Министерство образования и науки Российской Федерации

УДК: 004.021
ГРНТИ: 20.01.01, 34.05.25
Инв. №: 310276

УТВЕРЖДЕНО:
Исполнитель: **Государственное образовательное учреждение высшего профессионального образования «Санкт-Петербургский государственный университет информационных технологий, механики и оптики»**
Ректор ГОУВПО «СПбГУ ИТМО» **_____/В.Н. Васильев/** **М.П.**

НАУЧНО-ТЕХНИЧЕСКИЙ ОТЧЕТ

о выполнении первого этапа Государственного контракта
№ 16.740.11.0495 от 16 мая 2011 г.

Исполнитель: Государственное образовательное учреждение высшего профессионального образования «Санкт-Петербургский государственный университет информационных технологий, механики и оптики»
Программа (мероприятие): Федеральная целевая программа «Научные и научно-педагогические кадры инновационной России» на 2009-2013 гг., в рамках реализации мероприятия № 1.2.1 Проведение научных исследований научными группами под руководством докторов наук.
Проект: Разработка метода сборки геномных последовательностей на основе восстановления фрагментов по парным чтениям
Руководитель проекта: **_____/Шалыто Анатолий Абрамович** (подпись)

Санкт-Петербург
2011 г.

СПИСОК ОСНОВНЫХ ИСПОЛНИТЕЛЕЙ

по Государственному контракту 16.740.11.0495 от 16 мая 2011 на выполнение поисковых научно-исследовательских работ для государственных нужд

Организация-Исполнитель: Государственное образовательное учреждение высшего профессионального образования «Санкт-Петербургский государственный университет информационных технологий, механики и оптики»

Руководитель темы:

Доктор технических наук, профессор	_____ подпись, дата	Шалыто А. А.

Исполнители темы:

без ученой степени, без ученого звания	_____ подпись, дата	Царев Ф. Н.
без ученой степени, без ученого звания	_____ подпись, дата	Александров А. В.
без ученой степени, без ученого звания	_____ подпись, дата	Сергушичев А. А.
без ученой степени, без ученого звания	_____ подпись, дата	Мельников С. В.
без ученой степени, без ученого звания	_____ подпись, дата	Буздалов М. В.
без ученой степени, без ученого звания	_____ подпись, дата	Казаков С. В.

РЕФЕРАТ

Отчет 42 с., 4 гл., 15 рис., 6 табл., 13 источников.

Ключевые слова: парные чтения, восстановление фрагментов, сборка генома.

В настоящем отчете излагаются результаты выполнения *поисковых научно-исследовательских работ по теме «Разработка метода сборки геномных последовательностей на основе восстановления фрагментов по парным чтениям»*, выполняемых в рамках государственного контракта, заключенного между Министерством образования и науки Российской Федерации и государственным образовательным учреждением высшего профессионального образования «Санкт-Петербургский государственный университет информационных технологий, механики и оптики» в соответствии с решением Конкурсной комиссии Министерства образования и науки Российской Федерации № 1 (протокол от 26.04.2011 г. № 3/01731000037110000032) по лоту шифр «2011–1.2.1–201–007» «Проведение научных исследований научными группами под руководством докторов наук в следующих областях: – биокаталитические, биосинтетические и биосенсорные технологии; – биомедицинские и ветеринарные технологии жизнеобеспечения и защиты человека и животных; – геномные и постгеномные технологии создания лекарственных средств; – клеточные технологии; – биоинженерия; – биоинформационные технологии» в рамках федеральной целевой программы «Научные и научно-педагогические кадры инновационной России» на 2009 – 2013 годы, утвержденной постановлением Правительства Российской Федерации от 28 июля 2008 года № 568 «О федеральной целевой программе «Научные и научно-педагогические кадры инновационной России» на 2009–2013 годы».

Целями настоящего этапа являются:

1. Выполнение аналитического обзора.
2. Проведение патентных исследований.
3. Выбор и обоснование оптимального варианта проведения исследований и выбор архитектуры выбор архитектуры метода сборки геномных последовательностей.
4. Подготовка плана проведения теоретических и экспериментальных исследований.

Излагаются результаты выполнения аналитического обзора по следующим направлениям: «Методы исправления ошибок в чтениях геномной последовательности», «Методы сборки генома» и «Форматы представления геномных данных».

Приводится обоснование выбора оптимального варианта направления исследований, а также план проведения теоретических и экспериментальных исследований.

Описывается архитектура предлагаемого метода сборки геномных последовательностей.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

В настоящем отчете излагаются результаты выполнения *поисковых научно-исследовательских работ по теме «Разработка метода сборки геномных последовательностей на основе восстановления фрагментов по парным чтениям»*, выполняемых в рамках государственного контракта, заключенного между Министерством образования и науки Российской Федерации и государственным образовательным учреждением высшего профессионального образования «Санкт-Петербургский государственный университет информационных технологий, механики и оптики» в соответствии с решением Конкурсной комиссии Министерства образования и науки Российской Федерации № 1 (протокол от 26.04.2011 г. № 3/0173100003711000032) по лоту шифр «2011–1.2.1-201-007» «Проведение научных исследований научными группами под руководством докторов наук в следующих областях: – биокаталитические, биосинтетические и биосенсорные технологии; – биомедицинские и ветеринарные технологии жизнеобеспечения и защиты человека и животных; – геномные и постгеномные технологии создания лекарственных средств; – клеточные технологии; – биоинженерия; – биоинформационные технологии» в рамках федеральной целевой программы «Научные и научно-педагогические кадры инновационной России» на 2009 – 2013 годы, утвержденной постановлением Правительства Российской Федерации от 28 июля 2008 года № 568 «О федеральной целевой программе «Научные и научно-педагогические кадры инновационной России» на 2009–2013 годы».

Целями настоящего этапа являются:

1. Выполнение аналитического обзора.
2. Проведение патентных исследований.
3. Выбор и обоснование оптимального варианта проведения исследований и выбор архитектуры выбор архитектуры метода сборки геномных последовательностей.
4. Подготовка плана проведения теоретических и экспериментальных исследований.

Отчет имеет следующую структуру. В первой главе приводятся результаты выполнения аналитического обзора по трем направлениям:

- методы исправления ошибок в чтениях геномной последовательности;
- методы сборки генома;
- форматы представления геномных данных.

Во второй главе обосновывается выбор оптимального направления исследований.

В третьей главе приводится план проведения теоретических и экспериментальных исследований.

В четвертой главе описывается архитектура разрабатываемого метода машинного обучения.

Каждая из глав снабжена выводами, кратко резюмирующими содержание главы. В заключении дается общая оценка выполненных работ по этапу.

Изучение генома человека и других живых существ имеет важное прикладное значение. На основании результатов сборки генома конкретного человека возможна реализация индивидуальной медицины – определения предрасположенности к различным болезням, создание индивидуальных лекарств и т. д. Кроме этого, на основе результатов исследования геномов растений и животных с использованием методов биоинженерии могут быть выведены новые их виды, обладающие определенными свойствами.

Задача разработки методов сборки геномных последовательностей является среди всех задач биоинформатики, в определенном смысле, центральной. Это объясняется тем, что без ее решения нельзя приступить к детальному изучению генома живого существа и его анализу с применением других алгоритмов биоинформатики.

По оценкам экспертов (www.dubna-oez.ru/images/data/gallery/10_2948_.pps) технологии секвенирования генома в настоящее время развиваются быстрее, чем методы сборки геномных последова-

тельностей. Таким образом, актуальной является задача разработки новых методов сборки генома, соответствующих по своим параметрам существующим методам секвенирования.

Сложность задачи сборки геномной последовательности обусловлена следующими факторами:

- большой объем входных данных, который составляет порядка десятков и сотен гигабайт;
- сложность структуры генома – наличие в нем повторов и полиморфизмов;
- наличие ошибок в исходных данных – чтениях, полученных с устройств-секвенаторов.

Для решения указанных проблем сборку геномной последовательности разбивают на три этапа:

- исправление ошибок в исходных данных – чтениях геномной последовательности;
- сборка контигов – достаточно длинных непрерывных фрагментов искомой геномной последовательности;
- построение скэффолдов – наборов контигов, для которых с большой степенью уверенности определено их взаимное расположение в геномной последовательности.

Существующие в настоящее время методы сборки генома обладают двумя недостатками:

- высокие требования к ресурсам – **для работы существующих программ требуется объем оперативной памяти в сотни гигабайт**, что может быть достигнуто только за счет использования весьма дорогостоящих суперкомпьютеров и кластеров;
- отсутствие методов внутреннего контроля качества процесса сборки контигов – сборка контигов практически во всех существующих методах проводится в один непрерывный этап, в процессе которого его качество контролировать нельзя.

Для исправления указанных недостатков будет разработан новый метод сборки генома, основанный на восстановлении фрагментов последовательности по парным чтениям. В этом методе сборка скэффолдов будет разбита на два подэтапа, между которыми будет проводиться внутренний контроль качества сборки.

Кроме этого, при разработке метода **будут использоваться алгоритмы, которые не требуют больших объемов оперативной памяти**. Таким образом, общие требования разрабатываемого метода к ресурсам будут существенно ниже, чем у существующих методов.

Изложенное позволяет утверждать, что результаты выполнения научно-исследовательской работы будут превышать мировой уровень разработок в рассматриваемой области.

1. ВЫПОЛНЕНИЕ АНАЛИТИЧЕСКОГО ОБЗОРА

В настоящем разделе приводятся результаты аналитического обзора. Обзор проводился по следующим направлениям:

- методы исправления ошибок в чтениях геномной последовательности;
- методы сборки генома;
- форматы представления геномных данных.

1.1. МЕТОДЫ ИСПРАВЛЕНИЯ ОШИБОК В ЧТЕНИЯХ ГЕНОМНОЙ ПОСЛЕДОВАТЕЛЬНОСТИ

Данный подход был изложен в статье «*Whole-Genome Sequencing and Assembly with High-Throughput, Short-Read Technologies*» 2007 года (Andreas Sundquist, Mostafa Ronaghi, Haixu Tang, Pavel Pevzner, Serafim Batzoglou) [9].

SHRAP (Short Reads Assembly Protocol) – протокол сборки длинных геномов, основанный на иерархическом секвенировании (рис. 1). Иерархическое секвенирование (*hierarchical sequencing*) – исторически один из первых способов секвенирования. Первый шаг протокола состоит в том, чтобы выделить много фрагментов генома длиной около 150 kb (*kilobase* – тысяч нуклеотидов). Эти фрагменты называются *клоны* (*clones*) и обеспечивают достаточно большое покрытие генома (порядка десятикратного). Затем из каждого клона получаются чтения длиной около 200 нуклеотидов. Чтения обеспечивают небольшое покрытие клона – порядка двукратного. Таким образом, в итоге чтения обеспечивают двадцатикратное покрытие генома.

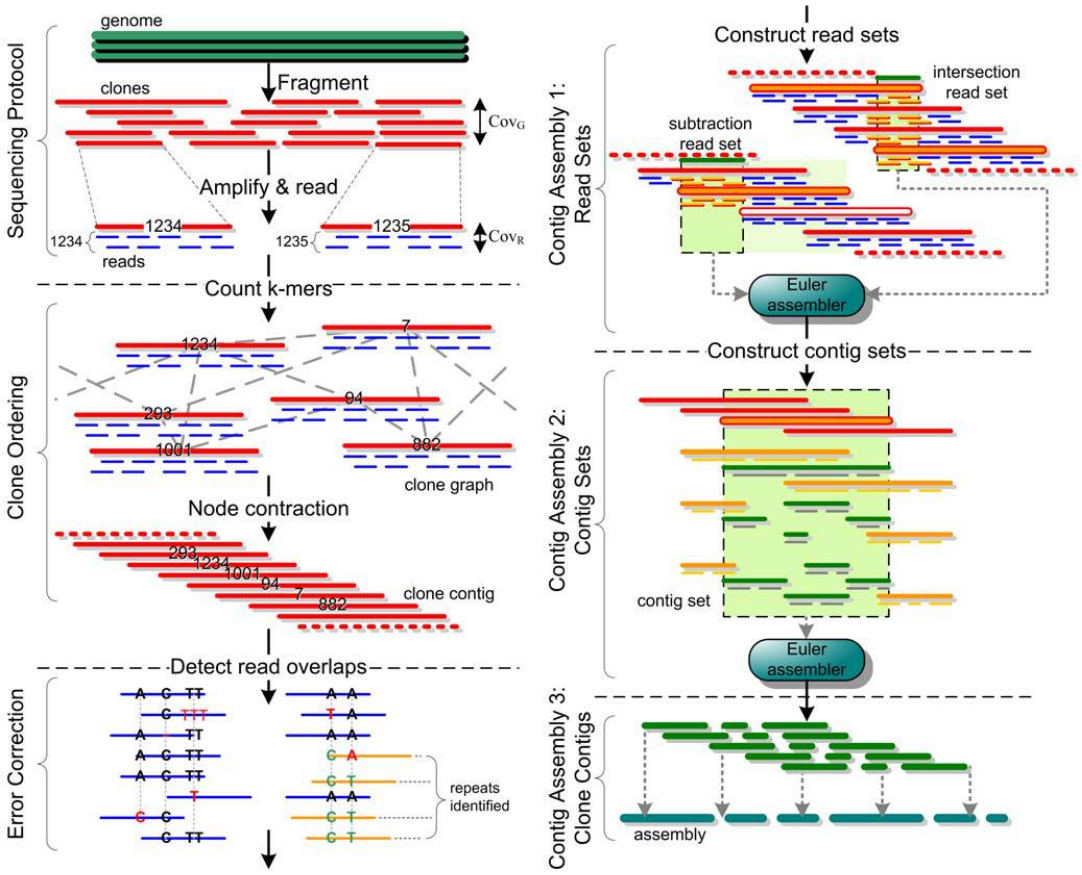

Рис. 1. Схема работы сборщика, основанного на иерархическом секвенировании

В традиционном иерархическом секвенировании примерное расположение клонов в геноме известно, поэтому сначала сами клоны собираются по чтениям, а затем информация о расположении

клонов используется для сборки их в более длинные куски. В предлагаемом методе информация о расположении клонов отсутствует, поэтому метод сборки совершенно другой.

