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

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к выпускной квалификационной работе

«Алгоритмы сравнительного анализа серий метагеномных образцов с использованием графов де Брейна для библиотек метагеномных чтений»

Автор: Иванов Артем Борисович _____
Направление подготовки (специальность): 01.03.02 Прикладная математика и информатика
Квалификация: Бакалавр
Руководитель: Ульянцев В.И., канд. техн. наук _____

К защите допустить
Руководитель ОП Парфенов В.Г., докт. техн. наук, проф. _____

«___» _____ 20___ г.

Санкт-Петербург, 2019 г.

Студент Иванов А.Б. Группа М3438 Факультет ИТиП

Направленность (профиль), специализация
Математические модели и алгоритмы в разработке программного обеспечения

Консультанты:

а) Олехнович Е.И., канд. биол. наук, без звания _____

ВКР принята «___» _____ 20___ г.

Оригинальность ВКР _____%

ВКР выполнена с оценкой _____

Дата защиты «___» _____ 20___ г.

Секретарь ГЭК *Павлова О.Н.* _____

Листов хранения _____

Демонстрационных материалов/Чертежей хранения _____

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

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

УТВЕРЖДАЮ
Руководитель ОП
докт. техн. наук, проф.
_____ Парфенов В.Г.
«___» _____ 20___ г.

ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ

Студент Иванов А.Б. **Группа** М3438 **Факультет** ИТиП
Руководитель Ульянцев В.И., канд. техн. наук, доцент ФИТиП
1 Наименование темы: Алгоритмы сравнительного анализа серий метагеномных образцов с использованием графов де Брейна для библиотек метагеномных чтений
Направление подготовки (специальность): 01.03.02 Прикладная математика и информатика
Направленность (профиль): Математические модели и алгоритмы в разработке программного обеспечения
Квалификация: Бакалавр

2 Срок сдачи студентом законченной работы: «___» _____ 20___ г.
3 Техническое задание и исходные данные к работе.

Требуется разработать и реализовать алгоритм, позволяющий проводить сравнительный анализ серий метагеномных образцов. Алгоритм должен разбивать входные образцы на категории на основании их близости для возможности их дальнейшего анализа и классификации. Отдельное внимание требуется уделить созданию правдоподобных тестовых метагеномов и валидации алгоритма на них.

4 Содержание выпускной квалификационной работы (перечень подлежащих разработке вопросов)

Описание состояния предметной области и существующих решений для обработки и анализа метагеномных данных. Разработка и реализация алгоритма, который позволит выявить схожие и различающиеся части в серии метагеномных образцов. Разработка тестовых образцов серий метагеномов и проверка на них работоспособности алгоритма.

5 Перечень графического материала (с указанием обязательного материала)

Не предусмотрено

6 Исходные материалы и пособия

1 Durable coexistence of donor and recipient strains after fecal microbiota transplantation / S. S. Li [et al.] // Science. — 2016. — Vol. 352, no. 6285. — P. 586–589.

2 MetaCherchant: analyzing genomic context of antibiotic resistance genes in gut microbiota / E. I. Olekhnovich [et al.] // Bioinformatics. — 2017. — Vol. 34, no. 3. — P. 434–444.

7 Дата выдачи задания: «___» _____ 20___ г.

Руководитель ВКР _____

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

Министерство науки и высшего образования Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

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

АННОТАЦИЯ
ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

Студент: Иванов Артем Борисович

Наименование темы ВКР: Алгоритмы сравнительного анализа серий метагеномных образцов с использованием графов де Брейна для библиотек метагеномных чтений

Наименование организации, где выполнена ВКР: Университет ИТМО

ХАРАКТЕРИСТИКА ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

1 Цель исследования: Разработка и реализация алгоритма, позволяющего проводить сравнительный анализ серий метагеномных образцов.

2 Задачи, решаемые в работе:

а) Разработка и реализация алгоритма обнаружения прочтений из одного метагенома в другом метагеноме.

б) Разработка и реализация классификатора прочтений для серии метагеномных образцов.

в) Тестирование нового алгоритма на сгенерированных и реальных метагеномных образцах.

3 Число источников, использованных при составлении обзора: 24

4 Полное число источников, использованных в работе: 32

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
1	0	0	20	8	3

6 Использование информационных ресурсов Internet: да, число ресурсов: 2

7 Использование современных пакетов компьютерных программ и технологий:

Пакеты компьютерных программ и технологий	**Параграф работы**
MetaCherchant	2.1, 2.2, 2.3
Среда разработки IntelliJ IDEA и язык программирования Java	2.1, 2.2, 2.3
python3 скрипты	2.1, 2.3, 3.1
Bash скрипты	2.4, 3.1
draw.io	2.4
MetaSim	3.1
BrownieAligner	3.2, 3.3
Commet	3.3
metaWRAP	3.3, 3.4
Kraken	3.3, 3.4, приложение Г
anvi'o	3.4
MetaFast	3.4
Mash	3.4
LaTeX	1, 2, 3

8 Краткая характеристика полученных результатов: Разработан и реализован новый алгоритм классификации прочтений из серий метагеномных образцов. Он разбивает прочтения на классы, которые можно использовать для более аккуратного по сравнению с исходными метагеномами анализа. Тестирование на сгенерированных метагеномных данных показало высокую точность классификации. Применение для анализа реальных метагеномных образцов показало результаты, согласующиеся с известными данными, и позволило сделать предположения о ранее не известных принципах изменения метагеномов.

9 Гранты, полученные при выполнении работы: Нет

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

1 *Иванов А. Б.* Алгоритмы сравнительного анализа серий метагеномных образцов с использованием графов де Брейна для библиотек метагеномных чтений // Сборник тезисов докладов конгресса молодых ученых. — Электронное издание. СПб: Университет ИТМО, 2019.

Студент Иванов А.Б. _____

Руководитель ВКР Ульянцев В.И. _____

«__» _____ 20__ г.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Наследственная информация о живом организме, необходимая для поддержания жизнеспособности, хранится в его геноме. Геном человека и других клеточных организмов хранится в виде пар нуклеотидов, образующих ДНК. Для изучения ДНК отдельного организма можно использовать образцы его генетического материала. В то же время существуют среды, которые одновременно населяют сотни и тысячи различных видов. Сложность их анализа состоит в том, что некоторые виды являются некультивируемыми, то есть они не способны расти и размножаться вне среды обитания. Это ставит перед исследователями задачу совместного изучения набора организмов, населяющих определенную среду – микробиом.

Исследование микробиоты кишечника человека является важной задачей, так как позволяет выявить зависимости между бактериями живущими в организме человека и его состоянием. Появление методов секвенирования нового поколения [1] способствовало бурному росту исследований в этой области. Они позволили классифицировать множество бактерий, населяющих кишечник человека, а также исследовать заболевания вызываемые этими бактериями.

Для борьбы с патогенными бактериями, как правило, применяются антибиотики. Однако, благодаря высокой степени мутации и горизонтальному переносу генов, появляются новые виды бактерий, устойчивые к антибиотикам. Они не поддаются лечению лекарственными средствами, что приводит к необходимости поиска альтернативных методов лечения. Одним из таких методов является трансплантация фекальной микробиоты (ТФМ), то есть пересадка кишечных бактерий от здорового человека к больному. Данный метод показывает высокую эффективность для борьбы с определенными видами бактерий (например, *Clostridium difficile*), однако принцип приживаемости бактерий и последующих изменений в метагеноме реципента остается до конца не исследованным [2].

Для изучения метагеномов, в том числе микробиоты кишечника человека, существуют различные методы. Одним из подходов является таксономическая аннотация организмов, содержащихся в метагеноме. Алгоритмы [3–6] позволяют с высокой точностью определить виды в средах с небольшим числом хорошо изученных организмов. Однако, микробиота кишечника

является средой с большим количеством бактерий, многие из которых мало изучены, поэтому данные алгоритмы не позволяют провести исчерпывающую классификацию метагенома.

Другой подход состоит в сборке длинных цепочек (контигов) или даже отдельных геномов (англ. Metagenome-Assembled Genomes) из коротких прочтений, которые затем могут таксономически аннотироваться или использоваться как самостоятельные единицы для сравнения метагеномов. Тем не менее, данный подход также не является надежным в связи с огромным числом прочтений в метагеномах. При построении длинных участков неизбежно будут появляться разветвления или связи между двумя участками за счет прочтений из других организмов. Правильное разрешение всех проблемных случаев и выделение отдельных цепочек является сложной задачей, а также ведет к потере информации, содержащейся в отброшенных прочтениях. Поэтому данный метод классификации также не может считаться достаточно надежным.

Интерес в изучении серий метагеномов и недостатки в существующих алгоритмах приводят к необходимости разработки новых методов для анализа серий метагеномных образцов. Решения, предлагаемые в данной работе, позволят классифицировать каждое прочтение из метагенома, тем самым сохранив всю исходную информацию. Разбиение прочтений на классы позволит уменьшить количество прочтений и видов для изучения в каждом классе, что упростит использование и улучшит качество указанных выше методов. Все это расширит возможности для анализа метагеномов и серий метагеномов и поможет в исследовании принципов изменения микробиоты после фекальной трансплантации.

ГЛАВА 1. ОБЗОР СОВРЕМЕННЫХ РЕЗУЛЬТАТОВ В ОБЛАСТИ МЕТАГЕНОМИКИ

Метагеномные данные это совокупность информации о живых организмах (например прокариотах – бактериях, археях; эукариотах – грибах) и вирусах (например – бактериофагах), обитающих в образце, взятом из определенной среды. Примерами сред для взятия образцов могут служить почва, вода из открытых водоемов, кишечник человека и др. Метагеномика исследует тотальную ДНК, выделенную из образца для определения таксономического состава среды и раскрытия его функционального потенциала. Активное развитие метагеномики стало возможным благодаря развитию относительно дешевых методов секвенирования нового поколения [1], наиболее популярные из которых рассмотрены ниже.

1.1. Методы секвенирования нового поколения

Среди методов секвенирования нового поколения выделяют две большие категории. Методы второго поколения позволяют получать короткие прочтения, в то время как методы третьего поколения могут генерировать гораздо более длинные прочтения. Ко второму поколению относятся пиросеквенирование и секвенирование с помощью синтеза, а к третьему поколению одномолекулярное и нанопоровое секвенирование.

1.1.1. Пиросеквенирование (Roche/454 Life Sciences)

Метод основан на идее обнаружения пирофосфата [1, 7]. Закрепленные на ячеистой подложке последовательности ДНК пытаются удлинить, поочередно вводя один из четырех нуклеотидов. Если нуклеотид присоединяется с помощью полимеразы к существующей цепи, то происходит выделение пирофосфата, который преобразуется в свет различной интенсивности путем химической реакции. Далее побочные продукты реакции вымываются и цикл повторяется. На основании зафиксированных световых сигналов реконструируется последовательность нуклеотидов, интенсивность сигнала соответствует количеству одинаковых присоединенных за один цикл нуклеотидов.

1.1.2. Секвенирование с помощью синтеза (Illumina)

Данный метод основан на регистрации флюоресцентной метки при присоединении нуклеотида к цепи ДНК [1, 8]. В ячейку с цепью ДНК

добавляются четыре нуклеотида (A, G, C, T), модифицированные так, что каждый имеет флюоресцентную метку своего цвета и возможно присоединение не более одного из них. После присоединения полимеразой одного из нуклеотидов остальные вымываются. Ячейка освещается лазером и камера фиксирует цвет присоединенного нуклеотида. Затем флюорофор вымывается и цикл повторяется.

1.1.3. Одномолекулярное секвенирование в реальном времени (Pacific Biosciences)

Данный метод секвенирования основан на наблюдении за работой полимеразы, достраивающей комплементарную цепь ДНК в реальном времени. Отдельные молекулы ДНК-полимеразы прикрепляются ко дну ячеек волноводной наноструктуры, которые позволяют зафиксировать свет, выделяющийся при встраивании нового нуклеотида. Каждому нуклеотиду сопоставлена своя флуоресцентная метка, прикрепленная к терминальной фосфатной группе. При прикреплении нуклеотида данная метка отрывается и не влияет на работу полимеразы, что позволяет наблюдать за синтезом большого числа оснований без помех и получать длинные прочтения. При пятнадцатикратном повторении секвенирования одной молекулы была получена точность 99,3% [9].

1.1.4. Нанопоровое секвенирование (Oxford Nanopore)

Молекулы ДНК помещаются в раствор электролита, который отделен перегородкой с нанопорой от второй части резервуара. Экзонуклеаза отщепляет отдельные нуклеотиды от одной молекулы ДНК, которые затем под действием поданного напряжения проходят через пору, где определяется каждое основание. Нуклеотиды имеют высокую вероятность прохода через нанопору, поэтому вероятность повторной регистрации одного и того же нуклеотида мала. В статье [10] показано, что точность определения оснований составляет в среднем 99,8%. Преимуществом данного подхода является тот факт, что нанопоровое секвенирование не требует флюоресцентных отметок для нуклеотидов, что может увеличить скорость секвенирования и снизить его стоимость.

1.2. Информация о качестве прочтений

При секвенировании метагеномов неизбежно возникают ошибки в получаемых прочтениях, однако современные секвенаторы (например, illumina [11]) позволяют получить информацию о качестве каждого основания в прочтении. Качество основания определяется с помощью шкалы Phred и равно $Q = -10 \log_{10} P$, где P – вероятность ошибки. Пример связи между вероятностью ошибки и качеством основания приведен в таблице 1. Качество $Q30$ является де-факто стандартом в методах секвенирования нового поколения, так как практически все прочтения длины 100 не будут содержать ошибок.

Таблица 1 – Оценка качества и точность оснований

Оценка качества	Точность основания, %	Вероятность неверного основания
$Q10$	90,00	1 на 10
$Q20$	99,00	1 на 100
$Q30$	99,90	1 на 1000
$Q40$	99,99	1 на 10000

Данная информация сохраняется в файле формата fastq, где каждому прочтению соответствует строка с закодированными качествами полученных оснований. Наиболее популярными для кодирования являются алгоритмы Phred+33 и Phred+64. К каждому численному значению качества Q прибавляется число 33 или 64 соответственно, и затем полученное число интерпретируется как символ с соответствующим номером в таблице ASCII. Таким образом информация о качестве каждого прочтения может быть использована при дальнейшей их обработке.

1.3. Алгоритмы для метагеномной сборки

Одним из подходов к изучению метагеномов является метагеномная сборка, которая позволяет выделять длинные последовательности нуклеотидов (контиги), которые могут рассматриваться как черновые геномы или части геномов организмов, обитающих в исследуемой среде. Анализ полученных последовательностей используется для воспроизведения геномов ранее не изученных видов и открывает возможности для пониманию различных сообществ микроорганизмов.

Одним из эффективных алгоритмов, решающих данную задачу, является MEGAHIT [12]. Принцип его работы основан на построении и

использовании сжатого графа де Брейна, что позволило значительно сократить вычислительные требования, необходимые ранее. Алгоритм последовательно строит граф для различных значений k, тем самым отфильтровывая ошибки и получая информацию из слабо покрытых регионов при малых значениях параметра, и разрешая повторы при больших значениях параметра. Также алгоритм может использовать графический процессор для ускорения вычислений в 3-5 раз.

Новым решением в этой области стал алгоритм metaSPAdes [13]. Он использует различные стратегии по разрешению сложных случаев в графе, таких как тупиковые ответвления или пузыри. Также, для определения продолжения пути на развилках используется предпросмотр вперед по всем разветвлениям. Результаты работы алгоритма на синтетических и реальных метагеномных данных различной сложности превзошли результаты ранних алгоритмов, в том числе MEGAHIT.

1.4. Алгоритмы для таксономической аннотации метагеномов

Для определения таксономического состава метагеномов используются разные стратегии. Одна из них это выравнивание прочтений на референсную базу для выявления схожих последовательностей [3]. Алгоритм Kraken использует точное выравнивание k-меров из прочтений на референсную базу. В случае обнаружения нескольких подходящих организмов алгоритм возвращает наименьшего общего таксономического предка, лучше всего классифицирующего данное прочтение. Однако 68,2 % прочтений из образцов слюны участников проекта «Микробиом человека» (англ. Human Microbiome Project) не были классифицированы, так как они не относились ни к одному из известных видов, содержащихся в базе данных. Другим известным алгоритмом локального выравнивания является BLAST [4]. Алгоритм составляет матрицу оценок для возможных замен и затем ищет локальные выравнивания последовательности на базу данных с максимальной оценкой похожести.

