(19) Евразийское патентное ведомство

(21) 201991908 (13) A1

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ЕВРАЗИЙСКОЙ ЗАЯВКЕ

(43) Дата публикации заявки
2020.01.21

(22) Дата подачи заявки
2018.02.14

(51) Int. Cl. ***G06F 7/00*** (2006.01)

(54) СПОСОБ И УСТРОЙСТВО ДЛЯ КОМПАКТНОГО ПРЕДСТАВЛЕНИЯ БИОИНФОРМАЦИОННЫХ ДАННЫХ С ПОМОЩЬЮ НЕСКОЛЬКИХ ГЕНОМНЫХ ДЕСКРИПТОРОВ

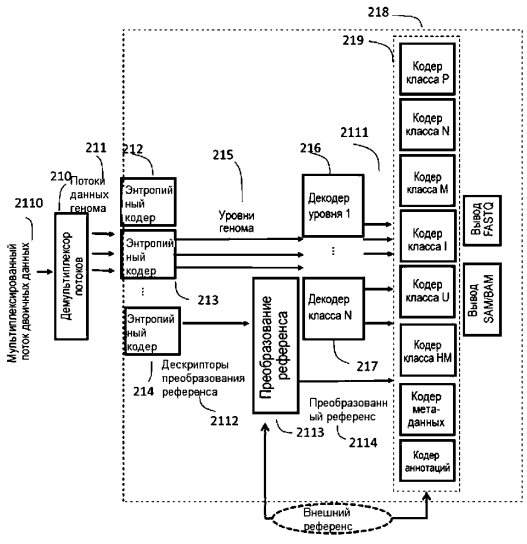
(31) **PCT/US2017/017842; PCT/US2017/041591**
(32) **2017.02.14; 2017.07.11**
(33) **US**
(86) **PCT/US2018/018092**
(87) **WO 2018/152143 2018.08.23**

(71) Заявитель:
ГЕНОМСЫС СА (CH)

(72) Изобретатель:
Алберти Клаудио, Зоиа Гиоргио, Рензи Даниэле (CH), Балуч Мохамед Хосо (US)

(74) Представитель:
Пронин В.О. (RU)

(57) Способ и устройство для сжатия данных геномной последовательности, созданных секвенаторами генома. Прочтения последовательности кодируют путем выравнивания их относительно ранее существующих или построенных референсных последовательностей, причем процесс кодирования состоит из классифицирования прочтений в классы данных с последующим кодированием каждого класса посредством множества блоков дескрипторов. Для каждого класса данных, на которые разбивают данные, и каждого соответствующего блока дескрипторов используют специальные модели источников и энтропийные кодеры.



201991908 A1

201991908 A1

СПОСОБ И УСТРОЙСТВО ДЛЯ КОМПАКТНОГО ПРЕДСТАВЛЕНИЯ БИОИНФОРМАЦИОННЫХ ДАННЫХ С ПОМОЩЬЮ НЕСКОЛЬКИХ ГЕНОМНЫХ ДЕСКРИПТОРОВ

ПЕРЕКРЕСТНАЯ ССЫЛКА НА РОДСТВЕННЫЕ ЗАЯВКИ

Настоящая заявка испрашивает приоритет и преимущество следующих заявок на патент: PCT/US2017/017842, поданной 14 февраля 2017 г., и PCT/US2017/041591, поданной 11 июля 2017 г.

ОБЛАСТЬ ТЕХНИКИ

В соответствии с настоящим изобретением предлагается новый способ представления данных секвенирования генома, который сокращает используемый объем памяти и улучшает характеристики доступа за счет обеспечения новых функциональных возможностей, которые недоступны в способах представления известного уровня техники.

УРОВЕНЬ ТЕХНИКИ

Надлежащая форма представления данных секвенирования генома является основополагающей для обеспечения эффективных приложений геномного анализа, таких как определение вариантов генома и все другие анализы, выполняемые в различных целях путем обработки данных секвенирования и метаданных.

Секвенирование генома человека стало доступным после появления недорогих технологий секвенирования с высокой пропускной способностью. Такая возможность открывает новые перспективы в нескольких областях, от диагностики и лечения рака до выявления генетических заболеваний, от патогенетического надзора за идентификацией антител до создания новых вакцин, лекарственных препаратов и адаптации персонализированных лечебных процедур.

Больницы, поставщики услуг по анализу геномных данных, специалисты в области биоинформатики и крупные центры хранения биологических данных ищут недорогие, быстрые, надежные и взаимосвязанные решения для обработки геномной информации, которые позволили бы распространить геномную медицину в мировом масштабе. Поскольку одним из узких мест в процессе секвенирования стало хранение данных, все больше внимания уделяется способам представления данных секвенирования генома в сжатой форме.

Наиболее используемые представления геномной информации данных секвенирования основываются на форматах архивирования FASTQ и SAM. Цель состоит в сжатии традиционно используемых форматов файлов (соответственно, FASTQ и SAM для невыровненных и выровненных данных). Такие файлы составляют из символов обычного текста и сжимают, как упоминалось выше, с использованием подходов общего назначения, таких как схемы LZ (от Lempel и Ziv, фамилий авторов, опубликовавших первые версии), например, широко известные zip, gzip и т. д. При использовании средств сжатия общего назначения, таких как gzip, результатом сжатия обычно бывает большой двоичный объект. Получаемую в результате информацию в такой монолитной форме довольно трудно архивировать, передавать и обрабатывать, в частности, как в случае высокопроизводительного секвенирования, когда объем данных чрезвычайно велик. Формат BAM отличается низкими характеристиками сжатия, так как он сосредоточен на сжатии неэффективного и избыточного формата SAM, а не на выделении фактической геномной информации, содержащейся в файлах SAM, и адаптирует алгоритмы сжатия текста общего назначения, такие как gzip, вместо использования специфического характера каждого источника данных (самих геномных данных).

Более сложным подходом к сжатию геномных данных, который применяется реже, но более эффективен по сравнению с BAM, является CRAM. CRAM обеспечивает более эффективное сжатие за счет внедрения дифференциального кодирования относительно референса (в частности, использования избыточности источника данных), но ему все же недостает функций, таких как инкрементальные обновления, поддержка потоковой обработки данных и выборочного доступа к конкретным классам сжатых данных.

Эти подходы обеспечивают плохие коэффициенты сжатия и структуры данных, которые сложны для перемещения по ним и манипулирования после сжатия. Последующий анализ может быть сильно замедлен ввиду необходимости обработки больших и жестко заданных структур данных даже для выполнения простой операции или получения доступа к выбранным областям геномного набора данных. CRAM опирается на концепцию записи CRAM. Каждая запись CRAM представляет одно картированное или некартированное прочтение путем кодирования всех элементов, необходимых для его восстановления.

CRAM демонстрирует следующие ограничения и недостатки, которые решаются и преодолеваются изобретением, описанным в данном документе.

1. CRAM не поддерживает индексацию данных и произвольный доступ к подмножествам данных с общими специфическими признаками. Индексация данных выходит за рамки данной спецификации (см. раздел 12 Спецификации CRAM v3.0) и реализуется в виде отдельного файла. Напротив, в подходе согласно изобретению, описанному в данном документе, используется способ индексации данных, который интегрирован с процессом кодирования, и индексы встраиваются в закодированный (то есть, сжатый) двоичный поток.

2. CRAM строят из блоков основных данных, которые могут содержать картированные прочтения любого типа (идеально совпадающие прочтения, прочтения только с заменами, прочтения с инсерциями или делециями (называемыми также «инделами»)). Не существует понимания классификации и группирования прочтений в классы в соответствии с результатом картирования относительно референсной последовательности. То есть, проверять нужно все данные, даже в случае поиска прочтений с определенными характеристиками. Такое ограничение снимается с помощью данного изобретения за счет классифицирования и разбиения данных на классы перед кодированием.

3. CRAM основан на принципе инкапсуляции каждого прочтения в «запись CRAM». Это означает, что при поиске прочтений, отличающихся определенными биологическими характеристиками (например, прочтений с заменами, но без «инделов», или идеально картированных прочтений) нужно проверять каждую полную «запись».

В отличие от этого, в настоящем изобретении существует понятие классов данных, кодируемых в отдельных информационных блоках, и нет понятия записи, инкапсулирующей каждое прочтения. Это позволяет эффективнее получать доступ к наборам прочтений с определенными биологическими характеристиками (например, к прочтениям с заменами, но без «инделов», или к идеально картированным прочтениям) без необходимости декодирования каждого прочтения (блока прочтений) для проверки его характеристик.

4. В записи CRAM каждое поле записи связано с определенным флагом, и каждый флаг должен всегда иметь одно и то же значение ввиду отсутствия понятия контекста, так как каждая запись CRAM может содержать данные какого-либо другого типа. Данный механизм кодирования вносит избыточную информацию и препятствует использованию эффективного контекстного энтропийного кодирования.

Вместо этого в настоящем изобретении не существует понятия флага, обозначающего данные, так как это по сути определено информационным «блоком», которому

принадлежат эти данные. Отсюда значительное сокращение символов, которые нужно использовать, и, как следствие, снижение энтропии источника информации, что приводит к более эффективному сжатию. Такое улучшение возможно за счет использования различных «блоков», позволяющих кодеру повторно использовать один и тот же символ в пределах каждого блока в разных значениях в соответствии с контекстом. В CRAM каждый флаг должен всегда иметь одно и то же значение ввиду отсутствия понятия контекстов и каждая запись CRAM может содержать данные любого типа.

5. В CRAM замены, инсерции и делеции представляют с помощью различных дескрипторов, и именно такой выбор увеличивает размер алфавита источника информации и дает повышенную энтропию источника. В подходе описываемого здесь изобретения, напротив, используют один алфавит и одно кодирование для замен, инсерций и делеций. Это упрощается процесс кодирования и декодирования и создает модель источника с пониженной энтропией, кодирование которой обеспечивает потоки двоичных данных, отличающиеся высокими характеристиками сжатия.

Целью настоящего изобретения является сжатие геномных последовательностей путем классифицирования и разделения данных секвенирования так, чтобы избыточная информация, подлежащая кодированию, была сведена к минимуму, а функции, такие как выборочный доступ и поддержка инкрементальных обновлений, действовали непосредственно в сжатой области.

Один из аспектов представленного подхода состоит в определение классов данных и метаданных, структурируемых в различных блоках и кодируемых по отдельности. Более актуальные улучшения такого подхода по сравнению с существующими способами, заключаются в следующем:

1. улучшение характеристик сжатия путем снижения энтропии источника информации за счет обеспечения эффективной модели источника для каждого класса данных или метаданных;

2. возможность осуществления выборочного доступа к частям сжатых данных и метаданных в целях любой дальнейшей обработки непосредственно в сжатой области;

3. возможность инкрементального (т. е. без необходимости в декодировании и повторном кодировании) обновления сжатых данных и метаданных новыми данными секвенирования, и/или метаданными, и/или новыми результатами анализа, связанными с определенными наборами прочтений секвенирования.

КРАТКОЕ ОПИСАНИЕ ЧЕРТЕЖЕЙ

На Фиг. 1 показано, как позицию картированных пар прочтений кодируют в блоке «pos» в виде разницы относительно абсолютного положения первого картированного прочтения.

На Фиг. 2 показано, как два прочтения в паре могут происходить из двух цепочек ДНК.

На Фиг. 3 показано, как кодируют обратный комплемент прочтения 2, если цепочку 1 используют в качестве референса.

На Фиг. 4 показаны четыре возможные комбинации прочтений, составляющих пару прочтений, и соответствующее кодирование в блоке «rcomp».

На Фиг. 5 для трех пар прочтений показано, как вычислять расстояние спаривания в случае прочтений постоянной длины.

На Фиг. 6 показано, как ошибки спаривания, кодированные в блоке «pair», позволяют декодеру восстанавливать правильное спаривание пары с помощью кодированного «MPPPD».

На Фиг. 7 показано кодирование расстояния спаривания, когда прочтение картировано не на тот референс, что его парное прочтение. В этом случае к расстоянию картирования добавляют дополнительные дескрипторы. Одним из них является флаг сигнализации, вторым — идентификатор референса, и еще одним — расстояние спаривания.

На Фиг. 8 показано кодирование несовпадение «n-типа» в блоке «nmis».

На Фиг. 9 показана картированная пара прочтений, в которой присутствуют замены по сравнению с референсной последовательностью.

На Фиг. 10 показано, как вычислять позиции замен в виде абсолютных или разностных значений.

На Фиг. 11 показано, как вычислять символы, кодирующие типы замен, когда используются коды IUPAC. Символы представляют расстояние, в кольцевом векторе замены, между молекулой, присутствующей в прочтении, и молекулой, присутствующей на референсе в этой позиции.

На Фиг. 12 показано, как кодировать замены в блоке «snpt».

На Фиг. 13 показано, как вычислять коды замены, когда используются коды неоднозначности IUPAC.

На Фиг. 14 показано, как кодировать блок «snpt», когда используются коды неоднозначности IUPAC.

На Фиг. 15 показано, как для прочтений класса I используют вектор замены, тот же самый, что и для класса M, с добавлением специальных кодов для вставок символов A, C, G, T, N.

На Фиг. 16 показаны некоторые примеры кодирования несовпадений и инделов в случае кодов неоднозначности IUPAC. В этом случае вектор замены намного длиннее, и поэтому возможных вычисляемых символов больше, чем в случае с пятью символами.

На Фиг. 17 показан другая модель источника для несовпадений и инделов, где каждый блок содержит позицию несовпадений или инсерций одного типа. В этом случае никаких символов для типа несовпадения или индела не кодируют.

На Фиг. 18 показан пример кодирования несовпадений и инделов. Когда в прочтении нет несовпадений или инделов данного типа, в соответствующем блоке кодируют 0. 0 действует в качестве разделителя прочтений и признака конца в каждом блоке.

На Фиг. 19 показано, как изменение в референсной последовательности может преобразовать прочтения M в прочтения P. Эта операция может снизить информационную энтропию структуры данных, особенно в случае данных с высоким покрытием.

На Фиг. 20 показан геномный кодер 2010 в соответствии с одним вариантом осуществления настоящего изобретения.

На Фиг. 21 показан геномный декодер 218 в соответствии с одним вариантом осуществления настоящего изобретения.

На Фиг. 22 показано, как «внутренний» референс может быть построен путем кластеризации прочтений и сборки сегментов, взятых из каждого кластера.

На Фиг. 23 показано, как стратегия построения референса заключается в сохранении самых последних прочтений после того, как к прочтениям была примерена определенная сортировка (например, в лексикографическом порядке).

На Фиг. 24 показано, как прочтение, принадлежащее классу «некартированных» прочтений (класс U), может быть кодировано с помощью шести дескрипторов, хранящихся или содержащихся в соответствующих блоках.

На Фиг. 25 показано, как альтернативное кодирование прочтений, принадлежащих классу U, где дескриптор pos имеет знак, используют для кодирования позиции картирования прочтения на построенном референсе.

На Фиг. 26 показано, как преобразования референса могут быть применены для удаления несовпадений из прочтений. В некоторых случаях преобразования референса могут

приводить к образованию новых несовпадений или изменению типа несовпадений, обнаруженных при сравнении с референсом до выполнения преобразования.

На Фиг. 27 показано, как преобразования референса могут изменить принадлежность прочтений к классу при устранении всех или подмножества несовпадений (т. е. прочтение, принадлежащее классу M до преобразования, назначают классу P после того, как преобразование референса было применено).

На фиг. 28 показано, как полукартированные пары прочтений (класс HM) могут быть использованы для заполнения неизвестных областей референсной последовательности путем сборки более длинных контигов с помощью некартированных прочтений.

На Фиг. 29 показано, как кодеры данных класса N, M и I конфигурируют с помощью векторов порогов и формируют отдельные подклассы классов данных N, M и I.

На Фиг. 30 показано, как все классы данных могут использовать один и тот же преобразованный референс для повторного кодирования, или для каждого класса N, M и I или любой их комбинации может быть использовано свое преобразование.

На Фиг. 31 показана структура заголовка геномного набора данных.

На Фиг. 32 показана общая структура таблицы главного индекса (MIT), где каждая строка содержит геномные интервалы нескольких классов данных P, N, M, I, U, HM и, кроме того, указатели на метаданные и аннотации. Столбцы относятся к определенным позициям на референсных последовательностях, связанных с кодированными геномными данными.

На Фиг. 33 показан пример одной строки MIT, содержащей геномные интервалы, относящиеся к прочтениям класса P. Геномные области, относящиеся к различным референсным последовательностям, разделены специальным флагом («S» в данном примере).

На Фиг. 34 показаны общая структура таблицы локального индекса (LIT) и порядок ее использования для хранения указателей на физическое местоположение кодированной геномной информации в сохраняемых или передаваемых данных.

На Фиг. 35 показан пример LIT, используемой для доступа к блокам доступа №№ 7 и 8 в полезной нагрузке блока.

На Фиг. 36 показана функциональная взаимосвязь между несколькими строками MIT и LIT, содержащимися в заголовках геномных блоков.

На Фиг. 37 показано, как блок доступа составляют из нескольких блоков геномных данных, переносимых разными геномными потоками, содержащими данные, принадлежащие разным классам. Каждый блок дополнительно содержит пакеты данных, используемые в качестве блоков передачи данных.

На Фиг. 38 показано, как блоки доступа составляют из заголовка и мультиплексированных блоков, принадлежащих одному или более блоками однородных данных. Каждый блок может состоять из одного или более пакетов, содержащих фактические дескрипторы геномной информации.

На Фиг. 39 показаны множественные выравнивания без сплайсинга. Крайнее левое прочтение имеет N выравниваний. N является первым значением дескриптора mmap, подлежащим декодированию, и сигнализирует о количестве выравниваний первого прочтения. Следующие N значений дескриптора mmap декодируют и используют для вычисления P, представляющего собой количество выравниваний второго прочтения.

На Фиг. 40 показано, как дескрипторы pos, pair и mmap используют для кодирования множественных выравниваний без сплайсов. Крайнее левое прочтение имеет N выравниваний.

На Фиг. 41 показаны множественные выравнивания со сплайсами.

На Фиг. 42 показано использование дескрипторов pos, pair, mmap и msar для представления множественных выравниваний со сплайсами.

СУЩНОСТЬ ИЗОБРЕТЕНИЯ

Признаки приведенной ниже формулы изобретения решают проблему решений известного уровня техники путем создания:

Способа кодирования данных геномной последовательности, где данные геномной последовательности содержат прочтения последовательностей нуклеотидов, причем способ включает в себя этапы:

выравнивания прочтений на одну или более референсных последовательностей с созданием тем самым выровненных прочтений,

классифицирования выровненных последовательностей согласно заданным правилам сопоставления с одной или более референсных последовательностей с созданием тем самым классов выровненных прочтений,

кодирование классифицированных выровненных прочтений в виде множества блоков дескрипторов,

причем кодирование классифицированных выровненных прочтений в виде множества блоков дескрипторов включает в себя выбор дескрипторов в соответствии с классами выровненных прочтений,

структурирования блоков дескрипторов с помощью информации заголовка с созданием тем самым последовательных блоков доступа.

В соответствии с еще одним аспектом способ кодирования дополнительно включает в себя дальнейшее классифицирование прочтений, которые не удовлетворяют заданным правилам сопоставления, в класс некартированных прочтений,
построение набора референсных последовательностей с использованием некоторых некартированных прочтений,
выравнивание класса некартированных прочтений на набор построенных референсных последовательностей,
кодирование классифицированных выровненных прочтений в виде множества блоков дескрипторов,
кодирование набора построенных референсных последовательностей,
структурирование блоков дескрипторов и кодированных референсных последовательностей с помощью информации заголовка с созданием тем самым последовательных блоков доступа.

В соответствии с еще одним аспектов способ кодирования дополнительно включает в себя идентифицирование геномных прочтений без несовпадений с референсной последовательностью в качестве первого «класса P».

В соответствии с еще одним аспектом способ кодирования дополнительно включает в себя идентифицирование геномных прочтений в качестве второго «класса N», когда несовпадения обнаруживают только в позициях, где секвенатор оказался не в состоянии определить никакого «основания», и количество несовпадений в каждом прочтении не превышает данного порога.

В соответствии с еще одним аспектом способ кодирования дополнительно включает в себя идентифицирование геномных прочтений в качестве третьего «класса M», когда несовпадения обнаруживают в позициях, где секвенатор оказался не в состоянии определить никакого «основания» (называются несовпадениями «n-типа») и/или он определил другое «основание» по сравнению с референсной последовательностью (называются несовпадениями «s-типа»), и количество несовпадений не превышает данных порогов для количества несовпадений «n-типа», «s-типа» и порога, полученного

из данной функции (f(n,s)), вычисленной на количестве несовпадений «n-типа» и «s-типа».

В соответствии с еще одним аспектом способ кодирования дополнительно включает в себя идентифицирование геномных прочтений в качестве четвертого «класса I», когда они могут иметь несовпадения того же типа, что и несовпадения «класса M», и, дополнительно, по меньшей мере одно несовпадение следующего типа: «инсерция» («i-типа»), «делеция» («d-типа»), мягкие отсечения («c-типа»), и при этом количество несовпадений каждого типа не превышает соответствующего данного порога и порога, обеспечиваемого данной функций (w(n,s,i,d,c)), вычисленной на количестве несовпадений «n-типа», «s-типа», «i-типа», «d-типа » и «c-типа».

В соответствии с еще одним аспектом способ кодирования дополнительно включает в себя идентифицирование геномных прочтений в качестве пятого «класса U», который содержит все прочтения, не отнесенные ни к одному из классов P, N, M, I, определенных ранее.

В соответствии с еще одним аспектом способ кодирования дополнительное отличается тем, что прочтения геномной последовательности, подлежащие кодированию, являются парными.

В соответствии с еще одним аспектом способ кодирования дополнительно отличается тем, что классифицирование дополнительно включает в себя идентифицирование геномных последовательностей в качестве шестого «класса HM», как содержащего все пары прочтений, где одно прочтение принадлежит классу P, N, M или I, а другое прочтение принадлежит «классу U».

В соответствии с еще одним аспектом способ кодирования дополнительно включает в себя этапы определения того, классифицируются ли два парных прочтения в один и тот же класс (каждый из: P, N, M, I, U) с назначением в таком случае пары в этот же идентифицированный класс.
Определение того, относятся ли два парных прочтения к разным классам, и если ни одно из них не принадлежит «классу U», то назначение пары прочтений классу с самым высоким приоритетом, определяемым в соответствии со следующим приоритетом:

P < N < M < I

где «класс P» обладает самым низким приоритетом, а «класс I» — самым высоким приоритетом;

определение того, классифицировано ли только одно из парных прочтений как принадлежащее «классу U», и классифицирование

пары прочтений как принадлежащей последовательностям «класса HM».

В соответствие с еще одним аспектом способ кодирования дополнительно отличается том, что каждый класс прочтений N, M, I дополнительно разделяют на два или более подклассов (296, 297, 298) в соответствии с вектором порогов (292, 293, 294), определенным, соответственно, для каждого класса N, M и I посредством количества несовпадений (292) «n-типа», функции f(n,s) (293) и функции w(n,s,i,d,c) (294).

определение того, классифицированы ли два парных прочтения в один и тот же подкласс с назначением в таком случае пары в этот же подкласс;

определение того, классифицированы ли два парных прочтения в подклассы разных классов с назначением в таком случае пары в подкласс, принадлежащий классу более высокого приоритета в соответствии со следующим выражением:

N < M < I

где N имеет самый низкий приоритет, а I имеет самый высокий приоритет;

определение того, классифицированы ли два парных прочтения в один и тот же класс, причем этим классом является N, или M, или I, но в разные подклассы, с назначением в таком случае пары в подкласс с самым высоким приоритетом в соответствии о следующими выражениями:

$N_1 < N_2 < \ldots < N_k$

$M_1 < M_2 < \ldots < M_j$

$I_1 < I_2 < \ldots < I_h$

где самый высокий индекс имеет наивысший приоритет.

В соответствии с еще одним аспектом информацию о позиции картирования каждого прочтения кодируют посредством блока дескрипторов pos.

В соответствии с еще одним аспектом информацию о цепочечности (например, из какой цепочки ДНК секвенировали прочтение) каждого прочтения кодируют посредством блока дескрипторов rcomp.

В соответствии с еще одним аспектом информацию о парноконцевых прочтениях кодируют посредством блока дескрипторов pair.

В соответствии с еще одним аспектом дополнительную информацию о выравнивании, такую как, картировано ли прочтение в правильную пару, не прошло ли оно проверки качества платформы/поставщика, является ли оно дубликатом ПЦР или оптическим дубликатом, является ли это вспомогательным выравниванием, кодируют посредством блока дескрипторов flags.

В соответствии с еще одним аспектом информацию о неизвестных основаниях кодируют посредством блока дескрипторов nmis.

В соответствии с еще одним аспектом информацию о позиции замен кодируют посредством блока дескрипторов snpp.

В соответствии с еще одним аспектом информацию о типе замен кодируют посредством блока специальных дескрипторов snpt.

В соответствии с еще одним аспектом информацию о позиции несовпадений типа замен, инсерций или делеций кодируют посредством блока дескрипторов indp.

В соответствии с еще одним аспектом информацию о типе несовпадений, таких как замены, инсерции или делеции, кодируют посредством блока дескрипторов indt.

В соответствии с еще одним аспектом информацию об отсеченных основаниях картированного прочтения кодируют посредством блока дескрипторов indc.

В соответствии с еще одним аспектом информацию о некартированных прочтениях кодируют посредством блока дескрипторов ureads.

В соответствии с еще одним аспектом информацию о типе референсной последовательности, используемой для кодирования, кодируют посредством блока дескрипторов rtype.

В соответствии с еще одним аспектом информацию о множественных выравниваниях картированных прочтений кодируют посредством блока дескрипторов mmap.

В соответствии с еще одним аспектом информацию о сплайсированных выравниваниях и множественных выравниваниях одного и того же прочтения кодируют посредством блока дескрипторов msar и блока дескрипторов mmap.

В соответствии с еще одним аспектом информацию об оценках выравнивания прочтения кодируют посредством блока дескрипторов mscore.

В соответствии с еще одним аспектом информацию о группах, которым принадлежат прочтения, кодируют посредством блока специальных дескрипторов rgroup.

В соответствии с еще одним аспектом способ кодирования дополнительно отличается тем, что блоки дескрипторов содержат таблицу главного индекса, содержащую по одному разделу для каждого класса и подкласса выровненных прочтений, причем раздел содержит позиции картирования на одну или более референсных последовательностей первого прочтения каждого блока доступа каждого класса или подкласса данных; при этом таблицу главного индекса и данные блока доступа кодируют вместе.

В соответствии с еще одним аспектом способ кодирования дополнительно отличается тем, что блоки дескрипторов также содержат информацию, относящуюся к типу использованного референса (ранее существующий или построенный) и сегментам прочтения, которые не совпадают с референсной последовательностью.

В соответствии с еще одним аспектом способ кодирования дополнительно отличается тем, что референсные последовательности сначала преобразуют в другие референсные последовательности путем применения замен, инсерций, делеций и отсечений, а затем кодируют классифицированные выровненные прочтения в виде множества блоков дескрипторов, относящихся к преобразованным референсным последовательностям.

В соответствии с еще одним аспектов способ кодирования дополнительно отличается тем, что к референсным последовательностям для всех классов данных применяют одно и то же преобразование.

В соответствии с еще одним аспектов способ кодирования дополнительно отличается тем, что к референсным последовательностям каждого класса данных применяют разные преобразования.

В соответствии с еще одним аспектом способ кодирования дополнительно отличается тем, что преобразования референсных последовательностей кодируют в виде блоков дескрипторов и структурируют с помощью информации заголовка с созданием тем самым последовательных блоков доступа.

В соответствии с еще одним аспектом способ кодирования дополнительно отличается тем, что кодирование классифицированных выровненных прочтений и связанных преобразований референсных последовательностей в виде множества блоков дескрипторов включает в себя этап связывания конкретной модели источника и конкретного энтропийного кодера с каждом блоком дескрипторов.