Сначала при помощи чтений определяются пересекающиеся клоны, из которых в итоге строятся *клоны-контиги* (*clone contigs*) – цепочки клонов, сильно пересекающихся друг с другом, расположенные в том порядке, в котором они расположены в геноме. Информация о расположении клонов-контигов используется в дальнейшем для сборки чтений и исправления ошибок в них.

На стадии исправления ошибок информация о расположении клонов-контигов используется для ускорения вычисления наложений чтений друг на друга. Для каждого чтения рассматриваются только те из них, которые расположены в том же клоне-контиге, причем в клоне, пересекающимся с клоном обрабатываемого чтения. Для нахождения пересечения между чтениями применяются 16-меры.

На первой стадии для каждого чтения строится множество других чтений, пересекающихся с ним. Затем для каждого чтения проводится два теста – тест на ошибочность (*error-rate test*) и тест на коррелированность (*correlation test*). Тест на ошибочность отфильтровывает чтения, содержащие слишком много отличий от остальных чтений (больше утроенного ожидаемого числа ошибочно прочитанных нуклеотидов). В тесте на коррелированность выделяются чтения, повторяемость которых вызвана повторами в геноме. Чтения, не прошедшие хотя бы один тест, удаляются из множества.

После удаления не прошедших тесты чтений множество *транзитивно* пополняется. Назовем множество чтений, перекрывающихся с p, R_p. Для транзитивного пополнения рассматривается каждая пара чтений p и q из R_r, и если их расположение относительно чтения r влечет их перекрытие, то p добавляется в R_q, а q – в R_p.

На второй стадии чтения одного множества вновь выстраиваются в цепочку. После этого из множества снова удаляются не проходящие тесты чтения. На третьей стадии при помощи применения простого правила большинства к каждой позиции из чтений состваляется цепочка нуклеотидов.

1.1.1. Сборщик *AllPaths*

Подход к исправлению ошибок, примененный в сборщике *AllPaths*, изложен в статье «**ALLPATHS**: De novo assembly of whole-genome shotgun microreads» 2008 года (Jonathan Butler, Iain MacCallum, Michael Kleber, Ilya A. Shlyakhter, Matthew K. Belmonte, Eric S. Lander, Chad Nusbaum, David B. Jaffe) [1].

Алгоритм исправления ошибок в этом случае использует набор чтений и является первой частью алгоритма сборки. В процессе работы алгоритма чтения делятся на три группы: чтения, которые оставляются без изменений, чтения, в которых осуществляются исправления, и чтения, которые удаляются и больше не используются.

Для начала подсчитывается частотная статистика k-меров. Для каждого m вычисляется число k-меров, встречающихся в чтениях ровно m раз. Полученная функция имеет два пика. Первый, резкий, находится в точке $m=1$ и связан с ошибками в чтениях. Второй же, гораздо более гладкий, обусловлен статистическим распределением k-меров при большом покрытии и с ошибками не связан. Между этими двумя пиками есть минимум функции, располагающийся в точке m_1. Большинство k-меров, отвечающих точкам левее m_1, содержат ошибки, тогда как большинство располгающихся справа от нее ошибок не содержат. Правые k-меры называются надежными (strong).

Подсчет статистики производится для нескольких значений k: 16, 20, 24. Для каждого из значений определяется *m1*. Если для некоторого чтения все k-меры для всех значений k являются надежными, то это чтение оставляется без изменений и считается правльным. В противном случае делается попытка исправить один или два нуклеотида (большее число исправлений возможно, но также влечет за собой дополнительное увеличение времени работы алгоритма). Каждому изменению ставится в соответствие вероятность, пропорциональная качеству нуклеотида, на который

производится замена. Если наиболее вероятная замена оказывается хотя бы в 10 раз более вероятной, чем вторая по величине вероятности, то эта замена осуществляется и чтение считается исправленным. В противном случае чтение удаляется.

1.1.2. Сборщик *Velvet*

Подход, используемый в сборщике *Velvet*, изложен в статье «*Velvet*: Algorithms for de novo short read assembly using de Bruijn graphs» 2008 года (Daniel R. Zerbino, Ewan Birney) [12].

В этом подходе применяется разновидность графа де Брюина (de Bruijn graph). В этом графе каждая вершина хранит упорядоченный список *k*-меров, причем соседние *k*-меры в списке пересекаются по *k-1* символам. Вся информация, хранящаяся в вершине, может быть представлена первым *k*-мером в списке и списком последних нуклеотидов каждого *k*-мера.

Каждая вершина имеет «прикрепленную» к ней вершину-двойника, в которой хранится список тех же *k*-меров, но развернутых и комплементарных. Пара из вершины и ее двойника называется блоком. Вершину, являющуюся двойником вершины *A*, будем обозначать $\tilde{A}$.

Вершины в графе соединяются ориентированными ребрами по тому же принципу, по которому *k*-меры соседствуют в списках, хранящихся в вершинах, – суффикс длины *k-1* последнего *k*-мера вершины, из которой выходит ребро, совпадает с префиксом первого *k*-мера вершины, в которую входит это ребро. Из-за симметрии блоков наличие ребра из вершины *A* в вершину *B* всегда влечет наличие обратного ребра из $\tilde{B}$ в $\tilde{A}$ (рис. 2).

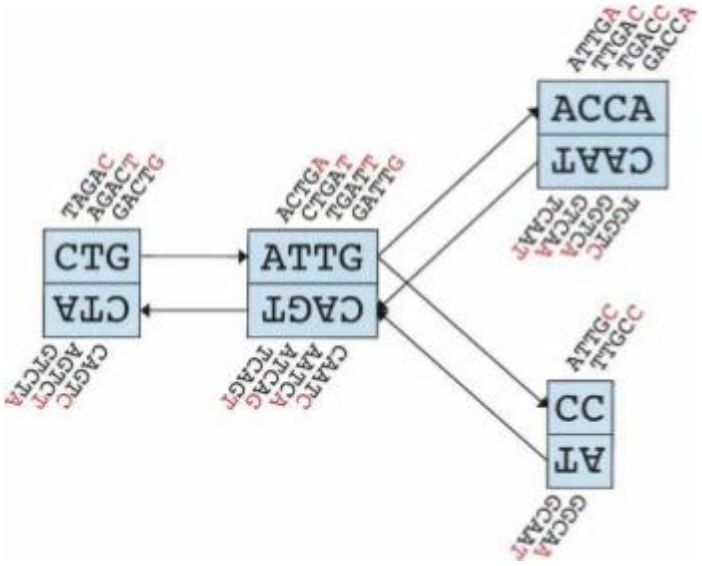

Рис. 2. Пример графа де Брюина для k = 5

Построение графа де Брюина – первая стадия работы сборщика. После построения граф упрощается. После этого на нем запускается алгоритм исправления ошибок.

Ошибки чтения порождают в графе структуры трех типов: отростки (tips), пузыри (bulges) и ошибочные ребра. Первые возникают из-за ошибок на концах чтений, вторые – из-за ошибок внутри чтений, а третьи – в результате повторных ошибок. Эти три типа ошибок удаляются одна за другой.

1.1.2.1. Удаление «отростков»

Ошибки данного типа порождают в графе обрывающиеся цепочки вершин, поэтому их удаление на первый взгляд не представляет особых трудностей. Однако не все обрывающиеся цепочки вершин небольшой длины вызваны ошибками чтения – некоторые из них могут быть

вызваны небольшим покрытием чтениями небольшого куска генома. Для того чтобы различать эти два случая, используются два критерия: длина цепочки и критерий альтернативности (minority count).

Критерий длины цепочки состоит в том, что удаляются только цепочки длины меньше *2k*. Такой выбор обусловлен максимальной длиной ошибочного куска, вызванного наличием двух идущих подряд ошибочно прочитанных нуклеотидов.

Критерий альтернативности состоит в том, что ребро, которое ведет в первую вершину цепочки, должно иметь меньший вес, чем любое другое ребро, ведущее из той же вершины. Иными словами, путь в отросток должен быть альтернативой более общему пути в графе.

Таким образом, отросток удаляется только при выполнении двух приведенных выше критериев.

1.1.2.2. Удаление «пузырей»

Пузырем называется несколько путей между одной парой вершин, содержащих схожие последовательности символов. Нахождение таких путей осуществляется при помощи модифицированного поиска в ширину. Запущенный из произвольной вершины, алгоритм посещает вершины в порядке увеличения расстояния от этой вершины, причем под расстоянием между парой вершин, соединенных ребром, понимается длина последовательности, хранящейся в вершине назначения, деленная на вес ребра, соединяющего эти вершины в этом направлении. Как только поиском обнаруживается ребро, ведущее в уже посещенную вершину, из нее и текущей вершины запускается поиск ближайшего общего предка. После нахождения общего предка строки, соответствующие путям из этого предка в рассматриваемую вершину, сравниваются. Если они достаточно похожи, то есть различия в них могут быть списаны на ошибки в чтениях, то пути объединяются, причем из двух путей выбирается кратчайший согласно выбранной метрике (рис. 3). Заметим, что модифицирование путей – операция довольно дорогостоящая из-за дополнительной информации, хранящейся в графе.

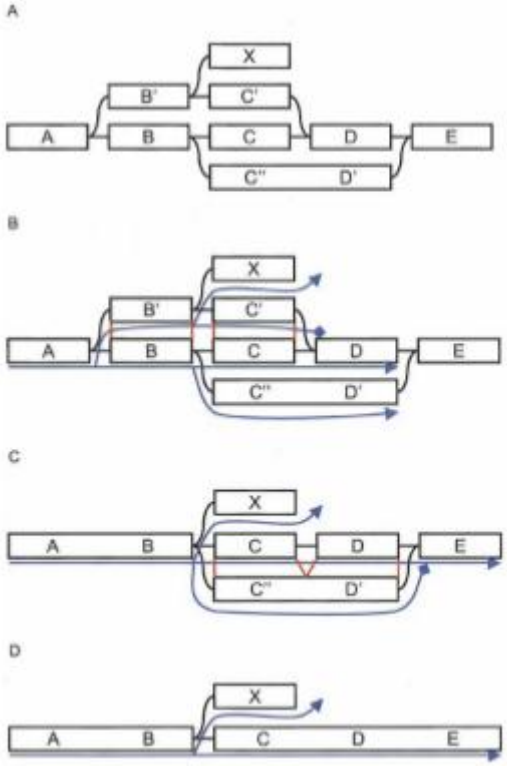

Рис. 3. Пример работы алгоритма удаления пузырей

1.1.2.3. Удаление ошибочных ребер

Удаления ошибочных ребер производится на основе их веса. Считается, что после работы предыдущих стадий ребра, имеющие слишком маленький вес – слишком мало покрытые, вызваны ошибочными чтениями.

1.1.3. Алгоритм *EULER-USR*

Алгоритм *EULER_USR* изложен в статье «De novo fragment assembly with short mate-paired reads: Does the read length matter?» 2009 года (Mark J. Chaisson, Dumitru Brinza and Pavel A. Pevzner) [2].

Алгоритм состоит из трех шагов:

- Определение длины длины префиксов с неплохим качеством и исправление ошибок в них на основе частотного анализа *k*-меров. После этой операции остается множество префиксов чтений, практически не содержащих ошибок.
- Построение на основе *k*-меров полученных префиксов *графа повторов* (*repeat-graph*).
- Упрощение построенного графа.