Однако с ростом размера базы данных нуклеотидных последовательностей и числом прочтений от секвенаторов в метагеномных исследованиях, такое выравнивание требует слишком больших вычислительных ресурсов. Для борьбы с этой проблемой был разработан алгоритм CensuScope [5], который выбирает подмножество из входных прочтений и на основе их таксономической аннотации делает выводы о

метагеноме в целом. Это позволяет значительно ускорить таксономическую аннотацию, сохранив при этом высокий процент достоверности результатов.

Алгоритм Centrifuge [6] показывает качество таксономической классификации сравнимое с результатами алгоритмов MegaBlast (оптимизированная версия BLAST) и Kraken, однако работает значительно быстрее первого и требует меньшего объема оперативной памяти, чем второй, что позволяет использовать его на персональных компьютерах. Маленький размер референсной базы достигается за счет отбрасывания частей геномов, совпадающих более чем на 99 % между штаммами одного вида, что может приводить к ошибочной классификации на штаммовом уровне. При поиске прочтения оно выравнивается без ошибок на базу и в качестве результата выдается набор организмов, которыму лучше всего соответствует прочтение на основании суммы квадратов длин найденных участков.

Использование референсных баз данных хорошо работает при анализе прочтений от одного организма, но дает неудовлетворительные результаты при анализе метагеномных данных из ранее неисследованных сред в связи с высокой вариативностью геномов бактерий и недостатком референсных геномов. Так в [14] было показано, что более половины геномов могут отсутствовать в референсных базах данных, что делает их применение для определения изменений в метагеноме ненадежным. Поэтому в [14] для определения видового разнообразия использовались уникальные маркерные гены. С помощью них строилось филогенетическое дерево, в которое попадали почти 95 % видов из метагенома как с референсом, так и без. Тем не менее такой подход не позволяет провести границы между различными штаммами, что ограничивает применимость данного подхода в случае необходимости внутривидовой дифференциации.

Еще один подход предложен в статье [15]. Авторы алгоритма MetaPhlAn отобрали подмножество специфичных маркерных генов, которые уникально характеризуют различные таксономические категории. Далее все прочтения выравниваются на полученную базу с помощью программы BLAST, однако время работы уменьшилось почти на два порядка по сравнению с оригинальным алгоритмом BLAST, преимущественно из-за малого количества референсных последовательностей. Алгоритм MetaPhlAn2 [16] является улучшением алгоритма MetaPhlAn. База маркерных генов значительно расширилась в связи

с десятикратным увеличением числа секвенированных геномов с момента первой публикации, а скорость выросла в 10 раз. Также алгоритм показал более высокую точность по сравнению с Kraken.

Другой стратегией является сравнение метагеномов без использования эталонных последовательностей [17, 18]. В crAss [18] из всех прочтений из входных метагеномов собираются контиги, которые интерпретируются как единицы информации о метагеномах. Далее используются различные метрики, которые по содержанию и покрытию контигов определяют близость метагеномов. Недостатком этого метода является тот факт, что классификатор работает с контигами, при сборке которых может теряться информация о небольших, но значимых перестройках в геномах, а также могут образовываться химерные контиги. Улучшение этого подхода представлено в алгоритме MetaFast [17]. Он строит граф де Брейна и собирает контиги для каждого метагенома раздельно, что уменьшает вычислительную сложность алгоритма. Затем контиги собираются в один граф, из которого выделяются компоненты, на основании покрытия k-мерами которых определяется близость метагеномов.

1.5. Алгоритмы для анализа метагеномов

Выделение генома отдельных организмов из метагенома позволяет проводить анализ на уровне одного генома. Это позволяет изучать метагеномы, состоящие из некультивированных организмов, для которых таксономический анализ не дает положительного результата. Процесс изучения метагенома состоит в выделении из него черновых геномов отдельных организмов (корзин, англ. bin) и дальнейшем изучении корзин по отдельности. Для разбиения метагенома на корзины существует программа metaWRAP [19]. Она состоит из набора модулей, которые позволяют делать сборку метагенома, разбиение на корзины, уточнение корзин, а также их дальнейшее изучение, включая оценку полноты, функциональную аннотацию и визуализацию. Ключевым улучшением по сравнению с более ранними методами разбиения на корзины является процесс уточнения корзин. Алгоритм принимает на вход корзины, полученные от различных программ (по-умолчанию используются CONCOCT, MaxBin и metaBAT), генерирует гибридные корзины, производит их попарное сравнение и выбирает лучшие корзины для результирующего множества.

Сравнение корзин производится при помощи программы CheckM [20]. Она позволяет оценить полноту корзины, то есть процент покрытия генома

в данной корзине, и загрязнение, то есть количество последовательностей в корзине, которые не принадлежат к данному геному. Анализ производится на основе уникальных маркерных генов, сгруппированных на основе их постоянного объединения в геномах. Также оценка может быть улучшена с помощью анализа маркерных генов, специфичных для определенной таксономической группы.

Другим удобным средством для анализа и визуализации метагеномных данных является программа anvi'o [21]. Она предоставляет возможность объединения различных данных (таких как покрытие контигов прочтениями, соотношение GC нуклеотидов, таксономию, разбиение на корзины) из нескольких метагеномов для их отображения на одном графике. Вся информация организована вокруг структуры дерева, построенного из входных контигов, и позволяет наглядно отобразить различную информацию о метагеномах для выявления зависимостей как между различными параметрами, так и между различными метагеномами. Кроме того, данная программа позволяет интерактивно разбивать контиги на корзины путем выделения их в дереве и автоматически обновляет информацию о полноте и загрязнении корзин. Это способствует более аккуратному, в отличие от автоматических методов, разбиению на корзины в сложных случаях.

1.6. Алгоритмы для изучения трансплантации фекальной микробиоты

Трансплантация фекальной микробиоты (ТФМ, англ. Fecal microbiota transplant, FMT) это процедура переноса кишечных бактерий, содержащихся в кале здорового человека – донора, реципиенту с каким-либо заболеванием (рис. 1). Она рекомендуется пациентам при рецидивах кишечных инфекций, не поддающихся лечению антибиотиками. В работе [22] было показано, что ТФМ является эффективным способом лечения для рецидивирующих случаев инфекции *Clostridium difficile*.

В исследовании [2] было обнаружено, что донорские бактерии приживаются у реципиента и обнаруживаются через три месяца после трансплантации. Там же было отмечено, что при трансплантации от одного донора разным реципиентам уровень приживаемости различался от 12 % до 56 %, что приводит к необходимости разработки методов детального анализа приживающихся бактерий и персональному подбору донора в зависимости от заболевания и текущего метагенома кишечника реципиента.

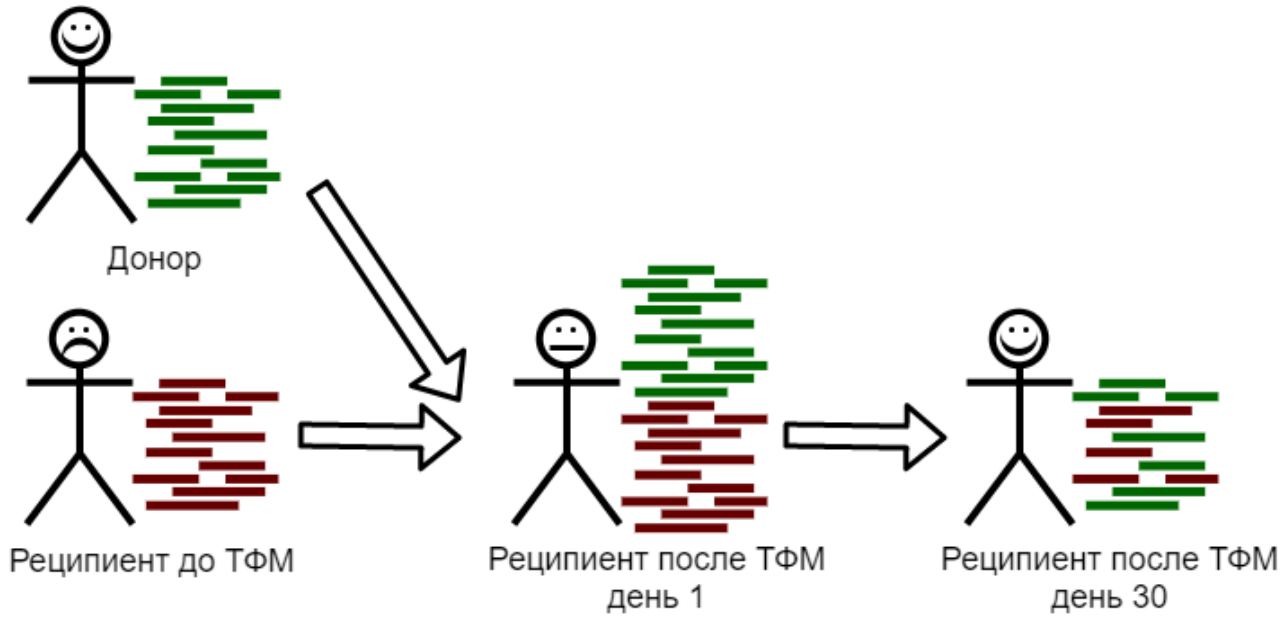

Рисунок 1 – Схема процедуры трансплантации фекальной микробиоты.

Для определения изменений в метагеноме реципиента использовался метод таксономических маркеров генов, описанный в [14], однако он не позволяет выявить события штаммового замещения, которые являются распространенными при ТФМ (см. [2]).

В работе [23] для определения приживаемости донорских бактерий у реципиента был использован другой подход. Для метагенома донора производилась сборка отдельных геномов (англ. Metagenome-Assembled Genomes – MAGs), на которые затем выравнивались короткие прочтения реципиента и принималось решение о приживаемости того или иного собранного генома. Была обнаружена положительная связь между приживаемостью бактерий и их нахождением у участников эксперимента «Микробиом человека» (англ. Human Microbiome Project). Однако при сборке геномов теряется значительное количество информации об отдельных прочтениях, а также данный подход не позволяет выявить различия между штаммами в метагеномах донора и реципиента.

Альтернативное решение для выявления штаммовых различий было предложено в работе [24]. Метод основан на анализе частоты встречаемости однонуклеотидных полиморфизмов. Алгоритм Strain Finder определяет генотипы штаммов путем максимизации функции правдоподобия с использованием алгоритма машинного обучения Expectation-Maximization (EM). Недостатками данного подхода являются необходимость

наличия референсного генома для поиска однонуклеотидных полиморфизмов и сходимость EM алгоритма к локальному оптимуму, что требует многократных запусков для аппроксимации глобального решения.

1.7. Постановка цели и задач ВКР

Целью работы является разработка и реализация алгоритма для проведения сравнительного анализа серий метагеномных образцов. Недостатками существующих алгоритмов является сборка контигов из метагеномов, что приводит к потере информации о части прочтений, содержащихся в метагеномах. Новый алгоритм должен позволить сохранить информацию о каждом прочтении из метагенома, тем самым повысив качество классификации.

Для достижения поставленной цели необходимо решить следующие задачи:

а) Разработка и реализация алгоритма обнаружения прочтений из одного метагенома в другом метагеноме.
б) Разработка и реализация классификатора прочтений для серии метагеномных образцов.
в) Тестирование нового алгоритма на сгенерированных и реальных метагеномных образцах.

Выводы по главе 1

В данной главе проведен обзор современных подходов, используемых для решения задач метагеномики. Введено понятие процедуры фекальной трансплантации и описаны существующие методы для идентификации приживаемости донорских бактерий. Отмечены недостатки этих методов, которые приводят к необходимости поиска новых решений.

ГЛАВА 2. ОПИСАНИЕ РАЗРАБОТАННЫХ АЛГОРИТМОВ

Для разработки нового алгоритма сравнительного анализа серий метагеномных образцов, который позволит получить информацию о каждом прочтении, необходимо решить задачу классификации отдельных прочтений из метагенома. Данная задача была решена с использованием графа де Брейна, реализованного в программе MetaCherchant [25].

2.1. Алгоритм обнаружения прочтений из одного метагенома в другом метагеноме

На вход данному алгоритму подается два метагенома в виде отдельных прочтений формата fasta/fastq. Метагеном в котором будет производиться поиск прочтений назовем источником. Он может состоять как из парных, так и из одноконцевых прочтений. Метагеном прочтения из которого будут анализироваться на вхождение в источник назовем анализируемым. Анализируемый метагеном должен быть задан парноконцевыми прочтениями. Этот факт будет использован в алгоритме обнаружения. Также обязательным входным параметром является число k – длина k-меров, на которые будут разбиты входные прочтения. Псевдокод алгоритма обнаружения прочтений представлен на листинге 1.

Листинг 1 – Псевдокод алгоритма обнаружения прочтений.

```
function ClassifyReads(k, inputFiles, readFiles)
    graph = BuildDeBruijnGraph(k, inputFiles)
    for ⟨forwardRead, backwardRead⟩ ← readFiles do
        forwardIsFound = FindRead(k, forwardRead)
        backwardIsFound = FindRead(k, backwardRead)
        if forwardIsFound = true & backwardIsFound = true then
            bothFound.append(⟨forwardRead, backwardRead⟩)
        else if forwardIsFound = true & backwardIsFound = false then
            forwardFound.append(⟨forwardRead, backwardRead⟩)
        else if forwardIsFound = false & backwardIsFound = true then
            backwardFound.append(⟨forwardRead, backwardRead⟩)
        else
            notFound.append(⟨forwardRead, backwardRead⟩)
        end if
    end for
    PrintClassifiedReads
end function
```

На первом шаге алгоритма из прочтений источника строится граф де Брейна с размером вершин k с помощью программы MetaCherchant.

Листинг 2 – Псевдокод поиска прочтения в графе де Брейна.

```
function FindRead(k, read)
    for kmer ← read do
        coverage.append(graph.get(kmer))
    end for
    if Criteria(coverage) = true then
        return true
    else
        return false
    end if
end function
```

На втором шаге алгоритма обрабатываются анализируемые прочтения. Последовательно обрабатываются по два прочтения, составляющих одно парноконцевое прочтение. Производится поиск каждого прочтения из пары в построенном графе де Брейна (листинг 2). По результатам поиска каждому прочтению присваивается статус обнаружен или не обнаружен. Всего для пары прочтений возможны четыре ситуации:

а) Оба прочтения обнаружены. В данном случае пара считается обнаруженной и прочтения записываются в выходной файл с обнаруженными парноконцевыми прочтениями (рис. 2).

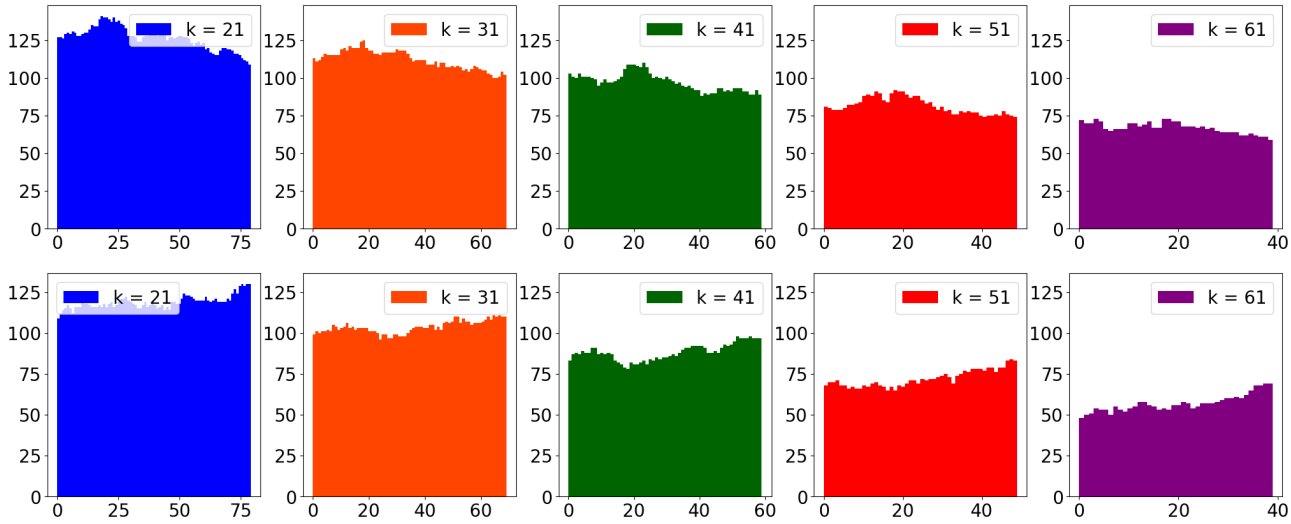

Рисунок 2 – Пример глубины покрытия парноконцевого прочтения k-мерами при различных значениях k. Оба прочтения обнаружены.

