Министерство образования и науки Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ»

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

«Алгоритмы локального анализа графа де-Брейна для метагеномных данных»

Автор: Васильев Артем Тарасович _____
Направление подготовки (специальность): 01.04.02 Прикладная математика и информатика
Квалификация: Магистр
Руководитель: Ульянцев В.И., канд. техн. наук, _____

К защите допустить
Зав. кафедрой Васильев В.Н., докт. техн. наук, проф. _____

«____» _____ 20_____ г.

Санкт-Петербург, 2016 г.

Студент Васильев А.Т. **Группа** М4239 **Кафедра** компьютерных технологий **Факультет** информационных технологий и программирования

Направленность (профиль), специализация Технологии проектирования и разработки программного обеспечения

Квалификационная работа выполнена с оценкой _____

Дата защиты «____» _____ 20_____ г.

Секретарь ГЭК _____

Листов хранения _____

Демонстрационных материалов/Чертежей хранения _____

Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ»

УТВЕРЖДАЮ
Зав. каф. компьютерных технологий
докт. техн. наук, проф.
_____ Васильев В.Н.
«____» _____ 20_____ г.

ЗАДАНИЕ
НА МАГИСТЕРСКУЮ ДИССЕРТАЦИЮ

Студент Васильев А.Т. **Группа** М4239 **Кафедра** компьютерных технологий **Факультет** информационных технологий и программирования **Руководитель** Ульянцев Владимир Игоревич, канд. техн. наук,, программист кафедры ИС Университета ИТМО

1 Наименование темы: Алгоритмы локального анализа графа де-Брейна для метагеномных данных

Направление подготовки (специальность): 01.04.02 Прикладная математика и информатика

Направленность (профиль): Технологии проектирования и разработки программного обеспечения

Квалификация: Магистр

2 Срок сдачи студентом законченной работы: «____» _____ 20_____ г.

3 Техническое задание и исходные данные к работе.

Требуется разработать программное обеспечение для поиска и анализа геномного окружения генов в метагеномных данных. Разработанное программное обеспечение должно по входной последовательности нуклеотидов и метагеномным чтениям находить и визуализировать геномное окружение этой последовательности. Исходные данные — метагеномные чтения и последовательности нуклеотидов.

4 Содержание магистерской диссертации (перечень подлежащих разработке вопросов)

а) Обзор предметной области
б) Разработка алгоритмов поиска геномного окружения
в) Результаты экспериментов

5 Перечень графического материала (с указанием обязательного материала)

Не предусмотрено

6 Исходные материалы и пособия

а) Метагеномные чтения [1]
б) База данных генов антибиотикорезистентности CARD [2]

7 Календарный план

№№ пп.	Наименование этапов магистерской диссертации	Срок выполнения этапов работы	Отметка о выполнении, подпись руков.
1	Изучение предметной области	31.12.2014	
2	Изучение существующих решений и форматов	01.03.2015	
3	Разработка алгоритма поиска окружения	01.07.2015	
4	Реализация алгоритма	31.12.2015	
5	Написание пояснительной записки	31.05.2016	

8 Дата выдачи задания: «____» _____ 20_____ г.

Руководитель _____

Задание принял к исполнению _____ «____» _____ 20_____ г.

Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ»

АННОТАЦИЯ
МАГИСТЕРСКОЙ ДИССЕРТАЦИИ

Студент: Васильев Артем Тарасович

Наименование темы работы: Алгоритмы локального анализа графа де-Брейна для метагеномных данных

Наименование организации, где выполнена работа: Университет ИТМО

ХАРАКТЕРИСТИКА МАГИСТЕРСКОЙ ДИССЕРТАЦИИ

1 Цель исследования: Разработка алгоритмов поиска и визуализации геномного окружения генов в метагеномных данных

2 Задачи, решаемые в работе:

а) разработка алгоритма поиска геномного окружения входной последовательности нуклеотидов

б) визуализация найденного геномного окружения

3 Число источников, использованных при составлении обзора: 12

4 Полное число источников, использованных в работе: 20

5 В том числе источников по годам

Отечественных			Иностранных		
Последние 5 лет	От 5 до 10 лет	Более 10 лет	Последние 5 лет	От 5 до 10 лет	Более 10 лет
4	0	0	11	2	3

6 Использование информационных ресурсов Internet: да, две ссылки

7 Использование современных пакетов компьютерных программ и технологий: Для написания пояснительной записки использовалась система подготовки документов LaTeX, для реализации алгоритмов использовались языки программирования Java и Python, для создания изображений графов использовался пакет Graphviz и утилита Bandage.

8 Краткая характеристика полученных результатов: Был разработан алгоритм поиска геномного окружения, а также способ визуализации результатов

9 Гранты, полученные при выполнении работы: заявки на получение грантов не подавались

10 Наличие публикаций и выступлений на конференциях по теме работы: нет

Выпускник: Васильев А.Т. _____

Руководитель: Ульянцев В.И. _____

«____» _____ 20_____ г.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ 6
1. Обзор предметной области 8
1.1. Биоинформатика 8
1.2. ДНК 8
1.3. Секвенирование ДНК 9
1.3.1. Метод Сэнгера 9
1.3.2. Метод дробовика 9
1.3.3. Методы нового поколения 9
1.4. Сборка генома 10
1.5. Метагеном 10
1.6. Граф де Брёйна 10
1.7. Форматы визуализации 11
1.7.1. Graphviz 11
1.7.2. Bandage 12
Выводы по главе 1 13
2. Описание и анализ алгоритма 15
2.1. Представление графа де Брёйна 15
2.2. Выбор параметра k 15
2.3. Фильтрация k-меров 17
2.4. Алгоритмы поиска окружения 17
2.4.1. Поиск в глубину 17
2.4.2. Поиск в ширину 19
2.5. Сжатие графа де Брёйна 20
2.6. Идентификация окружения 22
2.7. Комбинирование различных окружений 23
Выводы по главе 2 23
3. Описание и результаты экспериментов 24
3.1. Симулированные данные 24
3.2. Источник метагеномных чтений 24
3.3. Выбор минимального порога вхождения k-меров 25
3.3.1. $h = 1$ 25
3.3.2. $h = 5$ 25
3.3.3. $h = 10$ 26

3.4. Сравнение разных подходов к поиску в ширину. 26
3.5. Результаты выполнения алгоритма BLAST на базе NCBI. 26
Выводы по главе 3 27
ЗАКЛЮЧЕНИЕ. 33
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 34
ПРИЛОЖЕНИЕ А. Визуализация различий в геномном окружении в двух различных образцах 36

ВВЕДЕНИЕ

В последние годы проблема антибиотикорезистентности представляет собой глобальную проблему здравоохранения, в том числе представляет собой проблему национального масштаба в некоторых странах. Несмотря на общее признание этой проблемы, за последние 2 года статистика смертности от заболеваний, вызванных (прямо или косвенно) антибиотикорезистентными микробами не сократилось. Ожидается, что к 2050 году ещегодный показатель смертности достигнет 10 миллионов человек в год [3]. Таким образом, ввиду стремительного развития молекулярно-генетического вида анализа микробных организмов представляется чрезвычайно преспективным разработка новых эффективных методов анализа данных по отдельным бактериям и сложным микробным сообществам с прицелом на генные механизмы резистентности, в том числе прояснение механизмов переноса генов устойчивости между нормофлорой человека и патогенными микроорганизмами.

Биоинформатика также в последнее время наблюдает значительное увеличение в объеме метагеномных данных. Технологические продвижения и уменьшение стоимости секвенирования позволяют исследовать микроорганизмы из ранее неисследованных экологических ниш. Среднее покрытие метагенома выросло на несколько порядков, начиная с первых изучений метагеномов [4]. Новые микробные сообщества на большую часть состоят из некультивируемых бактерий, соответственно, для многих видов бактерий не существует референсного генома. Эта проблема существует не только для новых сообществ, но и для сообществ, которые изучались годами. В микробиоте человеческого кишечника, к примеру, неизвестные геномы составляют львиную долю всех чтений [5].

Одним из способов изучения таких сообществ является de novo сборка метагеномов (процесс, схожий с тем, что применяют к существующим индивидуальным геномам). Такая сборка осложняется широким многообразием видов бактерий и значительной внутривидовой геномной изменчивостью.

Кишечник человека образует динамический резервуар генов антибиотикорезистентности (генов АР, англ. ARG). Лечение антибиотиками имеет значительное воздействие на резистому кишечника и ведет к горизонтальному переносу генов антибиотикорезистентности. В связи с этим важно уметь выявлять

связь между набором принимаемых антибиотиков и увеличением представительности специфических групп генов антибиотикорезистентности.

В этой работе вышеописанная проблема подходится со стороны исследования геномного окружения отдельных участков возможного генома (в конкретном случае — генов антибиотикорезистентности).

В первой главе находится обзор предметной области: вводятся основные понятия биоинформатики, которые необходимы для понимания данной работы. Также вводятся термины, используемые в данной работе, вместе с их определениями.

Во второй главе описываются подходы к нахождению геномного окружения заданной последовательности нуклеотидов.

Третья глава содержит проведенные эксперименты вместе с их результатами.

ГЛАВА 1. ОБЗОР ПРЕДМЕТНОЙ ОБЛАСТИ

В данной главе находится обзор предметной области. Вводятся базовые моменты и определения биоинформатики, необходимые для понимания этой работы.

1.1. Биоинформатика

Также, как и в большинстве прикладных научных областей, последние продвижения в биологии и медицине открывают доступ к большему и большему набору данных, который необходимо уметь обрабатывать. В связи в этим на границе биологии, медицины, прикладной математики, математической статистики и информатики стала развиваться дисциплина под названием *биоинформатика*.

Биоинформатика занимается сбором и обработкой биологических данных вкупе с разработкой алгоритмов и программного обеспечения, решающих эти задачи. Биоинформатика решает множество задач, таких как расшифровка геномов живых организмов, аннотация генов, предсказание трехмерной структуры белка по последовательности аминокислот и много других.