В соответствии с еще одним аспектом способ кодирования дополнительно отличается тем, что энтропийный кодер представляет собой контекстно-адаптивный арифметический кодер, кодер переменной длины или кодер Голомба.

В настоящем изобретении также предложен способ декодирования кодированных геномных данных, включающий в себя следующие этапы:
синтаксический анализ блоков доступа, содержащих кодированные геномные данные, для выделения множества блоков дескрипторов путем использования информации заголовка;
декодирование множества блоков дескрипторов для выделения выровненных прочтений в соответствии с заданными правилами сопоставления, определяющими их классификацию относительно одного или более референсных прочтений.

В соответствии с еще одним аспектом способ декодирования дополнительно включает в себя декодирование некартированных геномных прочтений.

В соответствии с еще одним аспектом способ декодирования дополнительно включает в себя декодирование классифицированных геномных прочтений.

В соответствии с еще одним аспектом способ декодирования дополнительно включает в себя декодирование таблицы главного индекса, содержащей по одному разделу для каждого класса прочтений и связанные соответствующие позиции картирования.

В соответствии с еще одним аспектом способ декодирования дополнительно включает в себя декодирование информации, относящейся к типу используемого референса — ранее существующий, преобразованный или построенный.

В соответствии с еще одним аспектом способ декодирования дополнительно включает в себя декодирование информации, относящейся к одному или более преобразованиям, которые должны быть применены к ранее существующим референсным последовательностям.

В соответствии с еще одним аспектом способ декодирования отличается тем, что геномные прочтения являются парными.

В соответствии с еще одним аспектом способ декодирования дополнительно включает в себя случай, когда геномные данные энтропийно декодируют.

В настоящем изобретении также предложен геномный кодер (2010) для сжатия данных 209 геномной последовательности, причем данные 209 геномной последовательности содержат прочтения последовательностей нуклеотидов, при этом геномный кодер (2010) содержит:

узел (201) выравнивателя, выполненный с возможностью выравнивания прочтений на одну или более референсных последовательностей с созданием тем самым выровненных прочтений;

узел (202) генератора конструированной последовательности, выполненный с возможностью создания конструированных референсных последовательностей;
узел (204) классифицирования данных, выполненный с возможностью классифицирования выровненных прочтений в соответствии с заданными правилами сопоставления с одной или более ранее существующими референсными последовательностями или построенными референсными последовательностями с созданием тем самым классов выровненных прочтений (208);
один или более узлов (205–207) кодирования блоков, выполненных с возможностью кодирования классифицированных выровненных прочтений в виде блоков дескрипторов путем выбора дескрипторов в соответствии с классами выровненных прочтений;
мультиплексор (2016) для мультиплексирования сжатых геномных данных и метаданных.

В соответствии с еще одним аспектом геномный кодер дополнительно содержит
узел (2019) преобразования референсной последовательности, выполненный с возможностью преобразования ранее существующих референсов и классов (208) данных в преобразованные классы (2018) данных.

В соответствии с еще одним аспектом геномный кодер дополнительно отличается тем, что
узел (204) классифицирования данных содержит конфигурируемые с помощью векторов порогов кодеры классов данных N, M и I, формирующие подклассы данных классов N, M и I.

В соответствии с еще одним аспектом геномный кодер дополнительно отличается тем, что узел (2019) преобразования референса применяет одно и то же преобразование (300) референса для всех классов и подклассов данных.

В соответствии с еще одним аспектом геномный кодер дополнительно отличается тем, что декодер (2019) преобразования референса применяет разные преобразования (301, 302, 303) к разным классам и подклассам данных.

В соответствии с еще одним аспектом геномный кодер дополнительно содержит функции, пригодные для осуществления всех аспектов ранее упомянутых способов кодирования.

В настоящем изобретении также предложен геномный декодер (218) для разуплотнения сжатого геномного потока (211), причем геномный декодер (218) содержит:
демультиплексор (210) для демультиплексирования сжатых геномных данных и метаданных;
средства (212–214) синтаксического анализа, выполненные с возможностью синтаксического анализа сжатого геномного потока с получением геномных блоков дескрипторов (215),
один или более декодеров (216–217) блоков, выполненных с возможностью декодирования геномных блоков в классифицированные прочтения последовательностей нуклеотидов (2111);
декодеры (219) классов геномных данных, выполненные с возможностью выборочного декодирования классифицированных прочтений последовательностей нуклеотидов относительно одной или более референсных последовательностей с получением несжатых прочтений последовательностей нуклеотидов.

В соответствии с еще одним аспектом геномный декодер дополнительно содержит декодер (2113) преобразования референса, выполненный с возможностью декодирования дескрипторов (2112) преобразования референса и создания преобразованного референса (2114) для использования декодерами (219) классов геномных данных.

В соответствии с еще одним аспектом геномный декодер дополнительно отличается тем, что одну или более референсных последовательностей хранят в сжатом потоке (211) геномных данных.

В соответствии с еще одним аспектом геномный декодер дополнительно отличается тем, что одну или более референсных последовательностей подают в декодер посредством внеполосного механизма.

В соответствии с еще одним аспектом геномный декодер дополнительно отличается тем, что одну или более референсных последовательностей строят в декодере.

В соответствии с еще одним аспектом геномный декодер дополнительно отличается тем, что одну или более референсных последовательностей преобразуют в декодере с помощью декодера (2113) преобразования референса.

В настоящем изобретении также предложен машиночитаемый носитель информации, содержащий инструкции, исполнение которых вызывает осуществление по меньшей мере одним процессором всех аспектов ранее упомянутых способов кодирования.

В настоящем изобретении также предложен машиночитаемый носитель информации, содержащий инструкции, исполнение которых вызывает осуществление по меньшей мере одним процессором всех аспектов ранее упомянутых способов декодирования.

В настоящем изобретении также предложена поддержка хранения геномных данных, кодированных в соответствии с выполнением всех аспектов ранее упомянутых способов кодирования.

ПОДРОБНОЕ ОПИСАНИЕ

В число геномных или протеомных последовательностей, упоминаемых в данном изобретении, входят, например, помимо прочего, нуклеотидные последовательности, последовательности дезоксирибонуклеиновой кислоты (ДНК), рибонуклеиновой кислоты (РНК) и аминокислотные последовательности. Несмотря на то, что в данном документе достаточно подробно описана геномная информация в форме нуклеотидной последовательности, ясно, что способы и системы сжатия могут быть реализованы также для других геномных или протеомных последовательностей, хотя и с некоторыми изменениями, понятными специалисту в данной области.

Информация о секвенировании генома формируется приборами высокопроизводительного секвенирования (HTS) в виде последовательностей нуклеотидов (именуемых также «основаниями»), представленных строками буквенных символов из определенного словаря. Минимальный словарь представлен пятью символами: {A, C, G, T, N}, которые обозначают 4 типа нуклеотидов, присутствующих в ДНК, а именно: аденин, цитозин, гуанин и тимин. В рибонуклеиновой кислоте (РНК) тимин заменен урацилом (U). N указывает на то, что секвенатор не смог определить никакого

основания, из-за чего реальный характер нуклеотида в этой позиции не определен. Если в секвенаторе приняты коды неоднозначности IUPAC, то алфавит, используемый для символов, представляет собой (A, C, G, T, U, W, S, M, K, R, Y, B, D, H, V, N или -).

Последовательности нуклеотидов, создаваемые секвенаторами, называют «**прочтениями**». Прочтения последовательности могут быть длиной от нескольких десятков до нескольких тысяч нуклеотидов. Некоторые технологии секвенирования создают прочтения последовательностей в виде «**пар**», где одно из прочтений которых происходит из одной цепочки ДНК, а второе — из другой цепочки. В секвенировании генома термин «**покрытие**» используют для выражения уровня избыточности данных последовательности относительно «**референсной последовательности**». Например, чтобы достичь покрытия 30x генома человека (длиной 3,2 миллиарда оснований) секвенатор должен создать всего 30 × 3,2 миллиарда оснований, чтобы в среднем каждая позиция в референсе «покрывалась» 30 раз.

Везде в данном описании **референсная последовательность** — это любая последовательность, на которую выравнивают/картируют последовательности нуклеотидов, создаваемые секвенаторами. Одним примером последовательности может быть «**референсный геном**» — последовательность, собранная учеными в качестве репрезентативного примера наборов генов вида. Например, GRC37, геном человека Консорциума референсного генома (сборка 37), получен из образцов тринадцати анонимных добровольцев из г. Буффало, штат Нью-Йорк. Однако референсная последовательность может также состоять из синтетической последовательности, задуманной и попросту построенной для улучшения сжимаемости прочтений с учетом их дальнейшей обработки. Это более подробно описано в разделе «Дескрипторы класса U и построение "внутренних" референсов для некартированных прочтений "класса U" и "класса HM"» и изображено на Фиг. 22 и 23.

Секвенаторы могут вносить в прочтения последовательности ошибки, такие как:

1. Решение о пропуске определения основания ввиду отсутствия уверенности в определении какого-либо конкретного основания. Это называют неизвестным основанием и помечают как «N» (обозначают как несовпадение «n-типа»).
2. Использование неверного символа (т. е. представление другой нуклеиновой кислоты) для представления нуклеиновой кислоты, на самом деле присутствующей в

секвенированном образце; это обычно называют «ошибкой замены» (обозначают как несовпадение «s-типа»).

3. Вставка в одно прочтение последовательности дополнительных символов, которые не имеют отношения ни к одной действительно присутствующей нуклеиновой кислоте; это обычно называют «ошибкой вставки» (обозначают как несовпадение «i-типа»).
4. Удаление из одного прочтения последовательности символов, которые представляют нуклеиновые кислоты, действительно присутствующие в секвенированном образце; это обычно называют «ошибкой удаления» (обозначают как несовпадение «d-типа»).
5. Рекомбинация одного или более фрагментов в один фрагмент, который не отражает подлинную сущность исходной последовательности; этот обычно приводит к решению средства выравнивания отсечь основания (обозначают как несовпадение «c-типа»).

Термин «покрытие» используют в литературе для количественного определения степени, в которой референсный геном или его часть могут быть охвачены имеющимися прочтениями последовательности. Покрытие называют:

- частичным (менее 1X), когда на некоторые части референсного генома не картировано ни одного имеющегося прочтения последовательности;
- однократным (1X), когда на все нуклеотиды референсного генома картирован один и только один символ, представленный в прочтениях последовательности;
- многократным (2X, 3X, NX), когда каждый нуклеотид референсного генома картирован несколько раз.

Цель настоящего изобретения состоит в определении формата представления геномной информации, в котором можно эффективно получать доступ к релевантной информации и транспортировать ее, и в котором вес избыточной информации снижен.

Раскрытое изобретение имеет следующие элементы новизны.

1 Прочтения последовательности классифицируют и разделяют на классы данных в соответствии с результатами выравнивания относительно референсных последовательностей. Такие классификация и разделение обеспечивают возможность выборочного доступа к кодированным данным в соответствии с критериями, относящимися к результатам выравнивания и точности совпадения.

2 Классифицированные прочтения последовательности и связанные метаданные представляют в виде однородных блоков дескрипторов для получения четко различимых источников информации, отличающихся низкой информационной энтропией.

3 Возможность моделирования каждого отдельного источника информации с помощью четко различимой модели источника, адаптированной к статистическим характеристикам каждого класса, и возможность изменения модели источника в пределах каждого класса и в пределах блока дескрипторов для каждого индивидуально доступного блока данных (блока доступа). Внедрение подходящих контекстно-адаптивных вероятностных моделей и соответствующих энтропийных кодеров в соответствии со статистическими свойствами каждой модели источника.

4 Определение соответствий и зависимостей между блоками дескрипторов для обеспечения возможности выборочного доступа к данным секвенирования и связанным метаданным без необходимости декодирования всех блоков дескрипторов, если требуется не вся информация.

5 Кодирование каждого блока класса данных последовательности и связанных метаданных относительно «ранее существующих» (также называемых «внешними») референсных последовательностей или относительно «преобразованных» референсных последовательностей, получаемых путем применения надлежащих преобразований к «ранее существующим» референсным последовательностям, чтобы снизить энтропию источников информации блоков дескрипторов. Дескрипторы представляют прочтения, разделенные на различные классы данных. После любого кодирования прочтений с использованием соответствующих дескрипторов относительно «ранее существующего» референса или «преобразованной» «ранее существующей» референсной последовательности встречаемость различных несовпадений можно использовать для определения подходящих преобразований в референсные последовательности с целью нахождения конечного кодированного представления с низкой энтропией и достижения более высокой эффективности сжатия.

6 Построение одной или более референсных последовательностей (также называемых «внутренними» референсами для отличия их от «ранее существующих», также называемых «внешними» референсными последовательностями), используемых для кодирования класса прочтений, которые демонстрируют степень точности совпадения относительно ранее существующих референсных последовательностей, не удовлетворяющую установленным ограничениям. Такие ограничения устанавливают с той целью, чтобы затраты на кодирование представления в сжатой форме класса прочтений, выровненных относительно «внутренних» референсных последовательностей, и затраты на представление самих «внутренних» референсных последовательностей были ниже, чем при кодировании невыровненного класса

представлений буквально или с использованием «внешних» референсных последовательностей без преобразований или с преобразованиями.

Далее каждый из вышеупомянутых аспектов будет дополнительно описан в подробностях.

Классификация прочтений последовательности в соответствии с правилами сопоставления

Прочтения последовательности, формируемые секвенаторами, в соответствии с настоящим изобретением подразделяют на шесть различных «классов» в зависимости от результатов сопоставления при выравнивании относительно одной или более «ранее существующих» референсных последовательностей.

При выравнивании ДНК-последовательности нуклеотидов относительно референсной последовательности можно выделить следующие случаи:

1. Область в референсной последовательности совпадает с прочтением последовательности без единой ошибки (т. е. идеальное картирование). Такую последовательность нуклеотидов называют «идеально совпадающим прочтением» или обозначают как «класс P».

2. Область в референсной последовательности совпадает с прочтением последовательности, причем тип и количество несовпадений определяются только количеством позиций, в которых секвенатору, формировавшему это прочтение, не удалось определить никакого основания (или нуклеотида). Несовпадения такого типа обозначают буквой «N», указывающей на неопределенное нуклеотидное основание. В настоящем документе несовпадение этого типа называют несовпадением «n-типа». Такие последовательности принадлежат к прочтениям «класса N». После того, как прочтение классифицировано как принадлежащее «классу N», полезно ограничить степень неточности совпадения данной верхней границей и разграничить совпадения, считающиеся действительными и не считающиеся действительными. Поэтому прочтения, назначенные классу N, также ограничивают путем установки порога (MAXN), определяющего максимально допустимое количество неопределенных оснований (т. е. оснований, определенных как «N»), которое может содержать прочтение. Такая классификация неявно определяет требуемую минимальную точность совпадения (или максимальную степень несовпадения), общую для всех прочтений, принадлежащих классу N, при сравнении с соответствующей референсной

последовательностью, что дает полезный критерий для применения выборочного поиска данных к сжатым данным.

3. Область в референсной последовательности совпадает с прочтением последовательности, причем типы и количества несовпадений, определяются количеством позиций, в которых секвенатору, формирующему прочтение, не удалось определить никакого нуклеотидного основания, при наличии таковых (т. е. несовпадение «n-типа»), плюс количеством позиций, в которых было определено основание, отличное от присутствующего в референсе. Такой тип несовпадения, обозначаемый как «замена», называют также однонуклеотидной вариацией (ОНВ) или однонуклеотидным полиморфизмом (ОНП). В данном документе несовпадение этого типа называют также несовпадением «s-типа». В таком случае прочтение последовательности называют «прочтениями с M-несовпадением» и назначают «классу M». Как и в случае «класса N», для всех прочтений, принадлежащих «классу M», полезно ограничить степень неточности совпадения данной верхней границей и разграничить совпадения, считающиеся действительными и не считающиеся действительными. Поэтому прочтения, назначенные классу M, также ограничивают путем установки набора порогов, одного для количества «n» несовпадений «n-типа» (MAXN), при наличии таковых, а другого для количества замен «s» (MAXS). Третье ограничение представляет собой порог, определяемый любой функцией от обоих чисел «n» и «s», f(n,s). Такое третье ограничение позволяет формировать классы с верхней границей неточности совпадения в соответствии с любым значимым критерием выборочного доступа. Например, без ограничений, f(n,s) может быть (n+s)1/2, или (n+s), или линейным или нелинейным выражением, которое устанавливает границу для максимального уровня неточности совпадения, принятого для прочтения, принадлежащего «классу M». Такая граница представляет собой очень полезный критерий для применения требуемого выборочного поиска данных к сжатым данным при анализе прочтений последовательности в различных целях, так как она позволяет устанавливать дополнительную границу для любой возможной комбинации количеств несовпадений «n-типа» и несовпадений (замен) «s-типа» сверх простого порога, применяемого к одному типу или другому типу.

4. Четвертый класс составляют прочтения последовательности, представляющие по меньшей мере одно несовпадение любого типа из числа «инсерции», «делеции»

(иначе называемых *инделами*) и «отсечения», плюс (при наличии таковых) любые несовпадения типа, принадлежащего классу N или M. Такие последовательности называют «прочтениями с I-несовпадениями» и назначают «классу I». Инсерции образуются дополнительной последовательностью из одного или более нуклеотидов, отсутствующих в референсе, но присутствующих в последовательности прочтения. В данном документе несовпадение этого типа называют несовпадением «i-типа». В литературе, когда вставленная последовательность, находится на краях последовательности, ее называют также «мягко отсеченной» (то есть, нуклеотиды не совпадают с референсной последовательностью, но остаются в выровненных прочтениях, в отличие от «жестко отсеченных» нуклеотидов, которые отбрасывают). В данном документе несовпадение этого типа называют несовпадением «c-типа». Решения оставлять или отбрасывать нуклеотиды принимаются на этапе средства выравнивания, а не раскрытым в данном изобретении средством классифицирования прочтений, которое принимает и обрабатывает прочтения в том виде, в котором они определены секвенатором или на следующей стадии выравнивания. Делеции представляют собой «дыры» (отсутствующие нуклеотиды) в прочтении в сравнении с референсной последовательностью. В данном документе несовпадения этого типа называют несовпадениями «d-типа». Как и в случае классов «N» и «M» можно и нужно определять передел неточности несовпадения. Определение набора ограничений для «класса I» основывается на тех же принципах, которые используются для «класса M» и указаны в последних строках Таблица 1. Помимо порога для каждого типа несовпадения, допустимого для данных класса I, определяют дополнительное ограничение с помощью порога, определяемого посредством любой функции от количеств несовпадений «n», «s», «d», «i» и «c» — w(n,s,d,i,c). Такое дополнительное ограничение позволяет формировать классы с верхней границей неточности совпадения в соответствии с любым значащим критерием выборочного доступа, определяемым пользователем. В качестве примера, но не ограничения, w(n,s,d,i,c) может представлять собой (n+s+d+i+c)1/5, или (n+s+d+i+c), или любое линейное или нелинейное выражение, устанавливающее границу для максимального уровня неточности совпадения, который принимают для прочтения, принадлежащего «классу I». Такая граница представляет собой очень полезный критерий для применения требуемого выборочного поиска данных к сжатым данным при анализе прочтений последовательности в различных целях, так как она позволяет устанавливать дополнительную границу для любой возможной комбинации количеств несовпадений,

допустимых в прочтениях «класса I» сверх простого порога, применяемого к каждому типу допустимого несоответствия.

5. Пятый класс включает в себя все прочтения, для которых не найдено ни одного картирования, считающегося действительным (то есть, не удовлетворяющие набору правил, определяющих верхнюю границу для максимальной неточности совпадения, как указано в Таблица 1) для каждого класса данных при сравнении с референсной последовательностью. Такие последовательности называют «некартированными» относительно референсных последовательностей и классифицирую как принадлежащие «классу U».

Классификация пар прочтений в соответствии с правилами сопоставления

Классификация, определенная в предыдущем разделе, относится к одиночным прочтениям последовательности. В случае технологий секвенирования, формирующих прочтения парами (то есть, технологий Illumina Inc.), в отношении двух составляющих прочтений которых известно, что они разделены неизвестной последовательностью переменной длины, целесообразно рассматривать отнесение всей пары к одному классу данных. Прочтение, которое сопряжено с другим прочтением, называют его «парным прочтением».

Если оба спаренных прочтения принадлежат одному и тому же классу, назначение пары классу очевидно: всею пару назначают тому же классу, что и любую из частей пары (т. е, P, N, M, I, U). Если два прочтения принадлежат разным классам, но ни одно из них не принадлежит «классу U», то всю пару назначают классу с наивысшим приоритетом, определяемым согласно следующему выражению:

$$P < N < M < I$$

где «класс P» обладает самым низким приоритетом, а «класс I» — самым высоким приоритетом.

В случае, когда одно из прочтений принадлежит «классу U», а его парное прочтение принадлежит любому из классов P, N, M, I, определяют шестой класс как «класс HM», что означает «наполовину картированный».

Определение такого специфического класса прочтений мотивировано тем фактом, что его используют в попытке определить гэпы или неизвестные области, существующие в референсных геномах (называемые также малоизвестными или неизвестными областями). Такие области восстанавливают путем картирования пар на краях с использованием парного прочтения, которое может быть картировано на неизвестные области. Затем некартированное прочтение пары используют для построения так называемых «контигов» неизвестной области, как показано на Фиг. 28. Поэтому обеспечение выборочного доступа только к парам прочтений такого типа сильно сокращает соответствующую вычислительную нагрузку, позволяя обрабатывать такие данные, происходящие из больших количеств наборов данных, значительно эффективнее, чем с использованием решений известного уровня техники, потребовавших бы полной проверки.

В приведенной ниже таблице обобщены правила сопоставления, применяемые к прочтениям с целью определения класса данных, которому принадлежит каждое прочтение. Правила определены в первых пяти столбцах таблицы в форме наличия или отсутствия типов несовпадений (несовпадений типов n, s, d, i и c). В шестом столбце приведены правила в виде максимального порога для каждого типа несовпадения и какой-либо функции (при наличии таковой) f(n,s) и w(n,s,d,i,c) от возможных типов несовпадения.

Количество и типы несовпадений, обнаруживаемых при сопоставлении прочтения с референсной последовательностью					Набор ограничений на точность совпадения	Класс назначения
Кол-во Неизвест-ных основа-ний («N»)	Кол-во замен	Кол-во делеций	Кол-во инсер-ций	Кол-во отсечен-ных основа-ний		
0	0	0	0	0	0	P
n > 0	0	0	0	0	n ≤ MAXN	N
					n > MAXN	U

n ≥ 0	s > 0	0	0	0	n ≤ MAXN и s ≤ MAXS и f(n,s) ≤ MAXM	M
					n > MAXN или s > MAXS или f(n,s) > MAXM	U
n ≥ 0	s ≥ 0	d ≥ 0*	i ≥ 0*	c ≥ 0*	n ≤ MAXN и s ≤ MAXS и d ≤ MAXD и i ≤ MAXI и c ≤ MAXC w(n,s,d,i,c) ≤ MAXTOT	I
		***Должно быть по меньшей мере одно несовпадение типа d, i, c (т. е. d > 0, или i > 0, или c > 0)**				
		d ≥ 0	i ≥ 0	c ≥ 0	n > MAXN или s > MAXS или d > MAXD или i > MAXI или c > MAXC w(n,s,d,i,c) > MAXTOT	U

Таблица 1. Тип несовпадений и набор ограничений, которым должно удовлетворять каждое прочтение последовательности, относимое к классам данных, определенным в данном описании

Правила сопоставления для разделения классов данных прочтений N, M и I на подклассы с различными степенями точности совпадения

Классы данных типа N, M и I, которые определены в предыдущих разделах, могут быть дополнительно разделены на произвольное количество различных подклассов с разными степенями точности совпадения. Такая возможность является важным техническим преимуществом в обеспечении разделения на более мелкие подмножества и, как следствие, намного более эффективного выборочного доступа к каждому классу данных. В качестве примера, но не ограничения, для разбиения класса N на k подклассов

(подкласс N_1, …, подкласс N_k) необходимо определить вектор с соответствующими компонентами $MAXN_1$, $MAXN_2$, …, $MAXN_{(k-1)}$, $MAXN_{(k)}$ при условии, что $MAXN_1 < MAXN_2 < \ldots < MAXN_{(k-1)} < MAXN$, и назначить каждое прочтение подклассу самого низкого ранга, который удовлетворяет ограничениям, указанным в Таблица 1, при выполнении оценки для каждого элемента вектора. Это показано на Фиг. 29, где блок 291 классифицирования данных содержит кодер класса P, N, M, I, U, HM и кодеры для аннотаций и метаданных. Кодер класса N конфигурируют с помощью вектора 292 порогов $MAXN_1$–$MAXN_k$, который формирует k подклассов данных N (296).

В случае классов типа M и I применяют такой же принцип путем определения вектора с теми же свойствами для MAXM и MAXTOT, соответственно, и используют компоненты каждого вектора в качестве порога, чтобы проверять, удовлетворяют ли ограничению функции f(n,s) и w(n,s,d,i,c). Как и в случае подклассов типа N, назначают подклассу самого низкого ранга, удовлетворяющему ограничению. Количество подклассов для класса каждого типа является независимой величиной и допускается любая комбинация подразделений. Это показано на Фиг. 29, где кодер 293 класса M и кодер 294 класса I конфигурируют, соответственно, с помощью вектора порогов $MAXM_1$–$MAXM_j$ и $MAXTOT_1$–$MAXTOT_h$. Эти два кодера формируют, соответственно, j подклассов (297) данных M и h подклассов (298) данных I.

Если два прочтения в паре отнесены к одному и тому же подклассу, то пара принадлежит этому же подклассу.

Если два прочтения в паре отнесены к подклассам разных классов, то пара принадлежит подклассу класса с более высоким приоритетом в соответствии со следующим выражением:

$$N < M < I$$

где N имеет самый низкий приоритет, а I имеет самый высокий приоритет.

Если два прочтения принадлежат разным подклассам одного из классов N, M или I, то пара принадлежит подклассу с самым высоким приоритетом в соответствии со следующим выражениям:

$$N_1 < N_2 < \ldots < N_k$$

$$M_1 < M_2 < \ldots < M_j$$

$I_1 < I_2 < \ldots < I_h$

где самый высокий индекс имеет наивысший приоритет.

Преобразования «внешних» референсных последовательностей

Несовпадения, найденные для прочтений, отнесенных к классам N, M и I, могут быть использованы для создания «преобразованных» референсов, которые будут применены для более эффективного сжатия представления прочтения.

Прочтения, классифицированные как принадлежащие классам N, M или I (относительно «ранее существующей» (т. е. «внешней») референсной последовательности, обозначенной как RS_0), могут быть кодированы относительно «преобразованной» референсной последовательности RS_1 в соответствии с встречаемостью действительных несовпадений относительно «преобразованного» референса. Например, если прочтение$^{M}_{in}$, принадлежащее классу M (обозначено как i-е прочтение класса M), содержит несовпадения относительно референсной последовательности RS_n, то после «преобразования» прочтение$^{M}_{in}$ = прочтение$^{P}_{i(n+1)}$ может быть получено с помощью $A(Ref_n)=Ref_{n+1}$, где A является преобразованием из референсной последовательности RS_n в референсную последовательность RS_{n+1}.

На Фиг. 19 показан пример того, как прочтения, содержащие несовпадения (принадлежащие классу M) относительно референсной последовательности 1 (RS_1), могут быть преобразованы в прочтения, идеально совпадающие с референсной последовательностью 2 (RS_2), полученной из RS_1 путем изменения оснований, соответствующих позициям несовпадения. Они остаются классифицированными и их кодируют вместе с другими прочтениями в одном и том же блоке доступа, но кодирование выполняют с использованием только дескрипторов и значений дескрипторов, необходимых для прочтения класса P. Это преобразование можно представить в виде следующего выражения:

$RS_2 = A(RS_1)$

Когда представление преобразования A, которое формирует RS_2, будучи примененным к RS_1, и представление прочтений относительно RS_2 соответствуют более низкой энтропии, чем представление прочтений класса M относительно RS_1, целесообразно передавать представление преобразования A и соответствующее представление прочтения относительно RS_2, поскольку достигается более высокое сжатие представления данных.