б) Оба прочтения не обнаружены. В данном случае пара считается не обнаруженной и прочтения записываются в выходной файл с не обнаруженными парноконцевыми прочтениями.

в) Первое прочтение обнаружено, а второе не обнаружено. В данном случае парноконцевое прочтение не может быть классифицировано, поэтому оно разбивается на два одноконцевых прочтения. Обнаруженное прочтение записывается в выходной файл с обнаруженными одноконцевыми прочтениями. Не обнаруженное прочтение записывается в выходной файл с не обнаруженными одноконцевыми прочтениями.

г) Первое прочтение не обнаружено, а второе обнаружено. Данный случай симметричен случаю *в* и обрабатывается аналогично (рис. 3).

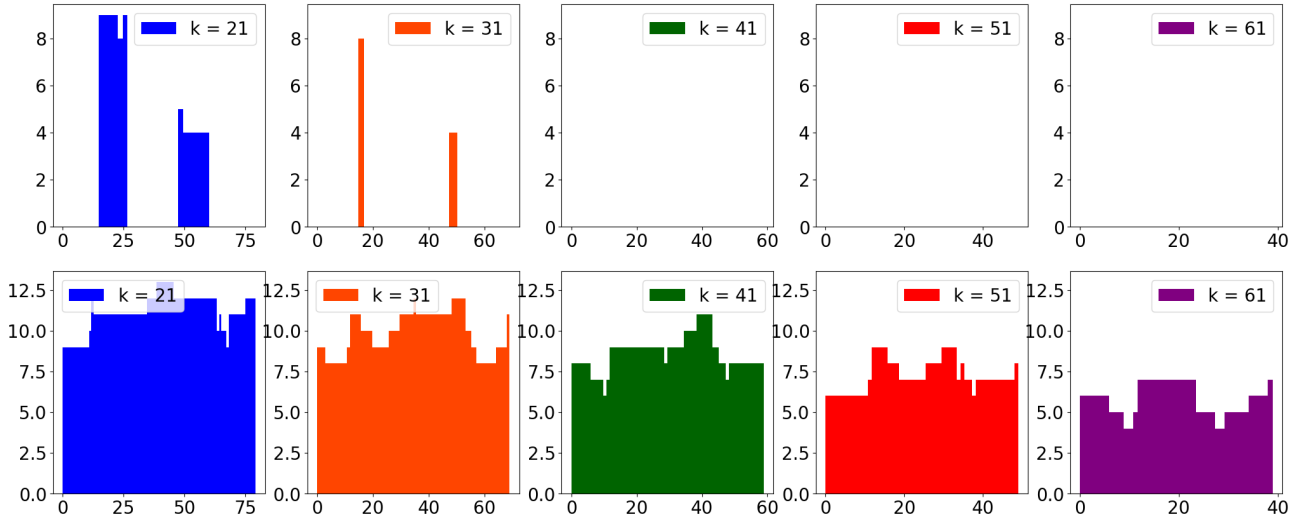

Рисунок 3 – Пример глубины покрытия парноконцевого прочтения k-мерами при различных значениях k. Обнаружено только второе прочтение.

Функция поиска одного прочтения в графе де Брейна работает следующим образом. Анализируемое прочтение разбивается на k-меры и каждому k-меру сопоставляется целое число – количество раз, которое он встретился в графе. По полученному списку чисел вычисляется их среднее, которое является средней глубиной покрытия прочтения k-мерами в графе. Также вычисляется ширина покрытия breadth – отношение количества нуклеотидов в прочтении покрытых хотя бы одним k-мером к длине прочтения.

Из найденного значения средней глубины покрытия вычисляется теоретическая ширина покрытия по формуле

$$\text{theoryBreadth} = 1 - e^{-\text{meanCoverage}}.$$

Предположим, что глубина покрытия одного нуклеотида в прочтении k-мерами подчиняется распределению Пуассона с функцией вероятности

$$p(n) = \frac{\lambda^n}{n!} e^{-\lambda}.$$

Для вычисления части прочтения с нулевой глубиной покрытия положим ожидаемое значение глубины покрытия λ равным средней глубине покрытия meanCoverage и количество событий покрытия нуклеотида k-мером $n = 0$. Тогда вероятность того, что нуклеотид не будет покрыт ни один раз равна

$$p(0) = e^{-\text{meanCoverage}}$$

и для достаточно длинного прочтения часть непокрытых нуклеотидов будет равна этому же значению. Тогда ширина покрытия равна разности единицы и непокрытой части прочтения, откуда получаем искомую формулу.

Для классификации прочтения теоретическая ширина покрытия сравнивается с реальной. Если разность попадает в доверительный интервал и при этом ширина покрытия прочтения breadth $\geqslant 0{,}9$, то прочтение классифицируется как обнаруженное, иначе как не обнаруженное. Для вычисления ширины доверительного интервала используется приближение, основанное на центральной предельной теореме. Распределение отклонения полученной ширины покрытия от теоретической аппроксимируется с помощью стандартного нормального распределения, откуда получается предельная оценка на вероятность отклонения

$$P\left(\left|\frac{\text{breadth} - \text{theoryBreadth}}{\sigma(X)}\sqrt{N}\right| < T\right) \xrightarrow[N\to\infty]{} \gamma,$$

где $\sigma(X) = \sqrt{(1 - p(0)) \cdot p(0)}$ – стандартное отклонение случайной величины, показывающей что один нуклеотид в прочтении покрыт, а T – квантиль стандартного нормального распределения, то есть

$$F_{N(0,1)}(T) = \frac{1+\gamma}{2}.$$

Тогда можно оценить ширину интервала, в который попадет реальная ширина покрытия с фиксированной вероятностью. Величина breadth принадлежит

интервалу $(\text{theoryBreadth} - \delta, \text{theoryBreadth} + \delta)$ с вероятностью γ при $\delta = \frac{T \cdot \sigma(X)}{\sqrt{N}}$. При вероятности $\gamma = 0{,}95$ получается $T = 1{,}96$. $T = 1$ соответствует вероятности $0{,}68$.

В результате работы данного алгоритма на двух метагеномах на выходе генерируются шесть файлов с прочтениями из входных анализируемых прочтений. В двух файлах содержится информация о парноконцевых прочтениях, классифицированных как обнаруженные. В двух файлах содержится информация о парноконцевых прочтениях, классифицированных как не обнаруженные. В одном файле содержится информация об одноконцевых прочтениях, классифицированных как обнаруженные. В одном файле содержится информация об одноконцевых прочтениях, классифицированных как не обнаруженные. Данный алгоритм реализован в виде класса в программе Metacherchant на языке программирования Java, исходный код доступен по ссылке `https://github.com/ivartb/metacherchant`.

2.2. Улучшение алгоритма обнаружения прочтений для прочтений с ошибками

Для улучшения работы алгоритма используется информация о качестве прочтений, описанная в разделе 1.2. На шаге поиска прочтения в графе де Брейна перед разбиением его на k-меры производится подсчет неверных оснований в прочтении. Основание считается ошибочным, если его оценка качества меньше 10. Если в прочтении все основания верные, то оно обрабатывается исходным алгоритмом без изменений. Если в прочтении более одной ошибки, то оно также обрабатывается исходным алгоритмом, так как вероятность такого события ничтожно мала и оно не внесет искажений в результат классификации. В случае одной ошибки в прочтении программа стремится определить правильный нуклеотид на ошибочной позиции путем поочередной подстановки всех четырех нуклеотидов и поиска нового полученного прочтения в графе де Брейна. При первой классификации прочтения как обнаруженного перебор нуклеотидов останавливается, и прочтение заменяется на обнаруженное. Таким образом, алгоритм стремится исправить прочтения с одной ошибкой, опираясь на предположение о близости метагеномов и присутствии прочтения в графе де Брейна.

2.3. Модификация алгоритма обнаружения прочтений

Алгоритм, описанный в разделе 2.1 имеет два недостатка. Во-первых, для классификации пользователь должен дать на вход алгоритму число k – длину k-меров, на которые будут разбиваться входные прочтения. В зависимости от его численного значения результаты классификации могут сильно различаться. Так, при выборе слишком малого значения параметра вероятность встретить любой k-мер становится достаточно высокой, что приводит к образованию в графе де Брейна химерных связей и ошибочной классификации прочтения как найденного. Наоборот, при выборе слишком большого значения данного параметра, алгоритм не сможет распознавать близкие прочтения, отличающиеся на один или несколько нуклеотидов, и, следовательно, не сможет классифицировать прочтения с небольшим количеством ошибок. Во-вторых, критерий обнаружения прочтений является достаточно жестким и разбивает множество прочтений на две непересекающихся класса: обнаружено и не обнаружено. В то же время, хочется иметь возможность получать информацию о близости некоторых прочтений. Например, для прочтений, которые отличаются на небольшое число нуклеотидов, можно сделать предположение, что они пришли из двух различных штаммов одного вида или подверглись мутации. Отслеживание таких событий является важной задачей при исследовании процедуры ТФМ, поэтому было принято решение доработать существующий алгоритм с учетом вышеуказанных замечаний.

Для решения второй проблемы, критерий обнаружения прочтения в графе был модифицирован следующим образом:

а) Если прочтение удовлетворяет критерию из оригинального алгоритма, то оно попадает в категорию *обнаружено*.

б) Если прочтение не попало в категорию *обнаружено*, но при этом его ширина лежит в доверительном интервале или оно покрыто на ширину breadth $\geqslant 0{,}4$, то прочтение классифицируется как *частично обнаружено*. В данную категорию должны попасть все прочтения, близкие к прочтениям в другом метагеноме (то есть возникшие в результате ошибок секвенирования или пришедшие из штамма близкородственной бактерии).

в) Все оставшиеся прочтения попадают в категорию *не обнаружено*.

Иллюстрация классификации прочтений в зависимости от глубины и ширины его покрытия представлена на рисунке 4.

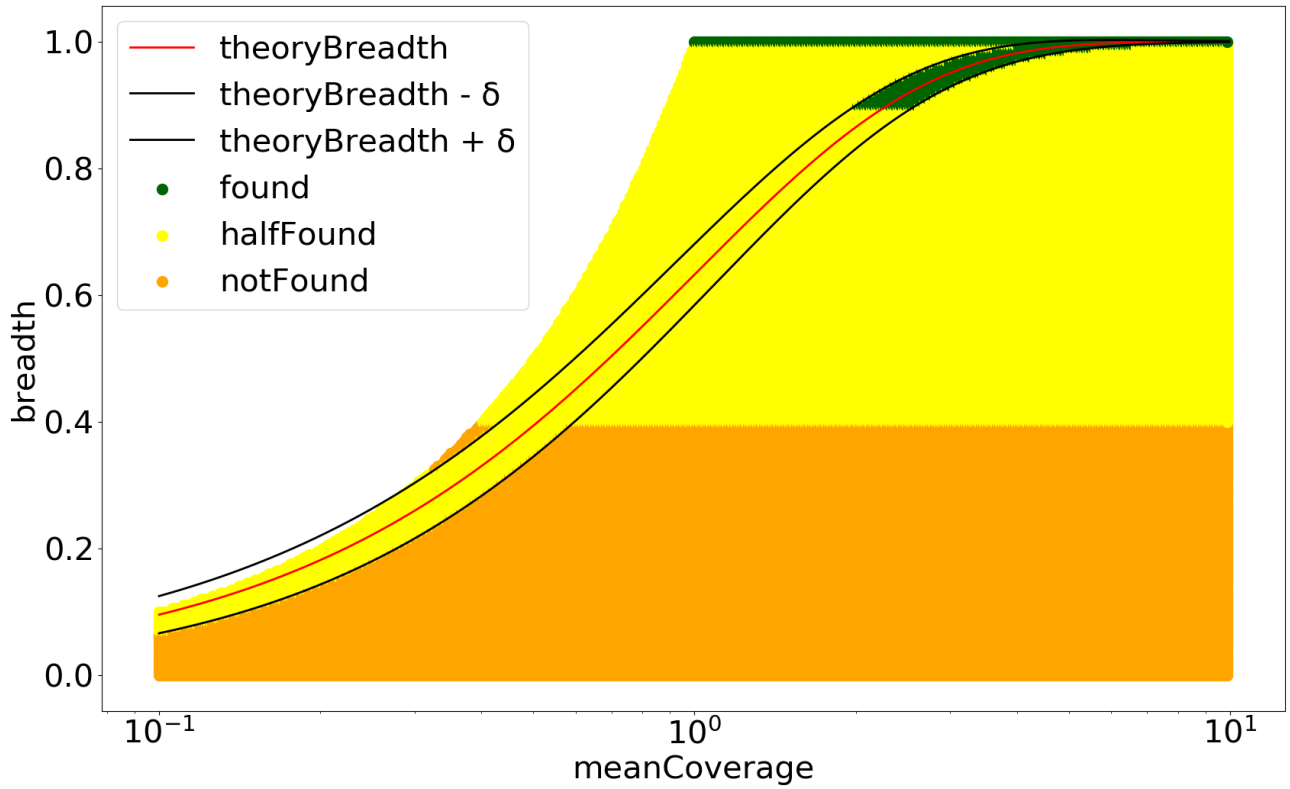

Рисунок 4 – Пример классификации прочтения в зависимости от его глубины и ширины покрытия. Зеленая область – прочтение обнаружено, желтая область – прочтение частично обнаружено, оранжевая область – прочтение не обнаружено.

Для решения первой проблемы в алгоритм была добавлена возможность поиска прочтения при двух различных параметрах k. Для каждого из них будет построен граф де Брейна и выполнен поиск прочтения в нем при заданном значении.

В результате для одного прочтения возможны следующие варианты:

а) Прочтение классифицируется как обнаруженное, если оно обнаружено при обоих значениях параметра k.

б) Прочтение классифицируется как частично обнаруженное, если выполняется одно из двух условий:

   1) Прочтение обнаружено при одном значении k (рис. 5).
   2) Прочтение частично обнаружено при обоих значениях параметра k (рис. 6).

в) В противном случае прочтение классифицируется как не обнаруженное.

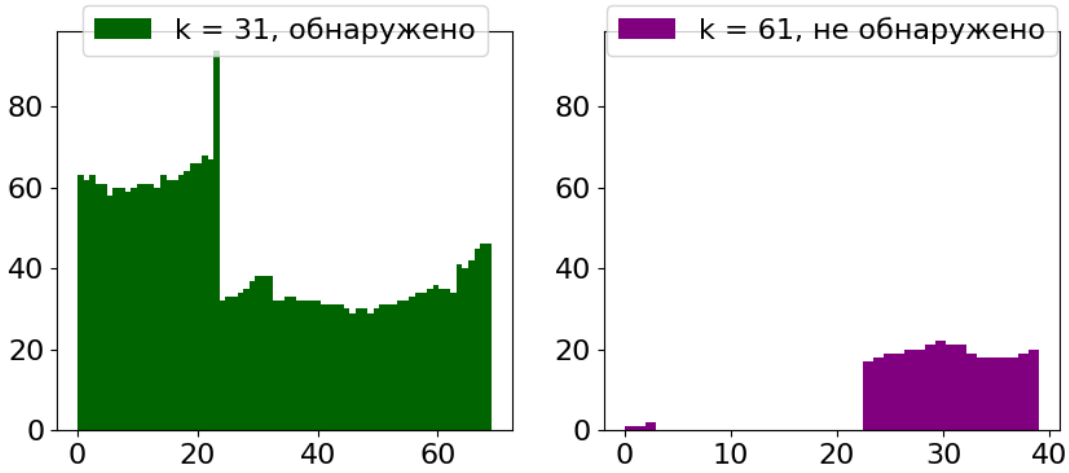

Рисунок 5 – Пример классификации прочтения как частично обнаруженного при обнаружении при одном значении k.