1.2. ДНК

Дезоксирибонуклеиновая кислота (ДНК) — молекула, обеспечивающая хранение и передачу генетической информации из поколения в поколение. Молекула ДНК состоит из двух цепочек, являющихся последовательностями *нуклеотидов*. В ДНК встречается четыре вида азотистых оснований : аденин (A), гуанин (G), цитозин (C) и тимин (T). Азотистые основания одной цепочки соединены с азотистыми основаниями другой цепочки согласно *принципу комплементарности*: аденин (A) соединяется только с тимином (T), а гуанин (G) соединяется только с цитозином (C). Таким образом, зная последовательность оснований в одной цепочке, возможно однозначно восстановить последовательность в другой цепочке. За расшифровку структуры ДНК в 1953 году Крик, Уотсон и Уилкинс получили Нобелевскую премию по физиологии или медицине.

Центральная догма молекулярной биологии — утверждение, сформулированное Френсисом Криком в 1956 году, которое объясняет процесс передачи генетической информации внутри биологической системы. Оно гласит, что такая информация не может передаваться от белка к нуклеиновым кислотам, а

только в обратном направлении. Эта догма также описывается как «ДНК создает РНК, а РНК создает белки».

Изучение ДНК важно и актуально потому, что информация, сохраненная в ДНК отвечает за одни из самых важных свойств живых организмов — изменчивость и наследственность. Это включает, в том числе, и описание возможных генетических болезней, и понимание процессов, происходящих в ДНК сильно поможет как в исследовании, так и в лечении подобных болезней.

1.3. Секвенирование ДНК

Секвенирование ДНК означает процесс нахождения точного порядка нуклеотидов в молекуле ДНК. Результатом секвенирования обычно является набор *чтений* — набор последовательностей нуклеотидов разных фрагментов ДНК, вырезанных из случайных мест в молекуле. Существует несколько различных технологий секвенирования, отличающихся стоимостью, точностью и длиной получаемых последовательностей ДНК.

1.3.1. Метод Сэнгера

Метод Сэнгера (англ. Sanger), также известный как метод обрыва цепей, был методом, использовавшимся в первом поколении секвенаторов ДНК. Процесс секвенирования разделен на 4 этапа, каждый из которых выявляет позиции каждого из четырех типов нуклеотидов. Секвенаторы такого типа позволяют получать чтения ДНК длиной порядка тысячи нуклеотидов.

1.3.2. Метод дробовика

В секвенировании методом дробовика ДНК случайно разбивается на большое количество коротких маленьких фрагментов, которые затем секвенируются методом обрыва цепей для получения чтений. Такой процесс повторяется некоторое количество раз для получения перекрывающихся чтений для последующий сборки на компьютере. Этот метод был одним из первых, что сделал возможным секвенирование полного генома человека.

1.3.3. Методы нового поколения

Методы секвенирования нового поколения (англ. next generation sequencing, NGS) — общее название методов секвенирования, разработанных, как правило, в течение последних пары десятилетий. Отличительной чертой этих методов является получение сравнительно коротких чтений (до 500

нуклеотидов), благодаря чему можно значительно повышая их количество и уменьшая стоимость.

1.4. Сборка генома

Сборкой генома называют получение длинных фрагментов генома из коротких чтений. Именно сборка генома позволяет детально изучить геном любого существа и анализировать его. Программы, занимающиеся сборкой генома (их называют сборщики) в качестве результата зачастую выдают *контиги* — длинные непрерывные последовательности нуклеотидов, которые, предположительно, присутствуют в полном геноме (возможно, с незначительными либо редкими ошибками). Контиги часто объединяют в *скэффолды* — последовательности контигов вместе с оценками на то, как далеко друг от друга эти контиги располагаются в геноме.

1.5. Метагеном

В отличие от традиционного секвенирования, метагеномика занимается изучением не отдельных организмов, а целых микробных сообществ. Метагеномный анализ позволяет изучить видовое разнообразие в тех случаях, когда культивирование отдельных организмов затруднительно или невозможно. Многие неизученные сообщества (а также и те, которые изучались десятилетиями) на большую часть состоят из некультивируемых организмов, поэтому метагеномный анализ является одним из немногих возможных методов анализа таких сообществ. Чаще всего метагеномные чтения получают при помощи метода дробовика.

Традиционно, к метагеномным данным применяется тот же подход сборки, что и к обычным данным: по чтениям строятся контиги, которые потом анализируются всеми возможными средствами. В этой работе предлагается несколько другой подход к изучению метагеномов с точки зрения графового окружения.

1.6. Граф де Брёйна

Графом де Брёйна называется ориентированный граф, вершинами которого являются k-меры, а ребрами которого являются $(k+1)$-меры. Два k-мера соединены ребром, если суффикс длины $(k-1)$ k-мера, являющегося началом ребра, совпадает с префиксом такой же длины k-мера, являющегося концом

ребра. Таким образом, любому пути из n последовательных вершин соответствует последовательность из $k + n - 1$ символов.

Поскольку все более популярными становятся секвенаторы ДНК, выдающие большое количество коротких чтений, все больше программных комплексов, работающих с такими данными, используют графы де Брёйна в качестве опорной структуры данных для дальнейшего анализа и сборки. На рисунке 1 изображен пример графа де Брёйна для $k = 5$. На ориентированных ребрах этого графа написан символ, который добавляется к строке, являющейся концом соответствующего ребра.

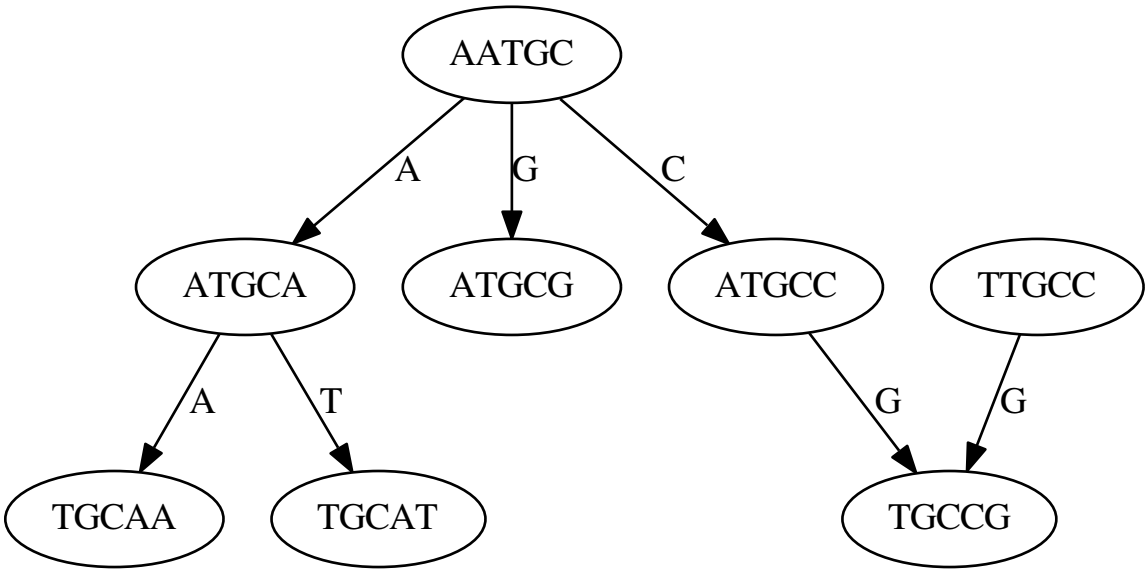

Рисунок 1 – Пример графа де Брёйна для $k = 5$

1.7. Форматы визуализации

Чтобы анализировать и работать с полученными окружениями, необходима качественная визуализация того, что выдал алгоритм поиска окружений. В данной работе было использовано два формата визуализации полученных геномных окружений.

1.7.1. Graphviz

Graphviz [6] это пакет программ для визуализации графов, разработанный отделением исследований и разработки компании AT&T. В составе этого

пакета поставляется утилита dot, которая обычно используется для отрисовки ориентированных графов в иерархическом формате. Она — стандартный выбор для изображения ориентированных графов.