Для исправления ошибок используется простые соображения, по которым *k*-меры делятся на две группы – надежные (solid) и все остальные. Надежными называются *k*-меры, которые встречаются в чтениях не меньше некоторого заранее выбранного числа *m*. Если все *k*-меры одного чтения являются надежными, то считается, что чтение не содержит ошибок. В противном случае производится попытка исправить ошибки в нем. Рассматриваются только ошибки замены, так как ошибки вставки и удаления редки в чтениях секвенатора *Illumina*.

Для исправления ошибок жадным образом ищется минимальное число исправлений, которое необходимо осуществить, чтобы сделать каждый *k*-мер в чтении надежным. Для каждого исправления на каждой позиции записывается, сколько *k*-меров становится надежными в результате этого исправления. Исправление, которое делает максимальное число *k*-меров надежными, применяется, если это число не меньше некоторого заранее выбранного порога *t*. Когда находится подходящее исправление, выводится максимальный по длине префикс чтения, содержащий только надежные *k*-меры. Суффикс *k*-мера, считающийся менее надежным, будет исправлен на более поздней стадии работы алгоритма.

Для выбора параметра *k* полагается, что распределение *k*-меров является смесью двух распределений – распределения ошибочных *k*-меров и распределения правильных (рис. 4). И те, и другие *k*-меры распределены по закону Пуассона, но распределение правильных *k*-меров имеет большое среднее, поэтому аппроксимируется распределением Гаусса. Параметр *m* выбирается как точка минимума распределения *k*-меров, параметры которых определяются исходя из частотного анализа *k*-меров.

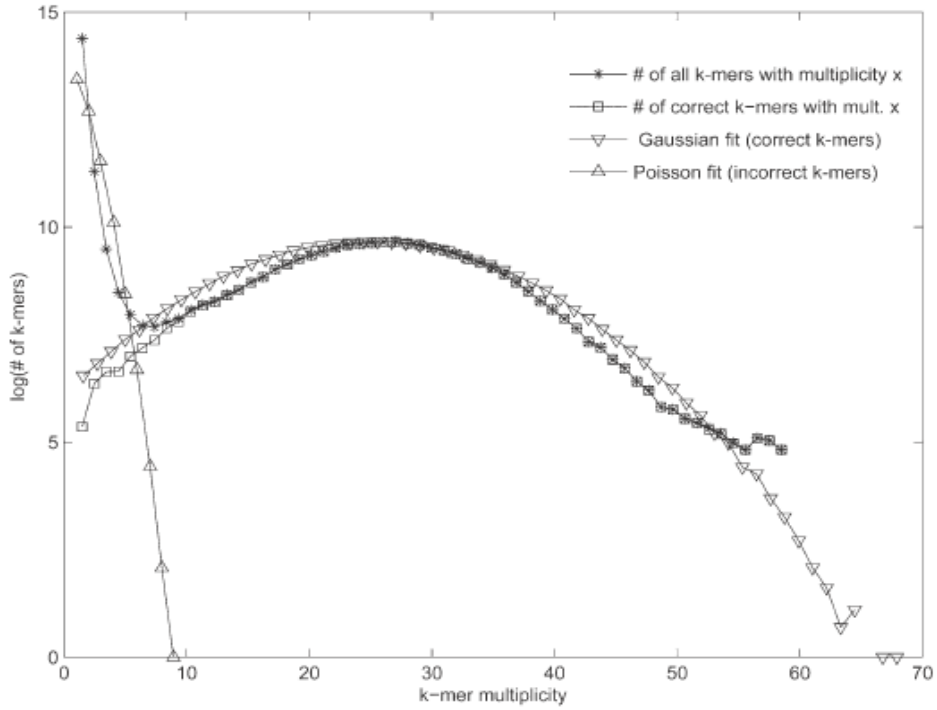

Рис. 4. Экспериментально полученное распределение k-меров

1.1.4. Сборщик *AbySS*

Подход, применяемый в сборщике ABySS (Assembly ByShort Sequences), изложен в статье «ABySS: A parallel assembler for short read sequence data» 2009 года (Jared T. Simpson, Kim Wong, Shaun D. Jackman, Jacqueline E. Schein, Steven J.M. Jones, Inanc Birol) [10]. По сути он ничем не отличается от подхода, применяемого в *Velvet'e.*

1.1.5. Сравнение рассмотренных методов исправления ошибок

В табл. 1 представлено сравнение рассмотренных в обзоре методов исправления ошибок. Сравнение проводилось по следующим пунктам:

- способ представления данных;
- возможность распараллеливания;
- участие в проекте de novo Genome Assembly Project (*dnGASP*, организован Национальным центром геномного анализа, Барселона, Испания, http://cnag.bsc.es);
- доступность.

Таблица 1. Сравнение рассмотренных в обзоре методов исправления ошибок

Название работы	Название сборщика	Способ представления данных	Возможность распараллеливания	Участие в проекте dnGASP	Доступность
Whole-Genome Sequencing and Assembly with High-Throughput, Short-Read Technologies (2007)	*SHRAP*	Работа напрямую с чтениями	Возможность работы в несколько потоков на одном узле	Нет	Исходный код предоставляется по запросу.
ALLPATHS: De novo assembly of whole-genome shotgun microreads (2008)	*ALLPATHS*	Частотный анализ *k*-меров	Возможность работы в несколько потоков на одном узле	Нет	Исходный код доступен для скачивания.
Velvet: Algorithms for de novo short read assembly using de Bruijn graphs (2008)	*Velvet*	Граф де Брюина	Возможность работы в несколько потоков на одном узле	Один из разработчиков был приглашенным докладчиком	Исходный код доступен для скачивания. Выпущен под лицензией «GPLv3»
De novo fragment assembly with short mate-paired reads: Does the read length matter? (2009)	*EULER-USR*	Частотный анализ *k*-меров	Возможность работы в несколько потоков на одном узле	Нет	Исходный код доступен для скачивания.
ABySS: A parallel assembler for short read sequence data (2009)	*ABySS*	Граф де Брюина	Возможность распараллеливания на несколько узлов	Да	Исходный код доступен для скачивания. Выпущен под лицензией «BCCA (academic use)»

1.2. Методы сборки генома

В настоящем разделе приводится обзор существующих методов сборки генома.

1.2.1. Программное средство *ABySS*

В статье [10] изложен подход к сборке контигов в программном средстве *ABySS*. Этот подход состоит из двух этапов:

- сборка контигов без учета парной информации;
- разрешение неоднозначностей с помощью парной информации и наращивание контигов.

В основе всего подхода лежит распределенный граф де Брюина.

Для того, чтобы собрать первоначальные версии контигов происходит объединение последовательностей смежных однозначных ребер – ребро называется однозначным, если исходящая степень его начальной вершины и входящая степень конечной вершины равны единице.

На втором этапе между контигами устанавливаются связи, используя парную информацию. Пара чтений называется связывающей два контига, если первое чтение картируется на первый контиг, а второе – на второй. Между двумя контигами устанавливается связь, если число связывающих их чтений больше некоторой константы p (по умолчанию используется $p = 5$). Для каждого контига C_i строится множество связанных с ним контигов P_i. Затем в графе связей контигов ищется уникальный путь, проходящий через все контиги из P_i. В качестве ограничений при поиске выступают оценка на расстояния между контигами на основе принципа максимального правдоподобия и эвристическая оценка на число посещенных вершин. После того, как поиск таких путей для каждого контига завершился (успешно или нет), согласующиеся пути сливаются, образуя конечные контиги.

1.2.2. Программное средство *ALLPATHS*

Алгоритм, лежащий в основе сборщика *ALLPATHS*, изложен в статье [1]. Он также основан на графе де Брюина и состоит из нескольких этапов.

Первым этапом является создание начального приближения контигов. Для этого, как и в *ABySS*, рассматриваются последовательности смежных однозначных ребер, которые и объявляются приближением контигов.

Вторым шагом является выбор контигов, вокруг которых будет происходить локализация – попытка выделить те чтения, которые картируются на реальный геном близко к положению контига. В качестве таких контигов лучше всего брать длинные контиги, уникальные в геноме. Для проверки уникальности можно использовать ожидаемое и реальное покрытие чтениями.

Затем для каждого выбранного контига – центра локализации – рассматривается множество последовательностей (чтений и контигов), лежащих в пределах, например, 10000 нуклеотидов от краев этого контига – окрестность контига. Если построить приближение этого множества, то можно будет собрать небольшой участок генома, практически только из тех чтений, которые в него картируются. Это позволяет существенно уменьшить объем данных. Для начала выделяется набор почти уникальных в геноме первоначальных контигов, лежащих в рассматриваемом промежутке. Это происходит с помощью итеративного добавления новых контигов, связанных с уже найденными (рис. 5). При этом для каждого контига можно узнать его смещение относительно центра (с какой-то погрешностью), что позволяет отбросить далеко отстоящие контиги. Затем строятся два множества чтений: первичное, состоящее из чтений картирующихся в рассматриваемую область генома, но, возможно, не всех таких чтений, и вторичное, которое кроме интересующих чтений, возможно, содержит и некоторые другие. В первичное множество входят, все пары чтений, хотя бы одно из которых картируется на какой-либо из выбранных контигов. Во вторичное множество чтений входят те пары чтений, которые могут быть собраны из чтений из первичного множества. Затем

перекрывающиеся пары чтений из вторичного множества сливаются, образуя пары более длинных последовательностей.

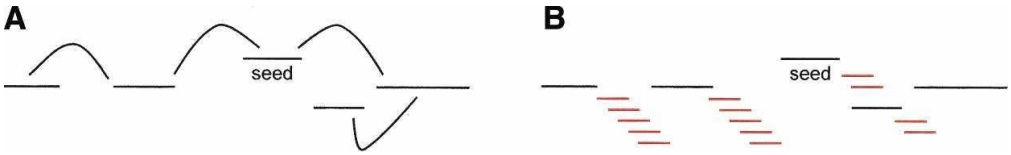

Рис. 5. Построение окрестности контига

Для каждой окрестности контига в полученных на предыдущем этапе парах последовательностей ищутся все пути, соединяющие первую последовательность со второй, состоящие из последовательностей из того же множества. Путем является последовательность последовательностей, перекрывающихся с предыдущей не менее чем на *K* нуклеотидов.

На следующем этапе происходит склеивание путей, найденных на предыдущем этапе, если они имеют длинный общий подпуть (рис. 6). Таким образом, осуществляется локальная сборка генома. Затем результаты локальных сборок сливаются между собой, образуя конечный вариант сборки генома.

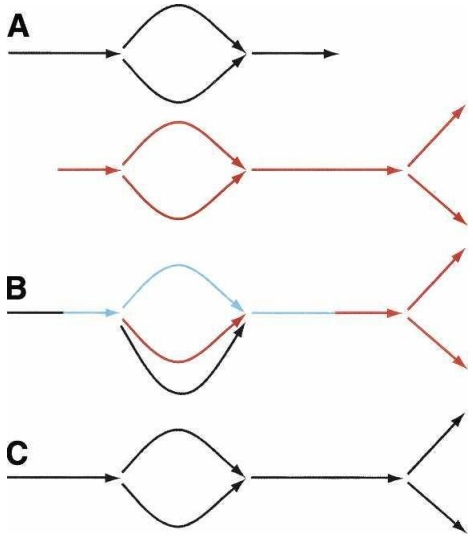

Рис. 6. Слияние путей

1.2.3. Программное средство *Velvet*

Алгоритм работы *Velvet* изложен в статье [12].

После исправления ошибок остается граф, вершинами которого являются последовательности, соответствующие вершинам графа де Брюина, объединенным по однозначным ребрам. С этого момента начинает работу модуль *Breadcrumb*, ответственный за разрешение проблемных участков с помощью парной информации.

Зная распределение длин пропусков в парных чтениях, можно выделить длину *L* такую, что число парных чтений, имеющих пропуск длиннее *L*, ничтожно мало. Можно выделить «длинные» вершины – вершины, длины которых не меньше чем *L*. Заметим, что таких «длинных» вершин должно быть достаточно много, поэтому значение *L* не должно быть слишком большим. Затем выделяются парные «длинные» вершины, связанные несколькими парными чтениями (рис. 7). Если после «длинной» вершины может следовать несколько «длинных» вершин, то она называется неоднозначной. Все узлы, на которые картируется чтение, парное к которому картируется на однозначную «длинную» вершину, помечаются. После этого уникальные узлы последовательно

наращиваются, добавлением помеченных узлов, связанных с текущим. Этот процесс продолжается до тех, пор пока не встречается узел, у которого нет продолжения или есть несколько вариантов продолжения. Если несколько «длинных» вершин последовательно соединились такими путями, они объединяются в контиг.