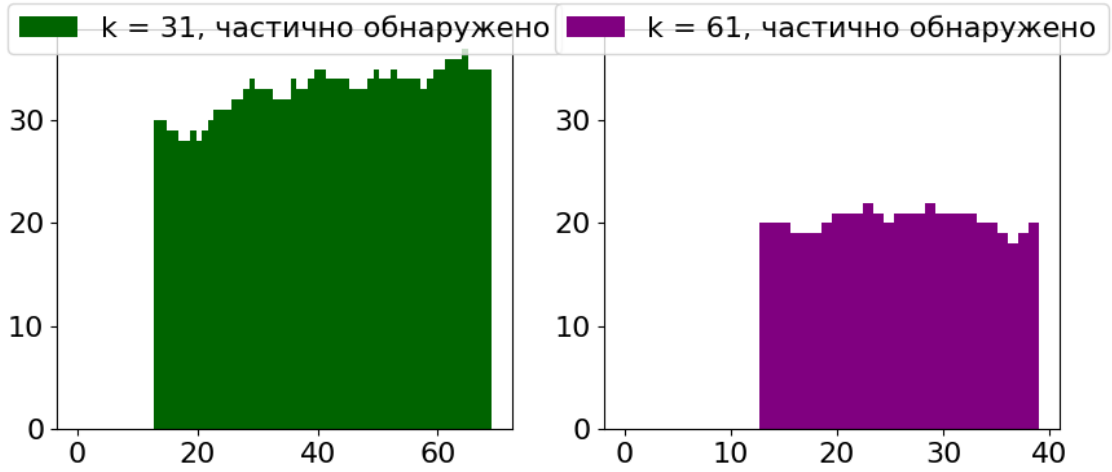

Рисунок 6 – Пример классификации прочтения как частично обнаруженного при частичном обнаружении при обоих значениях k.

Описанные улучшения должны повысить точность классификатора и позволить отслеживать приживаемость штаммов у реципиента.

2.4. Классификатор прочтений для серии метагеномных образцов

Следующей поставленной задачей была классификация прочтений из серии, содержащей три метагеномных образца: метагеном донора до забора материала для фекальной трансплантации, метагеном реципиента до фекальной трансплантации и метагеном реципиента после фекальной трансплантации. Данная задача была решена с помощью построения классификатора, использующего алгоритм обнаружения прочтений из одного метагенома в другом метагеноме.

Классификатор реализован в виде конвейера из пяти этапов, на каждом из которых запускается алгоритм обнаружения прочтения на парах из входных метагеномов (рис. 7).

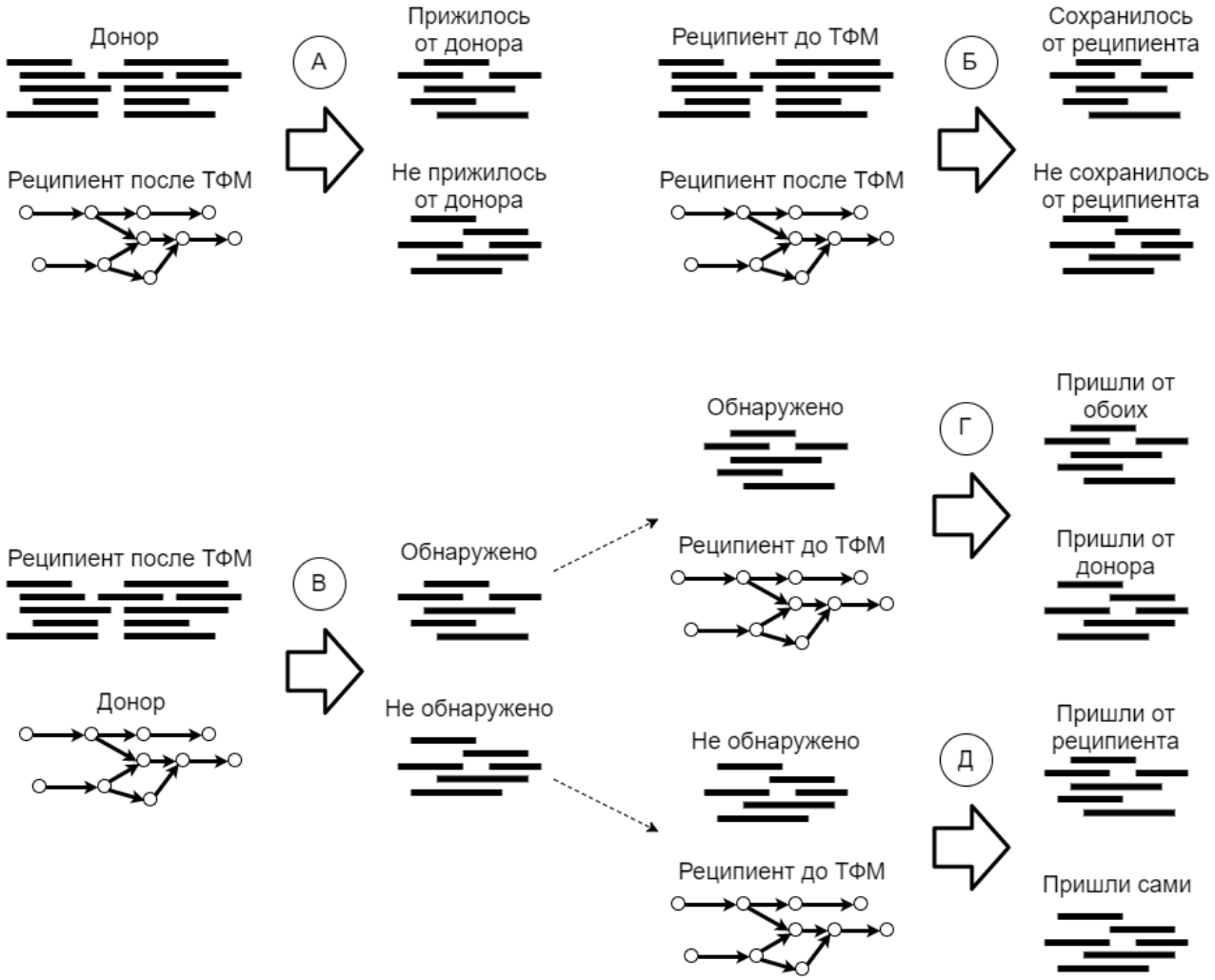

Рисунок 7 – Процесс работы алгоритма состоит из пяти этапов:
(А) классификация прочтений донора, (Б) классификация прочтений реципиента до ТФМ, (В, Г, Д) классификация прочтений реципиента после ТФМ.

Этап 1. На вход алгоритму подается метагеном реципиента после фекальной трансплантации в качестве источника и метагеном донора в качестве анализируемого. В результате работы алгоритма все прочтения из метагенома донора классифицируются на обнаруженные и не обнаруженные. Обнаруженные образуют класс прочтений, называемый *прижилось (от донора)*. Не обнаруженные образуют класс прочтений, называемый *не прижилось (от донора)*.

Этап 2. На вход алгоритму подается метагеном реципиента после фекальной трансплантации в качестве источника и метагеном реципиента до фекальной трансплантации в качестве анализируемого. В результате работы алгоритма все прочтения из метагенома реципиента до фекальной трансплантации классифицируются на обнаруженные и не обнаруженные. Обнаруженные образуют класс прочтений, называемый *сохранилось (от реципиента)*. Не обнаруженные образуют класс прочтений, называемый *не сохранилось (от реципиента)*.

Этап 3. На вход алгоритму подается метагеном донора в качестве источника и метагеном реципиента после фекальной трансплантации в качестве анализируемого. В результате работы алгоритма все прочтения из метагенома реципиента после фекальной трансплантации классифицируются на обнаруженные и не обнаруженные. Классы данного этапа являются промежуточными, поэтому они не именуются.

Этап 4. На вход алгоритму подается метагеном реципиента до фекальной трансплантации в качестве источника и прочтения обнаруженные на третьем этапе в качестве анализируемых. В результате работы алгоритма прочтения из метагенома реципиента после фекальной трансплантации обнаруженные у донора классифицируются на обнаруженные и не обнаруженные. Обнаруженные образуют класс прочтений, называемый *пришли от обоих (и от донора, и от реципиента)*. Не обнаруженные образуют класс прочтений, называемый *пришли от донора*.

Этап 5. На вход алгоритму подается метагеном реципиента до фекальной трансплантации в качестве источника и прочтения не обнаруженные на третьем этапе в качестве анализируемых. В результате работы алгоритма прочтения из метагенома реципиента после фекальной трансплантации не обнаруженные у донора классифицируются на обнаруженные и не обнаруженные. Обнаруженные образуют класс прочтений, называемый *пришли от реципиента*. Не обнаруженные образуют класс прочтений, называемый *пришли сами* (то есть эти прочтения не обнаружены ни у донора, ни у реципиента до фекальной трансплантации). Появление последней категории возможно благодаря тому, что реципиент взаимодействует с окружающей средой не только посредством фекальной трансплантации, но и другими путями

(например, употребляя пищу), что может влиять на видовой состав метагенома его кишечника.

В результате работы классификатора строятся восемь классов прочтений и классифицируются все прочтения из каждого из трех входных метагеномов. Классификатор реализован в виде скрипта для запуска конвейера на языке Bash, исходный код доступен в репозитории по ссылке `http://www.github.com/ivartb/fmt`.

В случае использования модифицированного алгоритма классификации прочтений, описанного в разделе 2.3, классификатор претерпит изменения, связанные с появлением нового класса прочтений (частично обнаруженные прочтения) на каждом этапе классификации. В процессе работы потребуется дополнительный этап, а в результате будут получены тринадцать классов прочтений, каждый из которых описан ниже. Схема работы классификатора изображена на рисунке 8.

Этап 1. На вход алгоритму подается метагеном реципиента после фекальной трансплантации в качестве источника и метагеном донора в качестве анализируемого. Обнаруженные прочтения образуют класс, называемый *прижилось (от донора)*. Не обнаруженные прочтения образуют класс, называемый *не прижилось (от донора)*. Частично обнаруженные прочтения образуют класс, называемый *частично прижилось (от донора)*. Последний класс соответствует обнаружению у донора такого организма, близкородственный штамм которого обнаружен у реципиента после процедуры ТФМ.

Этап 2. На вход алгоритму подается метагеном реципиента после фекальной трансплантации в качестве источника и метагеном реципиента до фекальной трансплантации в качестве анализируемого. Обнаруженные прочтения образуют класс, называемый *сохранилось (от реципиента)*. Не обнаруженные прочтения образуют класс, называемый *не сохранилось (от реципиента)*. Частично обнаруженные прочтения образуют класс, называемый *частично сохранилось (от реципиента)*. Последний класс соответствует обнаружению у реципиента до фекальной трансплантации такого организма, близкородственный штамм которого обнаружен у реципиента после процедуры ТФМ.

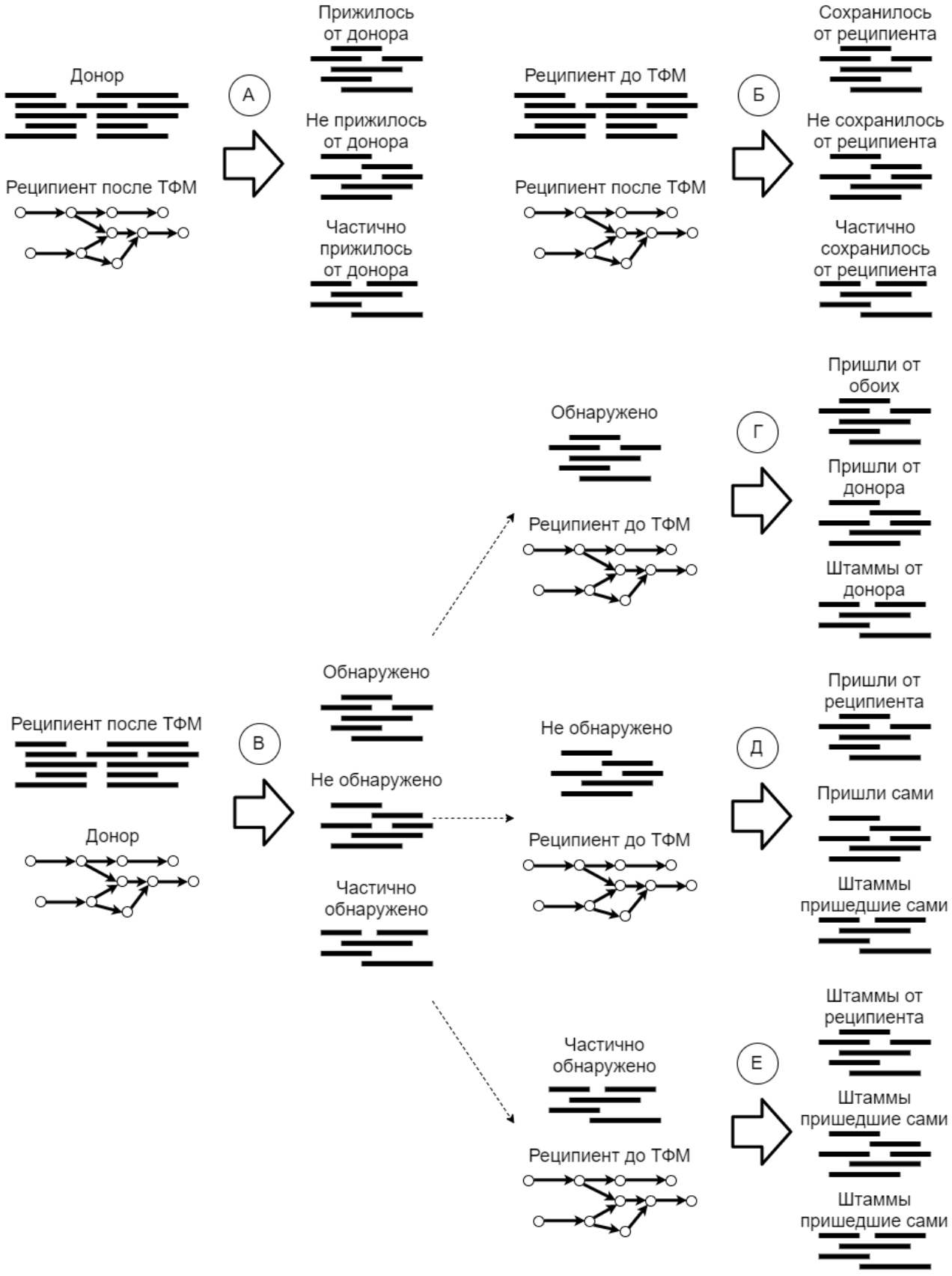

Рисунок 8 – Процесс работы алгоритма состоит из шести этапов: (А) классификация прочтений донора, (Б) классификация прочтений реципиента до ТФМ, (В, Г, Д, Е) классификация прочтений реципиента после ТФМ.

Этап 3. На вход алгоритму подается метагеном донора в качестве источника и метагеном реципиента после фекальной трансплантации в качестве анализируемого. В результате работы алгоритма все прочтения из метагенома реципиента после фекальной трансплантации классифицируются на обнаруженные, не обнаруженные и частично обнаруженные. Классы данного этапа являются промежуточными, поэтому они не именуются.