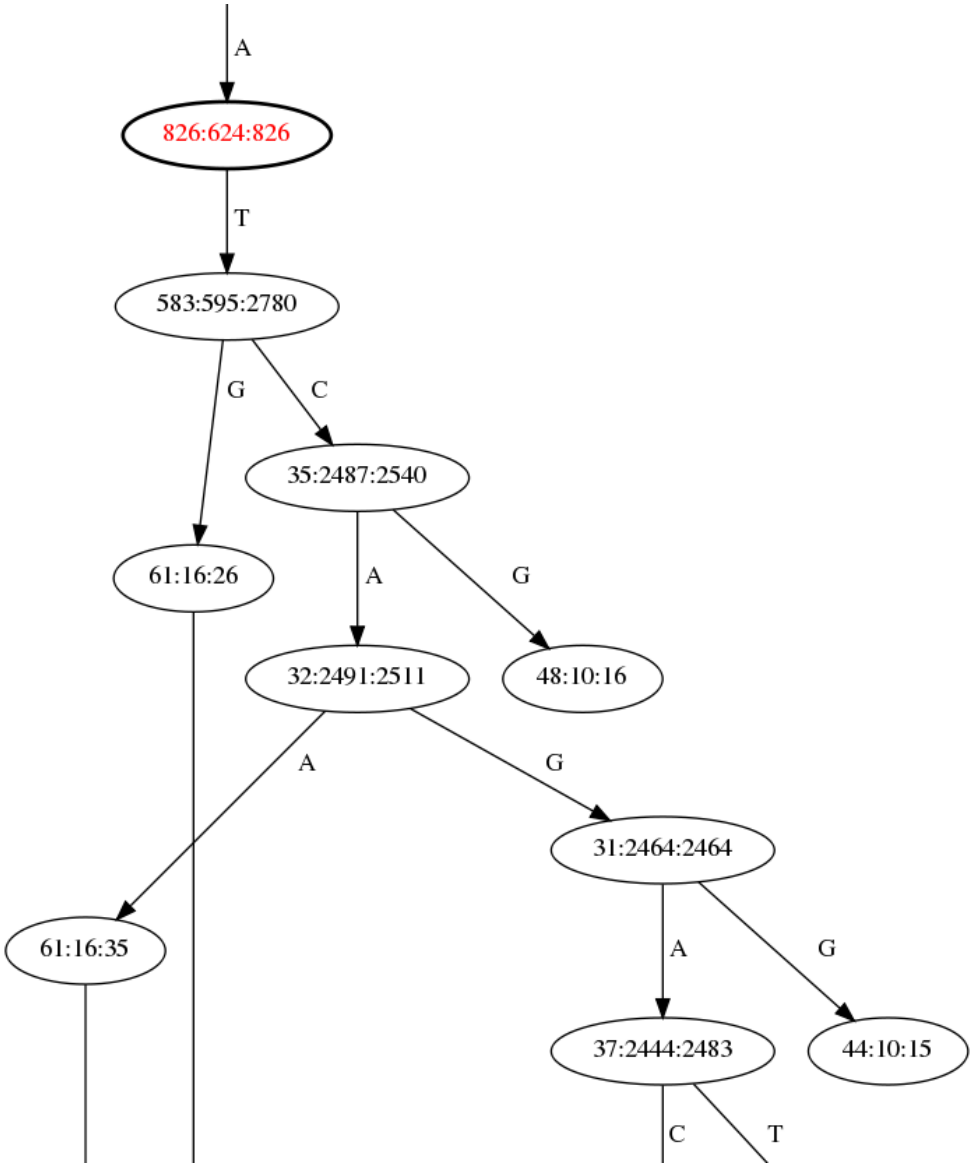

Рисунок 2 – Фрагмент графа геномного окружения, созданный graphviz

1.7.2. Bandage

Bandage [7] это программа, созданная для визуализации и интерактивного просмотра результатов de novo сборки современных сборщиков геномов. Несмотря на свое заявленное предназначение, программа отлично приспособлена для отрисовки графов де Брёйна, а соответственно, хорошо подходит для визуализации данных алгоритмов, описанных в этой работе. Bandage позво-

ляет отрисовывать графы де Брёйна размера порядка нескольких МБ всего за несколько секунд на среднем современном персональном компьютере.

Bandage поддерживает множество различных форматов графов от нескольких различных сборщиков, включая следующие:

— Формат LastGraph, использованный в сборщике Velvet [8]
— Формат FASTG, используемый, например, в сборщике SPAdes [9]
— Формат Trinity.fasta, разработанный в сборщике Trinity [10]

В данной работе для представления результатов алгоритма и последующей визуализации в Bandage использовался формат LastGraph.

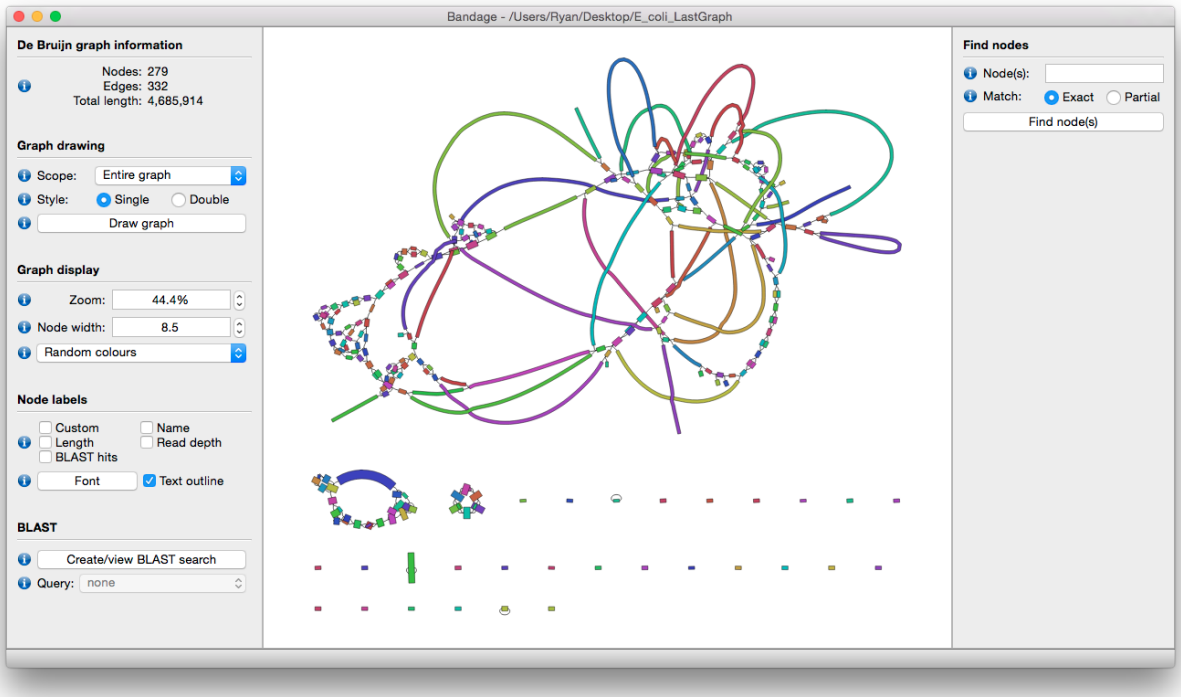

Рисунок 3 – Пользовательская оболочка утилиты Bandage [11]

На рисунке 3 изображена пользовательская графическая оболочка данной программы, изображающая сборку генома бактерии E.Coli сборщиком Velvet [8].

Выводы по главе 1

В первой главе были рассмотрены основные моменты биоинформатики: структура ДНК, методы секвенирования, особенности метагеномики. Были также описаны возможные форматы визуализации, с помощью которых можно

анализировать и изучать результаты запусков алгоритмов нахождения геномного окружения.

ГЛАВА 2. ОПИСАНИЕ И АНАЛИЗ АЛГОРИТМА

В этой главе будет описан используемый подход к нахождению геномного окружения заданных нуклеотидных последовательностей и методы их визуализации.

2.1. Представление графа де Брёйна

Непростую задачу представляет из себя хранение и представление графов де Брёйна в программном коде. Перед представлением этих графов ставится задача минимального потребления памяти наряду с возможностью быстрого определения частоты данного k-мера. С такой задачей традиционно хорошо справляются *хеш-таблицы* — структуры данных, занимающие линейное от размера входных данных память и позволяющие за ожидаемое константное время получать значение по ключу (в предположении того, что хеш-функция распределяет ключи достаточно равномерно, либо сами данные достаточно случайны).

В данной работе используются хеш-таблицы и утилиты для быстрого параллельного построения графа де Брёйна из входных чтений, реализованных в библиотеке itmo-assembler [12, 13], незначительно модифицированные под нужды данной работы, а также некоторые вспомогательные классы из программного комплекса Metafast [14].

2.2. Выбор параметра k

Как и в любом алгоритме, для достижения лучшего результата необходимо правильно подбирать параметры. От выбора параметра k очень сильно зависит то, как выглядит наш граф де Брёйна. При слишком маленьком параметре k могут появиться ребра, прохождение по которым не дает путей, которые можно получить из изначальных чтений. Такие ребра называют «химерными» ребрами. Если же установить k слишком большим, может получиться так, что какие-то чтения, пересечение которые было меньше, чем k, не будут соединены ребром в графе, а соответственно, не будет пути, связывающего два «соседних» чтения, что негативно отражается на работе алгоритмов. Естественно, выбирать k примерно равным длине чтения бессмысленно, поскольку в таком случае количество ребер в графе будет крайне мало, и из такого графа не удастся извлечь много интересной информации.

Другим аспектом выбора k является сложность программной реализации. Для хранения k-мера из символов A, G, C и T требуется $2k$ битов (для каждого из k символов существует 4 различных варианта, которые можно закодировать двумя битами). В 64-битной архитектуре один k-мер можно хранитть в одном машинном слове, если $k \leqslant 32$. В добавку к этому, зачастую принято брать k нечетным, поскольку при нечетных k не существует k-мера, который совпадает с k-мером, обратно комплементарный ему. Отсюда получается популярное ограничение $k \leqslant 31$, k нечетно, при котором алгоритмы, работающие с графом де Брёйна выполняются сравнительно быстро.

Для $k > 31$ такое представление невозможно, и приходится прибегать к другим методам хранения. Рассмотрим два метода хранения.

Первый метод заключается в хранении k-меров целиком. В языке Java нет примитивных типов разрядности больше, чем 64 бита, поэтому для хранения таких k-меров необходимо создавать отдельный класс, что ведет к дополнительным расходам памяти, сравнимым с размером изначальных данных, а также сильно замедляет работу с k-мерами.

Второй подход отличается от первого тем, что он не использует больше памяти, чем алгоритмы для $k \leqslant 31$, но при этом может ошибаться с ненулевой вероятностью. Такой подход заключается в том, что вместо подсчета количества вхождений k-меров будут считаться количества вхождений хешей этих k-меров. При таком подходе, очевидно, есть ненулевая вероятность коллизии — совпадения значения хеш-функции двух разных k-меров. Недостатком такого подхода является то, что по записи k-мера в хеш-таблице теперь невозможно однозначно восстановить сам k-мер. Однако, в алгоритмах поиска окружения этот недостаток восполняется тем, что все k-меры последовательности известны, и нет необходимости восстанавливать сами k-меры по их ключам в хеш-таблице. Для $k \leqslant 31$ такой проблемы не возникает, так как в качестве ключа в хеш-таблице можно использовать само $2k$-битное представление k-мера, которое вмещается в машинное слово.

В данной работе будут использоваться точный подход для $k \leqslant 31$ а также второй, неточный, подход для больших k. Для второго подхода будет оценена вероятность коллизии, которая может произойти при чтении k-меров.