Кодирование преобразования A для передачи в сжатом потоке двоичных данных требует определения двух дополнительных синтаксических элементов синтаксиса, как указано в приведенной ниже таблице.

Дескрипторы	Семантика	Комментарии
rftp	Позиция преобразования референса	Позиция отличия между референсом и контигом, используемым для предсказания
rftt	Тип преобразования референса	Тип отличия между референсом и контигом, используемым для предсказания Такой же синтаксис описан для дескриптора snpt, определенного далее в этом документе.

На фиг. 26 приведен пример порядка применения преобразования референса для уменьшения количества несовпадений, подлежащих кодированию, на картированных прочтениях.

Необходимо отметить последствия применения преобразования к референсу в некоторых случаях:

- возможно внесение несовпадений в представления прочтений, которые отсутствовали бы при сравнении с референсом до применения преобразования;
- возможно изменение типов несоответствий; прочтение может содержать A вместо G, в то время как все остальные прочтения содержат C вместо G, но несовпадения остаются в той же позиции;
- различные классы данных и подмножества данных каждого класса данных могут относиться к одним и тем же «преобразованным» референсным последовательностям или к референсным последовательностям, полученным применением разных преобразований к одной и той же ранее существующей референсной последовательности.

На Фиг. 27 также показан пример того, как прочтения могут менять тип кодирования с одного класса данных на другой с помощью надлежащего набора дескрипторов (например, с использованием дескрипторов класса P для кодирования прочтения из класса M) после применения преобразования и представления прочтения с помощью «преобразованного» референса. Это происходит, например, когда преобразование

меняет все основания, соответствующие несовпадениям прочтения, на основания, фактически присутствующие в прочтении, тем самым фактически преобразуя прочтение, принадлежащее классу M (если сравнивать с исходной не «преобразованной» референсной последовательностью), в виртуальное прочтение класса P (если сравнивать с «преобразованным» рефренсом). Определение набора дескрипторов, используемых для каждого класса данных, приведено в следующих разделах.

На Фиг. 30 показано, как разные классы данных могут использовать один и тот же «преобразованный» референс $R_1 = A_0(R_0)$ (300) для повторного кодирования прочтений, или к каждому классу данных могут быть по отдельности применены разные преобразования A_N (301), A_M (302), A_I (303).

Определение информации, необходимой для представления прочтений последовательности в виде блоков дескрипторов

После того, как классифицирование прочтений завершено определением классов, дальнейшая обработка состоит из определения набора различных дескрипторов, которые представляют остальную информацию, позволяющую восстанавливать последовательность прочтения, когда ее представляют как картируемую на данную референсную последовательность. Структура данных этих дескрипторов требует хранения глобальных параметров и метаданных, которые будут использованы механизмом декодирования. Эти данные структурированы в ***заголовке геномного набора данных***, описанном в приведенной ниже таблице. Набор данных определяют как совокупность элементов кодирования, необходимых для восстановления геномной информации, относящейся к одному прогону секвенирования генома и всему последующему анализу. Если один и тот же образец для генома секвенируют дважды за два разных прогона, полученные данные будут кодированы в двух разных наборах данных.

Элемент	Тип	Описание
Уникальный ИД	Массив байтов	Уникальный идентификатор для кодированного содержимого
Главная торговая марка	Массив байтов	Основная + дополнительная версии алгоритма кодирования
Дополнительная версия	Массив	

	байтов	
Размер заголовка	Целое число	Размер в байтах всего кодированного содержимого
Длина прочтений	Целое число	Размер прочтений в случае постоянной длины прочтений. Специальное значение (например, 0) зарезервировано для прочтений переменной длины
Счетчик реф.	Целое число	Количество используемых референсных последовательностей
Счетчики блоков доступа	Массив байтов (например, целые числа)	Общее количество кодированных блоков доступа на референсную последовательность
Идентификаторы референсов	Массив байтов	Уникальные идентификаторы для референсных последовательностей
for (i=0; i<Ref_count; i++) {		
Reference_genome:Ref_ID	Строк:строка	Однозначный ИД в виде строки символов, определяющий референсные последовательности, используемые в данном наборе данных
}		
for (i=0; i<Ref_count; i++) {		

Блоки реф.	Массив байтов	Количество кодированных блоков на каждый референс
}		
Размер метки набора данных	Целое число	Размер следующего элемента
Метка набора данных	Строка	Строка символов, используемая для идентификации набора данных
Тип набора данных	Целое число	Тип данных, кодированных в наборе данных (например, выровненные, невыровненные)
Таблица главного индекса Позиции выравнивания первого прочтения в каждом блоке (блоке доступа). Т. е. младшая позиция первого прочтения на референсном геноме для каждого блока из шести классов. 1 на класс pos (шесть) на референсную последовательность	Массив байтов	Это многомерный массив, поддерживающий произвольный доступ к блокам доступа
Список меток Подмножество главного заголовка, указывающее • количество меток; • для каждой метки: ○ ИД метки; ○ количество референсных последовательностей, имеющих отношение к метке; ○ для каждой референсной последовательности:	Массив байтов	Это список меток, каждая из которых представлена в виде многомерного массива для поддержки выборочно доступа к конкретным геномным областям или подобластям, или объединениям областей или подобластей

▪ идентификатор референса; ▪ количество областей, охватываемых меткой; ▪ для каждой области: • ИД класса; • начальная позиция в геномном диапазоне; • конечная позиция в геномном диапазоне. Начальную позицию и конечную позицию можно заменить «номерами блоков», вместе с ИД референсной последовательности и ИД класса составляющими трехмерный вектор для адресации координат таблицы главного индекса		
Набор параметров	Массив байтов	Параметры кодирования, используемые для конфигурирования процесса кодирования и отправляемые в декодер

Таблица 1. Структура заголовка геномного набора данных

Прочтение последовательности (т. е., сегмент ДНК), относящееся к данной референсной последовательности, может быть выражено с помощью следующих элементов:

- Начальная позиция на референсной последовательности (pos).
- Флаг, сигнализирующий, нужно ли последовательность рассматривать как обратный комплемент относительно референса (rcomp).
- Расстояние до парного прочтения в случае парных прочтений (pair).
- Значение длины прочтения в случае технологии секвенирования с получением прочтений переменной длины (len). В случае постоянной длины прочтений длину прочтений, связанную с каждым прочтением, очевидно можно опустить и сохранить в главном заголовке файла.
- Для каждого несовпадения:
    - позиция (*nmis* для класса N, *snpp* для класса M и *indp* для класса I);
    - тип несовпадения (отсутствует в классе N, *snpt* в классе M, *indt* в классе I).
- Флаги, указывающие определенные характеристики прочтения последовательности, такие как:
    - шаблон, имеющий множество сегментов при секвенировании;
    - каждый сегмент, надлежащим образом выровненный в соответствии со средством выравнивания;
    - некартированный сегмент;
    - следующий некартированный сегмент в шаблоне;
    - сигнализация о первом или последнем сегменте;
    - непрохождение контроля качества;
    - дубликат ПЦР или оптический дубликат;
    - вспомогательное выравнивание;
    - дополнительное выравнивание.
- Строка мягко отсеченных нуклеотидов, если присутствуют (*indc* в классе I).
- Флаг, указывающий референс, использованный для выравнивания и сжатия (например, «внутренний» референс для класса U), если применимо (дескриптор rtype).
- Для класса U дескриптор indc указывает те части прочтений (как правило, края), которые не соответствуют заданному набору ограничений на точность совпадения с «внутренними» референсами.
- Дескриптор ureads используют для буквального кодирования прочтений, которые невозможно картировать ни на один имеющийся референс, будь то ранее

существующая (т. е. «внешняя», такая как фактический референсный геном) или «внутренняя» референсная последовательность.

Данная классификация создает группы дескрипторов (дескрипторы), которые могут быть использованы для однозначного представления прочтений геномной последовательности. В приведенной ниже таблице обобщены дескрипторы, необходимые для каждого класса прочтений, выровненных относительно «внешних» (т. е. ранее существующих) или «внутренних» (т. е. построенных) референсов.

	P	N	M	I	U	HM
pos	X	X	X	X	X	X
pair	X	X	X	X		
rcomp	X	X	X	X	X	X
flags	X	X	X	X	X	X
rlen	X	X	X	X	X	X
nmis		X				
snpp			X		X	
snpt			X		X	
indp				X		X
indt				X		X
indc				X	X	X
ureads					X	X
rtype					X	
rgroup	X	X	X	X	X	X
mmap	X	X	X	X	X	
msar	X	X	X	X	X	
mscore	X	X	X	X	X	

Таблица 2. Определенные блоки дескрипторов для каждого класса данных

Прочтения, принадлежащие классу P, отличаются тем, что они могут быть идеально восстановлены с помощью лишь позиции, информации об обратном комплементе и

смещении между парными прочтениями в случае, когда они были получены с помощью технологии секвенирования, обеспечивающей парноконцевые прочтения с длинной вставкой, некоторых флагов и длины прочтения.

В следующем разделе подробно описано определение этих дескрипторов для классов P, N, M и I, а для класса U они описаны в последующем разделе.

Класс HM применим только к парам прочтений и представляет собой особый случай, в котором одно прочтение принадлежит классу P, N, M или I, а другое принадлежат классу U.

Дескриптор позиции

В блоке позиции (**pos**) хранится только позиция картирования первого картированного прочтения в виде абсолютного значения на референсной последовательности. Все остальные дескрипторы позиции принимают значение, выражающее разницу относительно предыдущей позиции. Такое моделирование источника информации, определяемого последовательностью дескрипторов позиции прочтения, как правило, отличается пониженной энтропией, особенно для процессов секвенирования, формирующих результаты с высоким покрытием.

Например, на Фиг. 1 показано, как после описания начальной позиции первого выравнивания как позиции «10 000» на референсной последовательности позиция второго прочтения, начинающегося в позиции 10 180, описана как «180». При высоких покрытиях (> 50x) большинство дескрипторов вектора позиции демонстрируют очень высокую встречаемость низких значений, таких как 0 и 1 и другие малые целые числа. На Фиг. 1 показано, как позиции трех парных прочтений описаны в блоке **pos**.

Дескриптор обратного комплемента

Каждое прочтение пар прочтений, создаваемых с помощью технологий секвенирования, может происходить из любой из двух цепочек генома секвенируемого органического образца. Однако в качестве референсной последовательности используют только одну из двух цепочек. На Фиг. 2 показано, как в паре прочтений одно прочтение (прочтение 1) может происходить из одной цепочки, а другое прочтение (прочтение 2) может происходить из другой цепочки.

Если цепочку 1 используют в качестве референсной последовательности, то прочтение 2 может быть кодировано в качестве обратного комплемента соответствующего фрагмента на цепочке 1. Это показано на Фиг. 3.

В случае сдвоенных прочтений возможны четыре комбинации пар из прямого и обратного парного комплемента пар. Это показано на Фиг. 4. Блок rcomp кодирует четыре возможные комбинации.

То же самое кодирование используют для информации об обратном комплементе прочтений, принадлежащих классам N, M, P и I. Чтобы обеспечить выборочный доступ к различным классам данных, информацию об обратных комплементах прочтений, принадлежащих четырем классам, кодируют в разных блоках, как показано в таблице 2.

Дескриптор информации о спаривании

Дескриптор спаривания сохраняют в блоке pair . Такой блок хранит дескрипторы, кодирующие информацию, необходимую для восстановления возникающих пар прочтения, когда используемая технология секвенирования создает прочтения парами. Хотя на момент раскрытия изобретения подавляющее большинство данных секвенирования формируют с использованием технологии, создающей парные прочтения, не все технологии являются таковыми. Именно поэтому наличие данного блока необязательно для восстановления всей информации данных секвенирования, если рассматриваемая технология секвенирования геномных данных не дает информации о парных прочтениях.

Определения:

- **парное прочтение**: прочтение, связанное с другим прочтением в паре прочтений (например, в предыдущем примере прочтение 2 является парным прочтением прочтения 1);
- **расстояние спаривания**: количество позиций нуклеотидов референсной последовательности, которые разделяют одну позицию в первом прочтении (якорь спаривания, например, последний нуклеотид первого прочтения) от одной позиции второго прочтения (например, первого нуклеотида второго прочтения);
- **наиболее вероятное расстояние спаривания (MPPD)**: это наиболее вероятное расстояние спаривания, выраженное в количестве позиций нуклеотидов;
- **расстояние спаривания позиций (PPD)**: PPD представляет собой способ выражения расстояния спаривания в количестве прочтений, отделяющих одно прочтение от его соответствующего парного прочтения, присутствующего в определенном блоке дескрипторов позиции;

- **наиболее вероятное расстояние спаривания позиций (MPPPD)**: это наиболее вероятное количество прочтений, отделяющих одно прочтение от его парного прочтения, присутствующего в определенном блоке дескрипторов позиции;
- **ошибка спаривания позиций (PPE)**: определяется как разница между MPPD или MPPPD и фактической позицией парного прочтения;
- **якорь спаривания**: позиция последнего нуклеотида первого прочтения в паре, используемая в качестве точки отсчета для вычисления расстояния до парного прочтения в количестве позиций нуклеотидов или количестве позиций прочтений.

На Фиг. 5 показано, как вычисляют расстояние спаривания среди пар прочтений.

Блок дескрипторов pair представляет собой вектор ошибок спаривания, вычисленных как количество прочтений, которые нужно пропустить, чтобы достичь парного прочтения первого прочтения пары относительно определенного расстояния спаривания декодирования.

На фиг. 6 показан пример порядка вычисления ошибок спаривания, как в виде абсолютного значения, так и в виде разностного вектора (отличающегося низкой энтропией при высоких покрытиях).

Те же самые дескрипторы используют для информации о спаривании прочтений, принадлежащих классам N, M, P и I. Чтобы обеспечить выборочный доступ к разным классам данных, информацию о спаривании прочтений, принадлежащих этим четырем классам, кодируют в разных блоках, как изображено на Фиг. 8 (класс N), Фиг. 10, 12 и 14 (класс M) и Фиг. 15 и 16 (класс I).

Информация о спаривании в случае прочтений, картированных на разные референсные последовательности

В процессе картирования прочтений последовательности на референсную последовательность нередко первое прочтение в паре картируют на одну референсную последовательность (например, хромосому 1), а вторую — на другую референсную последовательность (например, хромосому 4). В этом случае описанную выше информацию об образовании пар необходимо объединять с дополнительной информацией, относящейся к референсной последовательности, используемой для картирования одного из прочтений. Это достигают путем кодирования:

1. Зарезервированного значения (флага), указывающего, что пару картируют на две разных последовательности (разные значения указывают, картировано ли прочтение

1 или прочтение 2 на последовательность, которая не кодирована в настоящее время).

2. Уникального идентификатора референса, относящегося к идентификаторам референса, кодированным в главной структуре заголовка, как описано в таблице 1.
3. Третий элемент содержит информацию о картировании на референс, указанный в пункте 2, и выражен в виде смещения относительно последней кодированной позиции.

Пример данного сценария приведен на Фиг. 7.

Поскольку на Фиг. 7 прочтение 4 не картировано на текущую кодируемую референсную последовательность, геномный кодер сигнализирует об этой информации, создавая дополнительные дескрипторы в блоке pair. В примере, показанном ниже, прочтение 4 пары 2 картируют на референс № 4, тогда как в данный момент картируют референс № 1. Эту информацию кодируют с помощью 3 компонентов:

1) Одно специальное зарезервированное значение кодируют как расстояние спаривания (в данном случае 0xffffff).

2) Второй дескриптор обеспечивает ИД референса, который перечислен в главного заголовке (в данном случае 4).

3) Третий элемент содержит информацию о картировании на соответствующий референс (170).

Дескрипторы несовпадения для прочтений класса N

Класс N включает в себя все прочтения, в которых имеются только несовпадения «n-типа» — вместо основания A, C, G или T в качестве определенного основания обнаружено N. Все другие основания прочтения идеально совпадают с референсной последовательностью.

На Фиг. 8 показано, как:

позиции «N» в прочтении 1 кодируют в виде

- абсолютной позиции в прочтении 1 или
- виде разностной позиции относительно предыдущего «N» в этом же прочтении.

Позиции «N» в прочтении 2 кодируют в виде

- абсолютной позиции в прочтении 2 + длина прочтения 1 или
- разностная позиция относительно предыдущей N.

В блоке **nmis** кодирование каждой пары прочтений завершают специальным символом «разделителя».

Дескрипторы, кодирующие замены (несовпадения и ОНП), инсерции и делеции

Замену определяют как наличие в картированном прочтении другого основания нуклеотида по сравнению с присутствующим в референсной последовательности в этой же позиции.

На Фиг. 9 показаны примеры замен в картированной паре прочтений. Каждую замену кодируют в виде «позиции» (блок snpp) и «типа» (блок snpt). В зависимости от статистической встречаемости замен, инсерций или делеций можно определять разные модели источника соответствующих дескрипторов и формировать символы, кодируемые в соответствующем блоке.

Модель источника 1: замены в виде позиций типов

Дескрипторы позиций замен

Позицию замены вычисляют аналогично значениям блока nmis, а именно:

В прочтении 1 замены кодируют:

- как абсолютную позицию в прочтении 1, или
- как разностную позицию относительно предыдущей замены в этом же прочтении.

В прочтении 2 замены кодируют:

- как абсолютную позицию в прочтении 2 + длину прочтения 1, или
- как разностную позицию относительно предыдущей замены.

На Фиг. 10 показано, как замены (где в данной позиции картирования символ в прочтении отличается от символа в референсной последовательности) кодируют в виде:

1. позиции несовпадения:
    - относительно начала прочтения, или
    - относительно предыдущего несовпадения (дифференциальное кодирование).
2. типа несовпадения, представленного в виде кода, вычисленного, как описано на Фиг. 10.

В блоке snpp кодирование каждой пары прочтений завершают специальным символом «разделителя».

Дескрипторы типов замен

Для класса M (и для класса I, как описано в следующих разделах) несовпадения кодируют посредством индекса (перемещения справа налево) от фактического символа, присутствующего в референсной последовательности, к соответствующему символу замены, присутствующему в прочтении (A, C, G, T, N, Z). Например, если в выровненном прочтении присутствует C вместо T, которая присутствует в этой же позиции в референсной последовательности, индекс несовпадения будет обозначен как «4». Процесс декодирования считывает кодированный дескриптор, нуклеотид в данной позиции на референсе и перемещается слева направо для извлечения декодированного символа. Например, «2», принятое для позиции, где в референсе присутствует G, будет декодировано как «N». На Фиг. 11 показаны все возможные ситуации и соответствующие символы кодирования. Чтобы свести к минимуму энтропию дескрипторов, каждому индексу могут быть назначены явно отличающиеся друг от друга и контекстно-адаптивные вероятностные модели в соответствии со статистическими свойствами каждого типа замены для каждого класса данных.

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

На Фиг. 12 приведен пример кодирования типов замен в блоке ***snpt***.

Некоторые примеры кодирования замен при внедрении кодов неоднозначности IUPAC приведены на Фиг. 13. Еще один пример индексов замен приведен на Фиг. 14.

Кодирование инсерций и делеций

Для класса I несовпадения и делеции кодируют посредством индексов (перемещения справа налево) от фактического символа, присутствующего в референсной последовательности, к соответствующему символу замены, присутствующему в прочтении: {A, C, G, T, N, Z}. Например, если в выровненном прочтении присутствует C вместо T в той же позиции в референсе, индексом несовпадения будет «4». В случае, когда в прочтении присутствует делеция в позиции, где в референсе присутствует «A», кодированным символом будет «5». Процесс декодирования считывает кодированный дескриптор, нуклеотид в данной позиции на референсе и переходит слева направо для извлечения декодированного символа. Например, «3», принятое для позиции, где в референсе присутствует G, будет декодировано как «Z».

Для вставленных A, C, G, T, N инсерции кодируют как 6, 7, 8, 9, 10, соответственно.

На Фиг. 15 показан пример, как кодируют замены, инсерции и делеции в парах прочтений класса I. Для поддержки полного набора кодов неоднозначности IUPAC вектор замены S= {A, C, G, T, N, Z} следует заменить на S = {A, C, G, T, N, Z, M, R, W, S, Y, K, V, H, D, B}, как описано в предыдущем абзаце для несовпадений. В данном случае, если вектор замены имеет 16 элементов, коды инсерции должны иметь разные значения, а именно: 16, 17, 18, 19, 20. Этот механизм проиллюстрирован на Фиг. 16.

Модель источника 2: один блок на тип замены и инделы

Для некоторых статистик данных может быть разработана отличающаяся от описанной в предыдущем разделе модель кодирования замен и инделов, которая приводит к источнику с более низкой энтропией. Такая модель кодирования является альтернативой методам, описанным выше только для несовпадений и для несовпадений и инделов.

В данном случае определяют один блок данных для каждого возможного символа замены (5 без кодов IUPAC, 16 с кодами IUPAC) плюс один блок для делеций и еще 4 блока для инсерций. Для простоты объяснения, но не в качестве ограничения для применения модели, дальнейшее описание будет сконцентрировано на случае, когда коды IUPAC не поддерживаются.

На Фиг. 17 показано, как каждый блок содержит позицию несовпадений или инсерций одного типа. Если в кодируемой паре прочтений нет несовпадений или инсерций данного типа, в соответствующем блоке кодируют 0. Чтоб декодер мог начать процесс декодирования для блоков, описанных в данном разделе, заголовок каждого блока доступа содержит флаг, указывающий посредством сигнализации первый блок, подлежащий кодированию. В примере на Фиг. 18 первый элемент, подлежащий декодированию, находится в позиции 2 в блоке C. Когда в паре прочтений нет несовпадений или инделов данного типа, в соответствующие блоки добавляют 0. Когда на стороне декодирования указатель декодирования для каждого блока указывает на значение 0, процесс декодирования переходит к следующей паре прочтения.

Кодирование дополнительных флагов сигнализации

Для каждого класса данных, введенного выше (P, M, N, I), может потребоваться кодирование дополнительной информации о характере кодируемых прочтений. Эта информация может относиться, например, к эксперименту секвенирования (например, указание вероятности дублирования прочтения) или может выражать некоторые характеристики картирования прочтения (например, первое или второе в паре). В

контексте настоящего изобретения эту информацию кодируют в отдельном блоке для каждого класса данных. Основное преимущество такого подхода состоит в возможности выборочного доступа к этой информации только в случае необходимости и только в требуемой области референсной последовательности. Другие примеры использования таких флагов:

- парные прочтения;
- прочтение картированное в правильной паре;
- некартированное прочтение или парное прочтение;
- прочтение или парное прочтение из обратной цепочки;
- первое/второе прочтение в паре;
- не основное выравнивание;
- прочтение не проходит проверки качества платформы/поставщика;
- прочтение представляет собой дубликат ПЦР или оптический дубликат;
- дополнительное выравнивание.

Дескрипторы для класса U и построение «внутренних» референсов для некартированных прочтений «класса U» и «класса HM»

В случае прочтений, принадлежащих классу U или некартированной паре «класса HM», поскольку они не могут быть картированы ни на одну «внешнюю» референсную последовательность, удовлетворяющую заданному набору ограничений на точность совпадения для принадлежности классу P, N, M или I, для сжатого представления прочтений, принадлежащих этим классам данных, строят и используют одну или более «внутренних» референсных последовательностей.

К построению подходящих «внутренних» референсов возможны несколько подходов, таких как, для примера, но не ограничения:

- Разделение некартированных прочтений на кластеры, содержащие прочтения, у которых имеется общая непрерывная геномная последовательность по меньшей мере минимального размера (сигнатура). Каждый кластер может быть единственным образом идентифицирован по его сигнатуре, как показано на Фиг. 22.
- Сортировка прочтений в любом имеющем смысл порядке (например, в лексикографическом порядке) и использование последних N прочтений в качестве «внутреннего» референса для кодирования (N + 1)-го. Этот способ показан на Фиг. 23.

- Выполнение так называемой «сборки заново» на подмножестве прочтений класса U для получения возможности выравнивания и кодирования всех или соответствующего подмножества прочтений, принадлежащих этому классу в соответствии с заданными ограничениями на точность совпадения или новым набором ограничений.

Если кодируемое прочтение может быть картировано на «внутренний» референс, удовлетворяющий заданному набору ограничений на точность совпадения, информацию, необходимую для восстановления прочтения после сжатия, кодируют с помощью дескрипторов, которые могут быть одного из следующих типов:

1. Начальная позиция совпадающей части на внутреннем референсе в виде номера прочтения во внутреннем референсе (блок pos). Эту позицию можно кодировать в виде абсолютного или разностного значения относительно ранее кодированного прочтения.
2. Смещение начальной позиции от начала соответствующего прочтения во внутреннем референсе (блок pair). Например, в случае постоянной длины прочтения фактической позицией будет pos *length + pair.
3. Возможно присутствующие несовпадения, кодированные в виде позиции (блок snpp) и типа (блок snpt).
4. Те части прочтений (обычно края, указываемые посредством pair), которые не совпадают с внутренним референсом (или совпадают, но с количеством несовпадений выше заданного порога), кодируют в блоке indc. Для снижения энтропии несовпадений, кодируемых в блоке indc, можно применить операцию заполнения к краям части используемого внутреннего референса, как показано на Фиг. 24. Наиболее подходящая стратегия заполнения может быть выбрана кодером в соответствии со статистическими свойствами обрабатываемых геномных данных. Возможные стратегии заполнения:
    a. без заполнения;
    b. постоянный шаблон заполнения, выбираемый в соответствии с его частотой в текущих кодируемых данных;
    c. переменный шаблон заполнения в соответствии со статистическими свойствами текущего контекста, определенными посредством последних N кодированных прочтений.

Конкретный тип стратегии заполнения будет сигнализирован специальными значениями в заголовке блока indc.

5. Флаг, который указывает, кодировано ли прочтение с помощью внутреннего самоформируемого референса, внешнего референса или без референса (блок rtype).
6. Прочтения, которые кодируют буквально (ureads).

Пример такой процедуры кодирования приведен на Фиг. 24.

На Фиг. 25 показано альтернативное кодирование накартированных прочтений на внутренний референс, где дескрипторы pos + pair заменены дескриптором pos со знаком. В данном случае pos выражает расстояние, в единицах позиций, на референсной последовательности — позиции левого крайнего нуклеотида прочтения n относительно позиции крайнего левого нуклеотида прочтения n - 1.

В случае наличия прочтений переменной длины класса U для сохранения длины каждого прочтения используют дополнительный дескриптор rlen.

Этот подход к кодированию можно расширить для поддержки N начальных позиций для каждого прочтения, чтобы прочтения могли быть разбиты на две и более позиций референса. Это может быть особенно полезно для кодирования прочтений, формируемых с помощью тех технологий секвенирования (например, от Pacific Bioscience), которые создают очень длинные прочтения (свыше 50К оснований), в которых обычно присутствуют повторяющиеся структуры, образуемые за счет циклов в технологии секвенирования. Такой же подход можно использовать и для кодирования химерных прочтений последовательности, определяемых как прочтения, которые выравнены на две разные позиции генома с небольшим перекрытием или без него.

Ясно, что описанный выше подход может быть применен за рамками простого класса U и мог бы быть применен к любому блоку, содержащему дескрипторы, относящиеся к позициям прочтений (блоки pos).

Дескриптор оценки выравнивания

Дескриптор **mscore** обеспечивает оценку для каждого выравнивания. В контексте настоящего изобретения его используют для представления оценки картирования/выравнивания для каждого прочтения, формируемого средствами выравнивания прочтений геномной последовательности.

Оценку выражают с помощью степени и мантиссы. Количество битов, используемых для представления степени и мантиссы, передают в качестве параметров конфигурации. В качестве примера, но не ограничения, в таблице 2 показано, как это определено в стандарте IEEE RFC 754 для 11-битовой степени и 52-битовой мантиссы.

Оценку каждого выравнивания можно представить с помощью:

- одного бита знака (S);
- 11 битов для степени (E);
- 53 битов для мантиссы (M).