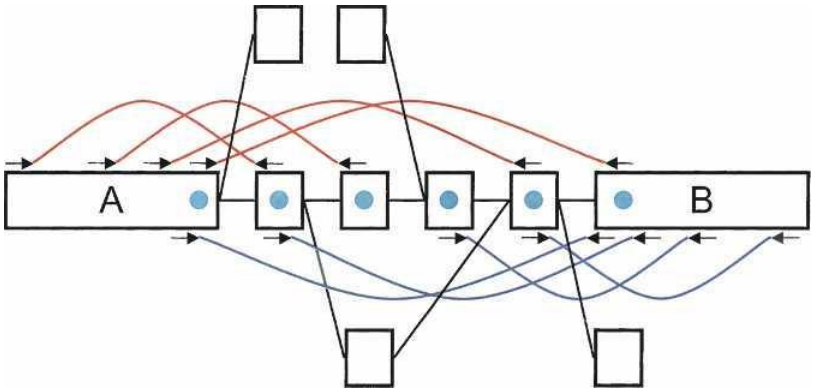

Рис. 7. Процесс работы алгоритма *Breadcrumb*

1.2.4. Программное средство *SOAPdenovo*

Способ сборки, используемый в *SOAPdenovo*, описан в статье [4]. Этот алгоритм разработан в Пекинском геномном институте (Китай).

Как и многие другие сборщики из коротких чтений *SOAPdenovo* использует граф де Брюина. После исправления ошибок пути, которые не содержат вершин с ветвлениями, объявляются первоначальным приближением контигов (рис. 8).

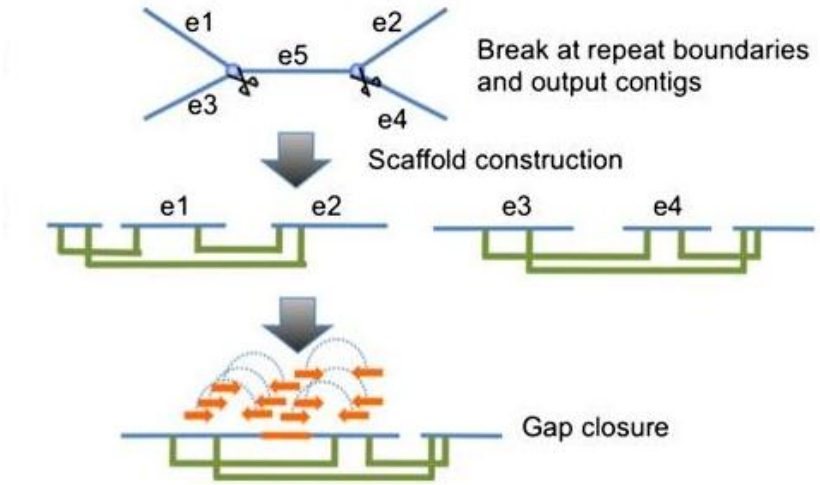

Рис. 8. Схема последних этапов работы *SOAPdenovo*

Затем происходит картирование парных чтений на контиги. Между двумя контигами устанавливается порядок и оценивается расстояние между ними, если их соединяют как минимум три пары чтений. Далее наборы контигов с совместимыми связями линеаризуются, образуя скэффолды.

Во время следующего этапа происходит заполнение промежутков между соседними контигами в скэффолдах. Для этого для каждой пары соседних контигов выбирается множество парных чтений, из котрых одно картируется на один из контигов. Далее проводится локальная сборка только из этих чтений. Если она завершается успешно, тогда получается соединить два контига в один, более длинный.

1.2.5. Программное средство *EULER-USR*

В статье [2] изложены основы работы средства *EULER-USR* из набора *EULER*.

Сборщик *EULER* использует для сборки сведение к задаче о поиске эйлерова суперпути – пути, проходящем через все ребра и содержащем в качестве подпутей все пути из наперед заданного множества *P*. Эта задача, в свою очередь, сводится к задаче поиска эйлерова пути. Такая постановка задачи была сформулирована в статье [7]. В этой статье предлагается в качестве графа использовать граф де Брюина, а в качестве множества *P* для начала использовать пути, соответствующие чтениям. Затем, после некоторых преобразований, упрощающих граф, рассмотреть все парные чтения (r_1 , r_2) и найти все пути, соединяющие r_1 и r_2 и имеющие длину около ожидаемой, определенной из априорного распределения длин. Если такой путь единственный, то он добавляется в множество *P*. Этот процесс повторяется до тех пор, пока в графе происходят изменения.

В статье [2] предлагается способ учета и тех парных чтений, для которых нашлось несколько путей. Пусть для парных чтений $(read_{start}, read_{end})$ нашлось несколько путей (рис. 9). Обозначим за e_{start} и e_{end} ребра, в которых начинаются и заканчиваются $read_{start}$ и $read_{end}$ соответственно. Для каждого пути *p*, соединяющего $read_{start}$ и $read_{end}$, рассмотрим величину $support(e_{start},e_{end},p)$ – число парных чтений, поддерживающих этот путь – таких парных чтений *(r1, r2)*, что одно из них картируется на ребро e_{start} или e_{end}, а другое на путь *p*. Из всех путей выбирается путь с максимальным значением *support*, если оно не меньше некоторого порогового значения *MinSupport*.

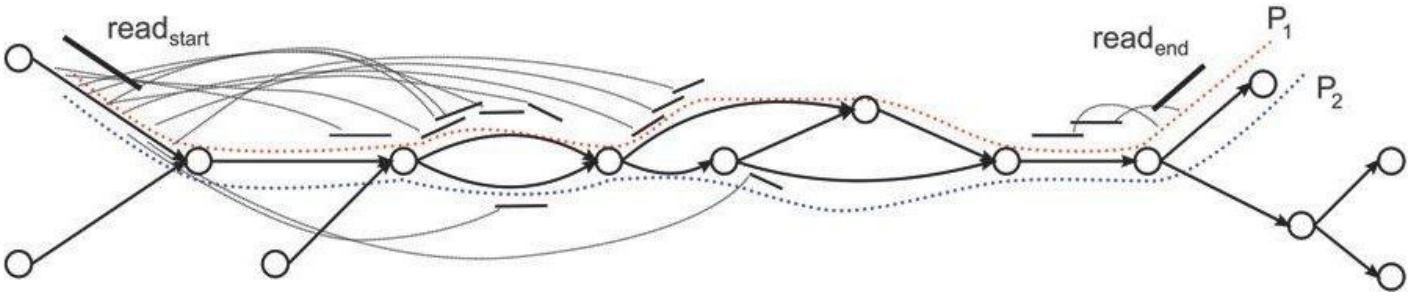

Рис. 9. Пути между $read_{start}$ и $read_{end}$.

1.2.6. Сравнение рассмотренных методов сборки контигов

В табл. 2 приведено сравнение рассмотренных в обзоре методов сборки контигов. Сравнение производилось по следующим пункта:

- способ представления данных;
- возможность распараллеливания;
- участие в проекте *dnGASP*;
- доступность.

Таблица 2. Сравнение рассмотренных в обзоре методов сборки контигов`

Название работы	Название сборщика	Способ представления данных	Возможность распараллеливания	Участие в проекте dnGASP	Доступность
ABySS: A parallel assembler for short read sequence data (2009)	*ABySS*	Граф де Брюина	Возможность распараллеливания на несколько узлов	Да	Исходный код доступен для скачивания. Выпущен под лицензией «BCCA (academic use)»
ALLPATHS: De novo assembly of whole-genome shotgun microreads (2008)	*ALLPATHS*	Граф де Брюина	Возможность работы в несколько потоков на одном узле	Нет	Исходный код доступен для скачивания.
Velvet: Algorithms for de novo short read assembly using de Bruijn graphs (2008)	*Velvet*	Граф де Брюина	Возможность работы в несколько потоков на одном узле	Один из разработчиков был приглашенным докладчиком	Исходный код доступен для скачивания. Выпущен под лицензией «GPLv3»
De novo assembly of human genomes with massively parallel short read sequencing (2010)	*SOAPdenovo*	Граф де Брюина	Возможность работы в несколько потоков на одном узле	Да	Исходный код доступен для скачивания. Выпущен под лицензией «GPLv3»
De novo fragment assembly with short mate-paired reads: Does the read length matter? (2009)	*EULER-USR*	Граф де Брюина	Возможность работы в несколько потоков на одном узле	Нет	Исходный код доступен для скачивания.

1.3. Форматы представления геномных данных

В настоящем разделе приведен обзор форматов представления геномных данных.

1.3.1. Шкала оценки качества геномных данных *Phred*

В большинстве форматов представления геномных данных для хранения качества отдельных нуклеотидов используется шкала качества *Phred* [8]. В ней качество одного нуклеотида кодируется одним символом. Для получения вероятности того, что конкретный нуклеотид был считан правильно, необходимо получить число P . Для этого из кода символа необходимо вычесть код базового символа. Базовый символ является разным в разных форматах. Вероятность ошибки составляет $10^{-P/10}$.

В файлах полученных на секвенаторе *Illumina* символ *B* обозначает, что данные о нуклеотиде являются абсолютно недостоверными, и не должны использоваться.

1.3.2. Формат *FASTQ*

Формат *FASTQ* используется для хранения нуклеотидных последовательностей их информации об их качестве. Файлы в этом формате имеют расширение `fastq`. Файл в формате *FASTQ* представляет собой текстовый файл, состоящих из нескольких блоков [6]. Каждый блок состоит из четырёх строк.

Первая строка начинается с символа @ и содержит идентификатор чтения и возможно его описание.

Вторая строка содержит нуклеотиды чтения, при этом символы *A, T, G, C* означают соответственно Аденин, Тимин, Гуанин, Цитозин, а символ N означает что соответствующий нуклеотид неизвестен.

Третья строка содержит один символ – плюс, и возможно идентификатор чтения, совпадающий с идентификатором чтения в первой строке.

Четвертая строка содержит описание качества чтения по одному символу на нуклеотид. Качество записано по шкале качества *Phred*.

Для описания парных чтений используется два файла: в первом из них записаны в описанном формате один конец каждого чтения, а во втором другой конец каждого чтения, при этом порядок чтений в этих двух файлах совпадает.

1.3.3. Формат *FASTA*

Формат *FASTA* используется для хранения нуклеотидных последовательностей или последовательностей аминокислот. Файлы в этом формате имеют расширение `fasta`. Файл в формате *FASTA* представляет собой текстовый файл, состоящих из нескольких блоков [3].

Первая строка блока начинается с символа > и в ней находится произвольное описание блока данных

Во второй и последующих строках находятся данные о нуклеотидах, при этом используются обозначения приведенные в табл. 3. При записи обозначений могут использоваться как прописные так и строчные буквы. Любые цифры должны быть интерпретированы как *N* (неизвестный нуклеотид).

Таблица 3. Обозначения возможных нуклеотидов в формате *FASTA*

Обозначение	Расшифровка
A	Аденин
C	Цитозин
G	Гуанин
T	Тимин
R	аденин или гуанин
Y	цитозин или тимин
K	гуанин или тимин
M	аденин или цитозин
S	гуанин или цитозин
W	аденин или тимин
B	гуанин или цитозин или тимин
D	аденин или гуанин или тимин
H	аденин или цитозин или тимин
V	аденин или гуанин или цитозин
N	аденин или гуанин или цитозин или тимин
–	неизвестный промежуток произвольной длины

1.3.4. Формат *Standard Flowgram*

Формат *Standard Flowgram* используется для хранения нуклеотидных последовательностей их информации об их качестве. Файлы в этом формате имеют расширение `sff`. Формат разработан в компании *454 Life Sciences* и институтах *Whitehead Institute for Biomedical Research* и *Sanger Institute* [5]. Файл в этом формате состоит из трех секций.

Заголовок файла, встречается один раз в начале файла, содержит информацию о числе чтений в файле, версию файла, и другую служебную информацию.

Заголовка чтения, присутствует перед каждым чтением, содержит информацию о длине чтения, и использованных фрагментах из которых будет состоять описание чтения.

Описание чтения представляет собой последовательность фрагментов, введенных в заголовке чтения.

Этот формат не является человеко-читаемым. Для работы с ним необходимо использование специальных библиотек.

1.3.5. Формат *SAM*

Формат *SAM* используется для описаний чтений их выравниваний [11]. Файлы в этом формате имеют расширение `sam`. Файл в этом формате состоит из двух частей: заголовка и описаний выравниваний чтений

Заголовок состоит из нескольких строк, начинающихся с символа @ и содержащих произвольную информацию о файле.