2.3. Фильтрация k-меров

Секвенирование ДНК неидеально, и в полученных чтения могут встречаться ошибки. Ошибки секвенирования бывают нескольких видов:

— Ошибки вставки — в линейную последовательность нуклеотидов вставляется нуклеотид, которого там изначально не было
— Ошибки удаления — в линейной последовательности нуклеотидов пропускается какой-то из существующих нуклеотидов
— Ошибки замены — вместо какого-то нуклеотида в последовательности записан другой

В предложенном методе не производится каких-либо попыток исправления ошибок, вместо этого производится попытка фильтровать k-меры, в которых произошла ошибка. Как видно из 4, огромное количество k-меров встречаются в чтениях лишь однажды, и, вероятнее всего, являются ошибками секвенирования. Для избавления от таких ошибочных k-меров, применяется фильтрация по частоте, то есть, k-меры, которые встречались меньше, чем h раз, выкидываются из рассмотрения. Часто на практике достаточно положить $h = 2$, но в данной работе было рассмотрено несколько возможных значений h и проанализировано, как выбор h влияет на результат выполнения.

2.4. Алгоритмы поиска окружения

Задачей алгоритма поиска окружения является по исходной геномной последовательности выдать подграф графа де Брёйна, являющийся «окружением» заданной последовательности в некотором смысле. В этой секции будут рассмотрено несколько алгоритмов поиска окружения, каждый из которых имеет ряд своих преимуществ и недостатков.

2.4.1. Поиск в глубину

Первый из подходов основан на алгоритме поиска в глубину и является самым простым из опробованных подходов. Этот подход находит некоторое фиксированное число путей, начинающихся в каждом из k-меров исходного гена и идущих в одном фиксированном направлении. Такой алгоритм запускается дважды для такого из направлений «вперед» и «назад». В качестве результата алгоритм выдает набор путей в графе, начинающихся или заканчивающихся в k-мерах искомой геномной последовательности.

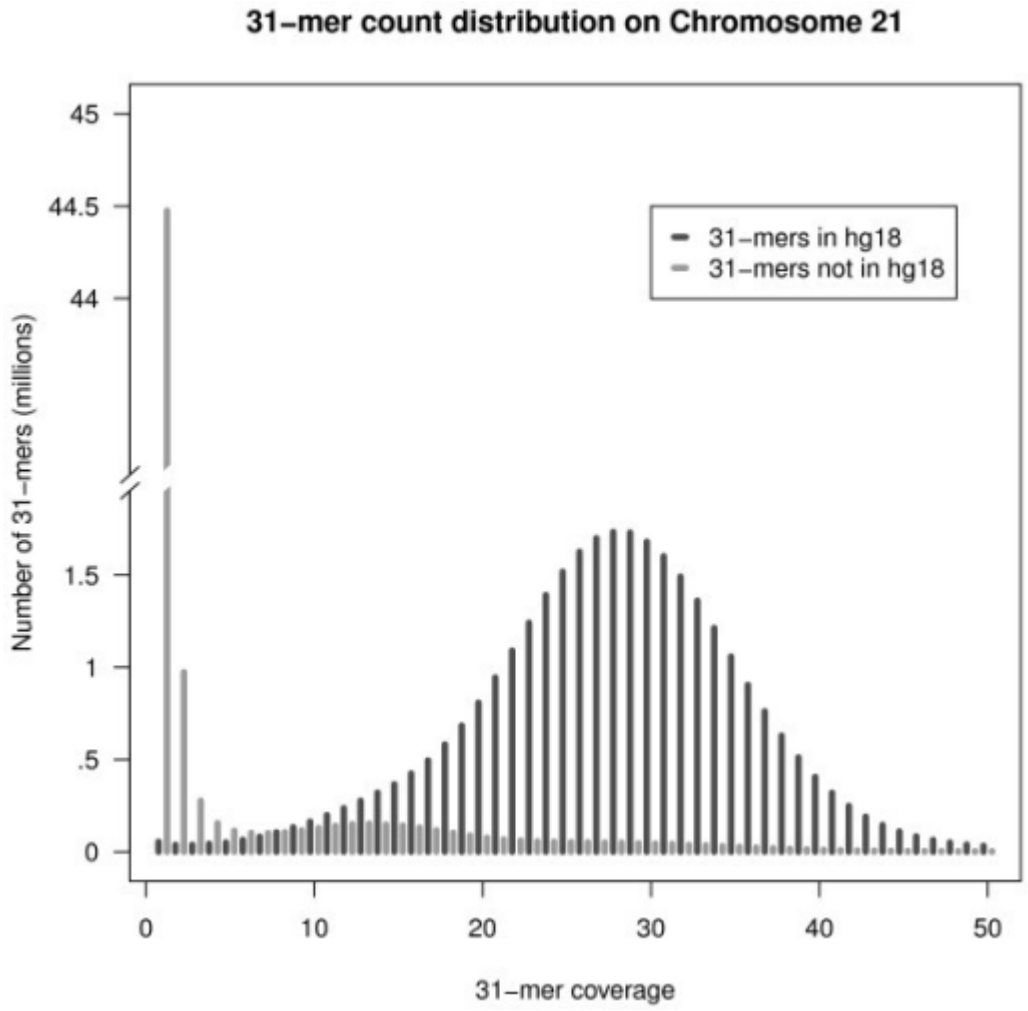

Рисунок 4 – Распределение частоты корректных и некорректных k-меров. Источник: [15]

Преимуществом этого метода является то, что его результатом является набор путей, по которому затем строится граф де Брёйна. Такой формат позволяет улучшить качество выдаваемых путей посредством *валидации чтениями*. Валидация чтениями данного пути заключается в том, что все метагеномные чтения наносятся на этот путь, и если существует часть пути, которая не была покрыта каким-либо чтением, то этот путь является заведомо некорректным или «химерным». Стоит отметить, что обратное неверно, и может существовать путь, который полностью покроется чтениями, но не будет частью генома. Такая эвристика помогает уменьшить размер графа и увеличить соответствие построенного графа реальным данным.

Недостатками такого подхода является то, что количество таких путей может возрастать экспоненциально относительно длины пути, а поэтому таким образом невозможно получить пути достаточно большой длины. К сожале-

нию, такой недостаток очень сильно влияет на производительность алгоритма на метагеномных данных из-за являения, называющегося *однонуклеотидный полиморфизм* (англ. single nucleotide polymorphism, SNP). Это явление состоит в изменении единственного нуклеотида в определенной позиции в геноме у небольшого числа особей, что влечет за собой ситуацию, показанную на рисунке 5. Такой подграф эффективно удваивает количество путей, проходящих через этот фрагмент, и достаточное количество проявлений однонуклеотидного полиморфизма заметно ухудшает работу данного алгоритма.

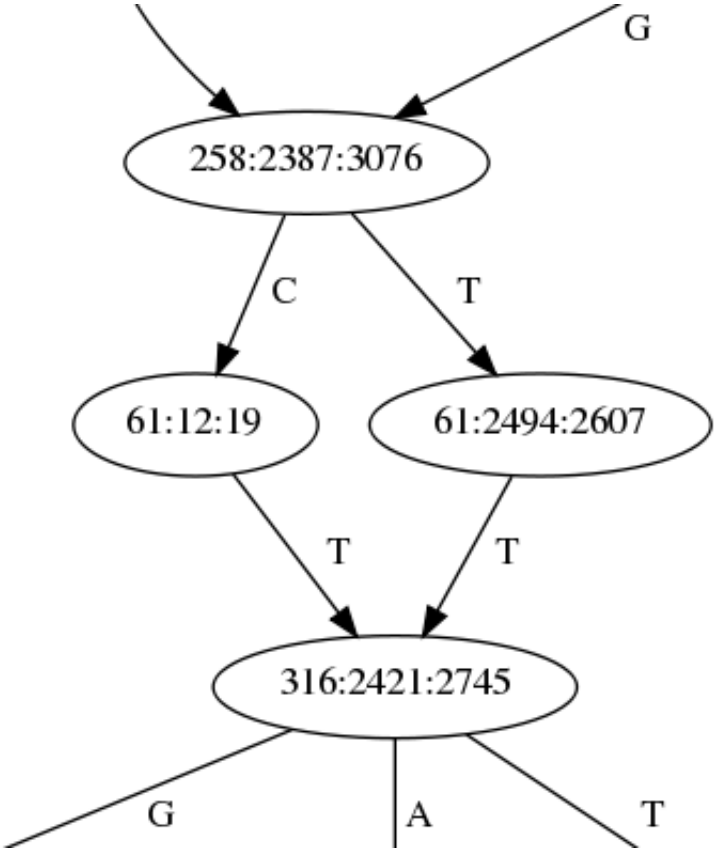

Рисунок 5 – Явление однонуклеотидного полиморфизма

2.4.2. Поиск в ширину

Поиск в ширину на графе — обход графа, посещающий вершины в порядке увеличения кратчайшего расстояния по невзвешенным ребрам графа. В отличие от поиска в глубину, время работы поиска в ширину не увеличивается с ростом количества путей, но и валидацию чтениями на результатах его запуска произвести сложнее. В данной работе было реализовано две вариации обхода в глубину. Первый из них, аналогично поиску в глубину запускается дважды, каждый раз рассматривая только правых или левых соседей k-мера в

графе де Брёйна соответственно. Второй же запускается лишь однажды, посещая каждый раз всех соседей каждый рассматриваемой вершины. Каждый из этих двух методов имеет свои преимущества и недостатки.

а) Первый из них пропускает участки метагенома, «параллельные» заданной геномной последовательности, то есть такие последовательности, что в графе де Брёйна существует их общий предок (вершина, достижимая из обеих заданных вершин) или общий потомок (вершина, из которой достижимы обе заданные вершины). Однако, такой подход приоретизирует ту часть графа, которая находится непосредственно за или предшествует искомой геномной последовательности, что более интересно для изучения в биологическом смысле.

б) Второй из методов находит геномное окружение в более полном смысле и, потенциально, может обнаружить то, что не обнаружил первый метод. Однако, эта вариация алгоритма имеет склонность уходить слишком далеко от изначальной последовательности, что может привести к появлению в окружении последовательностей, которые на самом деле не находятся в этом окружении.

В отличие от поиска в глубину, алгоритм поиска в ширину сразу выдает граф де Брёйна, являющийся геномным окружением. Критерием останова поиска в ширину является посещение определенного количества k-меров (в данной работе —- 20000).