Этап 4. На вход алгоритму подается метагеном реципиента до фекальной трансплантации в качестве источника и прочтения обнаруженные на третьем этапе в качестве анализируемых. Обнаруженные прочтения образуют класс, называемый *пришли от обоих (и от донора, и от реципиента)*. Не обнаруженные прочтения образуют класс, называемый *пришли от донора*. Частично обнаруженные прочтения образуют класс, называемый *штаммы от донора*. Последний класс соответствует обнаружению во всех трех временных точках штаммов одного организма. При этом штаммы у донора и реципиента после ТФМ совпадают, а штамм у реципиента до ТФМ отличается от них. Это позволяет сделать предположение, что штамм бактерии в организме реципиента был вытеснен штаммом из организма донора в процессе пересадки микробиоты.

Этап 5. На вход алгоритму подается метагеном реципиента до фекальной трансплантации в качестве источника и прочтения не обнаруженные на третьем этапе в качестве анализируемых. Обнаруженные прочтения образуют класс, называемый *пришли от реципиента*. Не обнаруженные прочтения образуют класс, называемый *пришли сами* (то есть эти прочтения не обнаружены ни у донора, ни у реципиента до фекальной трансплантации). Частично обнаруженные прочтения образуют класс, называемый *штаммы пришедшие сами*. Последний класс соответствует обнаружению у реципиента до ТФМ штамма организма, близкого к организму в метагеноме кишечника реципиента после ТФМ. Это может означать, что штамм бактерии в организме реципиента был вытеснен другим штаммом, пришедшим из взаимодействия реципиента с окружающей средой.

Этап 6. На вход алгоритму подается метагеном реципиента до фекальной трансплантации в качестве источника и прочтения частично обнаруженные на третьем этапе в качестве анализируемых. Обнаруженные прочтения образуют класс, называемый *штаммы от реципиента*. Он соответствует обнаружению

во всех трех временных точках штаммов одного организма. При этом штаммы у реципиента до и после ТФМ совпадают, а штамм у донора отличается от них. Это означает, что штамм донора не смог вытеснить штамм бактерии, находившейся в организме реципиента до фекальной трансплантации. Не обнаруженные и частично обнаруженные прочтения образуют класс, называемый *штаммы пришедшие сами*. Он соответствует обнаружению у донора и, возможно, реципиента до ТФМ штамма организма, близкого к организму в метагеноме кишечнике реципиента после ТФМ.

Выводы по главе 2

В данной главе предложен алгоритм решения задачи классификации прочтений из серии метагеномных образцов на основе графа де Брейна с классификацией каждого прочтения, удовлетворяющий условию задачи. В результате работы алгоритм выдает набор классов прочтений, которые в дальнейшем можно использовать для проведения различных биоинформатических экспериментов, например, запускать на них алгоритмы таксономии для определения видового состава классов.

ГЛАВА 3. ВАЛИДАЦИЯ АЛГОРИТМА И ПОЛУЧЕННЫЕ РЕЗУЛЬТАТЫ

В данной главе описаны методы, которые были применены для проверки корректности и оценки точности разработанного алгоритма. Проводится практическое сравнение с существующими программами. Обсуждаются результаты, полученные при запуске программы на реальных метагеномных данных пациентов.

3.1. Симуляции метагеномов с помощью MetaSim

Для проверки корректности работы алгоритма необходимо было сгенерировать набор прочтений, представляющий отдельный метагеном. Данная задача была решена с помощью симулятора последовательностей MetaSim [26]. Он принимает на вход геномные последовательности и профиль их содержания в метагеноме и генерирует набор прочтений для метагенома. Также программа позволяет применить к сгенерированным прочтениям модели ошибок, которые соответствуют существующим технологиям секвенирования. В результате получается метагеном с известными параметрами каждого прочтения, который используется для валидации разработанного алгоритма.

3.1.1. Метагеномы без ошибок в прочтениях

На первом этапе тестирования использовалась стратегия генерации метагеномов без ошибок в прочтениях. Были рассмотрены четыре варианта входных геномных последовательностей и профилей их представленности:

а) 30 видов кишечных бактерий (см. приложение А) с равномерным распределением представленности каждой бактерии в метагеноме
б) 30 видов кишечных бактерий (см. приложение А) с экспоненциальным распределением ($\lambda = 0{,}25$) представленности каждой бактерии в метагеноме
в) 61 штамм кишечной палочки (см. приложение Б) с равномерным распределением представленности каждого штамма в метагеноме
г) 61 штамм кишечной палочки (см. приложение Б) с экспоненциальным распределением ($\lambda = 0{,}25$) представленности каждого штамма в метагеноме

Для каждого из четырех вариантов проводилось по десять независимых экспериментов. В эксперименте генерировалось три метагенома, соответствующие реальным: метагеном донора, метагеном реципиента

до фекальной трансплантации и метагеном реципиента после фекальной трансплантации. При работе с видами для одного метагенома использовались 15 случайных видов с заданным распределением представленности. Примеры относительной представленности видов в метагеноме для экспериментов представлены на рис. 9. При работе со штаммами для одного эксперимента фиксировались 30 случайных штаммов, из которых выбирались 15 случайных с заданным распределением представленности для каждого метагенома. Всего для одного метагенома генерируется 10 млн парноконцевых прочтений длины 100 оснований каждое. После генерации серии из трех метагеномов на них запускался алгоритм классификаций, который выдавал восемь классов прочтений.

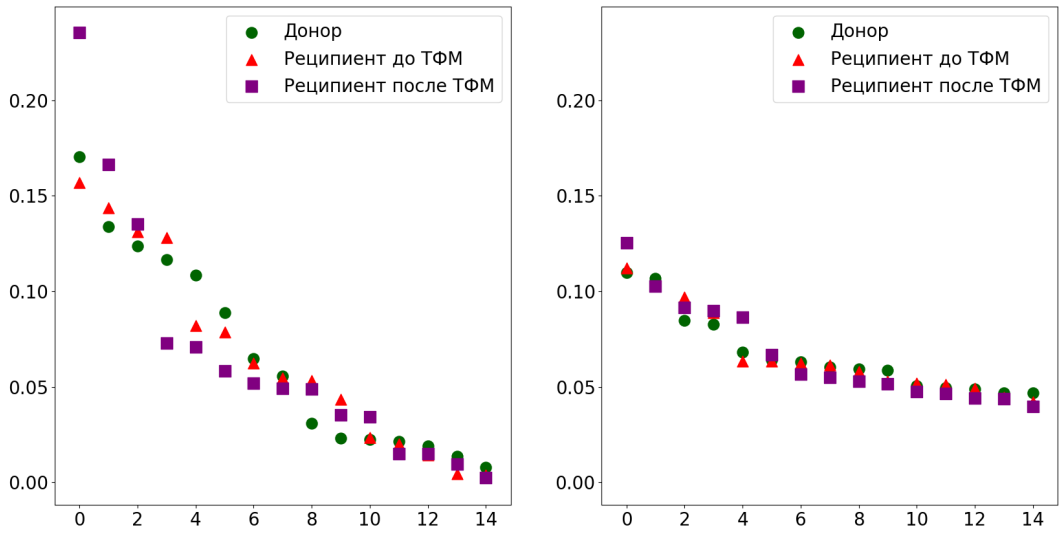

Рисунок 9 – Примеры относительной представленности видов в метагеноме для экспоненциального (слева) и равномерного (справа) распределения.

Так как при генерации было априори известно, какие геномы содержатся в каждом метагеноме, можно проверить насколько сгенерированные классы соответствуют теоретическим. Сначала из прочтений трех исходных метагеномов отфильтровываются те, которые соответствуют более чем одному геному, то есть остаются только прочтения уникально сопоставляемые с геномом. В результате этой операции остается 99 % прочтений в экспериментах с видами и 38 % в экспериментах со штаммами. Затем прочтения из каждого класса сравниваются с фильтрованными прочтениями исходных метагеномов. В результате каждому классу сопоставляются три числа – количество правильно

обнаруженных геномов, количество геномов ошибочно не обнаруженных в классе (ошибка второго рода) и количество геномов ошибочно обнаруженных в классе (ошибка первого рода). Также производится отсечение геномов, которые обнаружены в количестве прочтений меньше фиксированного уровня threshold от общего количества классифицированных прочтений в метагеноме. Необходимо отметить, что в связи с уменьшением количества прочтений после фильтрации, количество обнаруженных прочтений будет пропорционально уменьшаться. Соответственно, при уровне отсечении threshold в штаммах будут отсеиваться все геномы, содержание которых меньше $2{,}6 \cdot$ threshold относительно общего количества прочтений в метагеномах.

Результаты проверки качества полученных классов приведены в табл. 2. По ним можно сделать вывод, что алгоритм правильно классифицирует геномы различных видов, но может ошибаться на штаммах одного организма.

Таблица 2 – Результаты тестирования классификатора на сгенерированных метагеномах без ошибок в прочтениях. Параметр threshold $= 0{,}0001$

	Вероятность ошибки первого рода, %	Вероятность ошибки второго рода, %
Виды с равномерным распределением	$6{,}02$	$< 0{,}02$
Виды с экспоненциальным распределением	$9{,}81$	$< 0{,}02$
Штаммы с равномерным распределением	$21{,}68$	$0{,}38$
Штаммы с экспоненциальным распределением	$25{,}00$	$0{,}19$

3.1.2. Метагеномы с ошибками в прочтениях

На втором этапе тестирования использовалась стратегия генерации метагеномов с ошибками в прочтениях. Для генерации прочтений с ошибками использовалась встроенная в программу MetaSim эмпирическая модель ошибок. Для каждой позиции в прочтении вероятность ошибки на ней устанавливалась равной $0{,}001$, то есть одно ошибочное основание на 1000. Остальные параметры генерации метагеномов сохранены такие же, как и на первом этапе тестирования.

Результаты проверки качества полученных классов приведены в табл. 3 и табл. 4. При выборе параметре threshold $= 0{,}0001$ результаты классификации значительно хуже, чем при работе на прочтениях без ошибок, и полученные классы не могут считаться надежными. Однако, при выборе параметра threshold $= 0{,}001$ результаты близки по качеству к результатам классификации на прочтениях без ошибок. Из этого можно сделать выводы, что алгоритм ожидаемо работает хуже при наличии ошибок в прочтениях, однако он правильно классифицирует геномы с достаточным уровнем точности при выборе относительно малого параметра отсечения.

Таблица 3 – Результаты тестирования классификатора на сгенерированных метагеномах с эмпирической моделью ошибок в прочтениях. Параметр threshold $= 0{,}0001$

	Вероятность ошибки первого рода, %	Вероятность ошибки второго рода, %
Виды с равномерным распределением	26,37	$< 0{,}02$
Виды с экспоненциальным распределением	23,65	$< 0{,}02$
Штаммы с равномерным распределением	42,67	0,11
Штаммы с экспоненциальным распределением	38,62	0,03

Также были проведены запуски программы на сгенерированных метагеномах с моделью ошибок Sanger. Полученные результаты представлены в приложении В.

3.2. Сравнение с существующим алгоритмом выравнивания прочтений на граф де Брейна

Для выравнивания прочтений на геномы существуют различные программы, такие как [27], в которой используется геномный граф вариаций, и [28], предназначенная для выравнивания длинных прочтений с ошибками. Более близкой к разработанному алгоритму является программа BrownieAligner [29]. Она ищет лучшее точное совпадение по k-мерам из прочтения в графе де Брейна, а затем пытается продлить выравнивание. Также

Таблица 4 – Результаты тестирования классификатора на сгенерированных метагеномах с эмпирической моделью ошибок в прочтениях. Параметр threshold $= 0{,}001$

	Вероятность ошибки первого рода, %	Вероятность ошибки второго рода, %
Виды с равномерным распределением	$11{,}41$	$< 0{,}02$
Виды с экспоненциальным распределением	$11{,}95$	$< 0{,}02$
Штаммы с равномерным распределением	$23{,}52$	$0{,}11$
Штаммы с экспоненциальным распределением	$21{,}49$	$0{,}03$

для борьбы с химерными путями в графе в ней реализована марковская модель для определения наиболее вероятного продолжения на разветвлениях.

Для сравнения качества и скорости работы алгоритма, был запущен BrownieAligner на сгенерированных метагеномах из видов с экспоненциальным распределением без ошибок в прочтениях. Прочтение считалось обнаруженным программой BrownieAligner, когда оно было точно выровнено на граф или было выровнено на граф не более чем с фиксированным числом исправлений. Также была использована внутренняя метрика алгоритма, основанная на количестве исправлений и выборе пути при ветвлении, для определения обнаружения прочтения в графе. Порог обнаружения организмов threshold был установлен равным $0{,}0001$. Результаты сравнения представлены на рисунке 10. При увеличении порога threshold до $0{,}001$ процент ошибок падает почти в три раза. При этом соотношение процента ошибок в зависимости от выбранного метода сохраняется (см. рис. 11). Тем не менее необходимо понимать, что в сложных метагеномах для обнаружения малопредставленных видов может понадобиться более низкий порог.

Также алгоритм BrownieAligner был использован для обнаружения прочтений в сгенерированных метагеномах из видов с экспоненциальным распределением и вероятностью ошибки $0{,}001$ на каждой позиции в прочтении (имитация прочтений от секвенатора Illumina). При этом порог обнаружения прочтений threshold был выбран равным $0{,}001$, так как метагеномы из данной симуляции являются более сложными по сравнению с предыдущей симуляцией

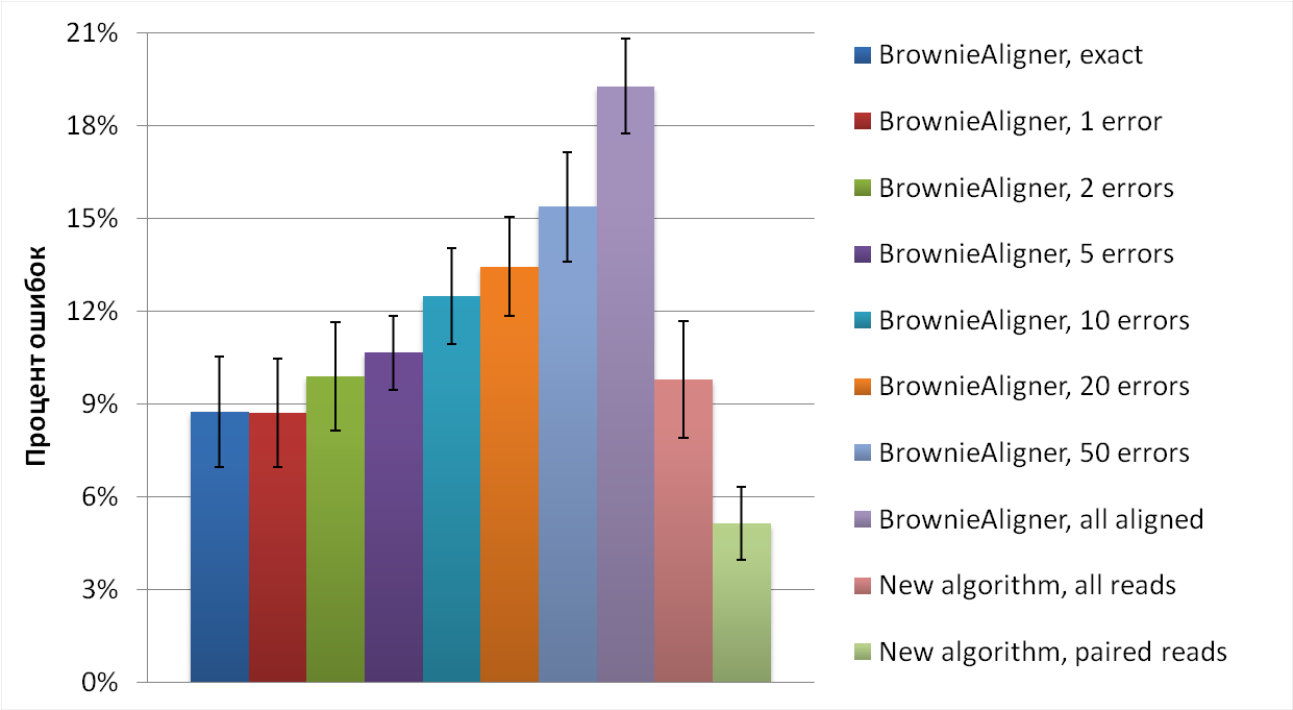

Рисунок 10 – Сравнение классификаторов на сгенерированных метагеномах из видов с экспоненциальным распределением без ошибок в прочтених. Параметр threshold $= 0{,}0001$.