Каждое описание выравнивания записывается в отдельной строке и состоит из нескольких полей разделенных символом табуляции. В этих полях хранится информация о нуклеотидах, их качестве, типе чтения и выравнивании чтения на геном. Выравнивания чтений могут быть упорядочены по позиции в исходном геноме.

1.3.6. Формат *BAM*

Формат *BAM* аналогичен формату SAM. Файл в формате *BAM* представляет собой файл формата *SAM* сжатый при помощи библиотеки *BGZF* аналогичной формату *ZIP* [13]. Отличие этого формата от простого сжатия формата *SAM* является быстрый доступ к произвольной позиции сжимаемого файла. Этот формат не является человеко-читаемым. Для работы с ним необходимо использование специальных библиотек.

1.3.7. Сравнение форматов

В табл. 4 приведено сравнение описанных форматов.

Таблица 4. Сравнение форматов представления геномных данных.

Формат	Хранение информации о качестве	Человеко-читаемость	Наличие дополнительной информации	Эффективность хранения
FASTQ	Да	Да	Нет	Нет
FASTA	Возможно хранение в отдельном файле	Да	Нет	Нет
Standard Flowgram Format	Да	Нет	Да	Нет
SAM	Да	Да	Да	Нет
BAM	Да	Нет	Да	Да

1.4. Выводы по главе 1

1. Алгоритмы исправления ошибок в наборе чтений геномной последовательности испоьзуют в качестве базового принципа работы граф де Брюина или частотный анализ встречаемости *k*-меров.
2. Алгоритмы сборки контигов (протяженных непрерывных фрагментов геномной последовательности), как правило, основаны на использовании графа де Брюина.
3. Одним из недостатков, которым обладают перечисленные программные средства, является большой объем оперативной памяти, необходимый им для сборки генома, сходного по размерам с геномом человека (2-3 миллиарда нуклеотидов). Так, например, *SOAPdenovo* необходимо порядка 140 гигабайт оперативной памяти, а *ABySS* – 21 компьютер с 16 гигабайтами каждый (всего – 336 гигабайт). Такие затраты памяти обусловлены наличием ошибок секвенирования в исходных данных (такие ошибки ведут к увеличению размера графа де Брюина), а также неоптимальным методом хранения этого графа.
4. Еще одним недостатком существующих методов сборки является отсутствие внутреннего контроля качества сборки.

2. ВЫБОР И ОБОСНОВАНИЕ ОПТИМАЛЬНОГО ВАРИАНТА ПРОВЕДЕНИЯ ИССЛЕДОВАНИЙ И ВЫБОР АРХИТЕКТУРЫ МЕТОДА СБОРКИ ГЕНОМНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

2.1. Выбор и обоснование оптимального варианта проведения исследований

В настоящем разделе представлено обоснование выбранного варианта проведения исследований.

Одним из недостатков, которым обладают перечисленные программные средства, является **большой объем оперативной памяти**, необходимый им для сборки генома, сходного по размерам с геномом человека (203 миллиарда нуклеотидов). Так, например, *SOAPdenovo* необходимо порядка 140 гигабайт оперативной памяти, а *ABySS* – 21 компьютер с 16 гигабайтами каждый (всего – 336 гигабайт). Такие затраты памяти обусловлены наличием ошибок секвенирования в исходных данных (такие ошибки ведут к увеличению размера графа де Брюина), а также неоптимальным методом хранения этого графа. Другим недостатком существующих методов сборки является отсутствие внутреннего контроля качества сборки.

В рамках настоящего исследования будет разработан метод сборки генома, который будет лишен указанных недостатков. Предварительные исследования показывают, что для сборки генома, сходного по размерам с геномом человека, будет достаточно 40-50 гигабайт оперативной памяти (в три – восемь раз меньше, чем у существующих методов), а внутренний контроль показывает высокое качество восстановления фрагментов геномной последовательности.

При проведении теоретических исследований в рамках настоящего Государственного контракта планируется разработать два алгоритма:

- алгоритм исправления ошибок в наборе чтений геномной последовательности;
- алгоритм восстановления фрагментов геномной последовательности.

При проведении экспериментальных исследований разработанный метод будет сравниваться с аналогами на различных примерах исходных данных. При этом будут рассматриваться геномы различных размеров – от нескольких миллионов до нескольких миллиардов нуклеотидов. На основании результатов экспериментов разработанный метод сборки геномных последовательностей может быть доработан, после чего будет реализован прототип инструментального средства.

2.2. Архитектура метода сборки геномных последовательностей

В настоящем разделе описывается архитекутра разрабатываемого метода сборки геномных последовательностей.

Сборка генома в предлагаемом методе будет осуществляться в четыре этапа (рис. 10):

- исправление ошибок в наборе чтений геномной последовательности;
- восстановление фрагментов геномной последовательности по чтениям;
- сборка контигов из восстановленных фрагментов;
- сборка скэффолдов из контигов.

При этом для первых двух этапов будут разработаны в рамках настоящей НИР алгоритмы и выполнена их программная реализация, а для третьего и четвертого этапов – будт использоваться сторнние средства, выбор которых будет осуществлен при планировании экспериментальных исследований.

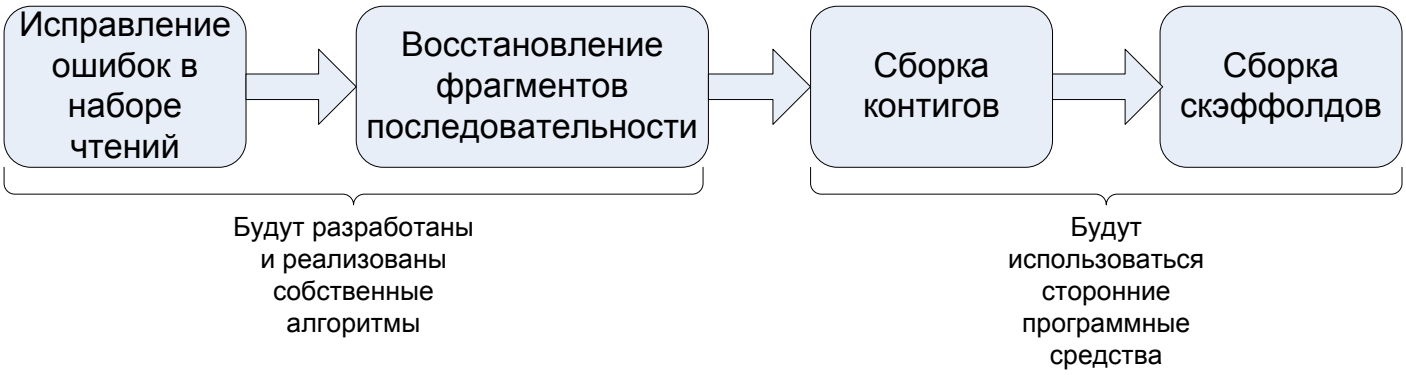

Рис. 10. Архитектура метода сборки геномных последовательностей

Для алгоритмов будут создаваться теоретические описания, затем будет выполняться их программная реализация на языке программирования высокого уровня, а далее будет выполняться ряд вычислительных экспериментов с разработанной программой.

При разработке алгоритма исправления ошибок в чтениях геномной последовательности будут использоваться методы частотного анализа и методы хеширования. Эти методы показали свою эффективность в процессе предварительных исследований.

При разработке алгоритма восстановления фрагментов геномной последовательности будут применяться методы обработки битовых строк, методы поиска путей в графах, методы обхода графов и методы перебора. Применение указанных методов обосновывается их высокой эффективностью, которую доказывают результаты НИР, проводящихся в ведущих научных центрах мира, а также результаты предварительных исследований, проведенных в СПбГУ ИТМО.

Выводы по главе 2

1. Выбрано и обосновано направление проведения исследований.
2. Описана архитектура разрабатываемого метода сборки геномных последовательностей.

3. ПОДГОТОВКА ПЛАНА ПРОВЕДЕНИЯ ТЕОРЕТИЧЕСКИХ И ЭКСПЕРИМЕНТАЛЬНЫХ ИССЛЕДОВАНИЙ

В настоящем разделе представлен план проведения теоретических и экспериментальных исследований.

3.1. План проведения первого этапа теоретических и экспериментальных исследований

На первом этапе (работы проводятся с даты подписания Государственного контракта по 12.07.2011 г.) проведения исследований планируется провести следующие работы:

- патентные исследования;
- разработка архитектуры метода сборки геномных последовательностей.

Результаты работ первого этапа:

- научно-технический отчет;
- отчет о патентных исследованиях.

3.2. План проведения второго этапа теоретических и экспериментальных исследований

На втором этапе (работы проводятся с 13.07.2011 г. по 18.11.2011 г.) проведения исследований планируется провести следующие работы:

- разработка алгоритма исправления ошибок в данных секвенирования;
- программная реализация алгоритма исправления ошибок в данных секвенирования.

Результаты работ второго этапа:

- научно-технический отчет.

3.3. План проведения третьего этапа теоретических и экспериментальных исследований

На третьем этапе (работы проводятся с 01.01.2012 г. по 12.07.2012 г.) проведения исследований планируется провести следующие работы:

- разработка алгоритма восстановления фрагментов геномной последовательности по парным чтениям;
- программная реализация алгоритма восстановления фрагментов геномной последовательности по парным чтениям.

Результаты работ третьего этапа:

- научно-технический отчет.

3.4. План проведения четвертого этапа теоретических и экспериментальных исследований

На четвертом этапе (работы проводятся с 13.07.2012 г. по 19.11.2012 г.) проведения исследований планируется провести следующие работы:

- составление плана проведения экспериментальных исследований;
- подготовка заявки на регистрацию программы для ЭВМ;
- проведение вычислительных экспериментов;
- анализ результатов вычислительных экспериментов;
- подготовка статьи для публикации в журнале из перечня ВАК.

Результаты работ четвертого этапа:

- научно-технический отчет;

- копия заявки на регистрацию программы для ЭВМ;
- копия статьи для публикации в журнале из перечня ВАК;
- протоколы экспериментов.

3.5. План проведения пятого этапа теоретических и экспериментальных исследований

На пятом этапе (работы проводятся с 01.01.2013 г. по 12.07.2013 г.) проведения исследований планируется провести следующие работы:

- корректировка разработанных алгоритмов с учетом результатов проведенных экспериментов;
- программная реализация прототипа инструментального средства, реализующего разработанный метод сборки геномных последовательностей;
- подача заявки на регистрацию программы для ЭВМ;
- подготовка статьи для публикации в журнале из перечня ВАК.

Результаты работ пятого этапа:

- научно-технический отчет;
- копия заявки на регистрацию программы для ЭВМ;
- копия статьи для публикации в журнале из перечня ВАК.

3.6. План проведения шестого этапа теоретических и экспериментальных исследований

На шестом этапе (работы проводятся с 13.07.2013 г. по 19.11.2013 г.) проведения исследований планируется провести следующие работы:

- разработка рекомендаций по возможности использования результатов проведенной поисковой научно–исследовательской работы в реальном секторе экономики;
- разработка рекомендаций по использованию результатов поисковой научно–исследовательской работы при разработке научно–образовательных курсов;
- разработка научно–образовательных курсов на электронных носителях по новейшим направлениям науки и технологий;
- разработка научно–популярных материалов для школьников и школьных учителей с участием руководителей поисковых научно–исследовательских работ.

Результаты работ шестом этапа:

- итоговый научно-технический отчет.

Выводы по главе 3

1. Разработан план проведения теоретических и экспериментальных исследований.
2. На первом этапе теоретических исследований разработана архитектура метода сборки геномных последовательностей.

4. ПРОВЕДЕНИЕ ПАТЕНТНЫХ ИССЛЕДОВАНИЙ

В данном разделе описываются результаты проведения патентных исследований.

4.1. Перечень сокращений, условных обозначений, символов, единиц, терминов

НИР – научно-исследовательская работа.
ПО – Программное обеспечение.
DNA – Deoxyribonucleic acid, дезоксирибонуклеиновая кислота.
ДНК – дезоксирибонуклеиновая кислота.
РНК – рибонуклеиновая кислота.
ЭВМ – электронно-вычислительная машина.
ПЦР – полимеразная цепная реакция.
UML – Unified Modeling Language.