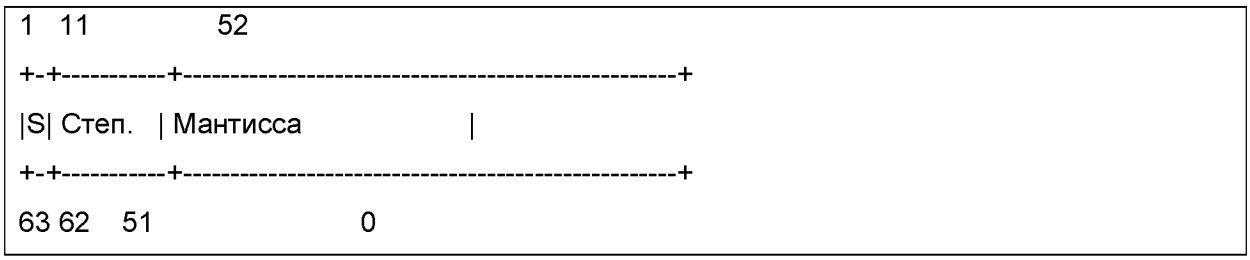


Таблица 2. Оценки выравнивания могут быть выражены в виде 64-битовых значений двойной точности с плавающей запятой

Основание (radix), предназначенное для вычисления оценок, равно 10, поэтому:

оценка = $-1^{s} \times 10^{E} \times M$

Группы прочтений

Во время процесса секвенирования могут быть получены секвенированные прочтения разных типов. В качестве примера, но не ограничения, типы могут относиться к разным секвенированным образцам, разным экспериментам, разным конфигурациям секвенатора. После секвенирования и выравнивания эту информацию сохраняют в соответствии с настоящим изобретением с помощью специально предназначенного дескриптора под названием **rgroup. rgroup** — это метка, связанная с каждым кодированным прочтением, которая позволяет декодирующему устройству разделять декодированные прочтения на группы после декодирования.

Дескрипторы для множественных выравниваний

Следующие дескрипторы определяют для поддержки множественных выравниваний. Для случая наличия сплайсированных прочтений в настоящем изобретении определен глобальный флаг **spliced_reads_flag**, который должен быть установлен на 1.

Дескриптор mmap

Дескриптор **mmap** используют для сигнализации о том, сколько позиций прочтения или крайнего левого прочтения пары были выровнены. Геномная запись, содержащая множественные выравнивания, связана с одним многобайтовым дескриптором **mmap**. Первые два байта дескриптора **mmap** представляют целое число без знака N, которое относится к прочтению как к единому сегменту (если в кодированном наборе данных нет сплайсов) или, напротив, ко всем сегментам, на которые было сплайсировано прочтение

для нескольких возможных выравниваний (если в наборе данных присутствуют сплайсы). Значение N указывает, сколько значений дескриптора **pos** кодированы для шаблона в данной записи. После N следуют одно или более целых чисел без знака M_i, как описано ниже.

Цепочечность множественных выравниваний

Дескриптор **rcomp**, описанный в настоящем изобретении, используют для указания цепочечности каждого выравнивания прочтения с помощью синтаксиса, определенного в настоящем изобретении.

Оценки множественных выравниваний

В случае множественных выравниваний каждому выравниванию назначают один дескриптор **mscore**, определенный в настоящем изобретении.

Множественные выравнивания без сплайсов

Если в блоке доступа нет сплайсов, **spliced_reads_flag** не установлен.

При парноконцевом секвенировании дескриптор **mmap** состоит из 16-битового целого числа без знака N, за которым следуют одно или более 8-битовых целых чисел без знака M_i, причем *i* принимает значения от 1 до количества полных выравниваний первого (в данном случае крайнего левого) прочтения. Для каждого выравнивания первого прочтения, сплайсированного или нет, M_i используют для сигнализации о том, сколько сегментов использованы для выравнивания второго прочтения (в данном случае, без сплайсов, оно равно количеству выравниваний), и, кроме того, сколько значений дескриптора **pair** кодированы для данного выравнивания первого прочтения.

Значения M_i используют для вычисления $P = \sum_{i=1}^{N} M_i$, которое указывет количество выравниваний второго прочтения.

Специальное значение M_i ($M_i = 0$) указывает, что i-е выравнивание крайнего левого прочтения спарено с выравниванием крайнего правого прочтения, которое уже спарено с k-м выравниванием крайнего левого прочтения, причем k < i (в таком случае *новое* выравнивание не обнаружено, что согласуется с приведенным выше уравнением).

В качестве примера в самых простых случаях:

1 Если имеются одно выравнивание для крайнего левого прочтения и два альтернативных выравнивания для крайнего правого прочтения, N будет равно 1, а M_1 будет равно 2.

2 Если обнаружены два альтернативных выравнивания для крайнего левого прочтения, но лишь одно для крайнего правого прочтения, N будет равно 2, M_1 будет равно 1 и M_2 будет равно 0.

Когда M_i равно 0, соответствующе значение **pair** должно быть привязано к существующему выравниванию второго прочтения; в противном случае возникнет синтаксическая ошибка, и выравнивание будет считаться нарушенным.

Пример: если первое прочтение имеет две позиции картирования, а второе прочтение только одну, N равно 2, M_1 равно 1 и M_2 равно 0, как указано выше. Если за этим следует другое альтернативное вспомогательное картирование для всего шаблона, N будет равно 3, а M_3 будет равно 1.

Фиг. 39 иллюстрирует значение N, P и M_i в случае множественных выравниваний без сплайсов, а на **Ошибка! Источник ссылки не найден.**Фиг. 10 показано, как дескрипторы **pos**, **pair** и **mmap** используют для кодирования информации о множественных выравниваниях.

В отношении Фиг. 40 справедливо следующее:

- Крайнее правое прочтение имеет $P = \sum_{i=1}^{N} M_i$ выравниваний.
- Некоторые значения M_i могут быть = 0, когда i-е выравнивание крайнего левого прочтения спарено с выравниванием крайнего правого прочтения, которое уже же спарено с k-м выравниванием крайнего левого прочтения, причем k < i.
- Может присутствовать одно зарезервированное значение дескриптора pair для сигнализации о принадлежности выравниваний другим диапазонам блоков доступа. При наличии такового оно всегда является первым дескриптором pair для текущей записи.

Множественные выравнивания со сплайсами

При кодировании набора данных со сплайсированными прочтениями дескриптор **msar** позволяет представлять длину и цепочечность сплайсов.

После декодирования дескрипторов **mmap** и **msar** декодер знает, сколько прочтений или пар прочтений было кодировано для представления нескольких картирований, и сколько сегментов составляют каждое картирование каждого прочтения или пары прочтений. Это показано на Фиг. 41 и Фиг. 42.

В отношении Фиг. 41 справедливо следующее:

- Крайнее левое прочтение имеет N_1 выравниваний с N сплайсами ($N_1 \leq N$).
- N представляет количество сплайсов, присутствующих во всех выравниваниях крайнего левого прочтения, и его кодируют в качестве первого значения дескриптора **mmap**.
- Крайнее правое прочтение имеет P = $\sum_{i=1}^{N1} Mi$ сплайсов, где M_i *представляет собой* количество сплайсов крайнего правого прочтения, которое связано в пару с i-м выравниванием крайнего левого прочтения ($1 \leq i \leq N_1$). Другими словами, P представляет количество сплайсов крайнего правого прочтения и вычисляется с помощью N значений, следующих за первым значением дескриптора **mmap**.
- N_1 и N_2 представляют количество выравниваний первого и второго прочтения и вычисляются с помощью N + P значений дескриптора **msar**.

В отношении Фиг. 42 справедливо следующее:

- Крайнее левое прочтение имеет N_1 выравниваний с N сплайсами($N_1 \leq N$). Если N_1 = N И N_2 = P, сплайсы будут отсутствовать.
- Крайнее правое прочтение имеет P = $\sum_{i=1}^{N1} Mi$ сплайсов (t_j $1 \leq j \leq P$) = и N_2 ($N_2 \leq P$) выравниваний.
- Количество дескрипторов **pair** можно вычислить как NP = Max(N1, P) + M_0 где:
    - M_0 — количество M_i со значением 0;
    - NP должно приращиваться на 1, и в таком случае один специальный дескриптор pair указывает на присутствие выравниваний в других БД.

Оценка выравнивания

Дескриптор **mscore** позволяет посредством сигнализации указывать оценку выравнивания. При одноконцевом секвенировании он будет иметь N_1 значений на шаблон; при парноконцевом секвенировании он будет иметь значение для каждого выравнивания всего шаблона (количество различных выравнивание первого прочтения и, возможно, плюс количество дополнительных выравниваний второго прочтения, т. е., когда $M_i - 1 > 0$).

Количество оценок = MAX(N_1, N_2) + M_0

где M_0 представляет общее количество M_i = 0.

В настоящем изобретении с каждым выравниванием могут быть связаны более одного значения оценки. Количество выравниваний сигнализирует параметр конфигурации **as_depth.**

Дескрипторы для множественных выравниваний без сплайсов

5

Семантика	mmap	mscore	Результат
Прочтение (или парноконцевое прочтение) с несколькими картированиями, не сплайсированное	Одиночное прочтение: N Пара прочтений: N, M_i где $1 \le i \le N$	Одиночное прочтение: N значений Пара прочтений: $MAX(N, \sum(M_i))$ значений + $\sum(M_i = 0)$ Введение разделителя позволяет иметь произвольное количество оценок. В противном случае, если он не присутствует, следует использовать 0. Это значения с плавающей запятой, как определено в настоящем изобретении.	Одиночное прочтение: прочтение имеет несколько картирований, и его кодируют в виде последовательности из N последовательных сегментов, принадлежащих классу с самым высоким идентификатором. Используют N дескрипторов pos. Пара прочтений: пара прочтений имеет несколько картирований, и ее кодируют в виде последовательности из: • N сегментов для первого прочтения; • $P = \sum(M_i)$ спариваний для выравниваний второго прочтения. Используют N дескрипторов **pos**. Используют N x P дескрипторов **pair**, причем

			NP = $\sum_{i=1}^{N} Mi$ + (к-во Mi = 0) + (необязательно) 1 Необязательный дескриптор спаривания используют, когда выравнивания присутствуют на других референсных последовательностях, а не на кодируемой в данный момент. N + 1 дескрипторов mmap
Прочтение (пара), картированное единственным образом	0	0	

Таблица 3. Определение количества дескрипторов, необходимых для представления множественных выравниваний в одной геномной записи в случае множественных выравниваний без сплайсов

Дескрипторы для множественных выравниваний со сплайсами

В таблице 4 показано, как определять количество дескрипторов, необходимых для представления множественных выравниваний в одной геномной записи в случае множественных выравниваний без сплайсов.

Семантика	mmap	mscore	Результат
Прочтение (или парноконцевое прочтение) с несколькими картированиями, со сплайсами	Одиночное прочтение: N Пара прочтений: N, M_i	Одиночное прочтение: N_1 значений Пара прочтений: $Max(N_1, N_2)$ + $\sum(M_i == 0)$ значений N_1 и N_2 с помощью N + P дескрипторов **msar**	Одиночное прочтение: прочтение имеет несколько картирований, и его кодируют в виде последовательности из N последовательных

	где 1 ≤ i ≤ N	$P = \sum_{i=1}^{N1} Mi$ Это значения с плавающей запятой, как определено в настоящем изобретении.	сегментов, принадлежащих классу с самым высоким идентификатором. Используют N дескрипторов **pos**. Пара прочтений: пара прочтений имеет несколько картирований, и ее кодируют в виде последовательности из: • N сегментов для первого прочтения; • P =∑(M_i) спариваний для выравниваний второго прочтения. Используют N дескрипторов pos.
Прочтение (пара), картированное единственным образом	0	0	

Таблица 4. Дескрипторы, используемые для представления множественных выравниваний и соответствующих оценок

Множественные выравнивания на разных последовательностях

Может получиться так, что в процессе выравнивания обнаружатся альтернативные картирования на другую референсную последовательность, а не на ту, на которой расположено основное картирование.

Для пар прочтений, которые выровнены единственным образом, дескриптор **pair** следует использовать для представления абсолютных позиций прочтения, когда имеется, например, химерическое выравнивание с парным прочтением на другой хромосоме. Дескриптор **pair** следует использовать для указания посредством сигнализации референса и позиции следующей записи, содержащей дополнительные выравнивания

для того же самого шаблона. Последняя запись (например, третья, если альтернативные картирования кодируют в 3 разных БД) должна содержать референс и позицию первой записи.

В случае, когда одно или более выравниваний для крайнего левого прочтения в паре присутствуют не на той референсной последовательности, которая относится к текущему кодируемому БД, для дескриптора **pair** использую зарезервированное значение. После зарезервированного значения следуют идентификатор референсной последовательности и позиция крайнего левого выравнивания среди всех содержащихся в следующем БД (т. е. первое декодированное значение дескриптора **pos** для этой записи).

Множественные выравнивания с инсерциями, делециями, некартированными частями

Когда альтернативное вспомогательное картирование не сохраняет непрерывность референсной области, где выравнивают последовательность, восстановление точного картирования, сформированного средством выравнивания, может оказаться невозможным, так как фактическая последовательность (а, значит, дескрипторы, связанные с несовпадениями, такими как замены или инделы) кодирована только для основного выравнивания. Для представления того, как вспомогательные выравнивания картированы на референсную последовательность в случае, когда они содержит инделы и/или мягкие отсечения, следует использовать дескриптор **msar**. Если **msar** представлен специальным символом «*» для вспомогательного выравнивания, декодер восстановит вспомогательное выравнивание из основного выравнивания и позиции картирования вспомогательного выравнивания.

Дескриптор msar

Дескриптор **msar** (Multiple Segments Alignment Record) поддерживает сплайсированные прочтения и альтернативные вспомогательные выравнивания, которые содержат инделы или мягкие отсечения.

msar предназначен для передачи информации о:

- длине картированного сегмента;
- другой непрерывности картирования (т. е. наличии инсерций, делеций или отсеченных оснований) для вспомогательного выравнивания и/или сплайсированного прочтения.

В **msar** используют синтаксис расширенной записи CIGAR, описанный ниже, плюс дополнительный символ, описанный в Таблица 5.

Таблица 5. Специальный символ, используемый для дескриптора msar в дополнение к синтаксису, описанному в таблице 6.

Символ	Семантика	Описание
*	Вспомогательное выравнивание не содержит инделов или мягких отсечений	Это используют, когда восстановление вспомогательного выравнивания не требует никакой дополнительной информации, кроме позиции выравнивания и основного выравнивания

Расширенный синтаксис cigar

В данном разделе определен расширенный синтаксис CIGAR (E-CIGAR) для строк, которые должны быть связаны с последовательностями и соответствующими несовпадениями, инделами, отсеченными основаниями и информацией о множественных выравниваниях и сплайсированных прочтениях.

Операции редактирования, описанные в настоящем изобретении, перечислены в таблице 6.

Операция	**Семантика**	**Представление E-CIGAR**	**Эквивалент представления CIGAR для SAM**
Приращение указателя на референс *R* и указателя на прочтение *r* на *n* позиций (совпадение)	n совпадающих оснований	n=	nM в более старых версиях (не эквивалентно), = в недавних версиях
Замена нуклеотида в прочтении основанием C из референса, приращение указателя на референс *R* и указателя на прочтение *r* на 1	Замена символа C (C присутствует в прочтении и отсутствует в референсе)	C	M в более старых версиях, X в недавних версиях (не эквивалентно)

Приращение указателя на прочтение *r* на *n* позиций (вставка из прочтения)	n оснований вставляют в прочтение (не присутствуют в референсе)	n+	nI
Приращение указателя на референс *R* на *n* позиций (удаление последовательности S в прочтении)	n оснований удаляют из прочтения (но присутствуют в референсе)	n-	nD
Приращение указателя на референс *R* на *n* позиций (вставка в прочтение). Может иметь место только в начале или конце прочтения	n мягких отсечений	(n)	nS
Жесткое обрезание. Может иметь место только в начале или конце прочтения	n жестких отсечений	[n]	nH
Приращение указателя на референс *R* на *n* позиций с соблюдением консенсуса сплайса (сплайс в прочтении)	Ненаправленный сплайс из n оснований	n*	nN
Приращение указателя на референс *R* на *n* позиций, соблюдение консенсуса сплайса на прямой цепочке (прямой сплайс в прочтении)	Прямой сплайс из n оснований	n/	Не существует
Приращение указателя на референс *R* на *n* позиций,	Обратный сплайс из n оснований	n%	Не существует

соблюдение консенсуса сплайса на обратной цепочке (обратный сплайс в прочтении)			

Таблица 6. Синтаксис строки E-CIGAR в формате MPEG-G

Модели источника, энтропийные кодеры и режимы кодирования

Для каждого класса данных, подкласса и соответствующего блока дескрипторов структуры геномных данных в настоящем изобретении могут быть внедрены различные алгоритмы кодирования в соответствии с конкретными особенностями данных или метаданных, содержащихся в каждом блоке, и их статистическими свойствами. «Алгоритм кодирования» следует понимать как взаимосвязь конкретной «модели источника» блока дескрипторов с конкретным «энтропийным кодером». Конкретная «модель источника» может быть определена и выбрана для получения наиболее эффективного кодирования данных в смысле минимизации энтропии источника. При выборе энтропийного кодера можно руководствоваться соображениями эффективности кодирования и/или характеристиками вероятностного распределения и соответствующими проблемами реализации. Каждый выбор специфического «алгоритма кодирования», также называемого «режимом кодирования» может применяться к всему «блоку дескрипторов», связанному с классом или подклассом данных для всего набора данных, или может быть применен свой «режим кодирования» к каждой части дескрипторов, разделенных на блоки доступа.

Каждую «модель источника», связанную с режимом кодирования, характеризуют посредством:

- Определения дескрипторов, выдаваемых каждым источником (т. е. набора дескрипторов, используемых для представления класса данных, таких как позиция прочтений, информация о спаривании прочтений, несовпадения с референсной последовательностью, как определено в таблице 2).
- Определения связанной вероятностной модели.
- Определения связанного энтропийного кодера.

Дальнейшие преимущества

Классифицирование данных последовательности на определенные классы и подклассы данных позволяет реализовывать эффективные режимы кодирования, использующие

более низкую энтропию источника информации, отличающиеся моделированием последовательностей дескрипторов одиночными отдельными источниками данных (например, расстояние, позиция и т. д.).

Другим преимуществом изобретения является возможность доступа только к подмножеству данных интересующего типа. Например, одной из наиболее важных областей применения в геномике является поиск отличий геномного образца по сравнению с референсом (ОНВ) или популяцией (ОНП). Сегодня анализ такого рода требует обработки полных прочтений последовательности, тогда как при внедрении представления данных, раскрытого в настоящем изобретении, несовпадения уже изолированы всего лишь в три класса данных (если представляют интерес, рассматриваются также несовпадения «n-типа» и «i-типа»).

Еще одно преимущество состоит в возможности выполнения эффективного транскодирования данных и метаданных, сжатых относительно определенной «внешней» референсной последовательности, в сжатые относительно другой «внешней» референсной последовательности при опубликовании новых референсных последовательностей или при выполнении повторного картирования на уже картированные данные (например, с помощью другого алгоритма картирования) с получением новых выравниваний.

На Фиг. 20 показано кодирующее устройство 207 в соответствии с принципами настоящего изобретения. Кодирующее устройство 207 принимает в качестве входа необработанные данные 209 последовательности, например, созданные устройством 200 секвенирования генома. В данной области известны устройства для секвенирования генома, такие как приборы Illumina HiSeq 2500, Thermo-Fisher Ion Torrent. Необработанные данные 209 последовательности подают в узел 201 выравнивателя, который подготавливает последовательности для кодирования путем выравнивания прочтений на референсную последовательность 2020. В альтернативном варианте осуществления может быть использован специально предназначенный модуль 202 для формирования референсной последовательности из имеющихся прочтений с использованием различных стратегий, как описано в настоящем документе в разделе «Построение внутренних референсов для некартированных прочтений "класса U" и "класса НМ"». После обработки генератором 202 референса прочтения могут быть картированы на полученную более длинную последовательность. Затем выровненные последовательности классифицируют с помощью модуля 204 классифицирования. После

этого к референсу применяют этап преобразования референса, чтобы снизить энтропию данных, сформированных узлом 204 классифицирования данных. Он включает в себя обработку внешнего референса 2020 в узле 2019 преобразования референса, который создает преобразованные классы 2018 данных и дескрипторы 2021 преобразования референса. Затем преобразованные классы 2018 данных подают в кодеры 205–207 блоков вместе с дескрипторами 2021 преобразования референса. Посл этого геномные блоки 2011 подают в арифметические кодеры 2012–2014, которые кодируют блоки в соответствии со статистическими свойствами данных или метаданных, содержащихся в блоке. В результате получается геномный поток 2015.

На Фиг. 21 показано декодирующее устройство 218 в соответствии с принципами настоящего изобретения. Декодирующее устройство 218 принимает мультиплексированный геномный поток 2110 двоичных данных из сети или элемента запоминающего устройства. Мультиплексированный геномный поток 2110 двоичных данных подают в демультиплексор 210 для создания отдельных потоков 211, которые затем подают в энтропийный декодеры 212–214 для получения геномных блоков 215 и дескрипторов 2112 преобразования референса. Выделенные геномные блоки подают в декодеры 216–217 блоков для дальнейшего декодирования в классы данных, а дескрипторы преобразования референса подают в узел 2113 преобразования референса. Декодеры 219 классов выполняют дальнейшую обработку геномных дескрипторов 2111 и преобразованного референса 2114 и объединяют результаты для получения несжатых прочтений последовательностей, которые затем также сохраняют в форматах, известных в данной области техники, например, в текстовом файле или сжатом файле формата zip, или в файлах FASTQ или SAM/BAM.

Декодеры 219 классов могут восстанавливать исходные геномные последовательности за счет использования информации об исходных референсных последовательностях, содержащейся в одном или более геномным потоках, и дескрипторов 2112 преобразования референса в кодированном потоке двоичных данных. Если референсные последовательности не транспортируют посредством геномных потоков, они должны быть в наличии на стороне декодирования и доступны для декодеров классов.

Обладающие признаками изобретения методы, описанные в настоящем документе, могут быть реализованы в оборудовании, программном обеспечении, встроенном программном обеспечении или любой их комбинации. При реализации в программном обеспечении они могут храниться на машиночитаемом носителе информации и исполняться аппаратным

процессорным устройством. Аппаратное процессорное устройство может содержать один или более процессоров, цифровых сигнальных процессоров, микропроцессоров общего назначения, специализированных интегральных схем или других дискретных логических схем.

Методы данного описания могут быть реализованы в самых разных устройствах и приборах, включая мобильные телефоны, настольные компьютеры, серверы, планшеты и подобные устройства.

Формат файла: выборочный доступ к областям геномных данных с помощью таблицы главного индекса

Для поддержки выборочного доступа к определенным областям выровненных данных структура данных, описанная в настоящем документе, реализует средство индексации, называемое таблицей главного индекса (MIT). Это многомерный массив, содержащий местоположения картирования конкретных прочтений на связанные референсные последовательности. Значения, содержащиеся в MIT, представляют собой позиции картирования первого прочтения в каждом блоке ***pos*** в целях поддержки непоследовательного доступа к каждому блоку доступа. MIT содержит один раздел на каждый класс данных (P, N, M, I, U и HM) и каждую референсную последовательность. MIT содержится в заголовке геномного набора данных кодированных данных. На Фиг. 31 показана структура заголовка геномного набора данных, на Фиг. 32 показано общее визуальное представление MIT, а на Фиг. 33 показан пример MIT для кодированных прочтений класса P.

Изображенные на Фиг. 33 значения, содержащиеся в MIT, используют для прямого доступа к интересующей области (и соответствующему БД) в сжатой области.

Например, как показано на Фиг. 33, если требуется получить доступ к области, содержащейся между позициями 150 000 и 250 000 на референсе 2, декодирующее приложение должно перейти сразу ко второму референсу в MIT и найти два значения k1 и k2 так, чтобы k1 < 150 000 и k2 > 250 000. Где k1 и k2 являются двумя индексами, считанными из MIT. В примере на Фиг. 33 в результате будут получены 3-я и 4-я позиции второго вектора MIT. Эти возвращенные значения будут затем использованы декодирующим приложением для извлечения позиций соответствующих данных из таблицы локального индекса блока ***pos***, как описано в следующем разделе.

Вместе с указателями на блок, содержащий данные, принадлежащие четырем классам геномных данных, описанным выше, MIT можно использовать в качестве индекса

дополнительных метаданных и/или аннотаций, добавленных к геномным данным в течение их срока службы.

Таблица локального указателя

В начале каждого блока геномных данных имеется структура данных, называемая ***локальным заголовком***. Локальный заголовок содержит уникальный идентификатор блока, вектор счетчиков блоков доступа для каждой референсной последовательности, таблицу локального индекса (LIT) и, необязательно, некоторые метаданные, специфичные для блока. LIT представляет собой вектор указателей на физическую позицию данных, принадлежащих каждому блоку доступа, в полезной нагрузке блока. На Фиг. 34 изображены общий заголовок блока и полезная нагрузка, где LIT используют для доступа к определенным областям кодированных данных непоследовательным образом.

В предыдущем примере для доступа к области в диапазоне от 150 000 до 250 000 прочтений, выровненных на референсную последовательность № 2, декодирующее приложение извлекло позиции 3 и 4 из MIT. Эти значения должны использоваться процессом декодирования для доступа к 3-му и 4-му элементам соответствующего раздела LIT. В примере, показанном на Фиг. 35, счетчики общего количества блоков доступа, содержащиеся в заголовке блока, используют для перехода сразу к индексам LIT, связанным с БД, относящимся к референсу 1 (5 в примере). Поэтому индексы, содержащие физические позиции запрашиваемых БД в кодированном потоке, вычисляют следующим образом:

позиция блоков данных, принадлежащих запрашиваемым БД = блокам данных, принадлежащим БД референса 1, которые нужно пропустить, + позиция, извлеченная с помощью MIT, т. е.:

Позиция первого блока: 5 + 3 = 8

Позиция последнего блока: 5 + 4 = 9

Блоки данных, извлеченные с помощью механизма индексации, называемого таблицей локального индекса, являются частью запрашиваемых блоков доступа.

На Фиг. 36 показано, как блоки, содержащиеся в MIT, соответствуют блокам LIT для каждого класса или подкласса данных.

На Фиг. 37 показано, как блоки данных, извлеченные с помощью MIT и LIT, образуют один или более блоков доступа, как определено в следующем разделе.

В варианте осуществления настоящего изобретения LIT может быть встроена в качестве подструктуры MIT. Преимуществом такого подхода является скорость доступа к

индексированным данным в случае последовательного синтаксического анализа сжатого файла. Если LIT встроена в MIT в заголовке файла, то в случае выборочного доступа декодирующему устройству нужно будет синтаксически анализировать только малую часть данных, чтобы извлекать запрашиваемую сжатую информацию. Другое преимущество, очевидное для специалиста в данной области, состоит в том, что в случае передачи в потоковом режиме по сети индексированная информация, содержащаяся в MIT и LIT, будет доставляться среди первых блоков данных, позволяя тем самым приемному устройству выполнять операции, такие как сортировка и выборочный доступ, до завершения передачи всех данных.

Блок доступа

Геномные данные, классифицированные в классы данных и структурированные в сжатые и несжатые блоки, организуют в разные блоки доступа.

Геномные блоки доступа (БД) определяют как разделы геномных данных (в сжатом и несжатом виде), которые восстанавливают нуклеотидные последовательности, и/или соответствующие метаданные, и/или последовательности ДНК/РНК (например, виртуальный референс), и/или данные аннотаций, формируемые секвенатором генома и/или устройством геномной обработки или приложением анализа. Пример блока доступа приведен на Фиг. 37.

Блок доступа представляет собой блок данных, который может быть декодирован либо независимо от других блоков доступа с использованием только глобально доступных данных (например, конфигурации декодера), либо с использованием информации, содержащейся в других блоках доступа.

Блоки доступа различают по:

- типу, характеризующему природу геномных данных и наборы данных, которые содержат эти блоки доступа, и возможный способ доступа к ним,
- порядку, обеспечивающему уникальный порядок блокам доступа, принадлежащим одному и тому же типу.