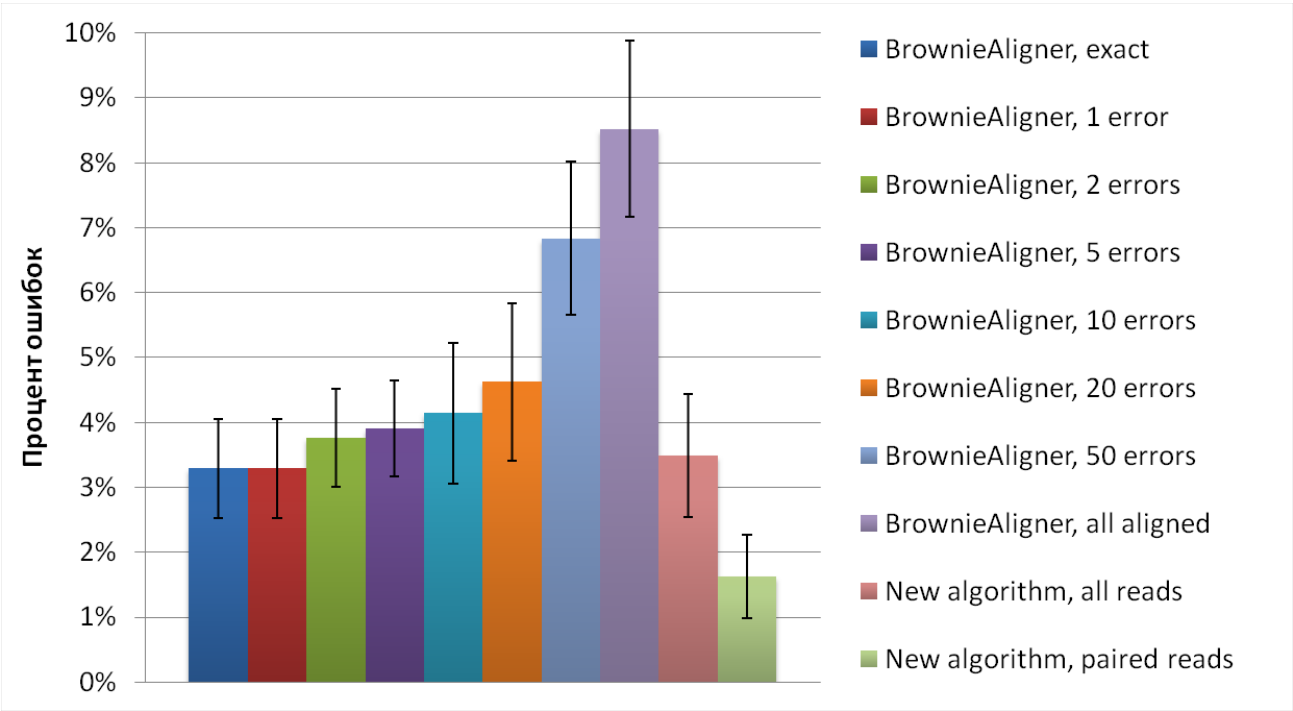

Рисунок 11 – Сравнение классификаторов на сгенерированных метагеномах из видов с экспоненциальным распределением без ошибок в прочтених. Параметр threshold $= 0{,}001$.

из-за наличия ошибок в прочтениях. Результаты обнаружения прочтений для разных режимов работы представлены на рисунке 12.

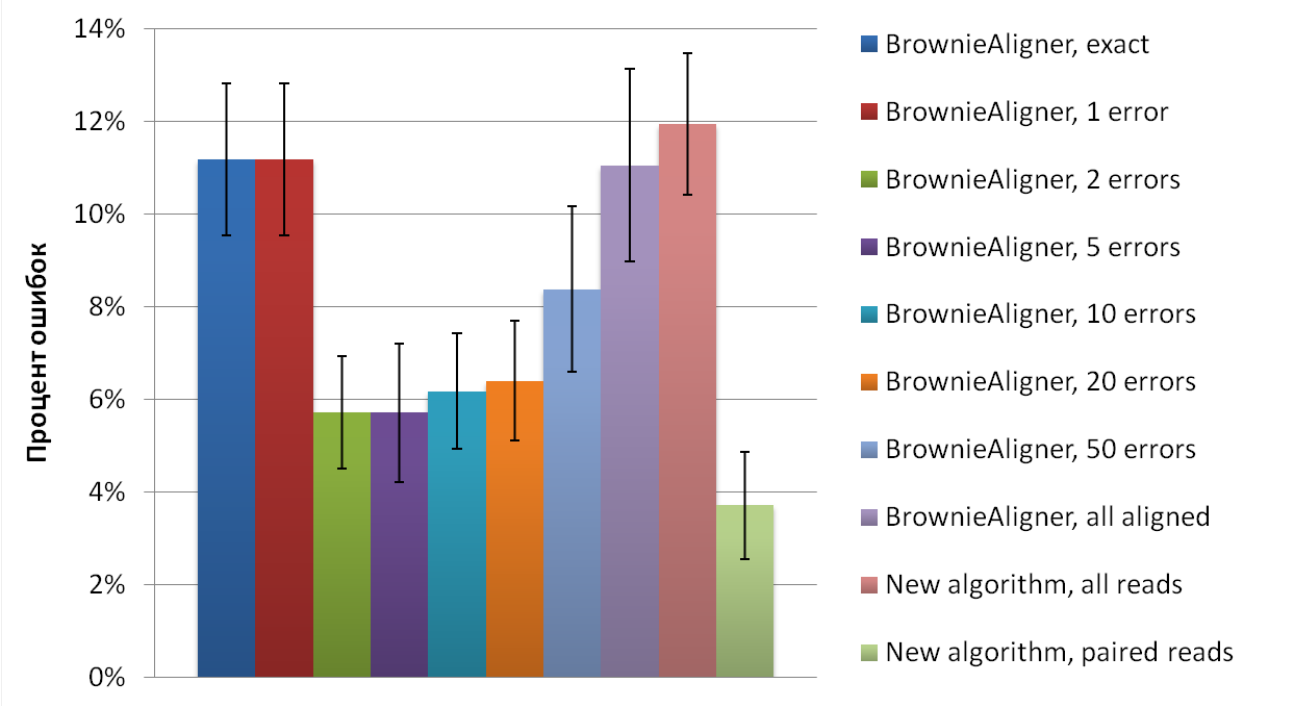

Рисунок 12 – Сравнение классификаторов на сгенерированных метагеномах из видов с экспоненциальным распределением и вероятностью ошибки $0{,}001$. Параметр threshold $= 0{,}001$.

По гистограммам на рисунках 10, 11 и 12 можно сделать вывод, что разработанный алгоритм показывает более высокую точность при использовании только парноконцевых классифицируемых прочтений и сопоставимую точность при использовании всех прочтений. Однако реализованный алгоритм более прямолинеен, а также парноконцевые прочтения содержат полезную информацию о метагеноме, что делает его более привлекательным для использования.

3.3. Сравнение с существующим средством сравнительного анализа метагеномов

Проведение сравнительного анализа метагеномов необходимо для понимания принципов существования микробных сообществ. Для решения этой задачи были созданы различные программы, одной из которых является Commet [30]. Принцип ее работы основан на поиске непересекающихся k-меров из прочтений в фильтре Блума. Ее преимуществом является экономия памяти дискового пространства, однако она не заточена для работы с зависимыми метагеномными данными. Для изучения образцов метагеномов из исследования ТФМ необходимо уметь точно отделять близкородственные организмы, так как их замещение может являться ключевым фактором, который

определяет результат процедуры фекальной трансплантации. В разработанном алгоритме для этой цели был введен отдельный класс частично обнаруженных прочтений. Однако, даже при жестком разделении прочтений на обнаруженные и необнаруженные, разработанный алгоритм по качеству превосходит Commet при анализе метагеномов из близкородственных штаммов вида E. Coli. Для анализа были использованы сгенерированные метагеномы из 100 тысяч парноконцевых прочтений длины 100, состоящие из 10 случайных штаммов E. Coli каждый. Результаты сравнения алгоритмов на сгенерированных метагеномах представлены на рисунке 13.

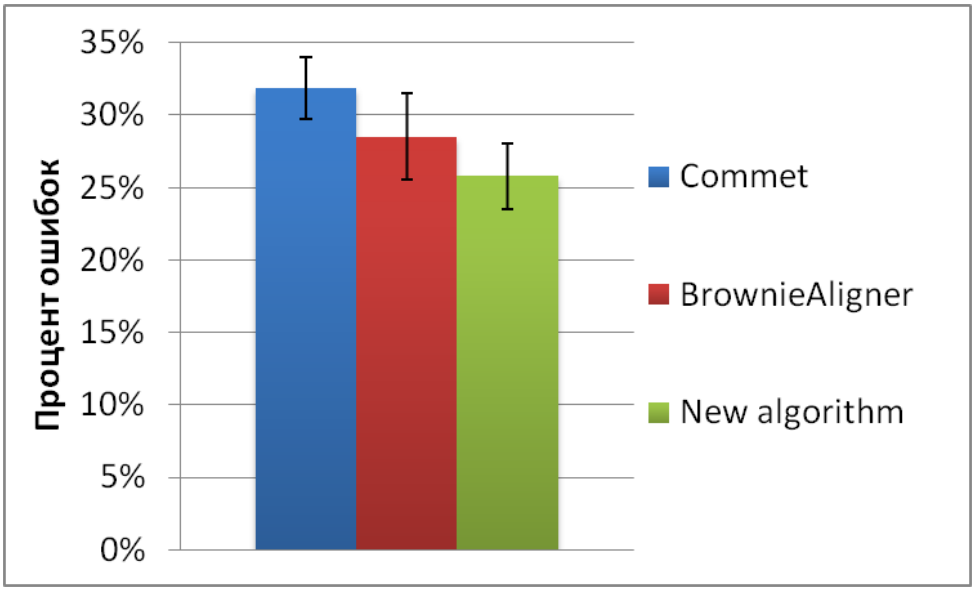

Рисунок 13 – Сравнение алгоритмов для сравнения метагеномов на сгенерированных образцах из штаммов E. Coli. Параметр threshold= 0,001.

Также было проведено сравнение Commet с разработанным алгоритмом на реальных метагеномных данных Lee2017 [23]. Из классов *пришли от донора* для первого и второго реципиентов были выделены высококачественные корзины (полнота не менее 70 %, загрязнение не более 10 %) с помощью алгоритма metaWRAP [19]. Далее был проведен таксономический анализ полученных корзин, а также исходных метагеномов для оценки качества полученных результатов. Из таблицы 5 видно, что предложенный алгоритм способствовал обнаружению большего числа корзин. При этом проверка корзин с известной таксономией выявила, что оба алгоритма показывают одинаковое качество. Следовательно, разработанный алгоритм предоставляет больше информации для исследования приживаемости бактерий, сравнимой

по качеству с Commet, и может помочь в выявлении ранее неизвестных зависимостей.

Таблица 5 – Сравнение корзин из класса *пришли от донора* у первого (R01) и второго (R02) реципиентов для предложенного алгоритма и Commet.

	Первый реципиент R01		Второй реципиент R02	
	New algorithm	Commet	New algorithm	Commet
Всего корзин	12	11	20	15
Таксономически аннотировано корзин	6	4	13	5
Ошибочных корзин	2	2	1	0

3.4. Запуск на реальных данных и сравнение результатов с существующими

Для проведения экспериментов на реальных данных были выбраны метагеномы, полученные в исследовании [23]. Двум реципиентам R01 и R02 производилась пересадка материала от одного донора с помощью одной процедуры колоноскопии. Для изучения были взяты три метагеномных образца от донора, а также по три метагеномных образца от каждого из реципиентов: один до трансплантации, один через четыре недели после трансплантации и один через восемь недель после трансплантации. Производился запуск разработанного алгоритма, на вход которому подавался метагеном донора, метагеном одного из реципиентов до трансплантации и метагеном соответствующего реципиента через четыре или восемь недель после трансплантации. В результате на выходе получены четыре набора классов прочтений, описывающие изменение микробиоты реципиентов после трансплантации (два реципиента по две временные точки после ТФМ).

В дальнейшем для анализа использовались полученные классы прочтений. Одним из проведенных экспериментов было сравнение приживаемости донорских бактерий в организме реципиентов. Для этого были проведены следующие этапы обработки данных:

а) Производилась совместная сборка прочтений из всех четырех классов *прижилось от донора* с помощью алгоритма megahit, запущенного со стандартными параметрами из оболочки metaWRAP.

б) Полученные на предыдущем шаге контиги разбивались на корзины с помощью модулей binning и bin_refinement из программы metaWRAP.

в) Для визуализации полученных корзин использовалась программа anvi'o. С помощью программы bowtie2 производилось выравнивание прочтений из входных метагеномов обратно на полученные контиги из сборки для получения информации о покрытии. Далее создавались профили покрытия полученных после сборки корзин прочтениями из каждого из входных метагеномов, объединялись в один профиль и по ним генерировалось совместное визуальное представление полученных результатов.

Пример такой визуализации представлен на рисунке 14. По нему можно сделать несколько выводов. Во-первых, каждая из корзин покрыта прочтениями хотя бы у одного реципиента после трансплантации. Это означает, что разработанный алгоритм правильно определил прижившиеся у реципиента корзины и смог отделить все не прижившиеся в отдельную категорию. Во-вторых, необходимо отметить, что более чем в половине случаев происходит сохранение организмов у обоих реципиентов. Это согласуется с результатами, полученными в оригинальной статье [23]. На основе этого факта можно сделать предположение, что некоторые виды лучше колонизируют реципиентов и могут быть полезны при разработке персонализированного набора бактерий для трансплантации. В-третьих, в большинстве случаев прижившиеся виды слабо представлены или вообще не представлены в организме реципиента до трансплантации. Это является весомым аргументом в пользу того, что состав микробиоты кишечника после трансплантации значительно изменяется. Возможно это связано с тем, что новые бактерии от донора колонизируют ранее не занятые экологические ниши.