После нахождения графового окружения любым из описанных алгоритмов, может случиться так, что какие-то длинные неветвящиеся последовательности входят в построенное окружение не полностью. Для этого, после построения графа де Брёйна окружения рассматриваются все стоки (вершины без исходящих ребер) и истоки (вершины без входящих ребер). Для этих вершин ищется продолжение, пока оно существует, либо пока не достигнется развилка (два или более входящих либо исходящих ребра), на этом поиск останавливается. Все найденные продолжения добавляются в результирующий граф.

2.5. Сжатие графа де Брёйна

Для визуализации найденного окружения использовать неизмененный граф де Брёйна непрактично, поскольку в графе будет содержаться большое количество линейных неразветвляющихся цепочек. Граф, в котором такие цепочки сжаты в одну вершину, называется *сжатым графом де Брёйна*. Более

формально, сжатым графом де Брёйна называют граф, в котором вершинам теперь соответствуют не обязательно k-меры, а неветвящиеся последовательности любой длины, большей или равной k, которые также называются *юнитигами*. Ребра в таком графе все еще представляют собой $(k+1)$-меры, и последние $k-1$ символов строки, соответствующие началу ребра совпадают с первым $k-1$ символами строки, соответствующей концу этого ребра. Заметим, что в такой постановке обычный граф де Брёйна является частным случаем сжатого графа де Брёйна.

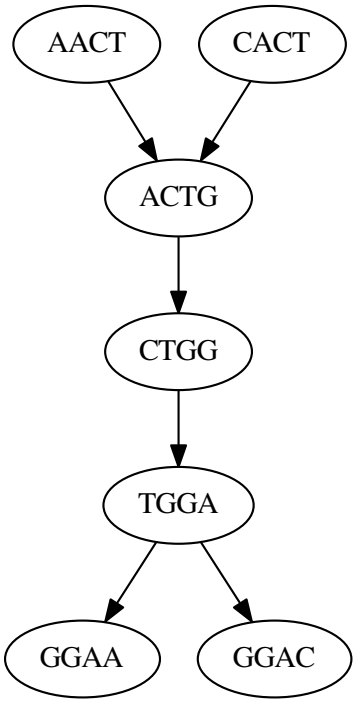

Рисунок 6 – Цепочка вершин, которую необходимо сжать в одну вершину

В этой работе был реализован алгоритм построения сжатого графа де Брёйна с минимальным количеством вершин, который работает за линейное от количества вершин и ребер в графе время. Стоит отметить, что данный алгоритм работает не только для графов де Брёйна, а для любых произвольных ориентированных графов. Алгоритм последовательно сжимает пары вершин, которые можно сжать в одну, пока такая пара существует. Пару вершин u и v можно сжать по ребру тогда и только тогда, когда это ребро является

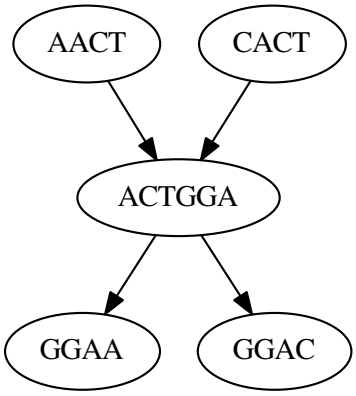

Рисунок 7 – Тот же самый граф, с неветвящимся путем, сжатым в одну длинную вершину

единственным исходящим ребром из первой вершины, а также единственным входящим ребром во вторую вершину.

Заметим, что сжатие двух вершин по ребру не меняет состояние всех остальных ребер: то, можно ли сжать конца ребра в одну вершину, не зависит от того, что по какому-то ребру произошло сжатие. Этот факт приводит к простому алгоритму построения сжатого графа де Брёйна, работающему за $O(V + E)$, где V — количество вершин в графе, а E — количество ребер.

Необходимо сказать, что существуют более продвинутые алгоритмы, которые позволяют построить сжатый граф де Брёйна быстрее и эффективнее, используя параллельные вычисления [16]. В этой работе сжимаемые графы де Брёйна, описывающие геномное окружение, имеют размер, на несколько порядков меньший, чем полный граф де Брёйна для метагенома, поэтому использование такого рода алгоритмов не было необходимо.

2.6. Идентификация окружения

После того, как сжатый граф де Брёйна был построен, следует понять функциональное устройство геномного окружения. Для этого в данной работе были взяты все достаточно длинные неветвящиеся последовательности, встречающиеся в этом графе, и запущен алгоритм BLAST [17] на базе последовательностей NCBI [18]. В результате этого было получено описание последова-

тельностей, встречающихся в окружении искомого гена, опираясь на которые, можно делать выводы о механизмах переноса этого гена.

Результаты выполнения алгоритма BLAST анализируются в третьей главе.

2.7. Комбинирование различных окружений

Одним из интересных применений построения геномных окружений является сравнение построенных окружений для разных метагеномных данных, таких как микробиоты кишечников разных людей или микробиота кишечника одного и того же человека в разные дни. В рамках данной работы была реализована одновременная визуализация нескольких окружений средствами пакета Graphviz. Вершины, присутствующие в обеих версия окружения, имеют черную границу, те, что присутствуют только в одном из окружений, имеют синюю или красную границу соответственно.

В Приложении А. можно увидеть результат фрагмент такой визуализации для двух взятий образцов микробиоты кишечника человека через 2 и 28 дней после принятия курса антибиотиков.

Выводы по главе 2

Во второй главе были описаны использованные алгоритмы нахождения геномного окружения вместе с описанием и сравнением различных тонкостей в их реализации.

ГЛАВА 3. ОПИСАНИЕ И РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ

В этой главе описываются проведенные эксперименты вместе с анализом их результатов. В конце главы на основе результатов работы алгоритма BLAST [17] проводится сравнение с выводами, полученными в статье [1].

3.1. Симулированные данные

Перед запуском реализованных алгоритмов на реальных метагеномных данных, первые версии алгоритмов тестировались на симулированных чтениях. Для этого был взят собранный геном Klebsiella pneumoniae strain MS6671 с таблицей генов антибиотикорезистентности [19]. Чтения строились из собранного геома следующим образом: 10^6 раз генерировалась случайная подстрока этого генома, с вероятностью 50% вместо нее выводилась обратно комплементарная к ней строка. Длины все чтений были равны 100.

В качестве исследуемого гена был выбран ген $bla_{OXA-181}$, имеющий длину в 798 нуклеотидов и встречающийся в трех местах в данном геноме: дважды встречается прямая версия и однажды — обратно-комплементарная версия.

3.2. Источник метагеномных чтений

В качестве входных данных алгоритма используются метагеномные чтения, полученные авторами статьи [1], а также база генов антибиотикорезистентности CARD [2]. Метагеномные данные, с которыми происходит анализ в этой работе, содержат порядка $450 \cdot 10^6$ парных чтений, каждое из которых имеет длину около сотни.

Метагеномные данные были получены от двух взрослых пациентов. Каждый из них в течение шести дней принимал ципрофлоксацин (антибиотик, предназначенный для лечения бактериальных кишечных инфекций). От каждого пациента было взято шесть образцов в различные дни: перед началом лечения (в день 0), во время лечения (в дни 1, 3 и 6), а также после окончания лечения (через 2 и 28 дней соответственно).

Авторы статьи [1] отмечают, что во время приема препарата, в метагеномах пациентов повысилось присутствие антибиотикорезистентных генов, таких как OXA-209, OXA-347, OXA-237, OXA-360. OXA-347 был найден на одном из контигов метагеномной сборки у второго пациента в третий день

лечения, а рядом с этим геном были обнаружены функциональные последовательности, которые свидетельствуют о том, что генный участок, где находился экземпляр этого гена — мобильный элемент ДНК и, соответственно, подвержен генному переносу.

Перед разработанным алгоритмом ставилась задача повторить вышеописанное открытие. В качестве изучаемой геномной последовательности был взят ген OXA-347 длиной в 826 нуклеотидов. После было найдено его геномное окружение во все дни взятия образцов второго пациента. Для первого пациента ни один k-мер этого гена в метагеномных чтениях не встретился. По умолчанию, если не сказано обратного, будет использоваться датасет TUE_S2-3 с параметром $k = 31$.

3.3. Выбор минимального порога вхождения k-меров

Как было описано во второй главе, для устранения ошибочных k-меров применяется фильтрация k-меров по частоте встречаемости. Выбор нижнего порога h связан с компромиссом между удалением неверных k-меров и сохранением реальных данных недопредставленных в метагеноме организмов. Для того, чтобы определить, какое значение h следует использовать, были проведены эксперименты с $h = 1, 5, 10$, и проанализированы полученные геномные окружения.

3.3.1. $h = 1$

В случае, когда никакие k-меры не фильтруются, граф окружения включает большое количество ответвлений с покрытием 1, и даже вершины, соответствующие k-мерам исследуемого гена, не образуют неветвящуюся последовательность. Фрагмент графа окружения, полученный при помощи Graphviz показан на рисунке 8. На этой картинке и далее во всех подобных, формат подписи к каждой вершине выглядит следующим образом: *length:min:max*, где length — длина последовательности, соответствующей данной вершине, min и max — минимальное и максимальное покрытие среди всех k-меров данной последовательности. Наблюдая значения min и max, можно делать выводы о том, следует ли изменить параметр h.

3.3.2. $h = 5$

При $h = 5$ большинство единичных помех уже устранено, и ген OXA-347 представлен единственным юнитигом. Однако теперь проявляется другой

эффект, когда от длинных линейных уч.астков появляются следующего рода ответвления:

3.3.3. $h = 10$

В связи с результатами для $h = 5$, было решено исследовать поведение алгоритма с параметром $h = 10$. Теперь такого спецэффекта не наблюдается, но нельзя достоверно сказать, были ли удалены события, связанные с редко представленными организмами.

3.4. Сравнение разных подходов к поиску в ширину

Во второй главе описывалось два разных подхода к реализации поиска в ширину. В этой секции будет визуально продемонстрирована разница между этими двумя подходами.