4.2. Общие данные об объекте исследований

Патентный поиск проводился с целью определения патентоспособности планируемых результатов научно-исследовательской работы **по лоту «Проведение научных исследований научными группами под руководством докторов наук в следующих областях:**

– биокаталитические, биосинтетические и биосенсорные технологии;

– биомедицинские и ветеринарные технологии жизнеобеспечения и защиты человека и животных;

– геномные и постгеномные технологии создания лекарственных средств;

– клеточные технологии;

– биоинженерия;

– биоинформационные технологии» – тема «Разработка метода сборки геномных последовательностей на основе восстановления фрагментов по парным чтениям» (шифр «2011-1.2.1-201-007-050»), выполняемой в рамках Федеральной целевой программы «Научные и научно-педагогические кадры инновационной России» на 2009–2013 годы **по государственному контракту № 16.740.11.0495**, заключенному между Министерством образования и науки Российской Федерации и Государственным образовательным учреждением высшего профессионального образования «Санкт-Петербургский государственный университет информационных технологий, механики и оптики» на основании решения Конкурсной комиссии Министерства образования и науки Российской Федерации (протокол от 26.04.2011 г. № 3/0173100003711000032), а также для получения сведений об охранных и иных документах, которые могут препятствовать применению результатов данной НИР в Российской Федерации, и условиях использования таких документов.

В соответствии с задачей исследования, разрабатываемый метод сборки геномных последовательностей должен удовлетворять следующим требованиям:

- работа с выходными данными секвенаторов второго поколения [14];
- сравнительно низкие требования к используемым ресурсам компьютера, таким как оперативная память.

Патентный поиск проводился в соответствии с ГОСТ Р. 15.011-96 «Система разработки и постановки продукции на производство. Патентные исследования» [1]. Проверка патентоспособности проводимой научно-исследовательской работы осуществлялась на основе поиска патентных и других открытых документов, описывающих решения, максимально полно удовлетворяющие задаче исследования. Поиск патентной информации проводился в Бюллетене Федеральной службы по интеллектуальной собственности, патентам и товарным знакам Российской Федерации «Программы для ЭВМ, базы данных, топологии интегральных микросхем» (Роспатент, www.fips.ru), Бюро по

патентам и товарным знакам США (USPTO, www.uspto.gov) и Европейского патентного бюро (EPO, ep.espacenet.com).

Патентный поиск проводился с 06 июня 2011 г. по 09 июня 2011 г.

4.3. Исследования патентов

Тема исследования ограничивается сборкой геномных последовательностей на основе восстановления фрагментов по парным чтениям. Однако, для повышения полноты результатов поиска, рассматриваемая тематическая область была расширена на следующие направления:

- сборка геномных последовательностей;
- секвенирование нуклеотидных последовательностей.

В случаях, когда в рамках некоторого направления была возможность выделить блоки меньшего объема, блоки, заведомо не соответствующие направлению исследования, из рассмотрения исключались. Таким образом, были выбраны элементы патентных классификаций для ограничения области поиска.

В рамках патентного поиска производился анализ патентов, принадлежащих следующим категориям международной патентной классификации (рассматривались патенты, принадлежащие самому низкому указанному уровню):

Категория	Содержание категории (рус.)	Содержание категории (англ.)
G	Физика	Physics
G06	Вычисление; счет	Computing; calculating; counting
G06F	Обработка цифровых данных с помощью электрических устройств	Electrical digital data processing
G06F 17/00	Устройства или методы цифровых вычислений или обработки данных, специально предназначенные для специфических функций	Digital computing or data processing equipment or methods, specially adapted for specific functions
G06F 17/30	Информационный поиск; структуры баз данных для этой цели.	Information retrieval; Database structures thereof
G06F 19/00	Устройства или методы цифровых вычислений или обработки данных, специально предназначенные для специфических приложений	Digital computing or data processing equipment or methods, specially adapted for specific applications

Кроме того, рассматривались следующие категории патентной классификации США (перевод на русский язык исполнителя отчета):

Категория	Содержание категории (рус.)	Содержание категории (англ.)
702	Обработка данных: измерение, калибровка, тестирование	Data Processing: Measuring, Calibrating, or Testing
702/1	Системы измерений в специфических окружениях	Measurement System in a Specific Environment
702/19	Биология и биохимия	Biological or Biochemical
702/20	Определение последовательности генов	Gene Sequence Determination
703	Обработка данных: проектирование структуры, моделирование,	Data Processing: Structural Design, Modeling, Simulation, and Emulation

	симуляция и эмуляция	
703/6	Симуляция неэлектрического устройства или системы	Simulation Nonelectrical Device or System
703/11	Биология и биохимия	Biological or Biochemical

Также производился поиск без ограничения разделов классификаций по следующим ключевым словам: *DNA, genome, assembly, sequencing*.

Так как патенты, полностью удовлетворяющие поставленным перед исследованием требованиям, найдены не были, в качестве результата поиска рассмотрим наиболее близкие к теме исследования патенты:

1. Патент US 7939264 от 10.05.2011 «DNA sequencing method».

В патенте описывается новый способ секвенирования генома (стадия – получение ридов из непосредственно молекул ДНК) с помощью специфических энзимов. Утверждается, что скорость метода доходит до сотен оснований в секунду, а длина производимых им ридов достигает тысяч оснований. Данный патент затрагивает иную стадию процесса сборки генома, чем исследования, проводимые по контракту.

2. Патент US 7939256 от 10.05.2011 «Composition and method for nucleic acid sequencing».

Как и в предыдущем патенте, описывается способ получения информации непосредственно из молекул ДНК. Этот способ основан на контактировании обездвиженных цепочек ДНК с маркированными трифосфатами различных нуклеотидов. Как и предыдущий, данный патент затрагивает иную стадию процесса сборки генома, чем исследования, проводимые по контракту.

3. Патент US 7890268 от 15.02.2011 «De-novo sequencing of nucleic acids».

В данном патенте описан метод применения многостадийной масс-спектрометрии, ранее использовавшейся только для секвенирования по образцу, для секвенирования генома с нуля (de novo). Как и предыдущие, данный патент затрагивает иную стадию процесса сборки генома, чем исследования, проводимые по контракту.

4. Патент US 7881873 от 01.02.2011 «Systems and methods for statistical genomic DNA based analysis and evaluation».

В данном патенте описываются методы статистического анализа ДНК в целях быстрого обнаружения аномалий, значимых при клинических исследованиях. Данный патент затрагивает область изучения генома, не пересекающуюся с исследованиями по контракту.

5. Патент US 7835871 от 16.11.2010 «Nucleic acid sequencing system and method».

В этом патенте описывается секвенатор компании *Illumina Inc.*, данные, производимые которым (короткие чтения), используются методом, исследуемым в рамках работ по контракту, для получения полного генома.

6. Патент US 7767400 от 03.08.2010 «Paired-end reads in sequencing by synthesis».

В этом патенте описываются методы генерации коротких парных чтений. Данные, производимые этими методом, используются методом, исследуемым в рамках работ по контракту, для получения полного генома.

7. Патент US 7720613 от 18.05.2010 «Method for sequencing nucleic acids».

В этом патенте описывается способ секвенирования малых отрезков ДНК (до 15 оснований) путем сравнения спектра секвенируемого отрезка со спектром предполагаемого значения, модификацией не менее одного основания предполагаемого значения и повторения описанного процесса до совпадения секвенируемого и предполагаемого отрезка. Метод, описанный в данном патенте, невозможно применить для секвенирования полного генома в силу его неэффективности. Поэтому данный патент не может повлиять на патентоспособность результатов, планируемых к получению в рамках контракта.

8. Патент US 7144699 от 05.12.2006 «Iterative resequencing».

В этом патенте описывается способ итеративного приближения для секвенирования последовательности нуклеотидов с использованием уже секвенированной и близкой к ней базовой последовательности. В отличие от данного патента, результаты исследований, проводимых по контракту, позволят секвенировать геном с нуля (de novo).

9. Патент US 6223128 от 24.04.2001 «DNA sequence assembly system».

В патенте описан метод секвенирования молекулы ДНК с помощью секвенаторов первого поколения, использующих длинные чтения. На настоящий момент секвенаторы первого поколения рассматриваются как медленные и дорогие – производительность секвенаторов второго поколения на несколько порядков выше. Исследования, проводимые в рамках работ по контракту, используют выходные данные секвенаторов второго порядка.

Сводная информация по основным различиям рассмотренных патентов и задач проверяемого на патентоспособность исследования приведена в табл. 5.

Из изложенного следует, что поиск не выявил наличия патентов, препятствующих проведению исследований по теме «Разработка метода сборки геномных последовательностей на основе восстановления фрагментов по парным чтениям».

Таблица 5. Отличия описываемых в патентах задач от темы исследования

Патент	Чтение непосредственно ДНК	Секвенирование по образцу	Статистический анализ	Секвенаторы первого порядка
US 7939264	Да			
US 7939256	Да			
US 7890268	Да			
US 7881873			Да	
US 7835871	Да			
US 7767400	Да			
US 7720613		Да		
US 7144699		Да		
US 6223128				Да

4.4. Исследования программ для ЭВМ

Согласно Приложению 2 к Государственному контракту, в ходе выполнения работ планируется регистрация программ для ЭВМ. В связи с этим, были рассмотрены данные о регистрации программ для ЭВМ из выпусков Электронного бюллетеня «Программы для ЭВМ, базы данных, топологии интегральных микросхем» за 2008-2011 годы. В табл. 6 приведена сводная информация по рассмотренным данным.

Таблица 6. Результаты рассмотрения данных о регистрации программ для ЭВМ

Рег. номер	Название программы	Краткое описание программы	Комментарий
2008614430	ПЦР-Дизайн	Назначение программы – выбор последовательностей праймеров и проб для измерения уровня мРНК и копийности генов с помощью полимеразной цепной реакции в реальном времени (ПЦР-РВ).	Выполняет иную задачу (планирование химических опытов, а не сборку генома).
2008612975	BioUML Enterprise Edition	Программа предназначена для интеграции данных, реконструкции, моделирования и анализа сложных биологических систем.	Выполняет роль интерфейса к базе данных. Может использоваться для хранения и отображения отдельных результатов, полученных с помощью разрабатываемой в рамках контракта технологии.
2008612150	Программа для анализа и статистической обработки нерекомбинирующих ДНК (мтФил)	Предназначена для проведения автоматического анализа нерекомбинирующих последовательностей ДНК.	При работе использует уже собранные оследовательности ДНК. Кроме того, скорее всего, умеет работать только с небольшими по объему данными митохондриальными ДНК).
2009615258	Программа, разрабатывающая праймеры для создания химерных ДНК конструкций методом ПЦР (PCRfusion)	Программа предназначена для автоматизированной разработки праймеров для объединения двух и более молекул ДНК в одну молекулу в процессе полимеразной цепной реакции (ПЦР).	Выполняет иную задачу (синтез молекул ДНК, а не их чтение и интерпретацию).
2009610058	Программа для расчета индекса и вероятности отцовства (материнства) при исследованиях ДНК	Программа предназначена для расчета индекса и вероятности отцовства (материнства) при ДНК-исследованиях.	При работе использует небольшие фрагменты (локусы) уже собранных последовательностей ДНК.
2009610743	Сравнение хромосом и полных геномов (GigaGene)	Программа предназначена для сравнения пар геномов или хромосом. Результаты сравнения выводятся на экран в графическом виде.	При работе использует уже собранные последовательности ДНК. Скорее всего, может работать только с небольшими по объему данными.
2009610744	Поиск нуклеотидных последовательностей в геноме человека (*MegaGene*)	Программа предназначена для поиска заданных нуклеотидных последовательностей в геноме человека. Результаты поиска	При работе использует уже собранные последовательности ДНК. Скорее всего, может работать

		всех вхождений выводятся на экран в графическом виде.	только с небольшими по объему данными.
2009610745	Графическое построение плотности распределения генетических элементов в геноме человека (*MegaDNA*)	Программа предназначениа для статистической обработки данных о генетических элементах в геноме человека и графического представления результатов обработки в виде плотности их распределения.	При работе использует уже собранные последовательности ДНК. Скорее всего, может работать только с небольшими по объему данными.
2010616107	Автоматизированный сбор данных о полиморфизмах индивидуальных геномов из *dbSNP*	Программа позволяет эксперту проводить автоматизированное извлечение информации о полиморфизмах индивидуальных геномов, представленной в базе данных *NCBI dbSNP*.	При работе использует уже собранные последовательности ДНК.
2010611465	Аналитическая система расчетов итоговых значений однонуклеотидных полиморфизмов	Программа представляет собой информационную систему по расчету рисков появления у человека определенных заболеваний, выявлению физических характеристик на основании данных ДНК-анализа человека и обработанных результатов исследований.	При работе использует уже собранные последовательности ДНК.
2009615516	Универсальная программа математического анализа структуры ДНК	Программа предназначена дял проведения анализа содержащейся в ДНК информации с использованием широкого круга современных физико-математических методов анализа.	При работе использует уже собранные последовательности ДНК.
2009615517	Универсальная программа для обработки и создания базы данных по типированию штаммов чумного микроба методом мультилокусного секвенирования	Программа предназначениа для обработки результатов мультилокусного секвенирования штаммов чумного микроба и создания на их основе базы данных.	При работе использует уже собранные фрагменты последовательностей ДНК.
2010616973	РегистрЛабПЦР	Программа предназначена для автоматизации накопления и обработки информации о пациентах и образцах материала, а также результатов исследований, при	Использует данные, полученные от детектора, однако не преобразует их в последовательность ДНК.