Для обоснования предположения о замещении донорскими бактериями бактерий в реципиенте было проведено сравнение всех метагеномных образцов из данного исследования с помощью программы MetaFast [17]. Она делает неполную сборку каждого метагенома в отдельности и выделяет из нее контиги, на основании покрытия которых k-мерами из каждого метагенома строится вектор признаков для каждого метагенома. Далее полученные вектора сравниваются, что позволяет провести оценку похожести метагеномов. Однако полученные результаты говорят лишь об общем сходстве или различии, не позволяя отслеживать изменения отдельных видов в метагеномных

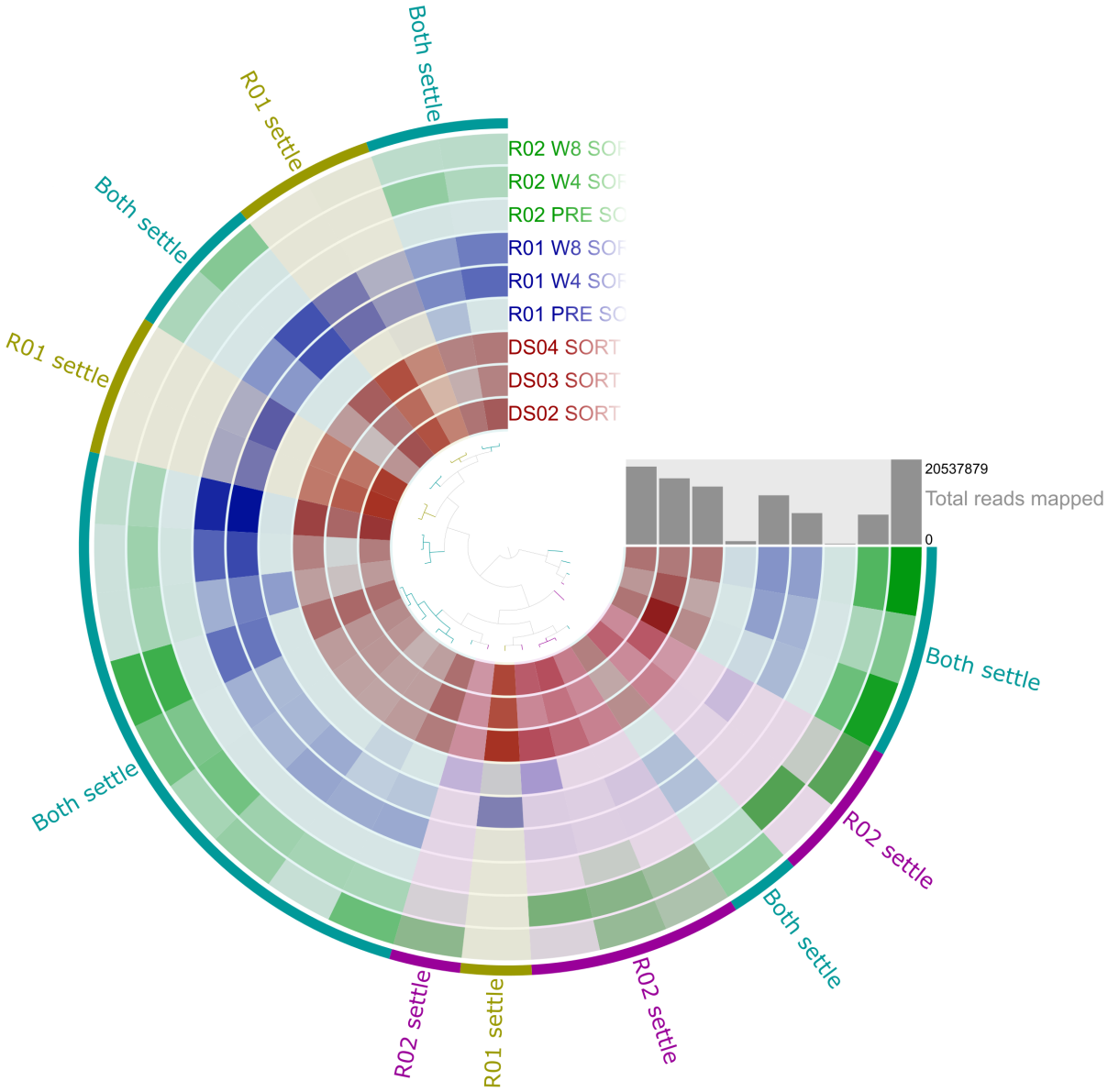

Рисунок 14 – Пример обнаружения корзин, полученных из прочтений прижившихся от донора, в метагеномных образцах донора (DS) и реципиентов (R01 и R02) до ТФМ (PRE), четыре недели после ТФМ (W4) и восемь недель после ТФМ (W8).

сериях. Визуализация полученных попарных расстояний между метагеномами совместно с дендрограммой представлена на рисунке 15.

Исходя из полученных расстояний видно, что метагеномы реципиентов после трансплантации больше похожи на донорские образцы, чем на себя самих до ТФМ. Значит донорские бактерии вытесняют живущие в кишечнике реципиента организмы и занимают образовавшиеся ниши.

Другим экспериментом был анализ класса прочтений реципиента R01, сохранившихся у реципиента в процессе трансплантации. Анализ проводился

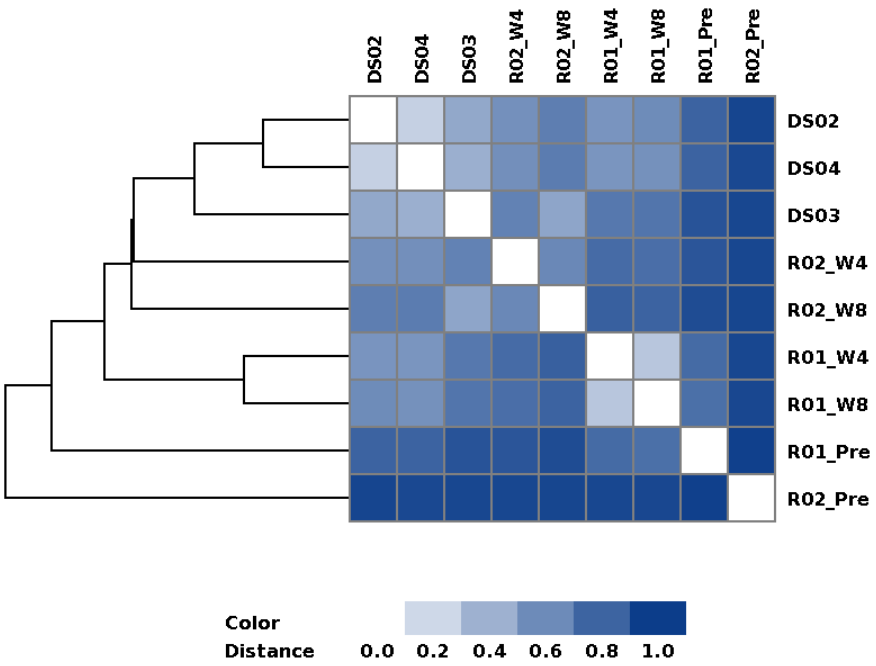

Рисунок 15 – Сравнение метагеномных образцов на основе их близости по расстоянию Брея-Кёртиса (англ. Bray-Curtis distance) с использованием программы MetaFast.

аналогично процессу, описанному выше. При этом производилась совместная сборка из двух классов *пришли от реципиента* реципиента R01. Пример визуализации полученных результатов анализа представлен на рисунке 16.

Из графика видно, что прочтения данного класса действительно отсутствуют в метагеномных образцах донора, но при это хорошо представлены у реципиента как до, так и после трансплантации. Это показывает, что разработанный алгоритм произвел правильный отбор прочтений для данного класса. Также важно отметить, что уровень покрытия корзин прочтениями у реципиента через восемь недель не уменьшился, а даже вырос по сравнению с образцом, взятым через четыре недели после ТФМ. Это говорит о том, что данные виды стабильно приживаются в организме реципиента и остаются там в течение продолжительного времени.

Также интересно отдельно проанализировать класс прочтений реципиента R02, прижившихся от донора в результате фекальной трансплантации. Эти прочтения составляют класс *пришли от донора* и были

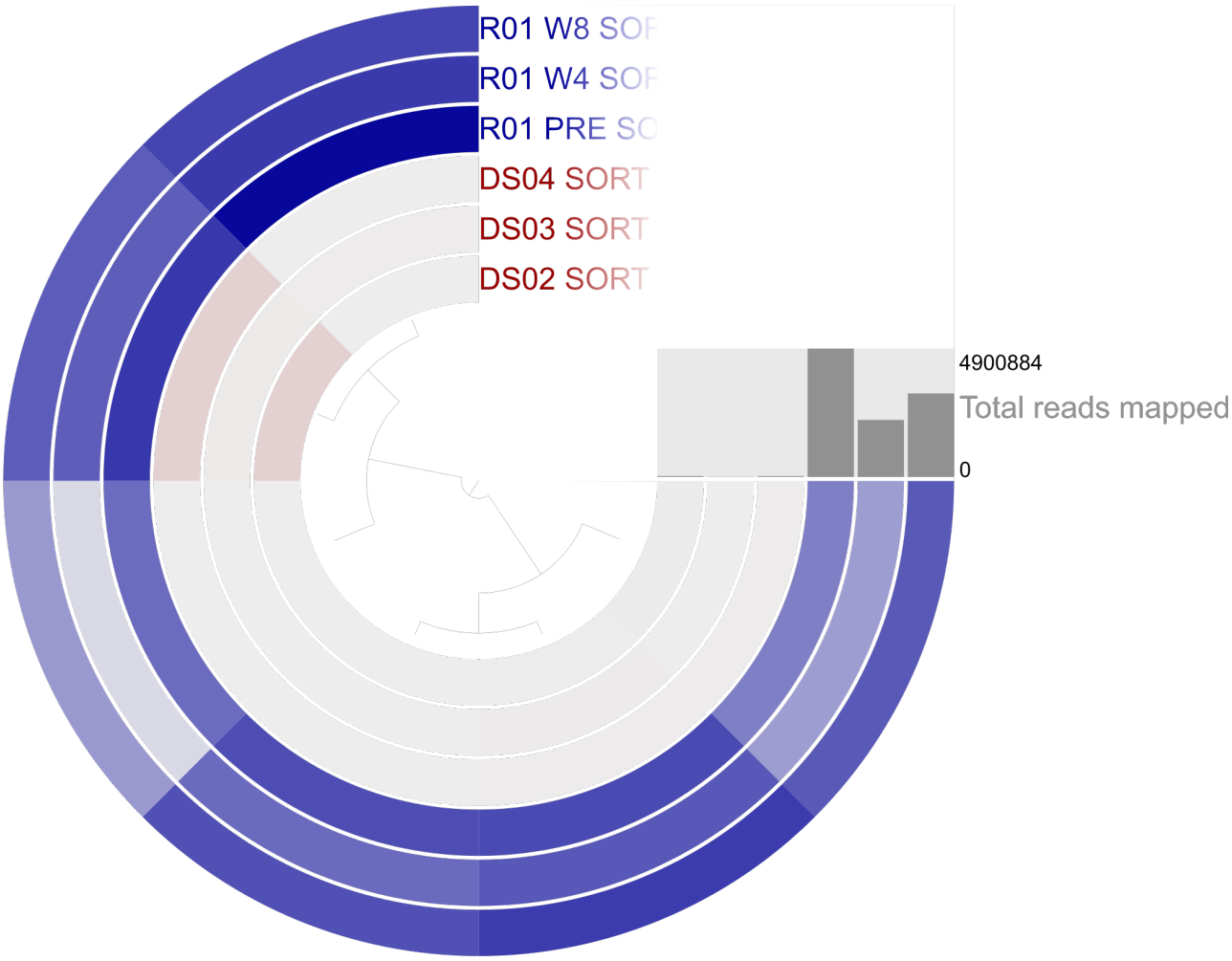

Рисунок 16 – Пример обнаружения корзин, полученных из прочтений сохранившихся у реципиента R01 после трансплантации, в метагеномных образцах донора (DS) и реципиента (R01) до ТФМ (PRE), четыре недели после ТФМ (W4) и восемь недель после ТФМ (W8).

проанализированы с помощью последовательной обработки, аналогичной описанной выше. Пример визуализации полученных результатов анализа представлен на рисунке 17.

Во-первых, каждая корзина не покрыта прочтениями реципиента до ТФМ, но при этом покрыта хотя бы в одной временной точке после ТФМ. Это означает, что предложенный алгоритм поместил в исследуемый класс прочтения, действительно соответствующие прижившимся от донора к реципиенту в результате трансплантации. Во-вторых, на графике видна зависимость между покрытием корзины прочтениями в метагеномных образцах донора и степенью приживаемости в реципиенте. Из этого наблюдения можно сделать предположение о том, что чем больше представителей одного вида

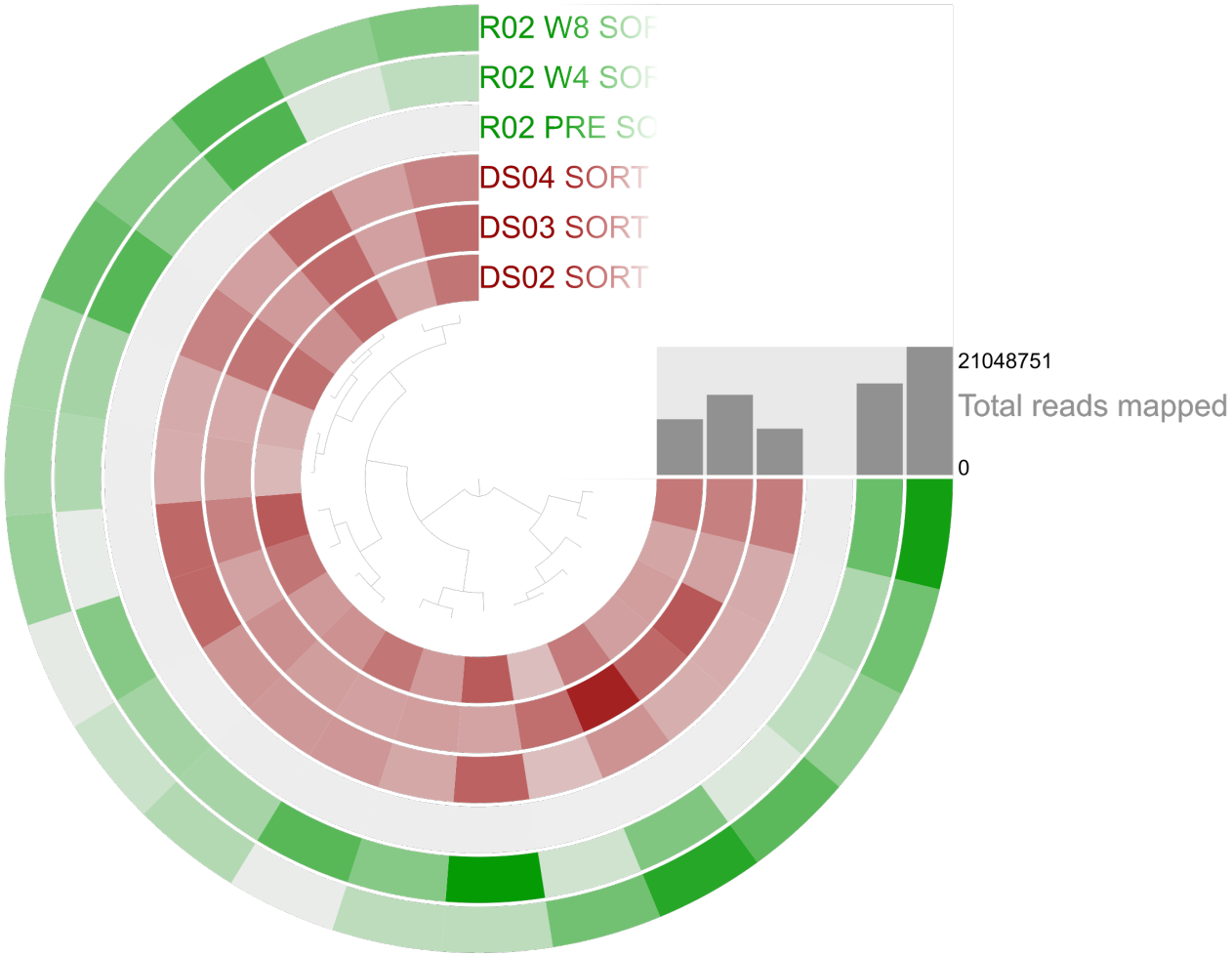

Рисунок 17 – Пример обнаружения корзин, полученных из прочтений пришедших от донора к реципиенту R02 после трансплантации, в метагеномных образцах донора (DS) и реципиента (R02) до ТФМ (PRE), четыре недели после ТФМ (W4) и восемь недель после ТФМ (W8).

приходит в организм реципиента в результате трансплантации, тем вероятнее его приживаемость и закрепление на длительный период.

Отдельно была изучена приживаемость донорских организмов у реципиентов. По рисунку 14 было отмечено, что многие корзины из класса *прижилось от донора* были покрыты прочтениями обоих реципиентов. Необходимо было проверить гипотезу о близости организмов, колонизирующих кишечник реципиентов. Для этого отдельно для реципиентов R01 и R02 из класса *пришли от донора* были собраны корзины аналогично алгоритму описанному выше. Далее между каждой парой корзин было посчитано *mash* расстояние (англ. mash distance), которое оценивает различие между геномами. Суть метода состоит в извлечении фиксированного числа k-меров из генома, вычислении их хэш-функции и поиска общих значений между

двумя геномами [31]. Визуализация полученных расстояний представлена на рисунке 18.