На графе, построенном двунаправленным поиском в ширину 12 виден цикл, что вероятнее всего означает встраивание гена антибиотикорезистентности в существующую структуру. На графе, построенном однонаправленным поиском в ширину 11, такого цикла не видно, но в этом случае наглядно показывается метагеномное разнообразие, где ген после некоторого линейного участка

3.5. Результаты выполнения алгоритма BLAST на базе NCBI

В этой части будут продемонстрированы результаты запуска алгоритма BLAST на базе NCBI по длинным (длины $\geqslant 200$) юнитигам, полученных при запусках двух разных версий поисков в ширину. В таблице 1 показано количество найденных в базе NCBI последовательностей, присутствующих в достаточно большой мере в юнитигах.

Таблица 1 – Результаты BLAST (только количество)

Проба	Однонаправленный поиск	Двунаправленный поиск
TUE_S2-1	3	3
TUE_S2-2	45	4
TUE_S2-3	57	32
TUE_S2-4	67	45
TUE_S2-5	81	11
TUE_S2-6	51	3

Из этой таблицы видно, что два запуска однонаправленного поиска находят больше аннотированных последовательностей, что дает больше информации для изучения окружения гена. Важно отметить, что среди найденных BLAST последовательностей недалеко от гена присутствует транспозаза бактерии *Bacteroides fragilis*, что свидетельствует о расположении гена на подвижном фрагменте ДНК [1]. Ровно такой же вывод был сделан авторами статьи [1], что показывает состоятельность предложенного метода.

Выводы по главе 3

В этой главе было описано несколько экспериментов, проведенных с разработанным и реализованным алгоритмом анализа геномного окружения генов. Последний шаг анализа — использование BLAST для поиска известных последовательностей в найденном окружении гена, подтвердил один из результатов в [1], чем показал состоятельность предложенного метода.