		использовании различных методик ДНК-диагностики в ПЦР лаборатории, формирования базы данных.	

Из изложенного следует, что поиск не выявил наличия программ для ЭВМ, реализующих требуемую функциональность и тем самым препятствующих проведению исследований по теме «Разработка метода сборки геномных последовательностей на основе восстановления фрагментов по парным чтениям».

4.5. Исследования непатентных источников

Согласно Патентному закону Российской Федерации [2], уровень техники, в сравнении с которым выявляется новизна изобретения, определяется не только зарегистрированными в России патентами, но также имеющейся во всем мире общедоступной информацией. В связи с этим был проведен анализ публикаций, содержащих упоминания построения конечных автоматов или подобных им моделей с применением алгоритмов решения задачи о выполнимости булевой формулы.

4.5.1. Общий обзор публикаций

Исследования в области секвенирования и сборки геномов различных видов живых существ ведутся в мире с 70–80-х годов XX века. Одним из наиболее значительных проектов в этой области «Геном человека» (Human Genome Project) был запущен в 1990 году и завершен в 2003 году получением геномной последовательности человека. В рамках этого проекта для секвенирования использовались так называемые технологии первого поколения [9].

В середине первого десятилетия XXI века широкое распространение получили технологии секвенирования второго поколения, разработанные, например, компанией *Illumina*. Эти технологии позволяют существенно быстрее и дешевле получать на порядок большие объемы данных о геномной последовательности. Их недостатком является то, что длины чтений при использовании составляют 80–100 нуклеотидов, а не 500–1000, как при использовании технологий второго поколения [14] Технологии секвенирования второго поколения активно развиваются, поэтому требуется постоянная разработка новых алгоритмов сборки генома по данным секвенирования и совершенствование существующих.

Одной из наиболее часто используемых при сборке генома математических моделей является так называемый граф де Брюина. На его использовании основаны следующие программные средства (со ссылками на описывающие их статьи): *Velvet* [12], *ALLPATHS* [1], *AbySS* [10], *SOAPdenovo* [4], *EULER* [8].

4.5.2. Обзор избранных работ

Далее будут подробно рассмотрены работы, наиболее близко соответствующие направлению проводимого по контракту исследования. Отличия их от проводимого исследования описаны в разд. 4.6 отчета.

4.5.2.1. ABySS

В статье [10] изложен подход к сборке контигов в программном средстве *ABySS*. Этот подход состоит из двух этапов:

- сборка контигов без учета парной информации;
- разрешение неоднозначностей с помощью парной информации и наращивание контигов.

В основе всего подхода лежит распределенный граф де Брюина [6].

Для того, чтобы собрать первоначальные версии контигов, происходит объединение последовательностей смежных однозначных ребер. Ребро называется однозначным, если исходящая степень его начальной вершины и входящая степень конечной вершины равны единице.

На втором этапе между контигами устанавливается связи, используя парную информацию. Пара чтений называется связывающей два контига, если первое чтение картируется на первый контиг, а второе – на второй. Между двумя контигами устанавливается связь, если число связывающих их чтений больше некоторой константы p (по умолчанию используется $p = 5$). Для каждого контига C_i строится множество связанных с ним контигов P_i. Затем в графе связей контигов ищется уникальный путь, проходящий через все контиги из P_i. В качестве ограничений при поиске выступают оценка на расстояния между контигами на основе принципа максимального правдоподобия и эвристическая оценка на число посещенных вершин. После того, как поиск таких путей для каждого контига завершился (успешно или нет), согласующиеся пути сливаются, образуя конечные контиги.

4.5.2.2. ALLPATHS

Алгоритм, лежащий в основе сборщика *ALLPATHS* изложен в статье [1]. Он также основан на графе де Брюина и состоит из нескольких этапов.

Первым этапом является создание начального приближения контигов. Для этого, как и в [10], рассматриваются последовательности смежных однозначных ребер, которые и объявляются приближением контигов.

Вторым шагом является выбор контигов, вокруг которых будет происходить локализация – попытка выделить те чтения, которые картируются на реальный геном близко к положению контига. В качестве таких контигов лучше всего брать длинные контиги, уникальны в геноме. Для проверки уникальности можно использовать ожидаемое и реальное покрытие чтениями.

Затем для каждого выбранного контига – центра локализации, рассматривается множество последовательностей – чтений и контигов, лежащих в пределах, например, 10 килобаз от краев этого контига, – окрестность контига. Если построить приближение этого множества, то можно будет собрать небольшой участок генома, практически только из тех чтений, которые в него картируются. Это позволяет существенно уменьшить объем данных. Для начала выделяется набор почти уникальных в геноме первоначальных контигов, лежащих в рассматриваемом промежутке. Это происходит с помощью итеративного добавления новых контигов, связанных с уже найденными (рис. 5). При этом для каждого контига можно узнать его смещение относительно центра (с какой-то погрешностью), что позволяет отбросить далеко отстоящие контиги. Затем строятся два множества чтений: первичное, состоящее из чтений картирующихся в рассматриваемую область генома, но, возможно, не всех таких чтений, и вторичное, которое, кроме интересующих чтений, возможно, содержит и некоторые другие. В первичное множество входят все пары чтений, хотя бы одно из которых картируется на какой-либо из выбранных контигов. Во вторичное множество чтений входят те пары чтений, которые могут быть собраны из чтений из первичного множества. Затем перекрывающиеся пары чтений из вторичного множества сливаются, образуя пары более длинных последовательностей.

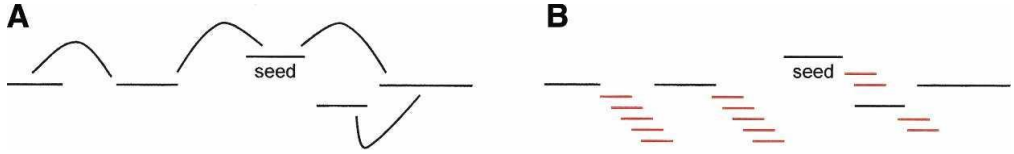

Рис. 11. Построение окрестности контига

Для каждой окрестности контига в полученных на предыдущем этапе парах последовательностей ищутся все пути, соединяющие первую последовательность со второй, которые состоят из последовательностей из того же множества. Путем является последовательность последовательностей, перекрывающихся с предыдущей не менее чем на *K* нуклеотидов.

На следующем этапе происходит склеивание путей, найденных на предыдущем этапе, если они имеют длинный общий подпуть (рис. 6). Таким образом, осуществляется локальная сборка генома. Затем результаты локальных сборок сливаются между собой, образуя конечный вариант сборки генома.

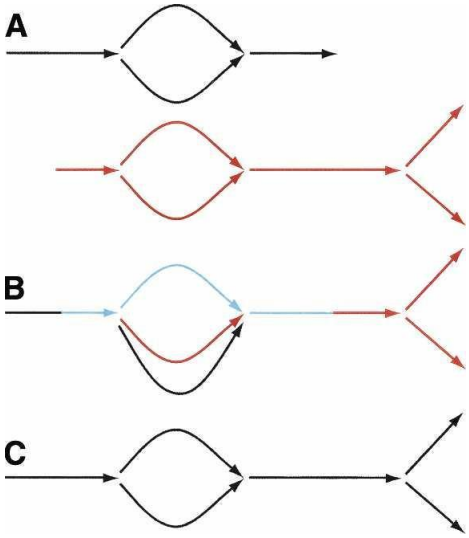

Рис. 12. Слияние путей

4.5.2.3. Velvet

Алгоритм работы *Velvet* изложен в статье [12].

После исправления ошибок остается граф, вершинами которого являются последовательности, соответствующие вершинам графа де Брюина, объединенным по однозначным ребрам. С этого момента начинает работу модуль *Breadcrumb*, ответственный за разрешение проблемных участков с помощью парной информации.

Зная распределение длин пропусков в парных чтениях можно выделить длину *L* такую, что число парных чтений, имеющих пропуск длиннее *L* ничтожно мало. Можно выделить «длинные» вершины – вершин, длины которых не меньше чем *L*. Затем на основе парных чтений выделяются парные «длинные» вершины, связанные несколькими парными чтениями (рис. 7). Если после «длинной» вершины может следовать несколько «длинных» вершин, то они называются неоднозначными. Все узлы, на которые картируется чтение, парное к которому картируется на однозначную «длинную» вершину, помечаются. После этого уникальные узлы последовательно наращиваются, добавлением помеченных узлов, связанных с текущим. Этот процесс продолжается до тех пор, пока не встречается узел, у которого нет продолжения или есть несколько вариантов продолжения. Если несколько «длинных» вершин последовательно соединились такими путями, то они объединяются в контиг.

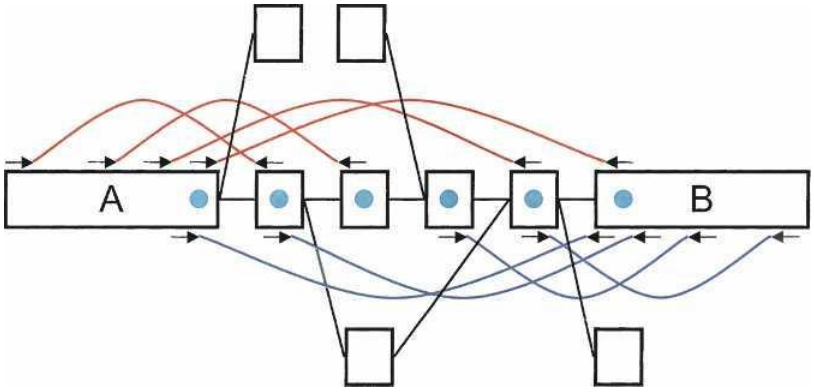

Рис. 13. Процесс работы алгоритма *Breadcrumb*

4.5.2.4. SOAPdenovo

Способ сборки, используемый в *SOAPdenovo*, описан в статье [4].

Как и многие другие сборщики из коротких чтений *SOAPdenovo* использует граф де Брюина. После исправления ошибок пути, не содержащие вершин с ветвлениями, объявляются первоначальным приближением контигов (рис. 8).

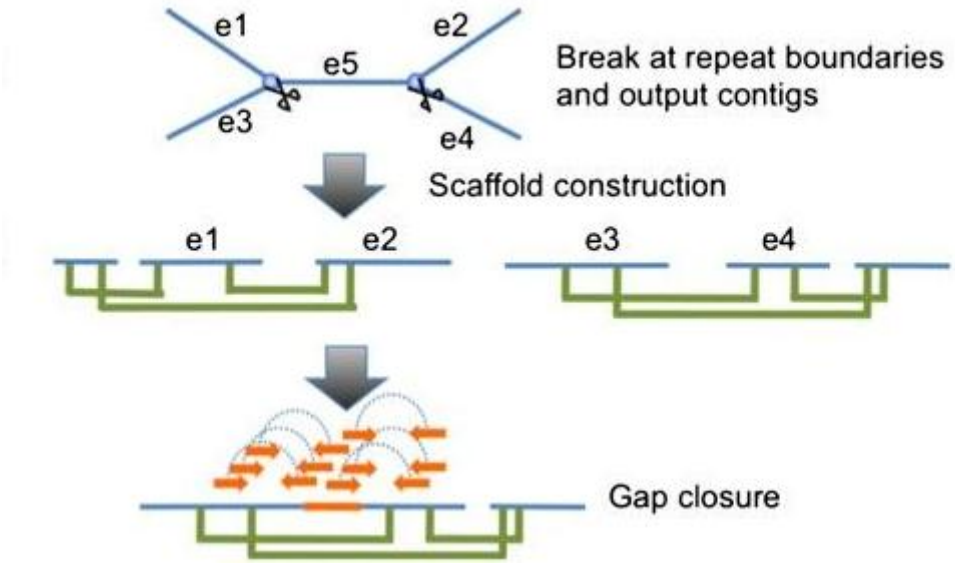

Рис. 14. Схема последних этапов работы *SOAPdenovo*

Затем производится картирование парных чтений на контиги. Между двумя контигами устанавливается порядок и оценивается расстояние между ними, если их соединяют как минимум три пары чтений. Далее наборы контигов с совместимыми связями линеаризуются, образуя скэффолды.

Во время следующего этапа происходит заполнение промежутков между соседними контигами в скэффолдах. Для этого для каждой пары соседних контигов выбирается множество парных чтений, из котрых одно картируется на один из контигов. Далее проводится локальная сборка только из этих чтений. Если она завершается успешно, тогда удается соединить два контига в один.