Блоки доступа любого типа можно дополнительно классифицировать на различные «категории».

Далее следует неисчерпывающий список определений геномных блоков доступа различных типов:

1) Для получения доступа или декодирования и получения доступа к блокам доступа типа 0 не требуется ссылка ни на какую информацию, поступающую из других блоков доступа. Вся информация в содержащихся в них данных или наборах данных, может быть независимо считана и обработана декодирующим устройством или приложением обработки.

2) Блоки доступа типа 1 содержат данные, которые ссылаются на данные, содержащиеся в блоках доступа типа 0. Для считывания или декодирования и обработки данных, содержащихся в блоках доступа типа 1, требуется наличие доступа к одному или более блокам доступа типа 0. Блок доступа типа 1 кодирует геномные данные, относящиеся к прочтениям последовательности «класса P».

3) Блоки доступа типа 2 содержат данные, которые ссылаются на данные, содержащиеся в блоках доступа типа 0. Для считывания или декодирования и обработки данных, содержащихся в блоках доступа типа 2, требуется наличие доступа к одному или более блокам доступа типа 0. Блок доступа типа 2 кодирует геномные данные, относящиеся к прочтениям последовательности «класса N».

4) Блоки доступа типа 3 содержат данные, которые ссылаются на данные, содержащиеся в блоках доступа типа 0. Для считывания или декодирования и обработки данных, содержащихся в блоках доступа типа 3, требуется наличие доступа к одному или более блокам доступа типа 0. Блоки доступа типа 3 кодируют геномные данные, относящиеся к прочтениям последовательности «класса M».

5) Блоки доступа типа 4 содержат данные, которые ссылаются на данные, содержащиеся в блоках доступа типа 0. Для считывания или декодирования и обработки данных, содержащихся в блоках доступа типа 4, требуется наличие доступа к одному или более блокам доступа типа 0. Блок доступа типа 4 кодирует геномные данные, относящиеся к прочтениям последовательности «класса I».

6) Блоки доступа типа 5 содержат прочтения, которые не могут быть картированы ни на одну из доступных референсных последовательностей («класс U»), и их кодируют с использованием внутренне построенной референсной

последовательности. Блоки доступа типа 5 содержат данные, которые ссылаются на данные, содержащиеся в блоках доступа типа 0. Для считывания или декодирования и обработки данных, содержащихся в блоках доступа типа 5, требуется наличие доступа к одному или более блокам доступа типа 0.

7) Блоки доступа типа 6 содержат пары прочтений, где одно прочтение может принадлежать любому из четырех классов P, N, M, I, а другое не может быть каритровано ни на одну из доступных референсных последовательностей («класс HM»). Блоки доступа типа 6 содержат данные, которые ссылаются на данные, содержащиеся в блоках доступа типа 0. Для считывания или декодирования и обработки данных, содержащихся в блоках доступа типа 6, требуется наличие доступа к одному или более блокам доступа типа 0.

8) Блоки доступа типа 7 содержат метаданные (например, оценки качества) и/или данные аннотаций, связанные с данными или наборами данных, содержащимися в блоке доступа типа 1. Блоки доступа типа 7 могут быть классифицированы и маркированы в различных блоках.

9) Блоки доступа типа 8 содержат данные или наборы данных, классифицированные как данные аннотаций. Блоки доступа типа 8 могут быть классифицированы и маркированы в блоках.

10) Блоки доступа дополнительных типов могут расширять описанные здесь структуры и механизмы. В качестве примера, но не ограничения, результаты определения геномных вариантов, структурного и функционального анализа могут быть кодированы в блоках доступа новых типов. Описанная в настоящем документе организация данных в виде блоков доступа не препятствует инкапсуляции данных любого типа в блоки доступа, что является механизмом, полностью прозрачным в отношении природы закодированных данных.

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, 4, 5 и 6 не нужно упорядочивать и не нужно хранить и/или передавать упорядоченным образом (техническое преимущество: параллельная обработка/параллельная потоковая передача).

На Фиг. 37 показано, как составляют блоки доступа с помощью заголовка и одного или более блоков однородных данных. Каждый блок может составлен из одного или более блоков. Каждый блок содержит несколько пакетов, а пакеты представляют собой структурированную последовательность дескрипторов, введенных выше для представления, например, позиций прочтений, информации о спаривании, информации об обратном комплементе, позиций и типов несовпадений и т. д.

У каждого блока доступа может быть разное количество пакетов в каждом блоке, но в пределах блока доступа у всех блоков одно и то же количество пакетов.

Каждый пакет данных может быть идентифицирован посредством комбинации из 3 идентификаторов X, Y и Z, где:

- X указывает блок доступа, которому он принадлежит;
- Y указывает блок, которому он принадлежит (т. е., тип данных, которые он инкапсулирует);
- Z представляет собой идентификатор, выражающий порядок пакета относительно других пакетов того же блока.

На фиг. 38 показан пример маркировки блоков доступа и пакетов, где ***AU_T_N*** — блок доступа типа T с идентификатором N, который может подразумевать или не подразумевать наличие порядка в соответствии с типом блока доступа. Идентификаторы используют для связывания единственным образом блоков доступа одного типа с блоками доступа других типов, требуемыми для полного декодирования содержащихся геномных данных.

Блоки доступа любого типа могут быть дополнительно классифицированы и маркированы в различных «категориях» в соответствии с различными процессами секвенирования. В качестве примера, но не ограничения, классификация и маркировка могут иметь место при

1. секвенировании одного и того же организма в разное время (блоки доступа содержат геномную информацию по геному с «временной» подоплекой),
2. секвенировании органических образцов различной природы некоторых организмов (например, кожи, крови, волос для человеческих образцов). Это блоки доступа с «биологической» подоплекой.

ФОРМУЛА ИЗОБРЕТЕНИЯ

1. Способ кодирования данных геномной последовательности, где данные геномной последовательности содержат прочтения последовательностей нуклеотидов, причем способ включает в себя этапы:

выравнивания прочтений на одну или более референсных последовательностей с созданием тем самым выровненных прочтений,

классифицирования выровненных последовательностей согласно заданным правилам сопоставления с одной или более референсных последовательностей с созданием тем самым классов выровненных прочтений,

кодирование классифицированных выровненных прочтений в виде множества блоков дескрипторов,

причем кодирование классифицированных выровненных прочтений в виде множества блоков дескрипторов включает в себя выбор дескрипторов в соответствии с классами выровненных прочтений,

структурирования блоков дескрипторов с помощью информации заголовка с созданием тем самым последовательных блоков доступа.

2. Способ кодирования по п. 1, дополнительно включающий в себя:

дальнейшее классифицирование прочтений, которые не удовлетворяют заданным правилам сопоставления, в класс некартированных прочтений,

построение набора референсных последовательностей с использованием некоторых некартированных прочтений,

выравнивание класса некартированных прочтений на набор построенных референсных последовательностей,

кодирование классифицированных выровненных прочтений в виде множества блоков дескрипторов,

кодирование набора построенных референсных последовательностей,

структурирование блоков дескрипторов и кодированных референсных последовательностей с помощью информации заголовка с созданием тем самым последовательных блоков доступа.

3. Способ по п. 2, в котором классифицирование включает в себя идентифицирование геномных прочтений без всяких несовпадений с референсной последовательностью в

качестве первого «класса P», когда в картированном прочтении нет несовпадений с референсной последовательностью, использованной для картирования.

4. Способ по п. 3, в котором классифицирование дополнительно включает в себя идентифицирование геномных прочтений в качестве второго «класса N», когда несовпадения обнаруживают только в позициях, где секвенатор оказался не в состоянии определить никакого «основания», и количество несовпадений в каждом прочтении не превышает данного порога.

5. Способ по п. 4, в котором классифицирование дополнительно включает в себя идентифицирование геномных прочтений в качестве третьего «класса M», когда несовпадения обнаруживают в позициях, где секвенатор оказался не в состоянии определить никакого «основания» (называются несовпадениями «n-типа») и/или он определил другое «основание» по сравнению с референсной последовательностью (называются несовпадениями «s-типа»), и количество несовпадений не превышает данных порогов для количества несовпадений «n-типа», «s-типа» и порога, полученного из данной функции (f(n,s)), вычисленной на количестве несовпадений «n-типа» и «s-типа».

6. Способ по п. 5, в котором классифицирование дополнительно включает в себя идентифицирование геномных прочтений в качестве четвертого «класса I», когда они могут иметь несовпадения того же типа, что и несовпадения «класса M», и, дополнительно, по меньшей мере одно несовпадение следующего типа: «инсерция» («i-типа»), «делеция» («d-типа»), мягкие отсечения («c-типа»), и при этом количество несовпадений для каждого типа не превышает соответствующего данного порога и порога, обеспечиваемого данной функцией (w(n,s,i,d,c)).

7. Способ по п. 6, в котором классифицирование дополнительно включает в себя идентифицирование геномных прочтений в качестве пятого «класса U», который содержит все прочтения, не отнесенные ни к одному из классов P, N, M, I.

8. Способ кодирования по п. 7, в котором прочтения геномной последовательности, подлежащие кодированию, являются парными.

9. Способ кодирования по п. 8, в котором классифицирование дополнительно включает в себя идентифицирование геномных последовательностей в качестве шестого «класса HM», как содержащего все пары прочтений, где одно прочтение принадлежит классу P, N, M или I, а другое прочтение принадлежит «классу U».

10. Способ кодирования по п. 9, дополнительно включающий в себя следующие этапы:
Определение того, классифицированы два парных прочтения в один и тот же класс (каждый из: P, N, M, I, U) с последующим назначением пары в этот же идентифицированный класс.
Определение того, относятся ли два парных прочтения к разным классам, и если ни одно из них не принадлежит «классу U», то назначение пары прочтений классу с самым высоким приоритетом, определяемым в соответствии со следующим приоритетом:
P < N < M < I
где «класс P» обладает самым низким приоритетом, а «класс I» — самым высоким приоритетом;
определение того, классифицировано ли только одно из парных прочтений как принадлежащее «классу U», и классифицирование пары прочтений как принадлежащей последовательностям «класса HM».

11. Способ по п. 11, в котором каждый класс прочтений N, M, I дополнительно разделяют на два или более подклассов (296, 297, 298) в соответствии с вектором порогов (292, 293, 294), определенным, соответственно, для каждого класса N, M и I посредством количества несовпадений (292) «n-типа», функции f(n,s) (293) и функции w(n,s,i,d,c) (294).

12. Способ кодирования по п. 11, дополнительно включающий в себя следующие этапы:
определение того, классифицированы ли два парных прочтения в один и тот же подкласс с назначением в таком случае пары в этот же подкласс,
определение того, классифицированы ли два парных прочтения в подклассы разных классов с назначением в таком случае пары в подкласс, принадлежащий классу более высокого приоритета в соответствии со следующим выражением:
N < M < I
где N имеет самый низкий приоритет, а I имеет самый высокий приоритет;
определение того, классифицированы ли два парных прочтения в один и тот же класс, причем этим классом является N, или M, или I, но в разные подклассы, с назначением в

таком случае пары в подкласс с самым высоким приоритетом в соответствии о следующими выражениями:

$N_1 < N_2 < \ldots < N_k$

$M_1 < M_2 < \ldots < M_j$

$I_1 < I_2 < \ldots < I_h$

где самый высокий индекс имеет наивысший приоритет.

13. Способ по п. 12, в котором информацию о позиции картирования каждого прочтения кодируют посредством блока дескрипторов «pos».

14. Способ по п. 13, в котором информацию о цепочечности (например, из какой цепочки ДНК секвенировали прочтение) каждого прочтения кодируют посредством блока дескрипторов «rcomp».

15. Способ по п. 14, в котором информацию о парноконцевых прочтениях кодируют посредством блока дескрипторов «pair».

16. Способ по п. 15, в котором В соответствии с еще одним аспектом дополнительную информацию о выравнивании, такую как, картировано ли прочтение в правильную пару, не прошло ли оно проверки качества платформы/поставщика, является ли оно дубликатом ПЦР или оптическим дубликатом, является ли это вспомогательным выравниванием, кодируют посредством блока дескрипторов «flags».

17. Способ по п. 16, в котором информацию о неизвестных основаниях кодируют посредством блока дескрипторов «nmis».

18. Способ по п. 17, в котором информацию о позиции замен кодируют посредством блока дескрипторов «snpp».

19. Способ по п. 18, в котором информацию о типе замен кодируют посредством блока дескрипторов «snpt».

20. Способ по п. 19, в котором информацию о позиции несовпадений типа замен, инсерций или делеций кодируют посредством блока дескрипторов «indp».

21. Способ по п. 20, в котором информацию о типе несовпадений, таких как замены, инсерции или делеции, кодируют посредством блока дескрипторов «indt».

22. Способ по п. 21, в котором информацию об отсеченных основаниях картированного прочтения кодируют посредством блока дескрипторов «indc».

23. Способ по п. 22, в котором информацию о некартированных прочтениях кодируют посредством блока дескрипторов «ureads».

24. Способ по п. 23, в котором информацию о типе референсной последовательности, используемой для кодирования, кодируют посредством блока дескрипторов «rtype».

25. Способ по п. 24, в котором информацию о множественных выравниваниях картированных прочтений кодируют посредством блока дескрипторов «mmap».

26. Способ по 25, в котором информацию о сплайсированных выравниваниях и множественных выравниваниях одного и того же прочтения кодируют посредством блока дескрипторов «msar» и блока дескрипторов «mmap».

27. Способ по п. 26, в котором информацию об оценках выравнивания прочтения кодируют посредством блока дескрипторов «mscore».

28. Способ по п. 27, в котором информацию о группах, которым принадлежат прочтения, кодируют посредством блока дескрипторов «rgroup».

29. Способ по п. 28, в котором блоки доступа класса P строят с помощью блоков дескрипторов типа «pos», «rcomp» и «flags».

30. Способ по п. 29, в котором блоки доступа класса P кодируют информацию о спаривании парноконцевых прочтений с помощью блока дескрипторов «pair».

31. Способ по п. 30, в котором блоки доступа класса N строят с помощью тех же самых блоков дескрипторов, что и для блоков доступа класса P, плюс блок дескриптора «nmis» для информации о позиции неизвестных оснований.

32. Способ по п. 30, в котором блоки доступа класса M строят с помощью тех же самых блоков дескрипторов, что для блоков доступа класса P, плюс блоки дескрипторов «snpp» и «snpt» для информации о позиции и типе замен.

33. Способ по п. 30, в котором блоки доступа класса I строят с помощью тех же самых блоков дескрипторов, что для блоков доступа класса P, плюс блоки дескрипторов «indp», «indt» и «indc» для информации о позиции и типе замен, инсерций, делеций и отсеченных оснований.

34. Способ по п. 33, в котором блоки доступа класса HM строят с помощью тех же самых блоков дескрипторов, что и блоки доступа класса I для картированных прочтений, и с помощью блоков дескриптора «ureads» для некартированных прочтений.

35. Способ по п. 33, в котором информацию о множественных выравниваниях передают с помощью блоков дескрипторов «mmap» и «msar».

36. Способ по п. 35, в котором информацию о сплайсированных выравниваниях передают с помощью расширенной строки cigar, содержащей:

- символ = для указания совпадающих оснований,
- символ + для указания инсерций,
- символ - для указания делеций,
- символ / для указания сплайса на прямой цепочке,
- символ % для указания сплайа на обратной цепочке,
- символ * для указания ненаправленного сплайса,
- текстовый символ из кодов IUPAC для ДНК с целью указания замены,
- символ (n) для указания n мягко отсеченных оснований, где n является целым числом,
- символ [n] для указания n жестко отсеченных оснований, где n является целым числом.

37. Способ по п. 36, в котором блоки дескрипторов содержат «таблицу главного индекса», содержащую по одному разделу для каждого класса и подкласса выровненных прочтений, причем раздел содержит позиции картирования на одну или более референсных последовательностей первого прочтения каждого блока доступа каждого класса или подкласса данных; при этом таблицу главного индекса и данные блока доступа кодируют вместе.

38. Способ по п. 37, в котором блоки дескрипторов также содержат информацию о типе использованного референса (ранее существующий или построенный) и сегментам прочтения, которые не картированы на референсную последовательность.

39. Способ по п. 38, в котором референсные последовательности сначала преобразуют в другие референсные последовательности путем применения замен, инсерций, делеций и отсечений, а затем кодируют классифицированные выровненные прочтения в виде множества блоков дескрипторов, относящихся к преобразованным референсным последовательностям.

40. Способ по п. 39, в котором к референсным последовательностям, используемым для всех классов данных, применяют одно и то же преобразование.

41. Способ по п. 40, в котором к референсным последовательностям, используемым для каждого класса данных, применяют разные преобразования.

42. Способ по п. 41, в котором преобразования референсных последовательностей кодируют в виде блоков дескрипторов и структурируют с помощью информации заголовка с созданием тем самым последовательных блоков доступа.

43. Способ по п. 42, в котором кодирование классифицированных выровненных прочтений и связанных преобразований референсных последовательностей в виде множества блоков дескрипторов включает в себя этап связывания конкретной модели источника и конкретного энтропийного кодера с каждом блоком дескрипторов.

44. Способ по п. 43, в котором энтропийный кодер представляет собой контекстно-адаптивный арифметический кодер, кодер переменной длины или кодер Голомба.

45. Способ декодирования кодированных геномных данных, включающий в себя следующие этапы:

синтаксический анализ блоков доступа, содержащих кодированные геномные данные, для выделения множества блоков дескрипторов путем использования информации заголовка,

декодирование множества блоков дескрипторов для выделения прочтений в соответствии с заданными правилами сопоставления, определяющими их классификацию относительно одного или более референсных прочтений.

46. Способ декодирования по п. 45, дополнительно включающий в себя декодирование таблицы главного индекса, содержащей по одному разделу для каждого класса прочтений и связанные соответствующие позиции картирования.

47. Способ декодирования по п. 46, дополнительно включающий в себя декодирование информации, относящейся к типу используемого референса — ранее существующий, преобразованный или построенный.

48. Способ декодирования по п. 47, дополнительно включающий в себя декодирование информации, относящейся к одному или более преобразованиям, которые должны быть применены к ранее существующим референсным последовательностям.

49. Способ декодирования по п. 48, в котором блоки дескрипторов энтропийно декодируют.

50. Способ декодирования по п. 49, в котором:
прочтения класса P получают декодированием блоков дескрипторов типа: «pos», «rcomp», «flags» и «rlen»,
прочтения класса N получают декодированием блоков дескрипторов типа: «pos», «rcomp», «flags», «rlen» и «nmis»,
прочтения класса M получают декодированием блоков дескрипторов типа: «pos», «rcomp», «flags», «rlen», «snpp» и «snpt»,

прочтения класса I получают декодированием блоков дескрипторов типа: «pos», «rcomp», «flags», «rlen», «indp», «indt» и «indc»,
прочтения класса U получают декодированием блоков дескрипторов типа: «pos», «rcomp», «flags», «rlen», «snpp», «snpt», «indc», «ureads» и «rtype».

51. Способ декодирования по п. 50, в котором парные прочтения:
класса P, N, M и I получают также декодированием блоков дескрипторов типа «pair»;
класса HM получают декодированием блоков дескрипторов типа «pos», «rcomp», «flags», «rlen», «indp», «indt», «indc» и «ureads».

52. Геномный кодер (2010) для сжатия данных 209 геномной последовательности, причем данные 209 геномной последовательности содержат прочтения последовательностей нуклеотидов,
при этом геномный кодер (2010) содержит:
узел (201) выравнивателя, выполненный с возможностью выравнивания прочтений на одну или более референсных последовательностей с созданием тем самым выровненных прочтений;
узел (202) генератора конструированной последовательности, выполненный с возможностью создания конструированных референсных последовательностей;
узел (204) классифицирования данных, выполненный с возможностью классифицирования выровненных прочтений в соответствии с заданными правилами сопоставления с одной или более ранее существующими референсными последовательностями или построенными референсными последовательностями с созданием тем самым классов выровненных прочтений (208);

один или более узлов (205–207) кодирования блоков, выполненных с возможностью кодирования классифицированных выровненных прочтений в виде блоков дескрипторов путем выбора дескрипторов в соответствии с классами выровненных прочтений;

мультиплексор (2016) для мультиплексирования сжатых геномных данных и метаданных.

53. Геномный кодер по п. 52, дополнительно содержащий

узел (2019) преобразования референсной последовательности, выполненный с возможностью преобразования ранее существующих референсов и классов (208) данных в преобразованные классы (2018) данных.

54. Геномный кодер по п. 53, в котором узел (204) классифицирования данных содержит конфигурируемые с помощью векторов порогов кодеры классов данных N, M и I, формирующие подклассы данных классов N, M и I.

55. Геномный кодер по п. 54, в котором узел (2019) преобразования референса применяет одно и то же преобразование (300) референса для всех классов и подклассов данных.

56. Геномный кодер по п. 54, в котором узел (2019) преобразования референса применяет разные преобразования (301, 302, 303) к разным классам и подклассам данных.

57. Геномный кодер по п. 54, дополнительно содержащий средство кодирования, пригодное для осуществления способа кодирования по п. 12.

58. Геномный декодер (218) для разуплотнения сжатого геномного потока (211), причем геномный декодер (218) содержит:
демультиплексор (210) для демультиплексирования сжатых геномных данных и метаданных,
средства (212–214) синтаксического анализа, выполненные с возможностью синтаксического анализа сжатого геномного потока с получением геномных блоков дескрипторов (215),
один или более декодеров (216–217) блоков, выполненных с возможностью декодирования геномных блоков дескрипторов в классифицированные прочтения последовательностей нуклеотидов (2111),
декодеры (219) классов геномных данных, выполненные с возможностью выборочного декодирования классифицированных прочтений последовательностей нуклеотидов относительно одной или более референсных последовательностей с получением несжатых прочтений последовательностей нуклеотидов.

59. Геномный декодер по п. 58, дополнительно содержащий декодер (2113) преобразования референса, выполненный с возможностью декодирования дескрипторов (2112) преобразования референса и создания преобразованного референса (2114) для использования декодерами (219) классов геномных данных.

60. Геномный декодер по п. 59, в котором одну или более референсных последовательностей хранят в сжатом потоке (211) геномных данных.

61. Геномный декодер по п. 59, в котором одну или более референсных последовательностей подают в декодер посредством внеполосного механизма.

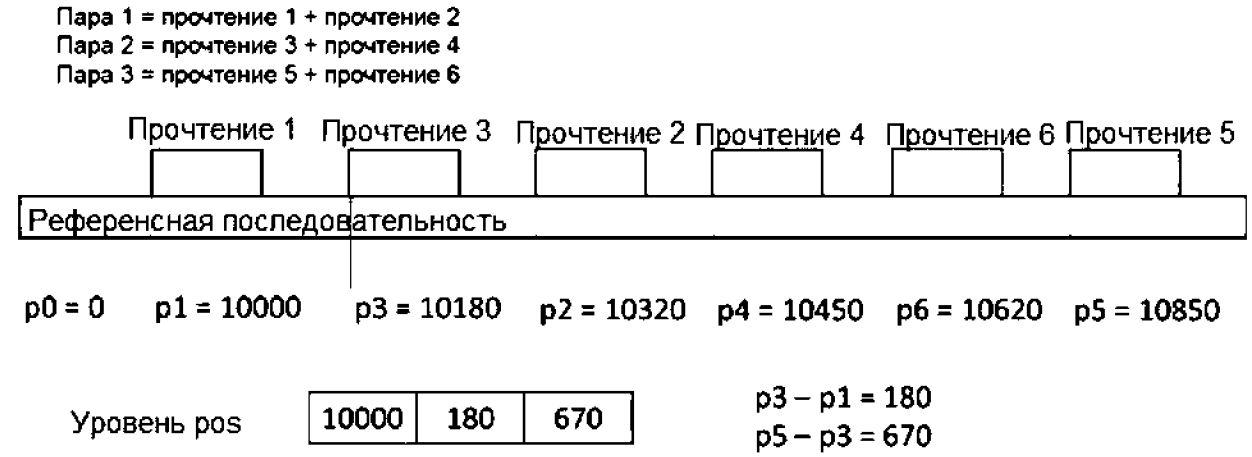
62. Геномный декодер по п. 59, в котором одну или более референсных последовательностей строят в декодере.

63. Геномный декодер по п. 59, в котором одну или более референсных последовательностей преобразуют в декодере с помощью декодера (2113) преобразования референса.

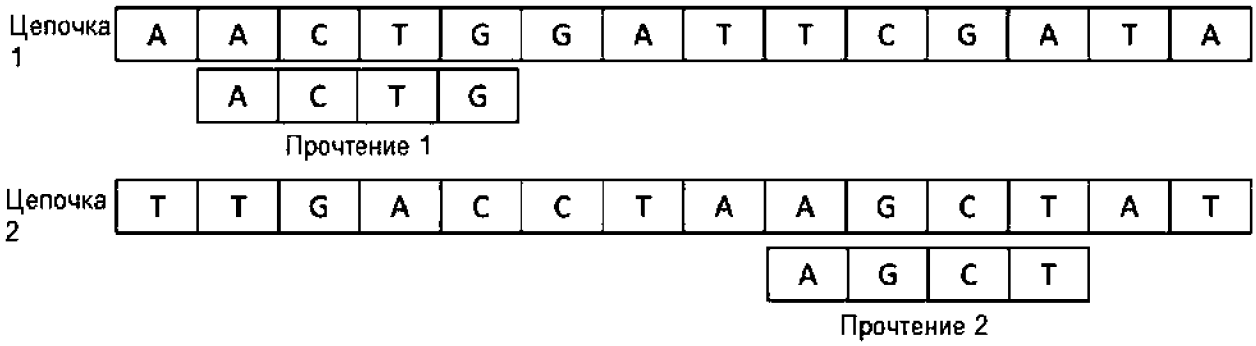
64. Машиночитаемый носитель информации, содержащий инструкции, исполнение которых вызывает осуществление по меньшей мере одним процессором способа кодирования по п. 12.

65. Машиночитаемый носитель информации, содержащий инструкции, исполнение которых вызывает осуществление по меньшей мере одним процессором способа декодирования по п. 59.

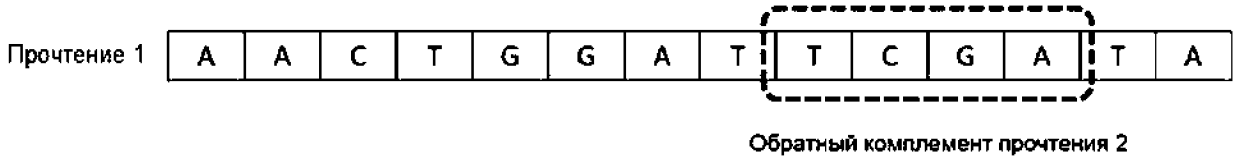
66. Поддержка хранения геномных данных в соответствии со способом по пр. 12.