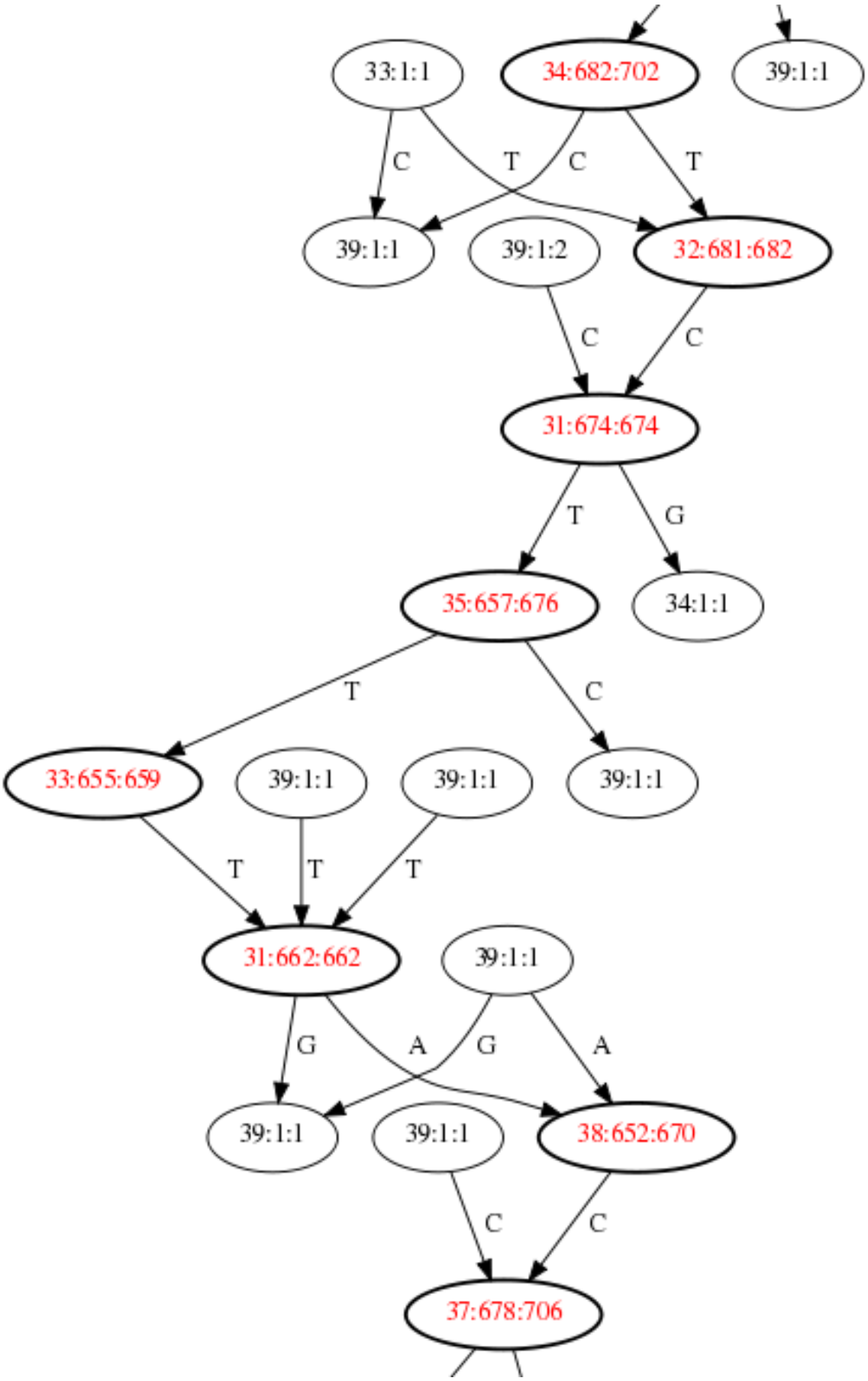

Рисунок 8 – Фрагмент графа окружения с $h = 1$

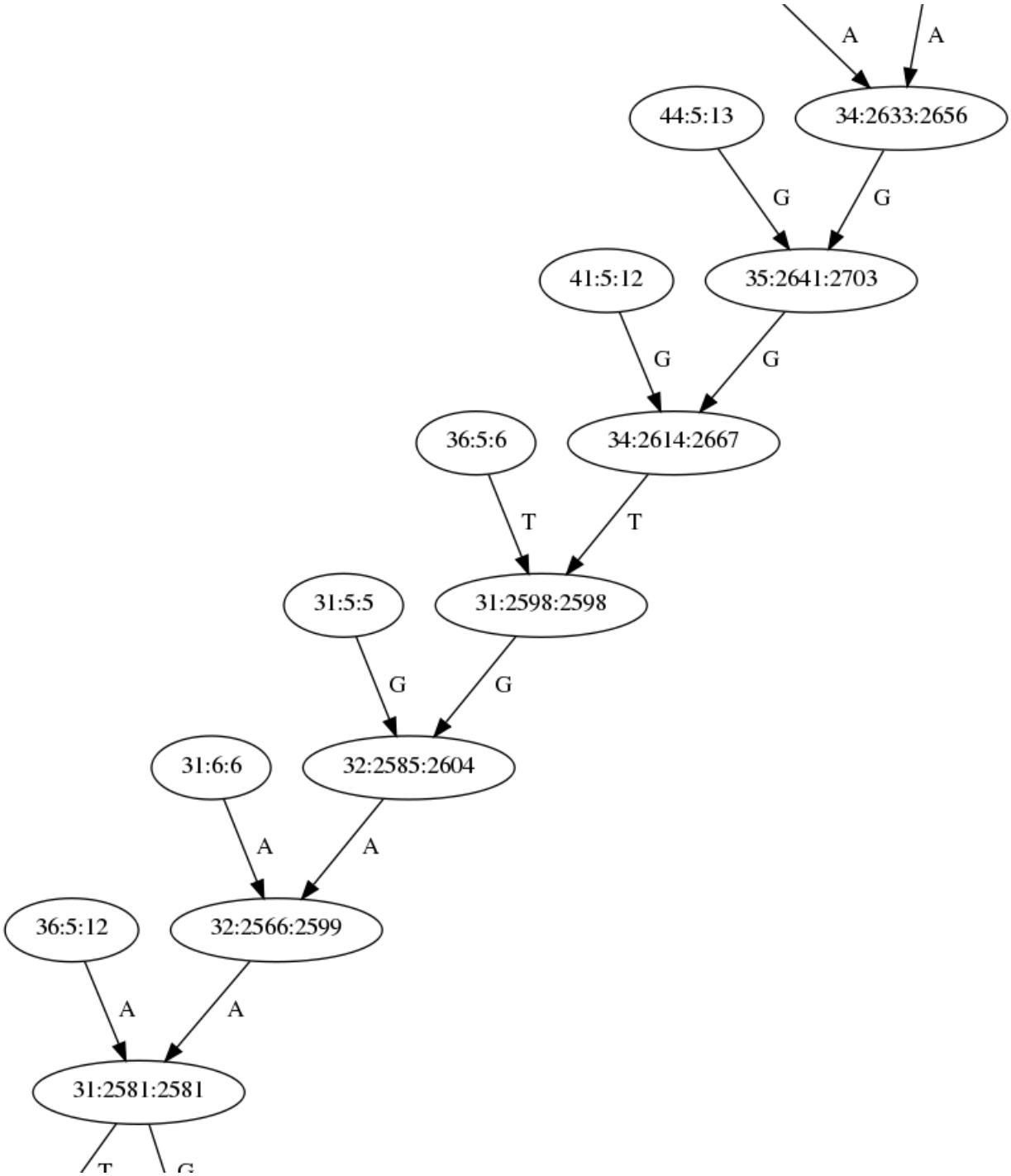

Рисунок 9 – Фрагмент графа окружения с $h = 5$

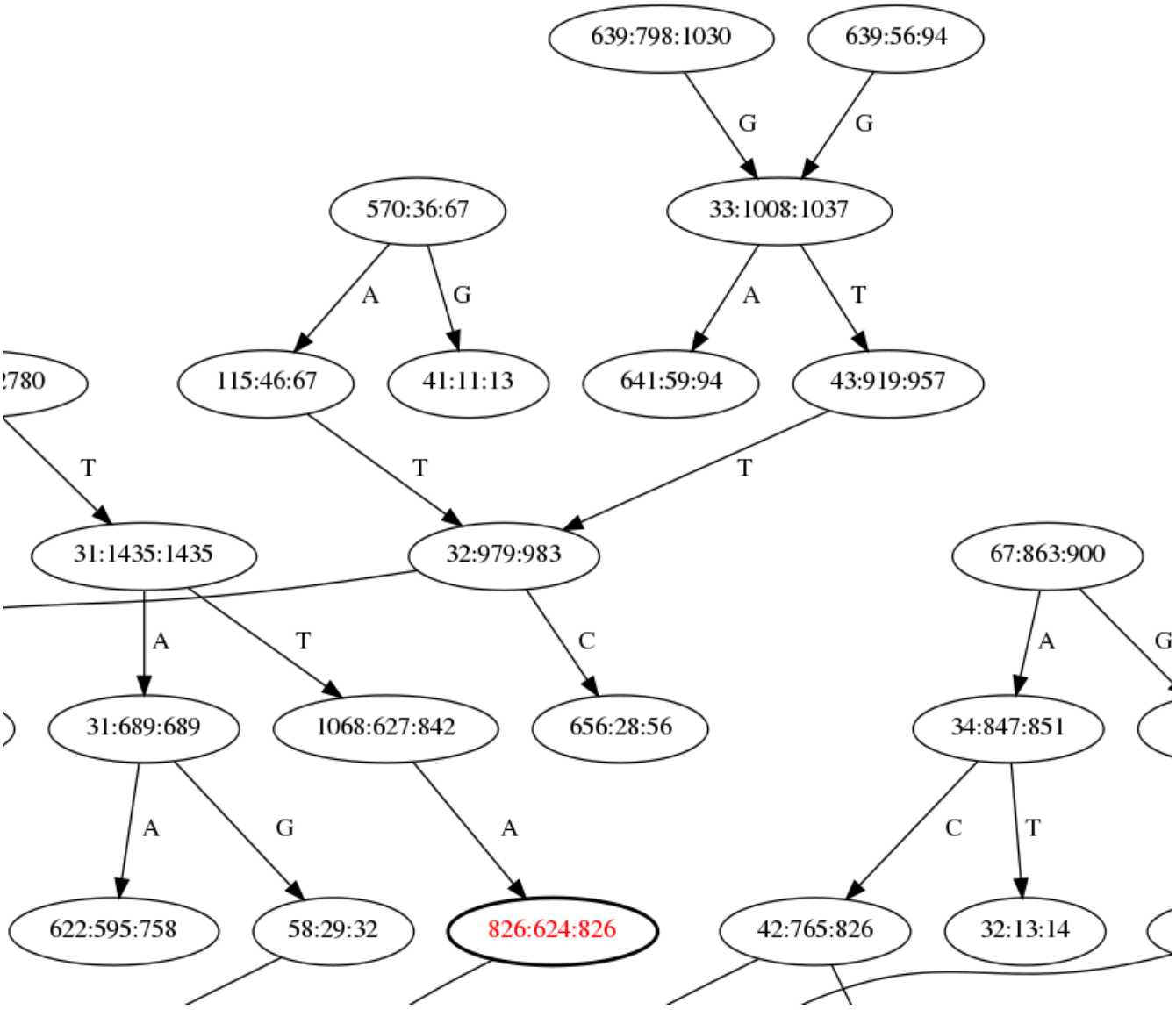

Рисунок 10 – Фрагмент графа окружения с $h = 10$

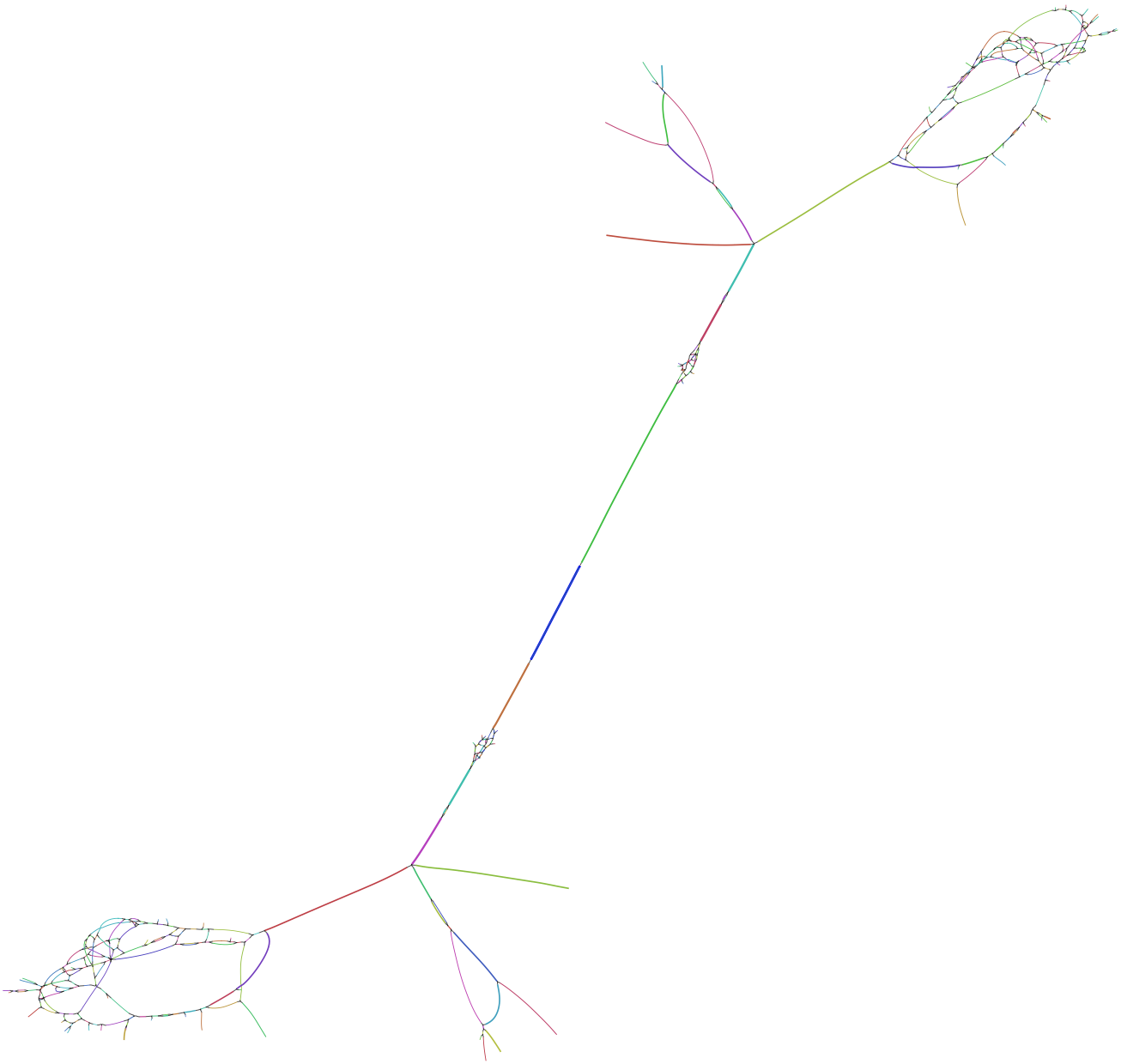

Рисунок 11 – Граф окружения, построенный однонаправленным поиском в ширину

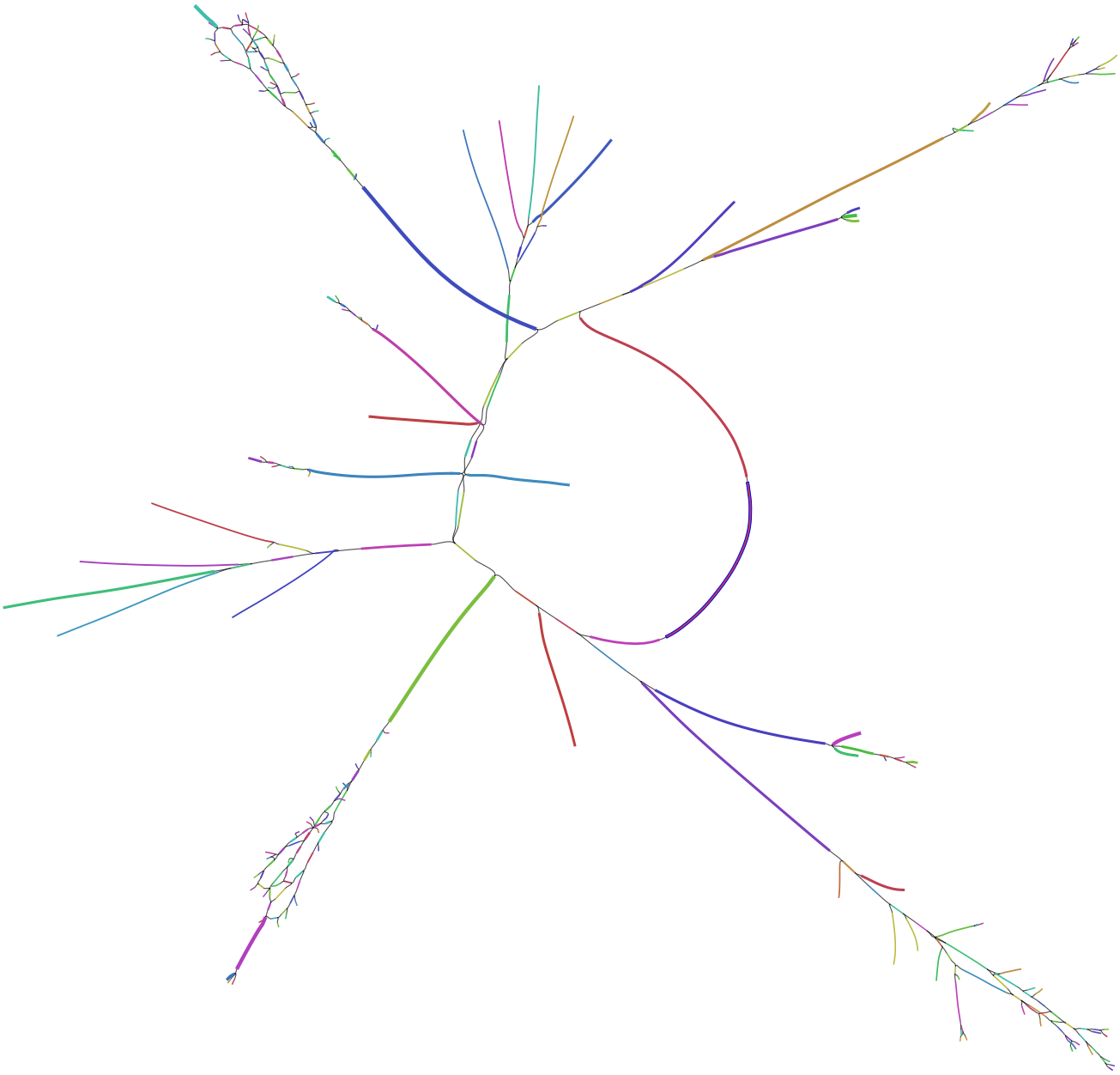

Рисунок 12 – Граф окружения, построенный двунаправленным поиском в ширину

ЗАКЛЮЧЕНИЕ

В этой работе была рассмотрена задача нахождения и идентификации графового окружения генов в метагеномных данных с последующим применением для изучения механизмов передачи антибиотикорезистентности в микробиоте кишечника человека. Проблема изучения сопротивления патогенных микроорганизмов антибиотикам становится все более актуальной ввиду увеличения использования антимикробных препаратов, что в свою очередь увеличило их шансы выработать резистентность. Такая ситуация заставляет разрабатывать новые лекарства на замену тем, что оказываются неэффективными из-за возросшего к ним сопротивления [20].

В данной работе был сформулирован новый подход к изучению передачи механизмов антибиотикорезистентности на основе изучения геномного окружения генов АР. Одним из компонентов такого подхода является построение графа де Брёйна для метагеномных данных и поиск геномного окружения внутри этого графа для заранее определенной последовательности. В качестве обоснованности такого метода был повторен эксперимент, проведенный в статье [1].

Однако в данной работе не удалось реализовать все изначально задуманные идеи, новых результатов получено не было. Существует несколько возможных векторов развития этой работы:

— Создание отдельного программного комплекса для работы с метагеномами и геномными окружениями.
— Разработка алгоритмов поиска геномных островов без опоры на известные гены АР
— Подтверждение явлений переноса антибиотикорезистентности на большем разнообразии входных данных

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Antibiotic selection pressure determination through sequence-based metagenomics / M. Willmann [и др.] // Antimicrobial Agents and Chemotherapy. — 2015. — Т. 59, № 12. — С. 7335–7345.

2 The comprehensive antibiotic resistance database / A. G. McArthur [и др.] // Antimicrobial agents and chemotherapy. — 2013. — Т. 57, № 7. — С. 3348–3357.

3 *Antimicrobial Resistance (London). R. on*, *Grande-Bretagne* Antimicrobial resistance: tackling a crisis for the health and wealth of nations. — Review on Antimicrobial Resistance, 2014.

4 Environmental genome shotgun sequencing of the Sargasso Sea / J. C. Venter [и др.] // Science. — 2004. — Т. 304, № 5667. — С. 66–74.