4.5.2.5. EULER

В статье [8] изложены основы работы набора алгоритмов *EULER*.

Сборщик *EULER* использует для сборки сведение к задаче о поиске в графе повторов Эйлерова суперпути – пути проходящем через все ребра и содержащем в качестве подпутей все пути из наперед заданного множества *P* [8]. Эта задача, в свою очередь, сводится к задаче поиска Эйлерова пути.

Изначально в качестве множества *P* использовались пути, соответствующие чтениям. Затем проводились преобразования, упрощающие граф. После этого рассматривалась каждые парные чтения (r_1, r_2) и искался путь, соединяющий r_1 и r_2 и имеющий длину около ожидаемой, определенной из априорного распределения длин. Если такой путь находился единственный, то он добавлялся в множество *P*. Этот процесс повторялся до тех пор, пока граф менялся.

В статье [8] предлагается способ учета и тех парных чтений, для которых нашлось несколько путей. Пусть для парных чтений $(read_{start}, read_{end})$ нашлось несколько путей (рис. 9). Обозначим за e_{start} и e_{end} ребра, в которых начинаются и заканчиваются $read_{start}$ и $read_{end}$ соответственно. Для каждого пути *p*, соединяющего $read_{start}$ и $read_{end}$, рассмотрим величину $support(e_{start}, e_{end}, p)$ – число парных чтений, поддерживающих этот путь – таких парных чтений (r_1, r_2), что одно из них картируется на ребро e_{start} или e_{end}, а другое на путь *p*. Из всех путей выбирается путь с максимальным значением *support*, если оно не меньше некоторого порогового значения *MinSupport.*

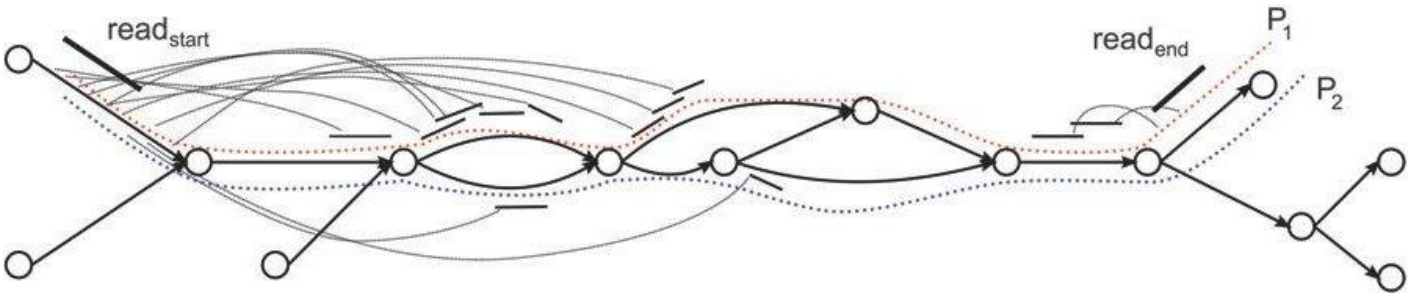

Рис. 15. Пути между $read_{start}$ и $read_{end}$

4.5.3. Результаты непатентных исследований

Исследование публикаций по сборке генома показало, что работы по этой тематике в мире проводятся. Работы по контракту являются дальнейшим совершенствованием некоторых из имеющихся работ, направленным на ускорение работы и снижение требований к ресурсам, таким как используемая память. Таким образом, препятствий к выполнению работ по контракту при рассмотрении публикаций не выявлено.

4.6. Отличия проводимого исследования

Проверяемое на патентоспособность исследование ориентировано на сборку геномных последовательностей на основе восстановления фрагментов по парным чтениям, при этом сами чтения предварительно выполняются с помощью таких секвенаторов, как секвенаторы компании *Illumina*. Из рассмотренных патентов, в большинстве осуществляется секвенирование самостоятельно либо с нуля (US 7835871, US 7767400, US 6223128, US 7939264, US 7939256, US 7890268), либо на основе образца (US 7720613, US 7144699). В патенте US 7881873 выполняется статистический анализ, и, следовательно, он неприменим в задаче секвенирования генома.

Подавляющее большинство зарегистрированных программ для ЭВМ, в той или иной мере работающих с генетическим материалом, используют ранее собранные последовательности ДНК (2008612150, 2009610058, 2009610743, 2009610744, 2009610745, 2010616107, 2010611465, 2009615516, 2009615517). Остальные программы либо работают с оптическими данными чтений ДНК (2010616973), либо используются для хранения и отображения данных (2008612975), планирования химических экспериментов (2008614430) или контроля синтеза ДНК (2009615258).

Одним из недостатков, которым обладают программные средства, описанные в статьях [1, 4, 8, 12], является большой объем оперативной памяти, необходимый им для сборки генома, сходного по размерам с геномом человека (203 миллиарда нуклеотидов). Так, например, *SOAPdenovo* [4] необходимо порядка 140 гигабайт оперативной памяти, а *ABySS* [10] – 21 компьютер с 16 гигабайтами каждый (всего – 336 гигабайт). Такие затраты памяти обусловлены наличием ошибок секвенирования в исходных данных (такие ошибки ведут к увеличению размера графа де Брюина), а также неоптимальным методом хранения этого графа. Другим недостатком существующих методов сборки является отсутствие внутреннего контроля качества сборки.

В рамках настоящего исследования будет разработан метод сборки генома, который будет лишен указанных недостатков. Предварительные исследования показывают, что для сборки генома, сходного по размерам с геномом человека, будет достаточно 40–50 гигабайт оперативной памяти (в три – восемь раз меньше, чем у существующих методов), а внутренний контроль показывает высокое качество восстановления фрагментов геномной последовательности.

Таким образом, выполненый анализ патентных и непатентных источников, а также зарегистрированных программ для ЭВМ, показал, что рассматриваемая в работе по контракту задача в настоящее время не решена.

4.7. Выводы по главе 4

Данное патентное исследование, проведенное в рамках первого этапа работы по государственному контракту № 16.740.11.0495, показало, что в настоящее время отсутствуют патенты и иные охранные документы, которые могут препятствовать применению в Российской Федерации метода сборки геномных последовательностей на основе восстановления фрагментов по парным чтениям, разрабатываемой в рамках указанной работы. Таким образом, в соответствии с пунктом 5 Технического задания и пунктом 6.1 Детализированного предложения о качестве поисковых научно-исследовательских работ обеспечена патентная чистота результатов и отсутствие препятствий для их применения.

4.8. Литература и источники

1. ГОСТ Р. 15.011-96 «Система разработки и постановки продукции на производство. Патентные исследования».
2. *Патентный закон РФ.* http://www.legal-support.ru/information/laws/intellect/patent-law.html.
3. *Watson J.D., Crick F.H.* Molecular structure of nucletic acids; a structure for deoxyribose nucleic acid // Nature. Vol. 171. 1953, pp. 737–738.
4. *Sanger F., Nicklen S., Coulson A.* DNA sequencing with chain-terminating inhibitors // Proc. Natl. Acad. Sci. USA. Vol. 74. 1977, pp. 5463–5467.
5. *Staden R.* A strategy of DNA sequencing employing computer programs // Nucleic Acids Res. Vol. 6. 1979, pp. 2601–2610.
6. *Pevzner P.* 1-Tuple DNA sequencing: computer analysis // J. Biomol. Struct. Dyn. 1989. V. 7, pp. 63–73.
7. *Pevzner P.* Computational molecular biology: an algorithmic approach. MIT Press, 2000.
8. *Pevzner P., Tang H., Waterman M.* An Eulerian path approach to DNA fragment assembly / Proc. Natl. Acad. Sci. USA. Vol. 98. 2001, pp. 9748 – 9753.
9. *International Human Genome Sequencing Consortium.* Initial sequencing and analysis of the human genome // Nature. Vol. 409. 2001, pp. 860 – 921.
10. *Butler J., MacCallum I., Kleber M., Shlyakhter I., Belmonte M., Lander E., Nusbaum C., Jaffe D.* ALLPATHS: de novo assembly of whole-genome shotgun microreads // Genome Res. 2008. No. 5, pp. 810– 820.
11. *Zerbino D., Birney E.* Velvet: algorithms for de novo short read assembly using de Bruijn graphs // Genome Res. No. 5, Vol. 18. 2008, pp. 821–829.

12. *Simpson J., Wong K., Jackman S., Schein J., Jones S., Birol I.* ABySS: a parallel assembler for short read sequence data // Genome Res. 2009. No. 6, pp. 1117 – 1123.
13. *Li R., Zhu H., Ruan J., Qian W., Fang X., Shi Z., Li Y., Li S., Shan G., Kristiansen K., et al.* De novo assembly of human genomes with massively parallel short read sequencing // Genome Res. 2010, No. 6, pp. 265 – 272.
14. Schatz M., Delcher A., Salzberg S. Assembly of large genomes using second-generation sequencing // Genome Res. 2010. No. 6, pp. 295 – 315.

ЗАКЛЮЧЕНИЕ

В результате исследований на первом этапе работ по контракту были выполнены следующие работы:

- выполнен аналитический обзор;
- проведены патентные исследования;
- выбран и обоснован оптимальный вариант проведения исследований;
- выбрана архитектура метода сборки геномных последовательностей;
- подготовлен план проведения теоретических и экспериментальных исследований.

Аналитический обзор был выполнен по следующим направлениям:

- методы исправления ошибок в чтениях геномной последовательности;
- методы сборки генома;
- форматы представления геномных данных.

В результате выполнения аналитического обзора разработан план проведения теоретических и экспериментальных исследований. В соответствии с этим планом в рамках теоретических исследований будут разработаны следующие алгоритмы, составляющие метод сборки геномных последовательностей на основе восстановления фрагментов по парным чтениям:

- алгоритм исправления ошибок в наборе парных чтений;
- алгоритм восстановления фрагментов геномной последовательности по набору парных чтений.

Разработанные алгоритмы будут реализованы в виде программ на языке программирования *Java*. После этого будет проведено экспериментальное исследование. По результатам этих исследований будет проведена модификация разработанных методов. В заключение работы будет проведено обобщение и оценка результатов исследований.

СПИСОК ЛИТЕРАТУРЫ

1. *Butler J., MacCallum I., Kleber M., Shlyakhter I. A., Belmonte M. K., Lander E. S., Nusbaum C., Jaffe D. B.* Allpaths: de novo assembly of whole-genome shotgun microreads // Genome Res. 2008. Vol. 18. No. 5, pp. 810 – 820.
2. *Chaisson M. J., Brinza D., Pevzner P. A.* De novo fragment assembly with short mate-paired reads: Does the read length matter? // Genome Research. 2009. Vol. 19. No. 2,. pp. 336 – 346.
3. *FASTA format*. http://zhanglab.ccmb.med.umich.edu/FASTA/
4. *Li R., Zhu H., Ruan J., Qian W., Fang X., Shi Z., Li Y., Li S., Shan G., Kristiansen K., Li S., Yang H., Wang J., Wang J.* De novo assembly of human genomes with massively parallel short read sequencing // Genome Research. 2010. Vol. 20, No. 2? pp. 265 – 272.
5. NCBI: Formats: Documentation: Trace Archive v4.2: NCBI/NLM/NI. http://www.ncbi.nlm.nih.gov/Traces/trace.cgi?cmd=show&f=formats&m=doc&s=format
6. *Cock P., Fields C., Goto N., Heuer M., Rice P.* The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants // Nucleic Acids Research, 2010, Vol. 38. No. 6, pp. 1767 – 1777.
7. *Pevzner P. A., Tang H., Waterman M. S.* An eulerian path approach to DNA fragment assembly / Proc. Natl Acad Sci U S A. 2001. Vol. 98. No. 17, pp. 9748 – 9753.
8. *Phred -Quality Base Calling*. http://www.phrap.com/phred/
9. *Sundquist A., Ronaghi M., Tang H., Pevzner P., Batzoglou S.* Whole-Genome Sequencing and Assembly with High-Throughput, Short-Read Technologies // PLosS ONE. 2007.
10. *Simpson J. T., Wong K., Jackman S. D., Schein J. E., Jones S. J. M., Birol I.* Abyss: a parallel assembler for short read sequence data // Genome Res. 2009. Vol. 19, No. 6, pp. 1117 – 1123.
11. *The SAM Format Specication*. http://samtools.sourceforge.net/SAM1.pdf
12. *Zerbino D. R., Birney E.* Velvet: algorithms for de novo short read assembly using de Bruijn graphs // Genome Res. 2008. Vol. 18. No. 5, pp. 821 – 829.
13. *UCSC Genome Browser: BAM Track Format*. https://cgwb.nci.nih.gov/goldenPath/help/bam.html