Пара 1 = прочтение 1 + прочтение 2
Пара 2 = прочтение 3 + прочтение 4
Пара 3 = прочтение 5 + прочтение 6
Прочтение 1
Прочтение 3
Прочтение 2
Прочтение 4
Прочтение 6
Прочтение 5
Референсная последовательность
p0 = 0
p1 = 10000
p3 = 10180
p2 = 10320
p4 = 10450
p6 = 10620
p5 = 10850
Уровень pos
10000
180
670
p3 – p1 = 180
p5 – p3 = 670

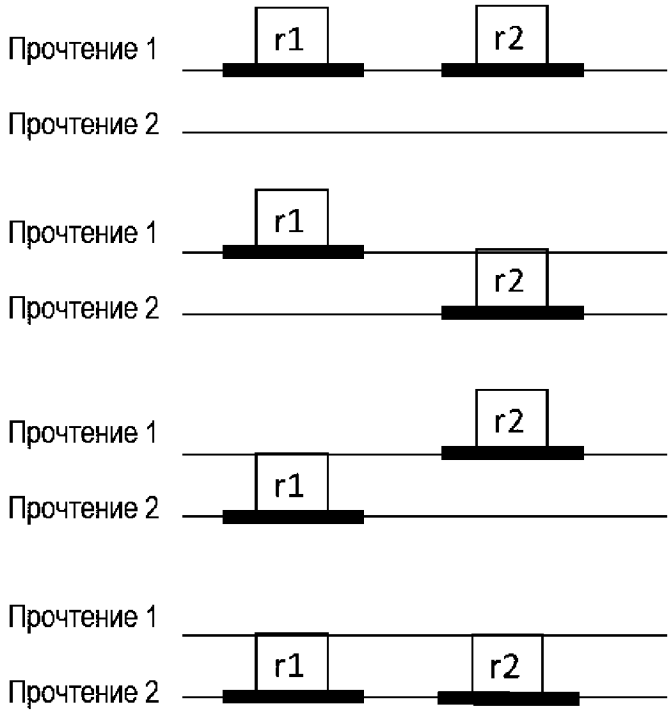
Фиг. 1


Цепочка 1
A A C T G G A T T C G A T A
A C T G
Прочтение 1
Цепочка 2
T T G A C C T A A G C T A T
A G C T
Прочтение 2

Фиг. 2

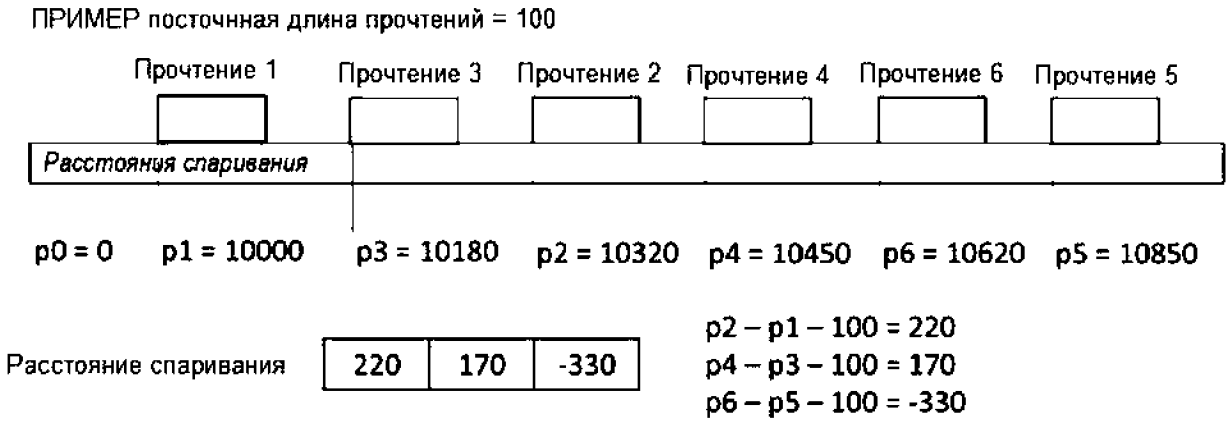


Фиг. 3

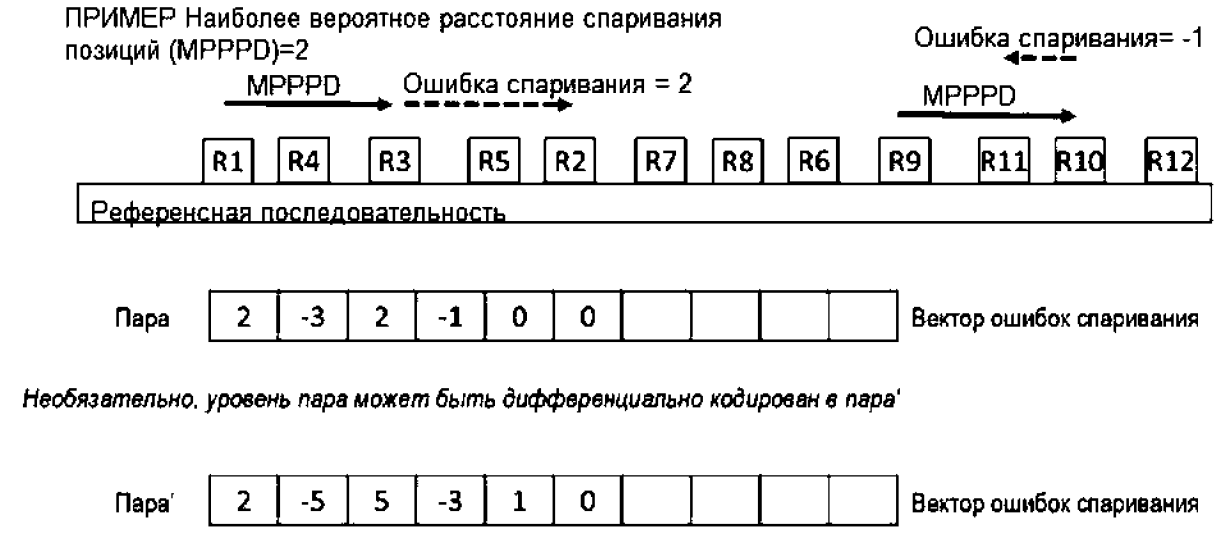


Элемент уровня rcomp
rc1
rc2
rc3
rc4

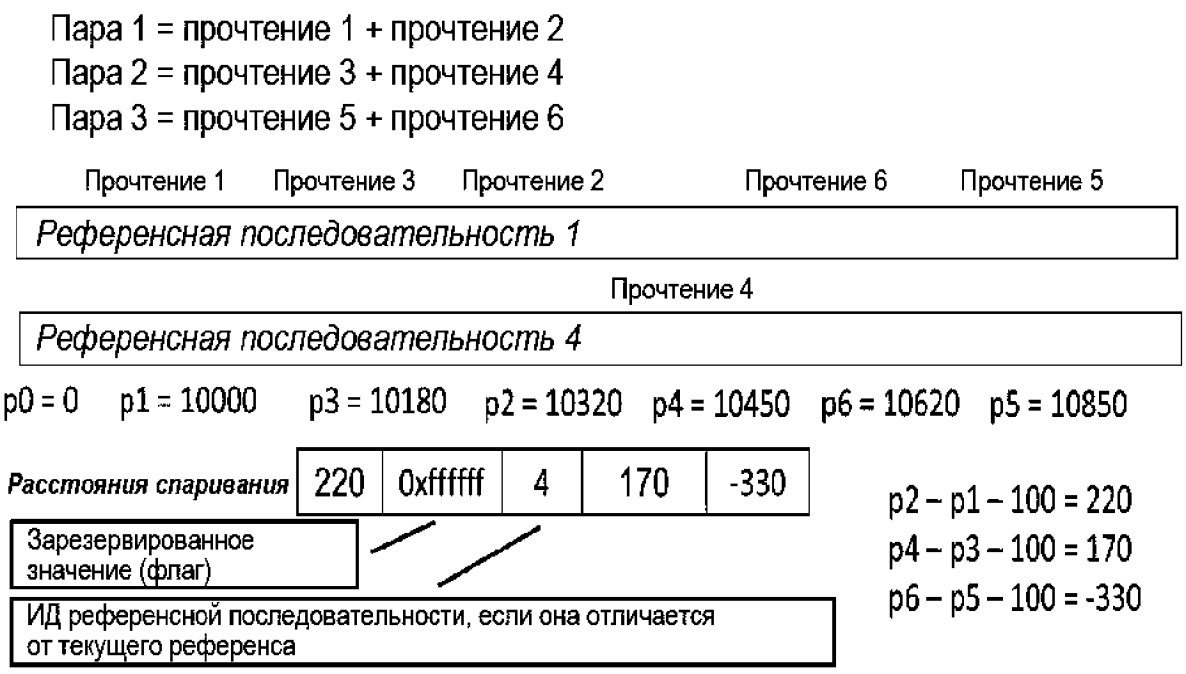
Фиг. 4


ПРИМЕР постоянная длина прочтений = 100
Прочтение 1
Прочтение 3
Прочтение 2
Прочтение 4
Прочтение 6
Прочтение 5
Расстояния спаривания
p0 = 0
p1 = 10000
p3 = 10180
p2 = 10320
p4 = 10450
p6 = 10620
p5 = 10850
Расстояние спаривания
220
170
-330
p2 – p1 – 100 = 220
p4 – p3 – 100 = 170
p6 – p5 – 100 = -330

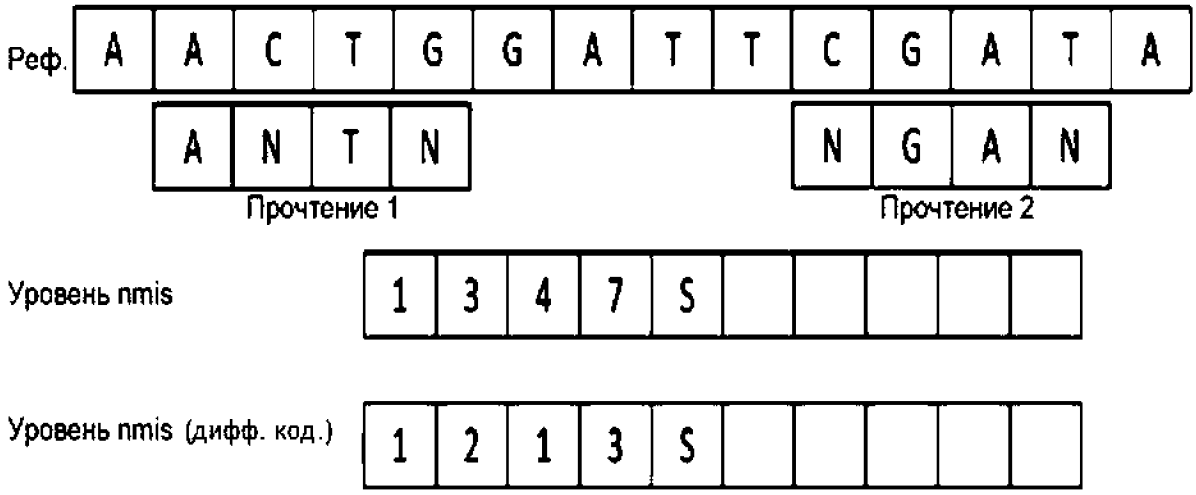
Фиг. 5


ПРИМЕР Наиболее вероятное расстояние спаривания позиций (MPPPD)=2
MPPPD
Ошибка спаривания = 2
Ошибка спаривания= -1
MPPPD
R1
R4
R3
R5
R2
R7
R8
R6
R9
R11
R10
R12
Референсная последовательность
Пара
2
-3
2
-1
0
0
Вектор ошибок спаривания
Необязательно, уровень пара может быть дифференциально кодирован в пара'
Пара'
2
-5
5
-3
1
0
Вектор ошибок спаривания

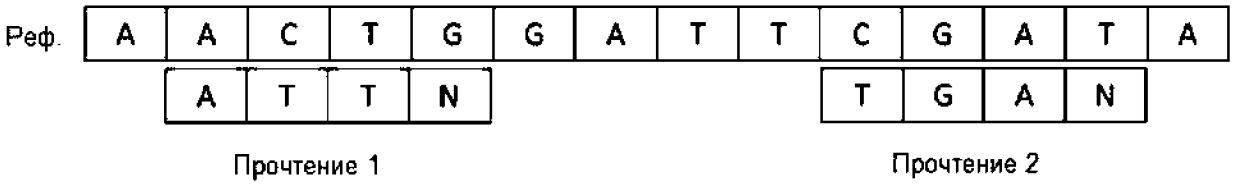
Фиг. 6


Пара 1 = прочтение 1 + прочтение 2
Пара 2 = прочтение 3 + прочтение 4
Пара 3 = прочтение 5 + прочтение 6
Прочтение 1
Прочтение 3
Прочтение 2
Прочтение 6
Прочтение 5
Референсная последовательность 1
Прочтение 4
Референсная последовательность 4
p0 = 0
p1 = 10000
p3 = 10180
p2 = 10320
p4 = 10450
p6 = 10620
p5 = 10850
Расстояния спаривания
220
0xffffff
4
170
-330
Зарезервированное значение (флаг)
ИД референсной последовательности, если она отличается от текущего референса
p2 – p1 – 100 = 220
p4 – p3 – 100 = 170
p6 – p5 – 100 = -330

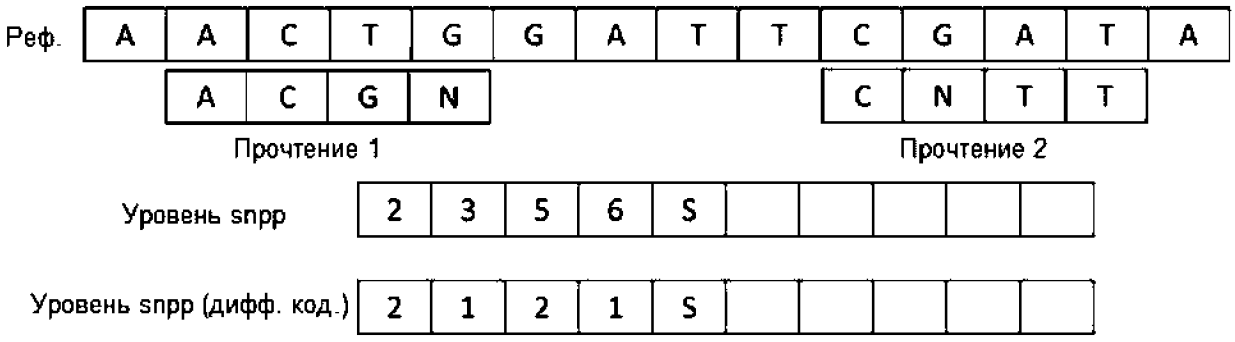
Фиг. 7


Реф.
A A C T G G A T T C G A T A
A N T N
N G A N
Прочтение 1
Прочтение 2
Уровень nmis
1 3 4 7 S
Уровень nmis (дифф. код.)
1 2 1 3 S

Фиг. 8


Реф.
A A C T G G A T T C G A T A
A T T N
T G A N
Прочтение 1
Прочтение 2

Фиг. 9


Реф.
A A C T G G A T T C G A T A
A C G N
C N T T
Прочтение 1
Прочтение 2
Уровень snpp
2 3 5 6 S
Уровень snpp (дифф. код.)
2 1 2 1 S

Фиг. 10

Уровень snpt (без кодов IUPAC)

Тип замены вычисляют как индекс вектора замен, составленного из всех возможных символов.

Например:

S = [A, C, G, T, N, Z], где **Z = направление**

Индекс делеции

• КОДИРОВАНИЕ справа налево

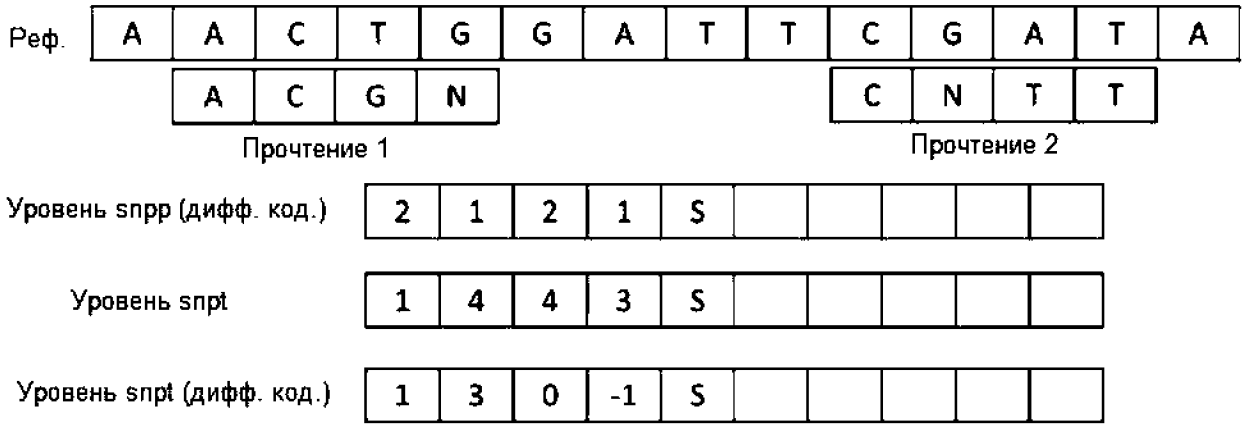
• ДЕКОДИРОВАНИЕ слева направо

Реф.	Прочт.	Кодированный символ
A		idx(A,Z) = 5
C		idx(C,Z) = 4
G		idx(G,Z) = 3
T		idx(T,Z) = 2

Реф.	Прочт.	Кодированный 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

Реф.	Прочт.	Кодированный 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

Фиг. 11



Фиг. 12

Уровень snpt (без кодов IUPAC)

Тип замены вычисляют 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

Реф.	Прочт.	Кодированный 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

Реф.	Прочт.	Кодированный 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

Фиг. 13

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

- КОДИРОВАНИЕ справа налево
- ДЕКОДИРОВАНИЕ слева направо

Фиг. 14

Уровень indt (без кодов IUPAC)

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

- КОДИРОВАНИЕ справа налево
- ДЕКОДИРОВАНИЕ слева направо

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

Фиг. 15

Уровень 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	Кодированный символ
A	16
C	17
G	18
T	19
N	20

Фиг. 16

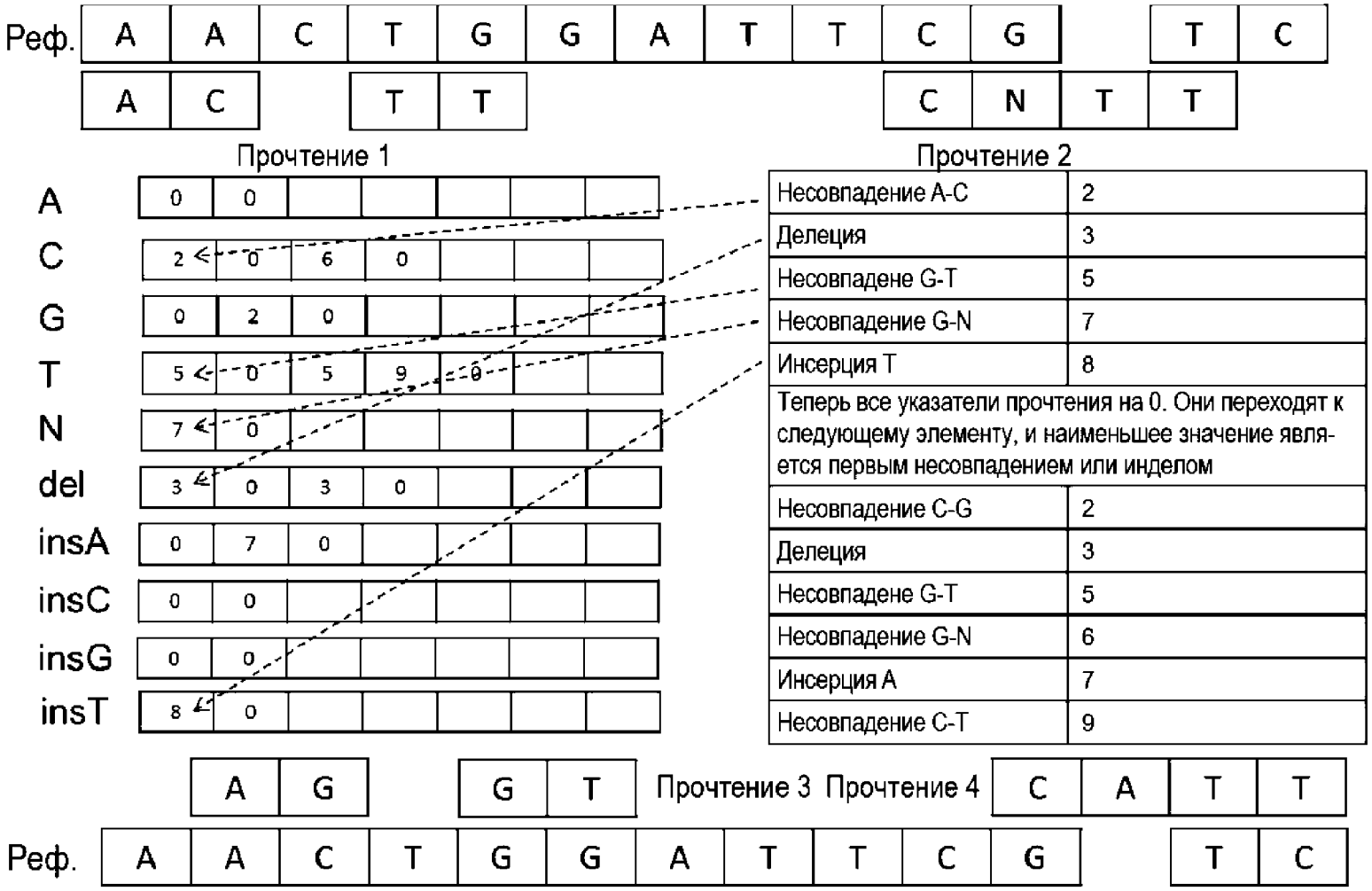
Модель источника 2 (без кодов IUPAC)

Один уровень позиции на тип замены, один на тип делеции и один на тип инсерции

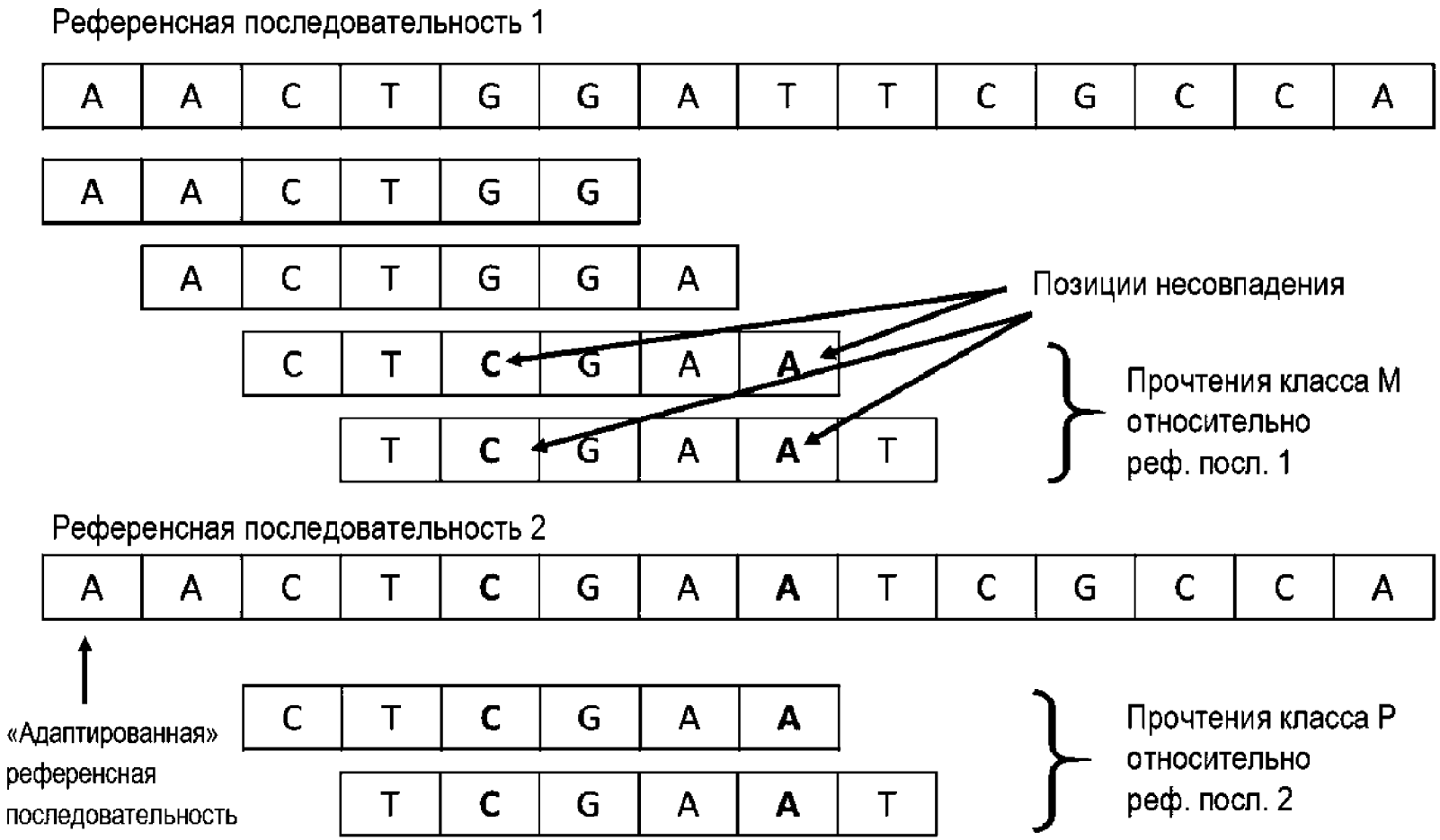
Фиг. 17

Модель источника 2 (без кодов IUPAC)

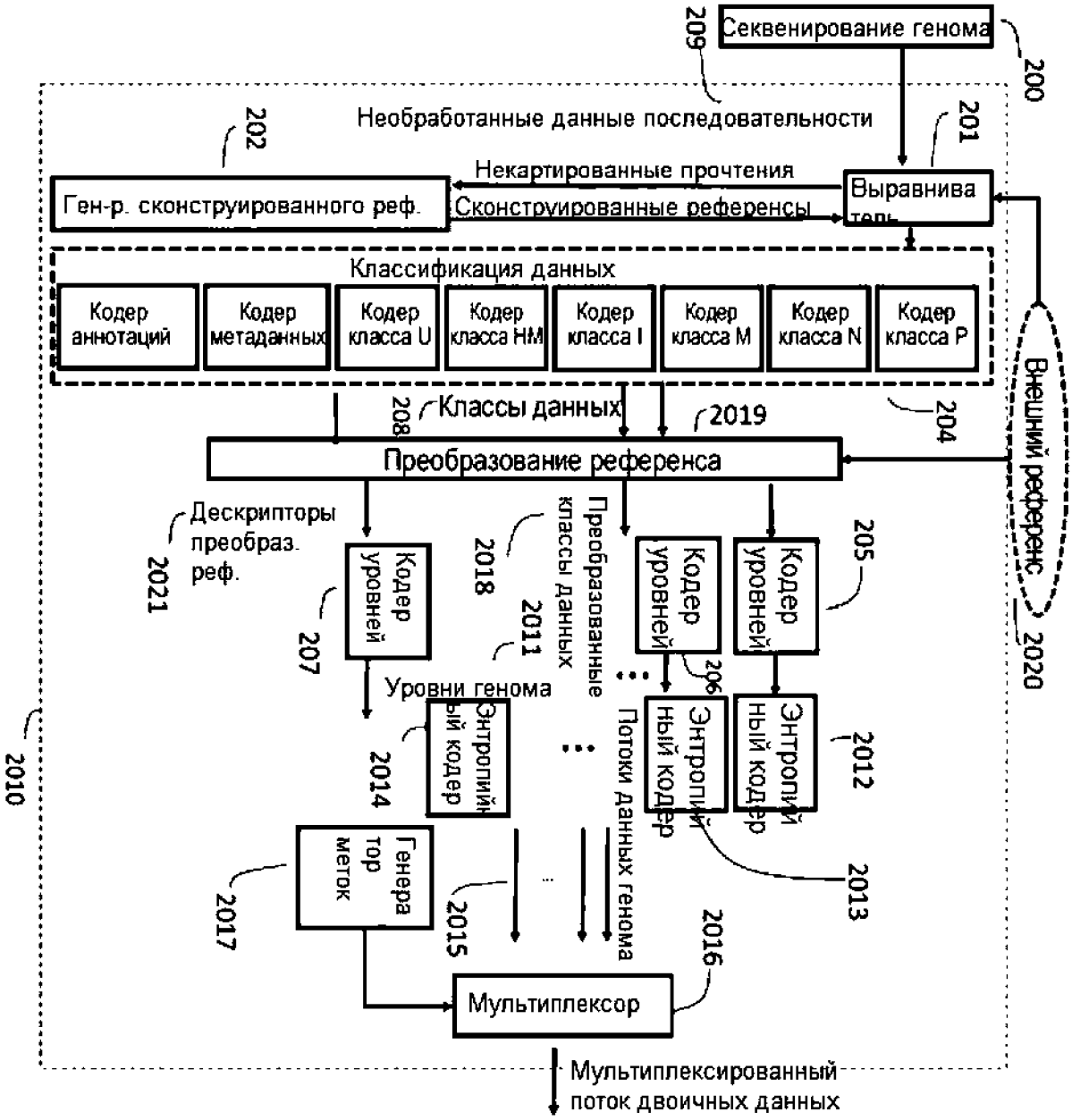
Один уровень позиции на тип замены, один на тип делеции и один на тип инсерции