5 Identification and assembly of genomes and genetic elements in complex metagenomic samples without using reference genomes / H. B. Nielsen [и др.] // Nature biotechnology. — 2014. — Т. 32, № 8. — С. 822–828.

6 Graphviz—open source graph drawing tools / J. Ellson [и др.] // Graph Drawing. — Springer. 2001. — С. 483–484.

7 Bandage: interactive visualization of de novo genome assemblies / R. R. Wick [и др.] // Bioinformatics. — 2015. — Окт. — Т. 31, № 20. — С. 3350–3352.

8 *Zerbino D. R.*, *Birney E.* Velvet: algorithms for de novo short read assembly using de Bruijn graphs // Genome Research. — 2008. — Т. 18, № 5. — С. 821–829.

9 SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing / A. Bankevich [и др.] // Journal of Computational Biology. — 2012. — Т. 19, № 5. — С. 455–477.

10 Full-length transcriptome assembly from RNA-Seq data without a reference genome / M. G. Grabherr [и др.] // Nature Biotechnology. — 2011. — Т. 29, № 7. — С. 644–652.

11 *Wick R.* Bandage: a Bioinformatics Application for Navigating De novo Assembly Graphs Easily. — 2015. — URL: `http://rrwick.github.io/Bandage/` (дата обр. 20.06.2016).

12 Combining de Bruijn graph, overlaps graph and microassembly for de novo genome assembly / A. Alexandrov [и др.] // Proceedings of «Bioinformatics. — 2012. — С. 72.

13 *Laboratory G. A. A.* Genome Assembler. — 2016. — URL: `http://genome.ifmo.ru/en/assembler` (дата обр. 20.06.2016).

14 MetaFast: fast reference-free graph-based comparison of shotgun metagenomic data / V. I. Ulyantsev [и др.] // Bioinformatics. — 2016. — btw312.

15 *Melsted P.*, *Pritchard J. K.* Efficient counting of k-mers in DNA sequences using a bloom filter // BMC Bioinformatics. — 2011. — Т. 12, № 1. — С. 1.

16 *Chikhi R.*, *Limasset A.*, *Medvedev P.* Compacting de Bruijn graphs from sequencing data quickly and in low memory // Bioinformatics. — 2016. — Июнь. — Т. 32, № 12. — С. i201–i208.

17 Basic local alignment search tool / S. F. Altschul [и др.] // Journal of Molecular Biology. — 1990. — Т. 215, № 3. — С. 403–410.

18 The NCBI biosystems database / L. Y. Geer [и др.] // Nucleic Acids Research. — 2009. — gkp858.

19 Stepwise evolution of pandrug-resistance in Klebsiella pneumoniae / H. M. Zowawi [и др.] // Scientific reports. — 2015. — Т. 5.

20 *AMR Team R. on* Review on Antimicrobial Resistance. — 2015. — URL: `http://amr-review.org/` (дата обр. 20.06.2016).

ПРИЛОЖЕНИЕ А. ВИЗУАЛИЗАЦИЯ РАЗЛИЧИЙ В ГЕНОМНОМ ОКРУЖЕНИИ В ДВУХ РАЗЛИЧНЫХ ОБРАЗЦАХ

В этом приложении продемонстрирована возможность визуализации двух геномных окружений одновременно. Вершины с черной границей принадлежат обоим окружениям, вершины с красной или синей — ровно одному из этих двух.

В качестве окружений взяты геномные окружения гена OXA-347 в образцах, взятых через 2 и 28 дней после курса антибиотика соответственно.

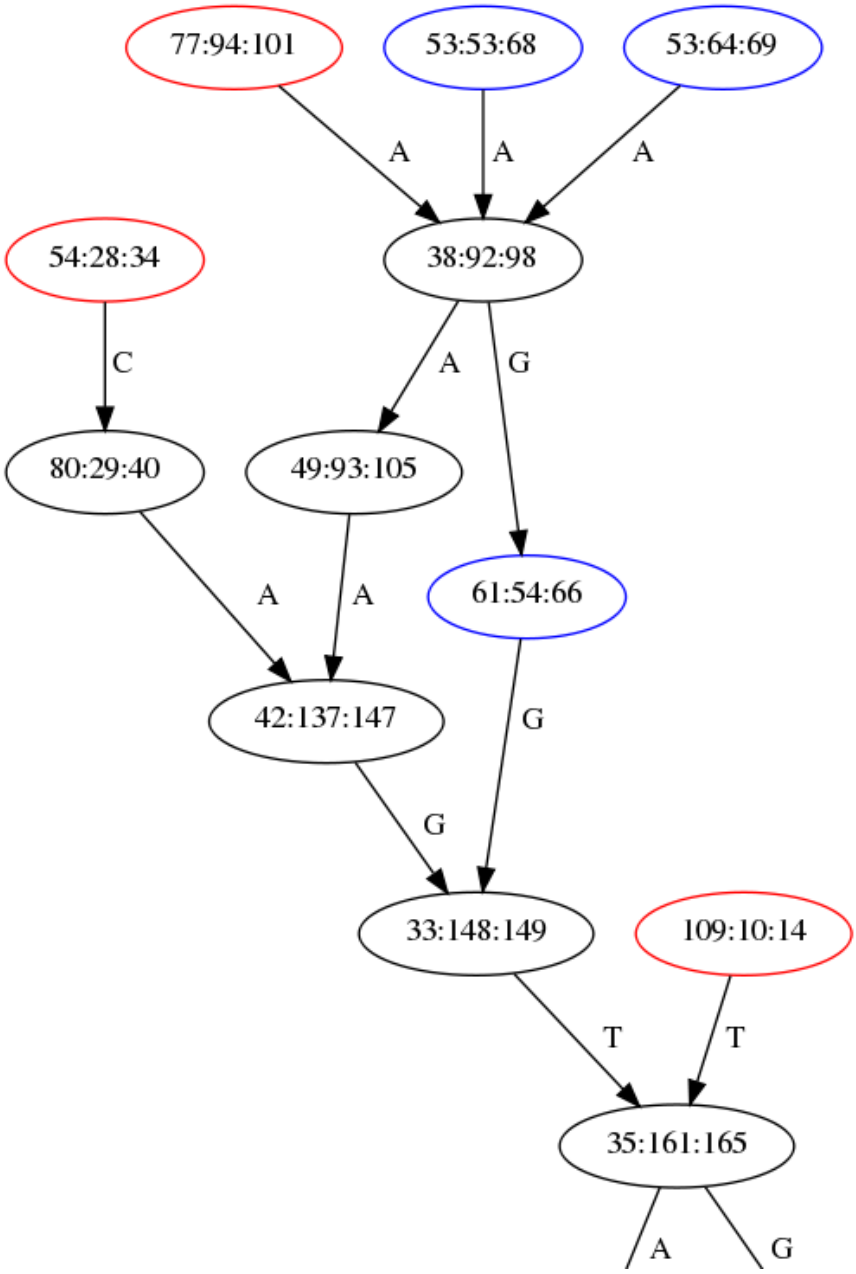

Рисунок А.1 – Фрагмент сдвоенного окружения №1