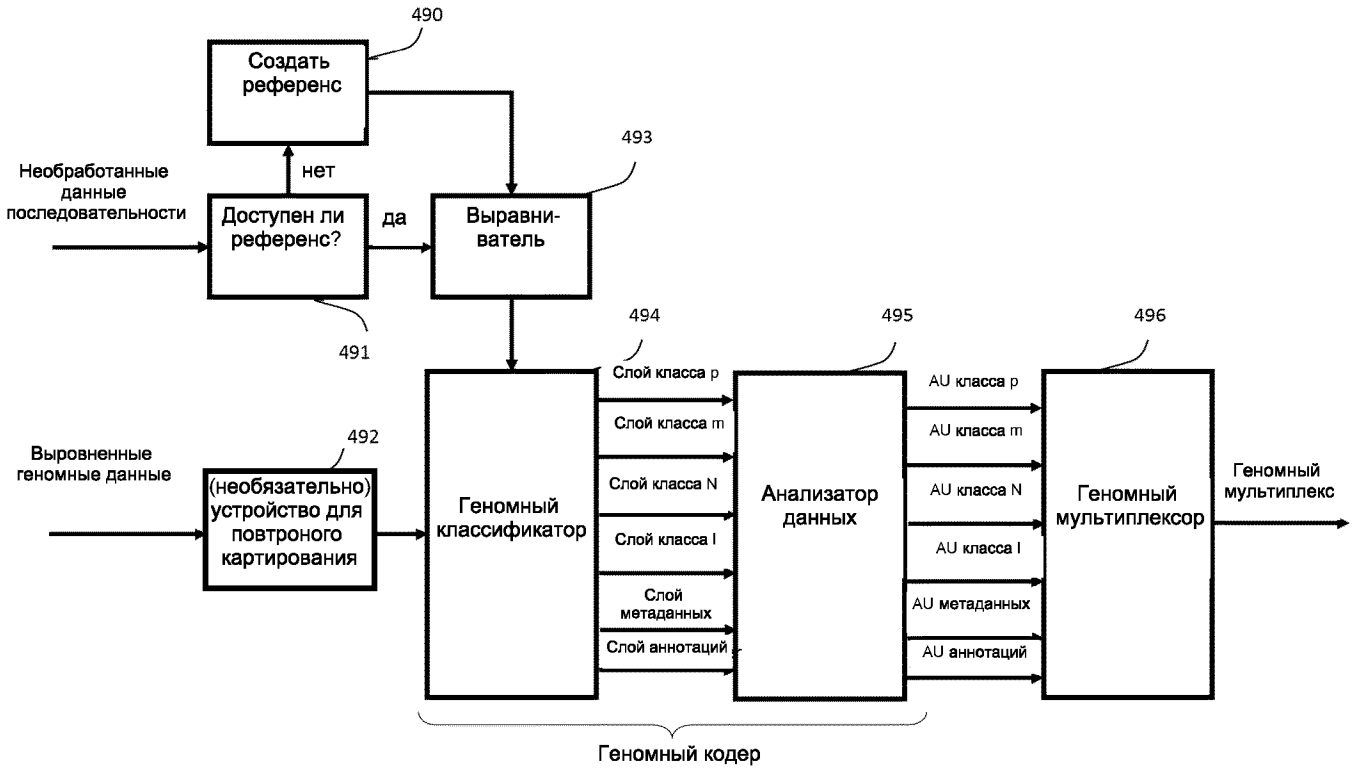
Фигура 46 – В случае обнаружения нового местоположения картирования (например, с меньшим количеством несовпадений) соответствующие прочтения могут быть перемещены из одного AU в другой того же типа.



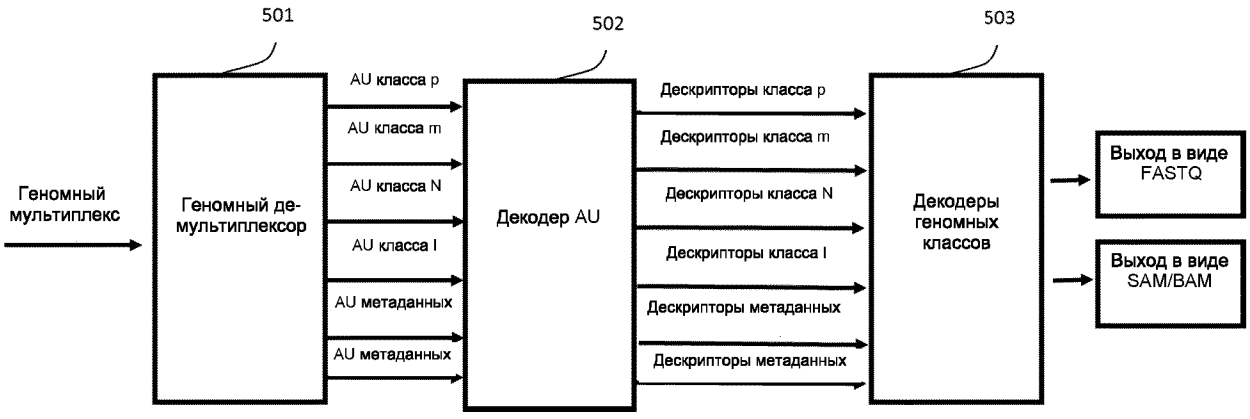
Фигура 47 – Избирательное шифрование может применяться только к блокам доступа типа 4, если они содержат важную информацию, подлежащую защите.



Фигура 48 – Инкапсуляция данных в геномном мультиплексе.



Фигура 49 - Геномный кодер представляет собой первый этап классификации данных и второй этап синтаксического анализа и кодирования данных. Ему может предшествовать выравнивание по внешней или внутренней сгенерированной референсной последовательности. Геномный кодер может использовать выравниватели и сборщики de-novo для подготовки данных к кодированию.



Фигура 50 – Геномный демультиплексор извлекает слои блоков доступа из геномного мультиплекса, один декодер для каждого типа AU извлекает геномные дескрипторы, которые затем декодируются в различные геномные форматы, такие как, например, FASTQ и SAM/BAM.