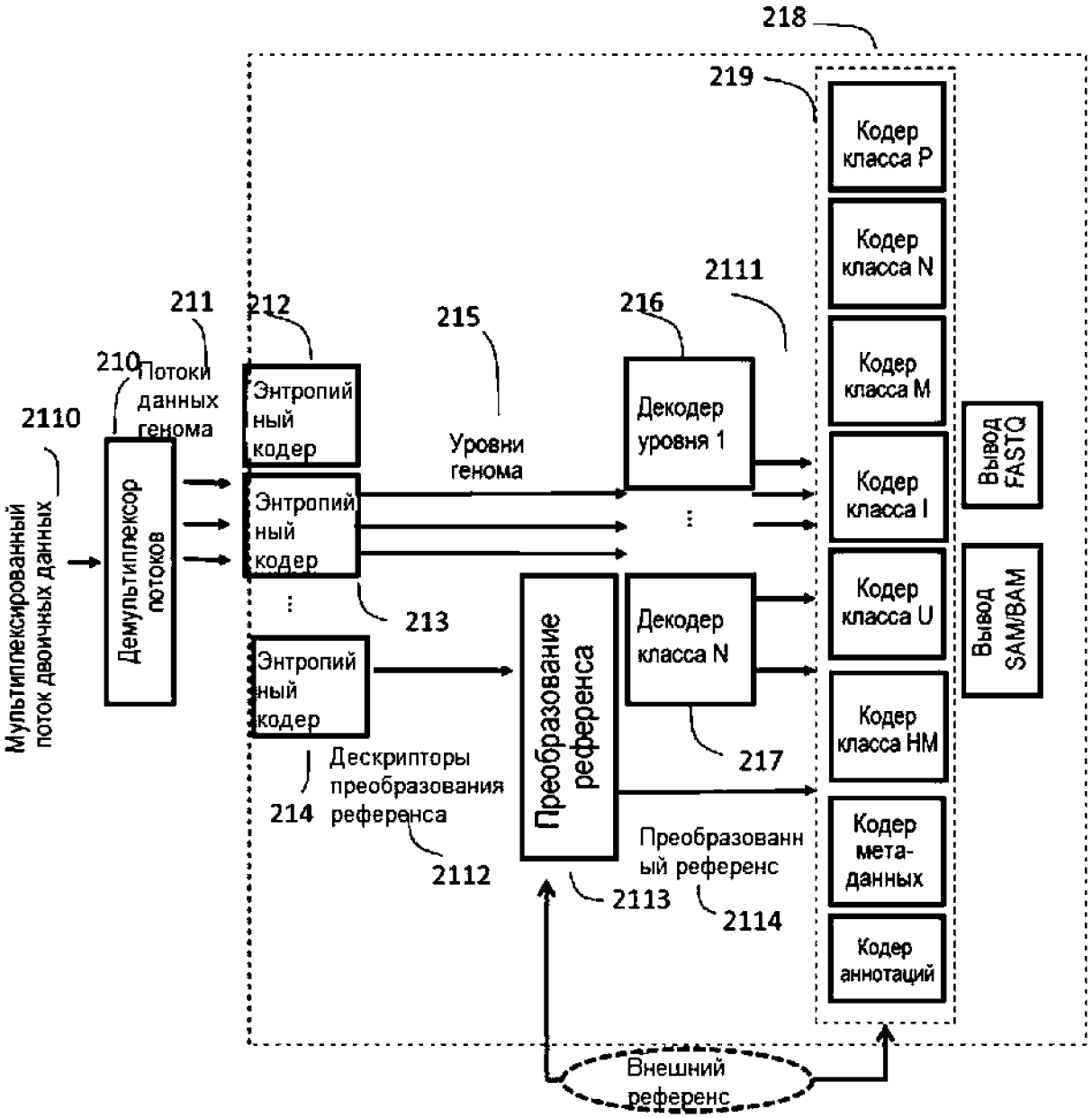
Фиг. 18


Референсная последовательность 1
A A C T G G A T T C G C C A
A A C T G G
A C T G G A
C T C G A A
T C G A A T
Позиции несовпадения
Прочтения класса M относительно реф. посл. 1
Референсная последовательность 2
A A C T C G A A T C G C C A
«Адаптированная» референсная последовательность
C T C G A A
T C G A A T
Прочтения класса P относительно реф. посл. 2

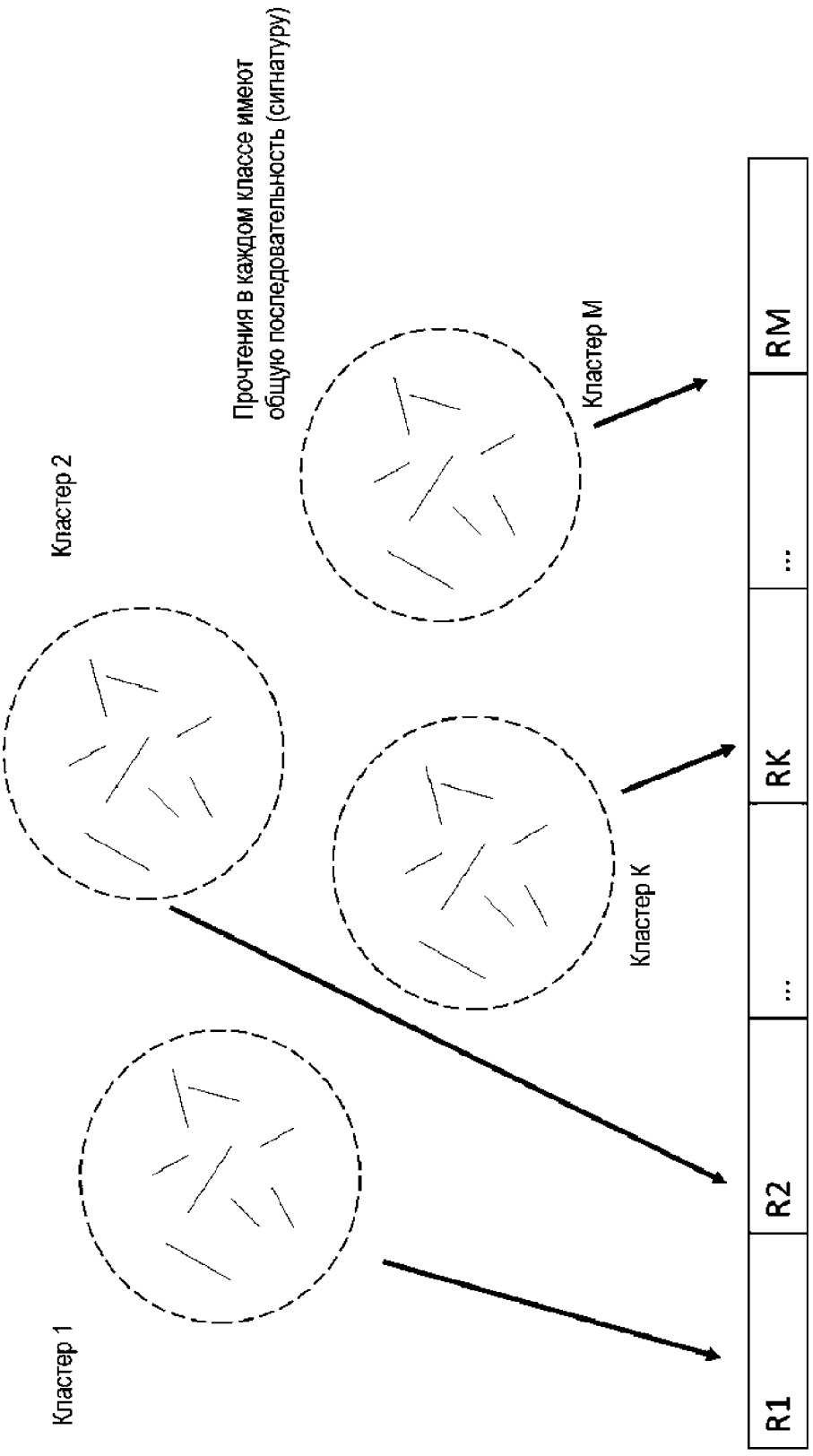
Фиг. 19


Секвенирование генома
200
209
Необработанные данные последовательности
201
202
Некартированные прочтения
Ген-р. сконструированного реф.
Сконструированные референсы
Выравнива
Классификация данных
Кодер аннотаций
Кодер метаданных
Кодер класса U
Кодер класса HM
Кодер класса I
Кодер класса M
Кодер класса N
Кодер класса P
Внешний референс
2020
208
Классы данных
2019
204
Преобразование референса
Дескрипторы преобраз. реф.
2021
2018
Преобразованные классы данных
2011
Кодер уровней
205
207
Уровни генома
206
Энтропийный кодер
2012
2014
Потоки данных генома
2013
Генератор меток
2017
2015
2016
Мультиплексор
2010
Мультиплексированный поток двоичных данных

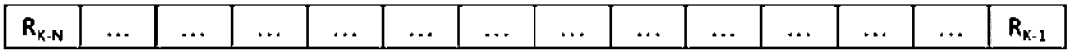
Фиг. 20


218
219
Кодер класса P
Кодер класса N
Кодер класса M
Кодер класса I
Кодер класса U
Кодер класса HM
Кодер мета-данных
Кодер аннотаций
Вывод FASTQ
Вывод SAM/BAM
2111
216
Декодер уровня 1
Декодер класса N
217
211
212
215
Потоки данных генома
Энтропийный кодер
Энтропийный кодер
Энтропийный кодер
213
Уровни генома
210
2110
Мультиплексированный поток двоичных данных
Демультиплексор потоков
Преобразование референса
Дескрипторы преобразования референса
214
2112
2113
Преобразованный референс
2114
Внешний референс

Фиг. 21

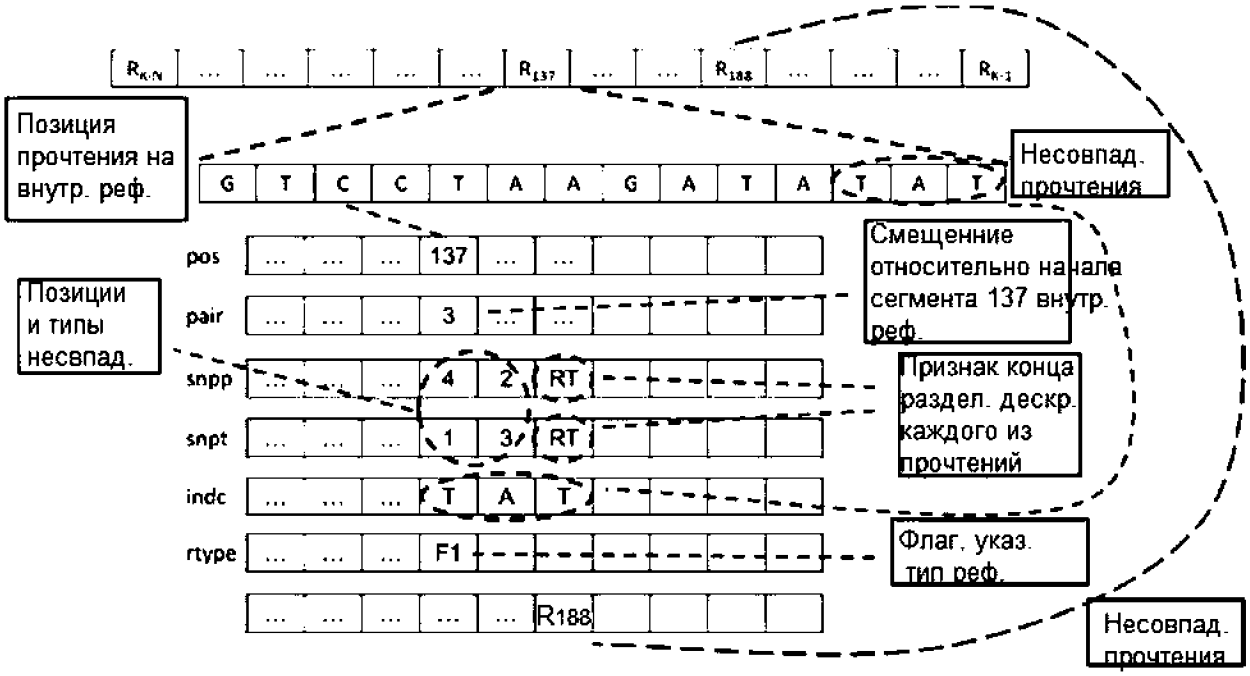

Прочтения в каждом классе имеют общую последовательность (сигнатуру)
Кластер 1
Кластер 2
Кластер K
Кластер M
R1
R2
...
RK
...
RM

Фиг. 22

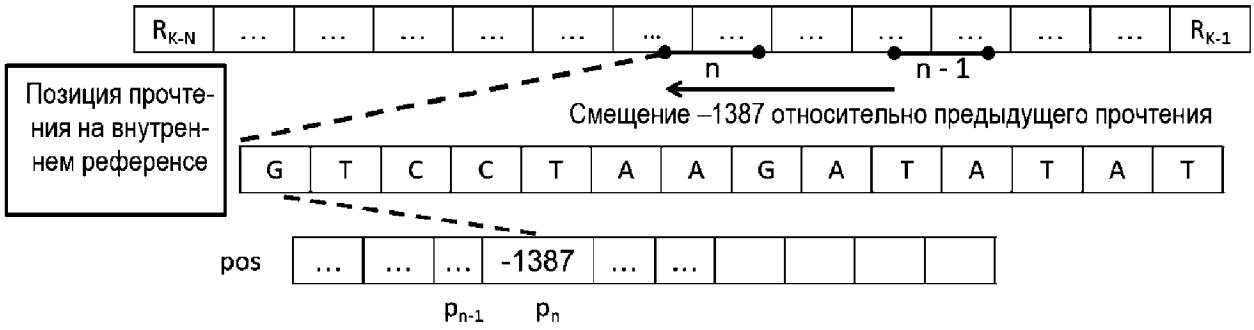


При кодировании прочтения К референс составляют из прочтений с К-1 по К-N. Когда имеющихся прочтений недостаточно, используют заполнение

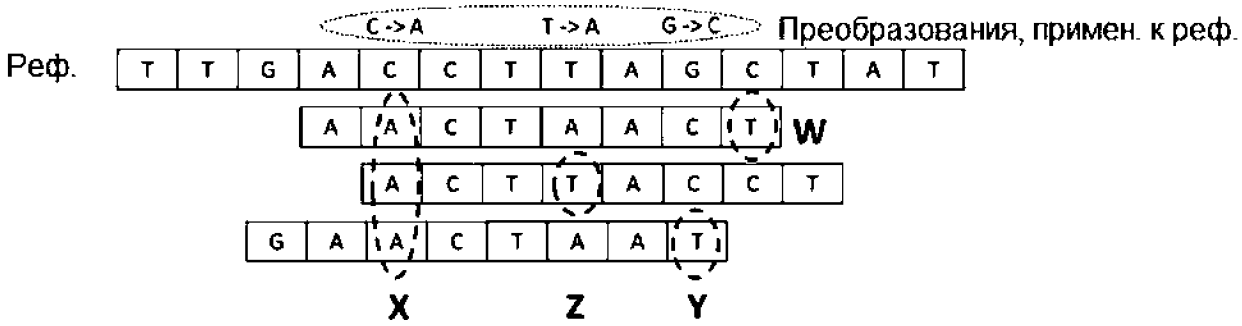
Фиг. 23



Фиг. 24

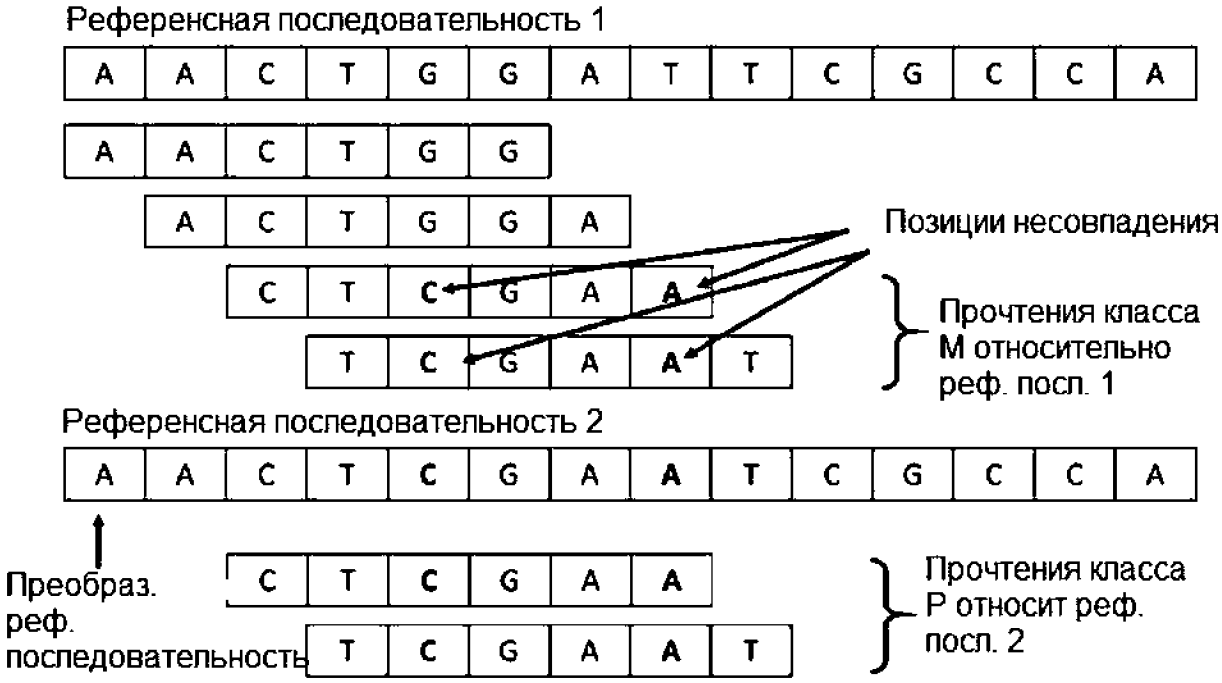


Фиг. 25

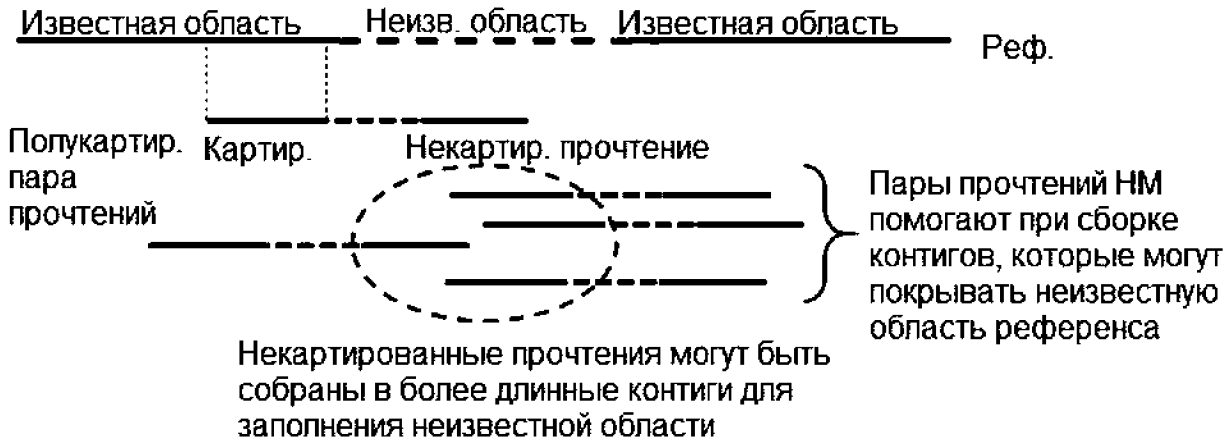


X = несовпадене, которое исчезнет
Y = несовпадение, которое не исчезнет, но код несовпадения измениться
W = несовпадение, которое остается неизменным
Z = заново введенное несовпадение после преобразования

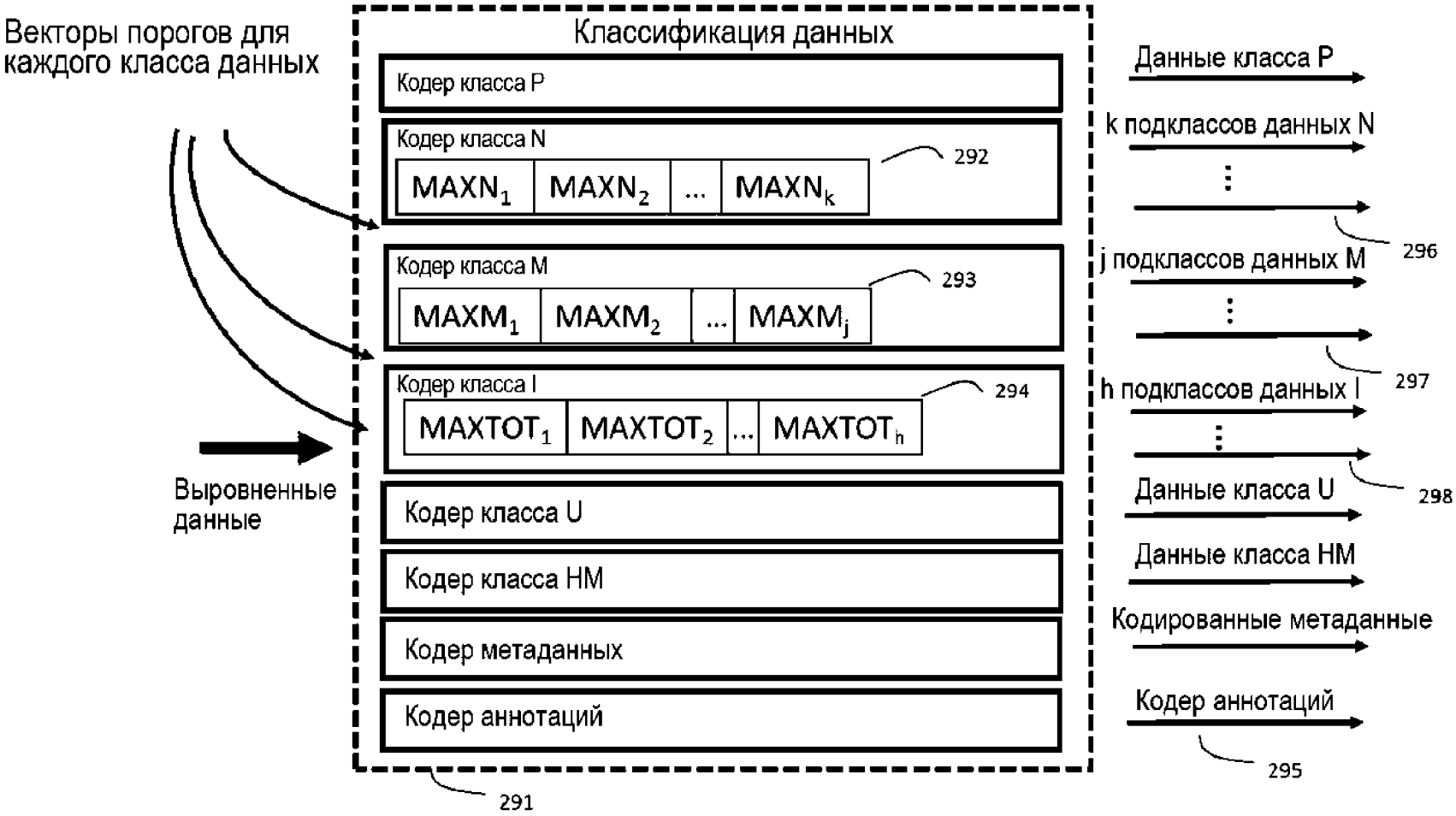
Фиг. 26


Референсная последовательность 1
A A C T G G A T T C G C C A
A A C T G G
A C T G G A
C T C G A A
T C G A A T
Позиции несовпадения
Прочтения класса M относительно реф. посл. 1
Референсная последовательность 2
A A C T C G A A T C G C C A
Преобраз. реф. последовательность
C T C G A A
T C G A A T
Прочтения класса P относит реф. посл. 2

Фиг. 27

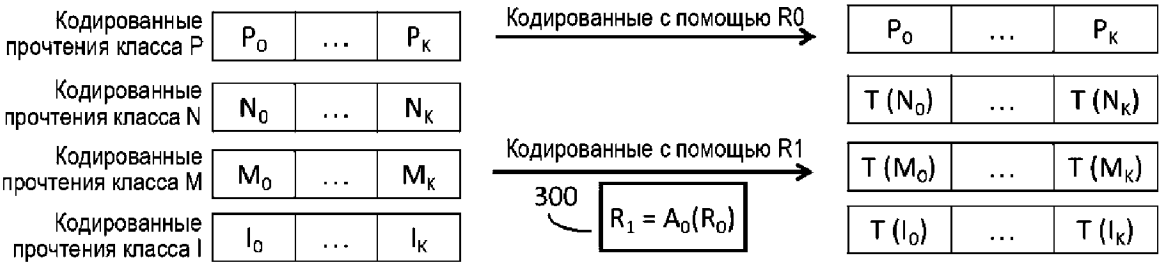

Известная область
Неизв. область
Известная область
Реф.
Полукартир. пара прочтений
Картир.
Некартир. прочтение
Пары прочтений НМ помогают при сборке контигов, которые могут покрывать неизвестную область референса
Некартированные прочтения могут быть собраны в более длинные контиги для заполнения неизвестной области

Фиг. 28


Векторы порогов для каждого класса данных
Классификация данных
Кодер класса P
Кодер класса N
MAXN1
MAXN2
...
MAXNk
292
Кодер класса M
MAXM1
MAXM2
...
MAXMj
293
Кодер класса I
MAXTOT1
MAXTOT2
...
MAXTOTh
294
Кодер класса U
Кодер класса HM
Кодер метаданных
Кодер аннотаций
291
Выровненные данные
Данные класса P
k подклассов данных N
296
j подклассов данных M
297
h подклассов данных I
298
Данные класса U
Данные класса HM
Кодированные метаданные
Кодер аннотаций
295

Фиг. 29

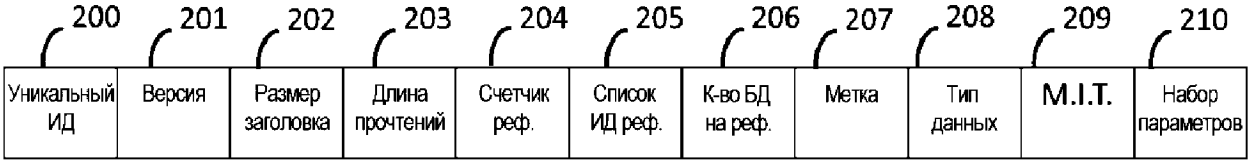
Классы N, M и I могут использовать одно и то же преобразование A0 референса или использовать разные преобразования AN, AM, AI



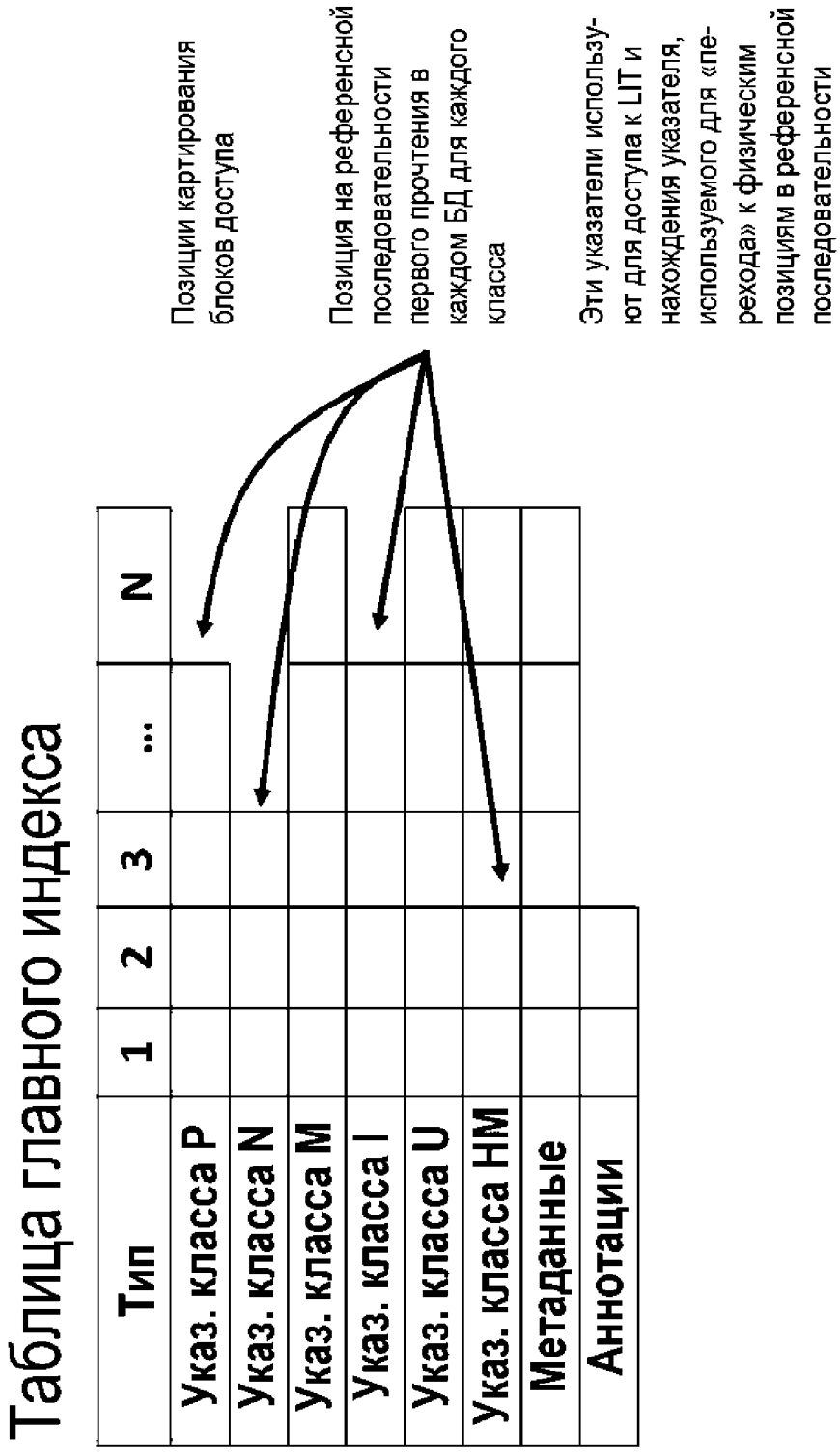
T представляет преобразование каждого кодированного прочтения при кодировании относительно R1 = A0(R0)

T- преобразование каждого кодированного прочтения при кодировании относительно R1=A0(R0)

Фиг. 30


200
201
202
203
204
205
206
207
208
209
210
Уникальный ИД
Версия
Размер заголовка
Длина прочтений
Счетчик реф.
Список ИД реф.
К-во БД на реф.
Метка
Тип данных
M.I.T.
Набор параметров

Фиг. 31


Позиции картирования блоков доступа
Позиция на референсной последовательности первого прочтения в каждом БД для каждого класса
Эти указатели используют для доступа к LIT и нахождения указателя, используемого для «перехода» к физическим позициям в референсной последовательности
Таблица главного индекса
Тип
1
2
3
...
N
Указ. класса P
Указ. класса N
Указ. класса M
Указ. класса I
Указ. класса U
Указ. класса HM
Метаданные
Аннотации

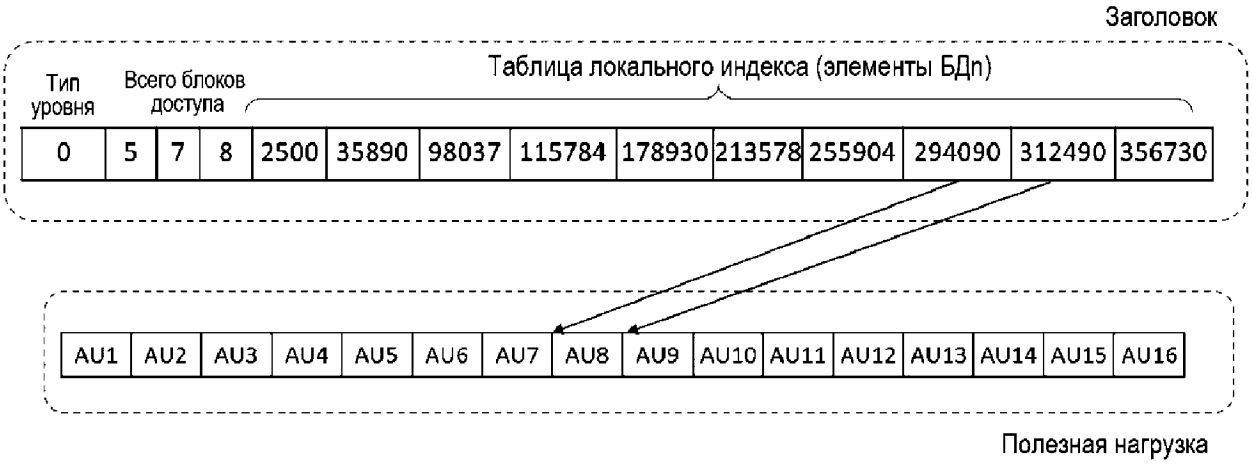
Фиг. 32


Означает, что 3-е прочтение в 3-м БД класса P картируется на референс 2 в позиции 130 298
Класс P
Разделитель
10000
73987
123897
....
S
12078
62656
130298
235019
....
S
....
Означает, что первое прочтение в первом БД класса P картируется на референс 1 в позиции 10 000

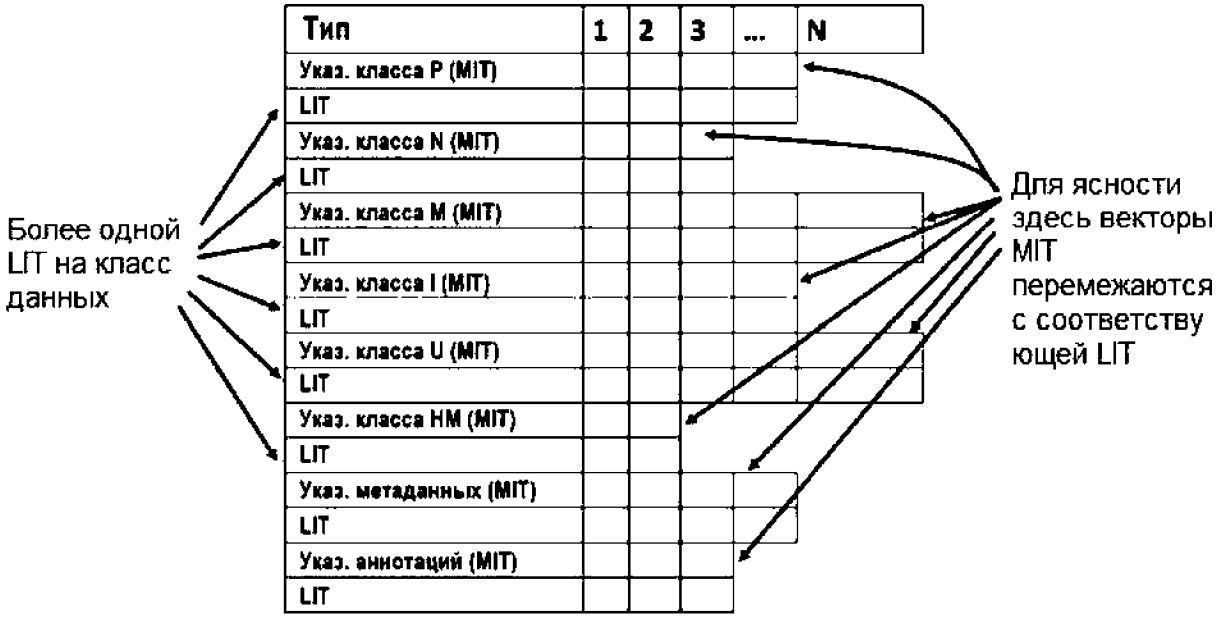
Фиг. 33


Тип уровня
Всего блоков доступа
Таблица локального индекса (элементы БДn)
Бит n
Целое число (БДn)
Указатели на БД в полезной нагрузке. Указатели на БД первого байта каждого БД в полезной нагрузке
Метаданные, специфичные для уровня
Полезная нагрузка
Блок 1
Блок 2
Блок 3
Блок 4
Блок 5
Блок 6
Блок 7
Блок 8
Блок 9
Блок10

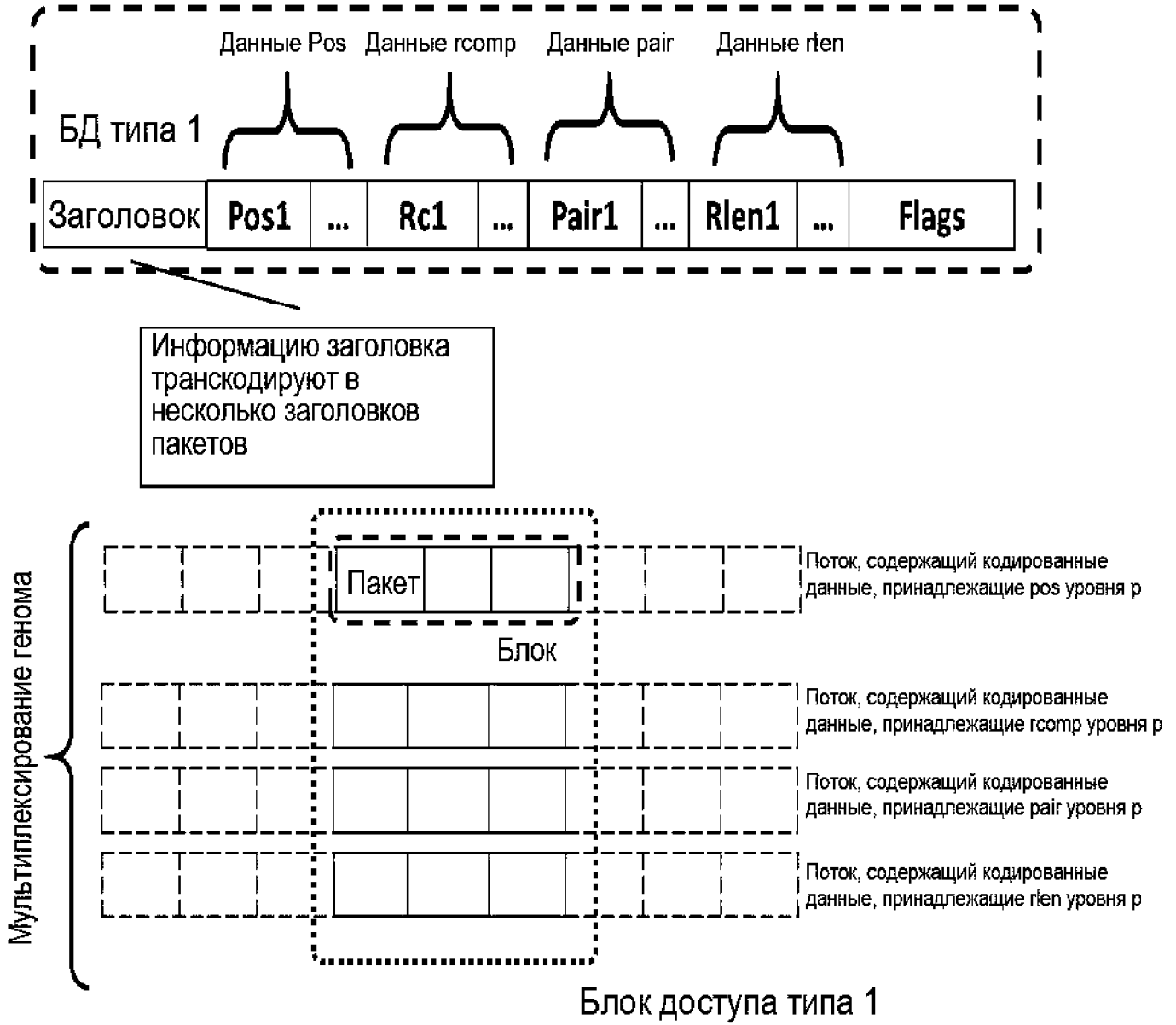
Фиг. 34


Заголовок
Тип уровня
Всего блоков доступа
Таблица локального индекса (элементы БДn)
0
5
7
8
2500
35890
98037
115784
178930
213578
255904
294090
312490
356730
AU1
AU2
AU3
AU4
AU5
AU6
AU7
AU8
AU9
AU10
AU11
AU12
AU13
AU14
AU15
AU16
Полезная нагрузка

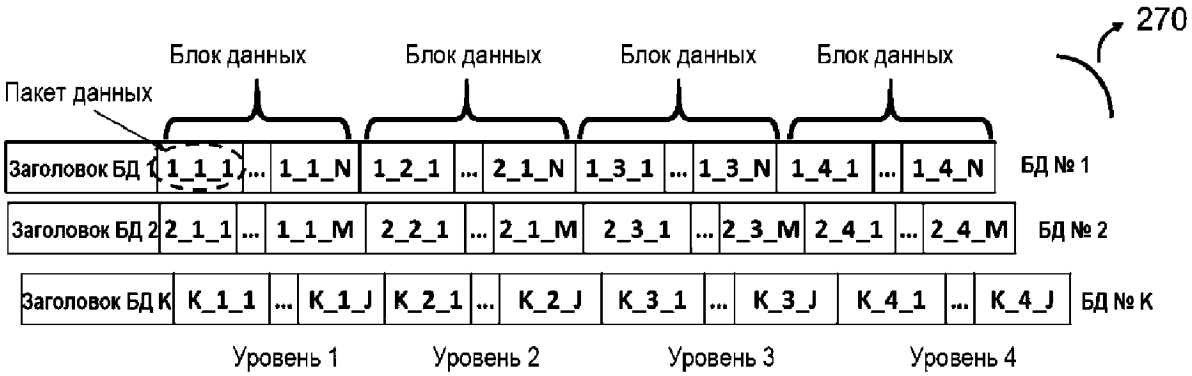
Фиг. 35


Тип
1
2
3
...
N
Указ. класса P (MIT)
LIT
Указ. класса N (MIT)
LIT
Указ. класса M (MIT)
LIT
Указ. класса I (MIT)
LIT
Указ. класса U (MIT)
LIT
Указ. класса HM (MIT)
LIT
Указ. метаданных (MIT)
LIT
Указ. аннотаций (MIT)
LIT
Более одной LIT на класс данных
Для ясности здесь векторы MIT перемежаются с соответству ющей LIT

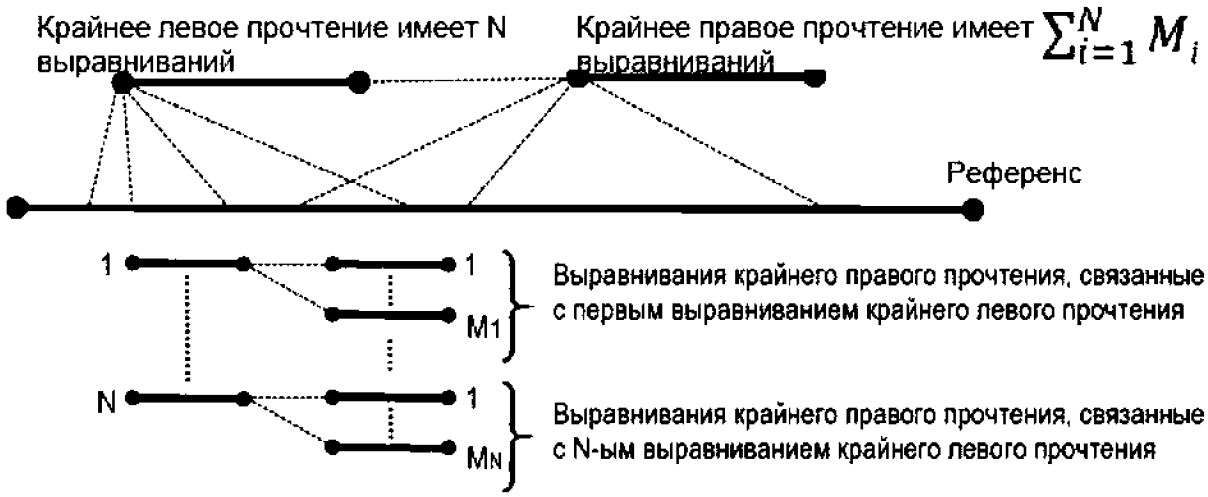
Фиг. 36


Данные Pos
Данные rcomp
Данные pair
Данные rlen
БД типа 1
Заголовок
Pos1
...
Rc1
...
Pair1
...
Rlen1
...
Flags
Информацию заголовка транскодируют в несколько заголовков пакетов
Мультиплексирование генома
Пакет
Блок
Поток, содержащий кодированные данные, принадлежащие pos уровня p
Поток, содержащий кодированные данные, принадлежащие rcomp уровня p
Поток, содержащий кодированные данные, принадлежащие pair уровня p
Поток, содержащий кодированные данные, принадлежащие rlen уровня p
Блок доступа типа 1

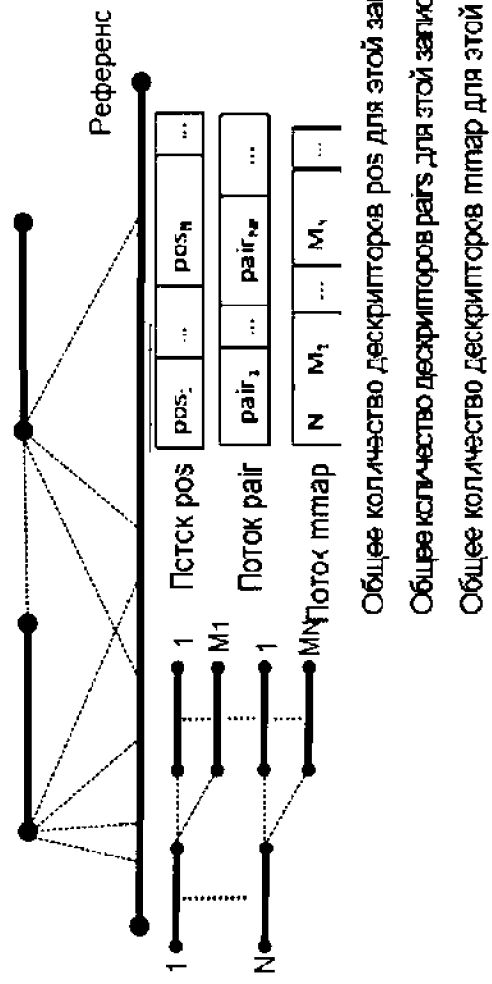
Фиг. 37


270
Блок данных
Блок данных
Блок данных
Блок данных
Пакет данных
Заголовок БД 1
1_1_1
1_1_N
1_2_1
2_1_N
1_3_1
1_3_N
1_4_1
1_4_N
БД № 1
Заголовок БД 2
2_1_1
1_1_M
2_2_1
2_1_M
2_3_1
2_3_M
2_4_1
2_4_M
БД № 2
Заголовок БД K
K_1_1
K_1_J
K_2_1
K_2_J
K_3_1
K_3_J
K_4_1
K_4_J
БД № K
Уровень 1
Уровень 2
Уровень 3
Уровень 4

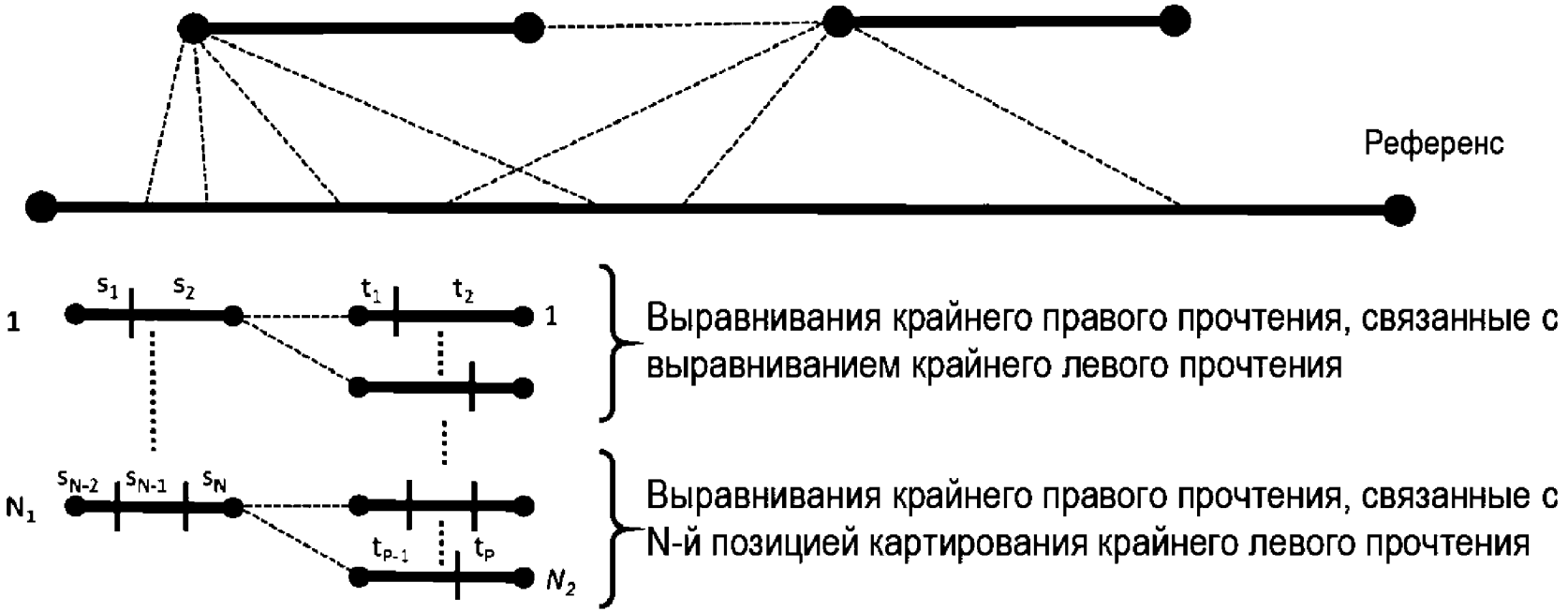
Фиг. 38


Крайнее левое прочтение имеет N выравниваний
Крайнее правое прочтение имеет $\sum_{i=1}^{N} M_i$ выравниваний
Референс
1
1
M1
Выравнивания крайнего правого прочтения, связанные с первым выравниванием крайнего левого прочтения
N
1
MN
Выравнивания крайнего правого прочтения, связанные с N-ым выравниванием крайнего левого прочтения

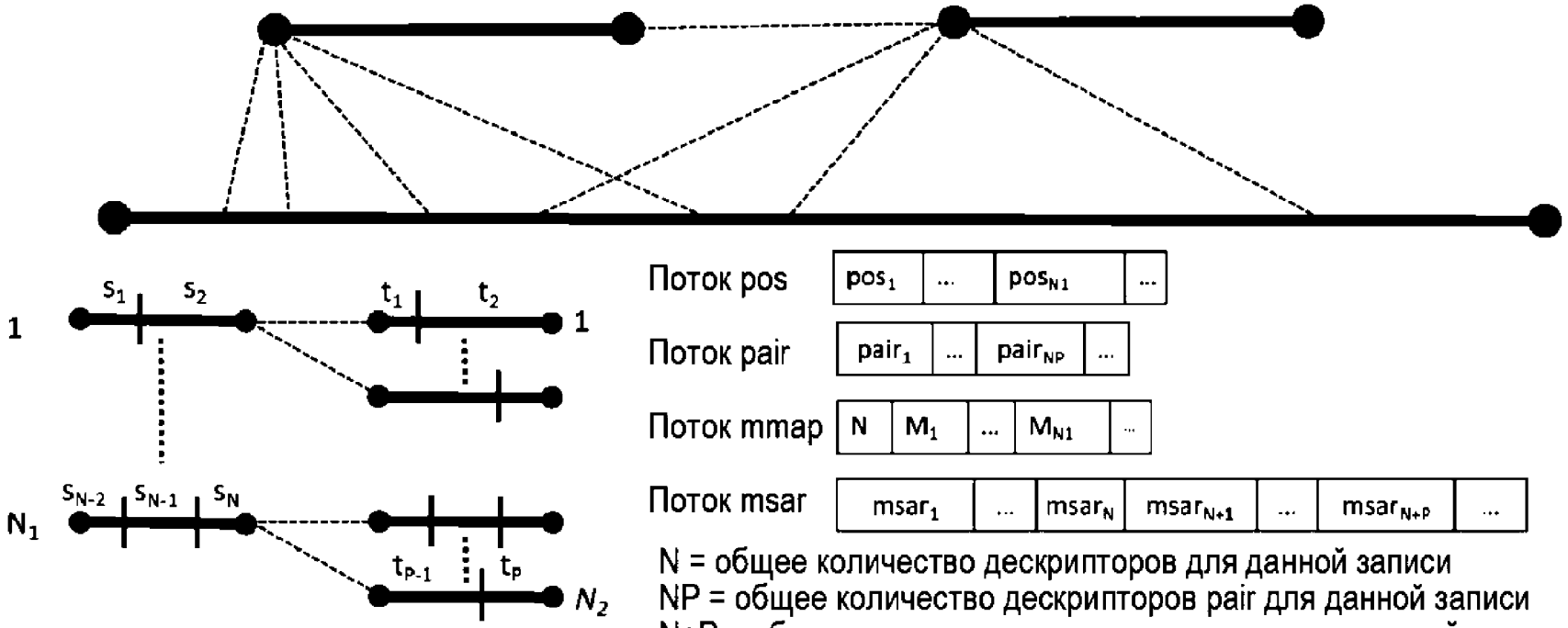
Фиг. 39


Референс
Поток pos
Поток pair
Поток mmap
pos1
posn
pair1
pairNP
N
M1
MN
1
1
1
N
Общее количество дескрипторов pos для этой записи = N
Общее количество дескрипторов pairs для этой записи = NP = $\sum_{i=1}^{N} M_i$ - (к-во Mi = 0) + (необязательно) 1
Общее количество дескрипторов mmap для этой записи = N + 1

Фиг. 40


Референс
1
s_1
s_2
t_1
t_2
1
N_1
s_{N-2}
s_{N-1}
s_N
t_{P-1}
t_P
N_2
Выравнивания крайнего правого прочтения, связанные с выравниванием крайнего левого прочтения
Выравнивания крайнего правого прочтения, связанные с N-й позицией картирования крайнего левого прочтения

Фиг. 41


1
s_1
s_2
t_1
t_2
1
N_1
s_{N-2}
s_{N-1}
s_N
t_{P-1}
t_P
N_2
Поток pos
pos_1 ... pos_{N1} ...
Поток pair
$pair_1$ ... $pair_{NP}$ ...
Поток mmap
N M_1 ... M_{N1} ...
Поток msar
$msar_1$ ... $msar_N$ $msar_{N+1}$ ... $msar_{N+P}$ ...
N = общее количество дескрипторов для данной записи
NP = общее количество дескрипторов pair для данной записи
N+P = общее количество дескрипторов msar для данной записи

Фиг. 42