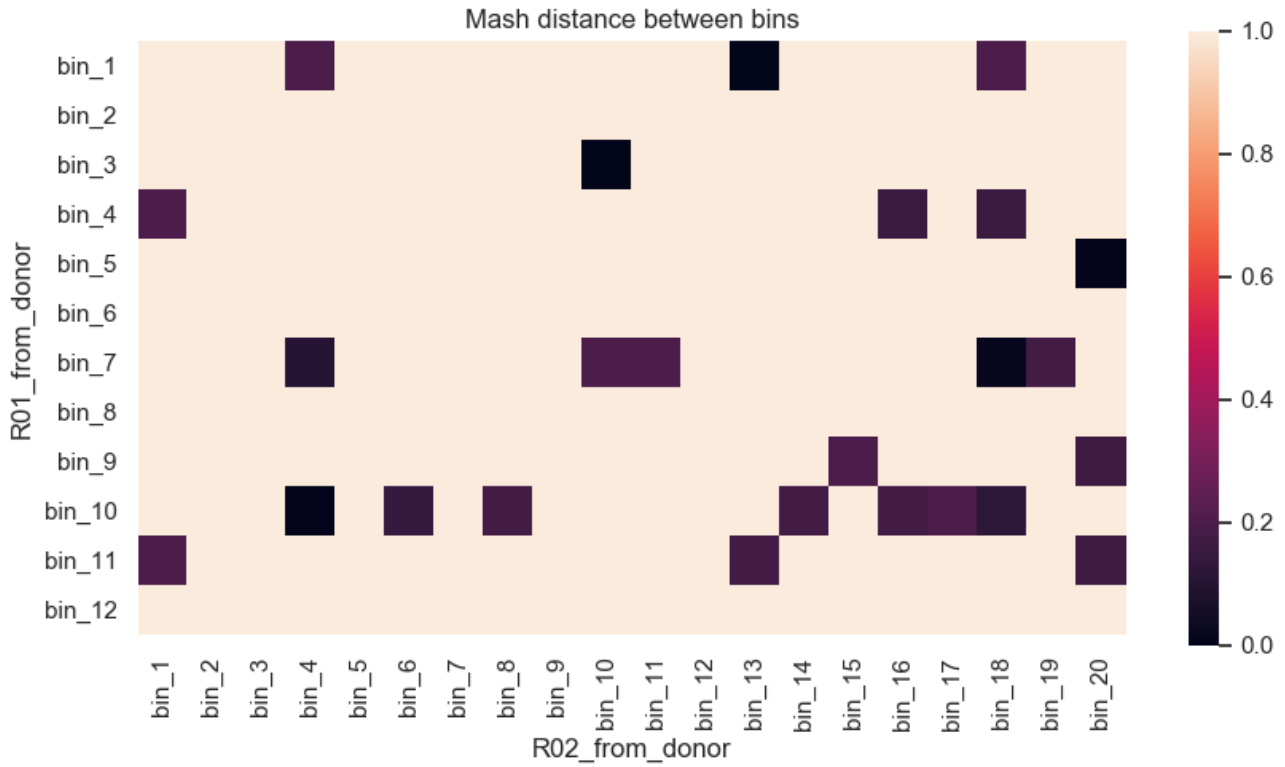

Рисунок 18 – Попарное расстояние между корзинами, полученными из класса *пришло от донора* для первого и второго реципиентов. Значения меньше 0,05 соответствуют одинаковому виду.

У пяти пар корзин значение расстояния не превышает 0,05 (пары bin_1 – bin_13, bin_3 – bin_10, bin_5 – bin_20, bin_7 – bin_18, bin_10 – bin_4), что позволяет сделать вывод об одном виде, находящемся в каждой паре корзин. Проверка предположения об одинаковых видах, содержащихся в корзинах с близким mash расстоянием была проведена путем таксономической аннотации контигов, содержащихся в этих корзинах, с помощью Kraken и определения наименьшего общего предка для каждой корзины с помощью алгоритма, реализованного в metaWRAP. Результаты таксономической аннотации корзин представлены в приложении Г. По ним видно, что корзины в каждой из пяти близких пар одинаково аннотированы, при этом у четырех из пяти классификация совпадает с точностью до вида. Это означает, что виды *Bacteroides helcogenes P 36-108*, *Alistipes finegoldii DSM 17242*, *Bacteroides vulgatus ATCC 8482* и *Oscillibacter valericigenes Sjm18-20*, отсутствовавшие в кишечнике реципиентов до трансплантации, успешно прижились после ТФМ. Такая точная классификация стала возможна благодаря выделению малых

классов с помощью предложенного алгоритма и, следовательно, уменьшению шума при разбиении на корзины и таксономическом анализе.

Часть корзин имеет попарное расстояние меньше $0{,}2$, что указывает на близкие таксономические категории. Однако некоторые корзины от одного реципиента не имеют пересечений ни с одной из корзин другого реципиента. Это говорит о том, что определенные виды организмов прижились только у одного из реципиентов. В целом, результаты этого эксперимента хорошо коррелируют с наблюдениями из первого опыта (рис. 14) и результатами статьи [23].

Выводы по главе 3

В данной главе описаны методы валидации корректности работы алгоритма. Приведены результаты симуляционных запусков программы для оценки точности ее работы. Также приведены результаты работы программы на реальных метагеномных данных, которые сопоставлены с известными научными результатами. Получены результаты согласующиеся с текущими исследованиями и открывающие возможности для дальнейшего углубленного анализа серий метагеномов.

ЗАКЛЮЧЕНИЕ

В данной работе был проведен обзор существующих решений для анализа метагеномных данных. Описаны их недостатки в применении к анализу микробиоты кишечника человека, связанные с высокой сложностью и малой степенью изученности данной среды. Поставлена задача разработки новых, более точных методов для анализа серий метагеномных образцов.

Предложено два алгоритма для обнаружения прочтений из одного метагенома в другом метагеноме, основанных на выравнивании прочтений на граф де Брейна. На основе полученных алгоритмов разработан и реализован классификатор прочтений для серий метагеномных образцов.

Для тестирования разработанных алгоритмов были сгенерированы искусственные серии метагеномов с различными параметрам. Классификатор показал высокую точность работы на синтетических метагеномах. Также было проведено сравнение с существующим средством для обнаружения прочтений в графе де Брейна и программой для сравнительного анализа метагеномов. Новый алгоритм показал более высокую степень точности, чем каждый из существующих методов по отдельности, при использовании более ясных методов поиска. Использование нового метода совместно с существующими может позволить получить больше информации для анализа серий метагеномов.

Были проведены эксперименты на реальных данных с помощью предложенного алгоритма. Полученные результаты позволили выявить новые интересные зависимости, которые не противоречат существующим исследованиям.

Разработанный классификатор расширяет возможности для анализа серий метагеномов и может быть применен для исследования метагеномов людей, подвергшихся трансплантации фекальной микробиоты, для обнаружения ранее неизвестных принципов переноса микробиоты.

По результатам работы сделано выступление с докладом на Конгрессе молодых ученых в Университете ИТМО, который выиграл конкурс «За лучший доклад» на X сессии научной школы «Технологии программирования, искусственный интеллект, биоинформатика».

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 *Ansorge W. J.* Next-generation DNA sequencing techniques // New biotechnology. — 2009. — Vol. 25, no. 4. — P. 195–203.

2 Durable coexistence of donor and recipient strains after fecal microbiota transplantation / S. S. Li [et al.] // Science. — 2016. — Vol. 352, no. 6285. — P. 586–589.

3 *Wood D. E.*, *Salzberg S. L.* Kraken: ultrafast metagenomic sequence classification using exact alignments // Genome biology. — 2014. — Vol. 15, no. 3. — R46.

4 Basic local alignment search tool / S. F. Altschul [et al.] // Journal of molecular biology. — 1990. — Vol. 215, no. 3. — P. 403–410.

5 Census-based rapid and accurate metagenome taxonomic profiling / A. Shamsaddini [et al.] // BMC genomics. — 2014. — Vol. 15, no. 1. — P. 918.

6 Centrifuge: rapid and sensitive classification of metagenomic sequences / D. Kim [et al.] // Genome research. — 2016. — Vol. 26, no. 12. — P. 1721–1729.

7 Real-time DNA sequencing using detection of pyrophosphate release / M. Ronaghi [et al.] // Analytical biochemistry. — 1996. — Vol. 242, no. 1. — P. 84–89.

8 *Illumina I.* An introduction to next-generation sequencing technology. — 2015. — URL: `https://www.illumina.com/Documents/products/illumina_sequencing_introduction.pdf`.

9 Real-time DNA sequencing from single polymerase molecules / J. Eid [et al.] // Science. — 2009. — Vol. 323, no. 5910. — P. 133–138.

10 Continuous base identification for single-molecule nanopore DNA sequencing / J. Clarke [et al.] // Nature nanotechnology. — 2009. — Vol. 4, no. 4. — P. 265.

11 *Illumina I.* Quality Scores for Next-Generation Sequencing. Technical Note. Pub. No. 770-2011-030. — 2011. — URL: `https://www.illumina.com/Documents/products/technotes/technote_Q-Scores.pdf`.

12 MEGAHIT: an ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph / D. Li [et al.] // Bioinformatics. — 2015. — Vol. 31, no. 10. — P. 1674–1676.

13 metaSPAdes: a new versatile metagenomic assembler / S. Nurk [et al.] // Genome research. — 2017. — Vol. 27, no. 5. — P. 824–834.

14 Metagenomic species profiling using universal phylogenetic marker genes / S. Sunagawa [et al.] // Nature methods. — 2013. — Vol. 10, no. 12. — P. 1196.

15 Metagenomic microbial community profiling using unique clade-specific marker genes / N. Segata [et al.] // Nature methods. — 2012. — Vol. 9, no. 8. — P. 811.

16 MetaPhlAn2 for enhanced metagenomic taxonomic profiling / D. T. Truong [et al.] // Nature methods. — 2015. — Vol. 12, no. 10. — P. 902.

17 MetaFast: fast reference-free graph-based comparison of shotgun metagenomic data / V. I. Ulyantsev [et al.] // Bioinformatics. — 2016. — Vol. 32, no. 18. — P. 2760–2767.

18 Reference-independent comparative metagenomics using cross-assembly: crAss / B. E. Dutilh [et al.] // Bioinformatics. — 2012. — Vol. 28, no. 24. — P. 3225–3231.

19 *Uritskiy G. V.*, *DiRuggiero J.*, *Taylor J.* MetaWRAP—a flexible pipeline for genome-resolved metagenomic data analysis // Microbiome. — 2018. — Vol. 6, no. 1. — P. 158.

20 CheckM: assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes / D. H. Parks [et al.] // Genome research. — 2015. — Vol. 25, no. 7. — P. 1043–1055.

21 Anvi'o: an advanced analysis and visualization platform for 'omics data / A. M. Eren [et al.] // PeerJ. — 2015. — Vol. 3. — e1319.

22 Duodenal infusion of donor feces for recurrent Clostridium difficile / E. Van Nood [et al.] // New England Journal of Medicine. — 2013. — Vol. 368, no. 5. — P. 407–415.

23 Tracking microbial colonization in fecal microbiota transplantation experiments via genome-resolved metagenomics / S. T. Lee [et al.] // Microbiome. — 2017. — Vol. 5, no. 1. — P. 50.

24 Strain tracking reveals the determinants of bacterial engraftment in the human gut following fecal microbiota transplantation / C. S. Smillie [et al.] // Cell host & microbe. — 2018. — Vol. 23, no. 2. — P. 229–240.

25 MetaCherchant: analyzing genomic context of antibiotic resistance genes in gut microbiota / E. I. Olekhnovich [et al.] // Bioinformatics. — 2017. — Vol. 34, no. 3. — P. 434–444.

26 MetaSim—a sequencing simulator for genomics and metagenomics / D. C. Richter [et al.] // PloS one. — 2008. — Vol. 3, no. 10. — e3373.

27 Variation graph toolkit improves read mapping by representing genetic variation in the reference / E. Garrison [et al.] // Nature biotechnology. — 2018.

28 *Rautiainen M.*, *Mäkinen V.*, *Marschall T.* Bit-parallel sequence-to-graph alignment // bioRxiv. — 2018. — P. 323063.

29 BrownieAligner: accurate alignment of Illumina sequencing data to de Bruijn graphs / M. Heydari [et al.] // BMC bioinformatics. — 2018. — Vol. 19, no. 1. — P. 311.

30 COMMET: comparing and combining multiple metagenomic datasets / N. Maillet [et al.] // 2014 IEEE international conference on bioinformatics and biomedicine (BIBM). — IEEE. 2014. — P. 94–98.

31 Mash: fast genome and metagenome distance estimation using MinHash / B. D. Ondov [et al.] // Genome biology. — 2016. — Vol. 17, no. 1. — P. 132.

32 *Иванов А. Б.* Алгоритмы сравнительного анализа серий метагеномных образцов с использованием графов де Брейна для библиотек метагеномных чтений // Сборник тезисов докладов конгресса молодых ученых. — Электронное издание. СПб: Университет ИТМО, 2019.

ПРИЛОЖЕНИЕ А. НАБОР ВИДОВ КИШЕЧНЫХ БАКТЕРИЙ ДЛЯ СИМУЛЯЦИИ МЕТАГЕНОМОВ

— NZ_DS264586.1 Actinomyces odontolyticus ATCC 17982
— NC_010655.1 Akkermansia muciniphila ATCC BAA-835, complete genome
— NZ_KB290627.1 Anaerostipes hadrus DSM 3319 Scfld0, whole genome shotgun sequence
— NZ_CP011531.1 Bacteroides dorei CL03T12C01, complete genome
— NZ_CP012938.1 Bacteroides ovatus strain ATCC 8483, complete genome
— NZ_DS499677.1 Bacteroides stercoris ATCC 43183 Scfld_02_16, whole genome shotgun sequence
— NZ_DS362249.1 Bacteroides uniformis ATCC 8492 Scfld_3.0.1_32, whole genome shotgun sequence
— NC_009614.1 Bacteroides vulgatus ATCC 8482, complete genome
— NC_008618.1 Bifidobacterium adolescentis ATCC 15703 DNA, complete genome
— NC_012815.1 Bifidobacterium animalis subsp. lactis DSM 10140, complete genome
— NZ_AP012324.1 Bifidobacterium breve DSM 20213 = JCM 1192 DNA, complete genome
— NC_004307.2 Bifidobacterium longum NCC2705 chromosome, complete genome
— NZ_AAVN02000022.1 Collinsella aerofaciens ATCC 25986 C_aerofaciens-2.0_Cont809, whole genome shotgun sequence
— NZ_DS264419.1 Dorea longicatena DSM 13814
— NC_004668.1 Enterococcus faecalis V583 chromosome, complete genome
— NC_017960.1 Enterococcus faecium DO chromosome, complete genome
— NC_000913.3 Escherichia coli str. K-12 substr. MG1655, complete genome
— NC_012778.1 [Eubacterium] eligens ATCC 27750, complete genome
— NZ_ACEP01000175.1 [Eubacterium] hallii DSM 3353 E_hallii-1.0_Cont496.1, whole genome shotgun sequence
— NC_012781.1 [Eubacterium] rectale ATCC 33656, complete genome
— NZ_DS264288.1 Eubacterium ventriosum ATCC 27560
— NC_014106.1 Lactobacillus crispatus ST1 complete genome, strain ST1
— NC_008530.1 Lactobacillus gasseri ATCC 33323, complete genome

— NC_008526.1 Lactobacillus paracasei ATCC 334 chromosome, complete genome
— NC_009513.1 Lactobacillus reuteri DSM 20016, complete genome
— NC_002662.1 Lactococcus lactis subsp. lactis Il1403 chromosome, complete genome
— NZ_JH976524.1 Parabacteroides merdae CL09T00C40
— NC_010554.1 Proteus mirabilis strain HI4320
— NZ_KI260285.1 Ruminococcus callidus ATCC 27760 Scaffold0, whole genome shotgun sequence
— NZ_DS990209.1 Ruminococcus lactaris ATCC 29176 Scfld_02_46, whole genome shotgun sequence

ПРИЛОЖЕНИЕ Б. НАБОР ШТАММОВ КИШЕЧНОЙ ПАЛОЧКИ ДЛЯ СИМУЛЯЦИИ МЕТАГЕНОМОВ

- Escherichia_coli_042_uid161985
- Escherichia_coli_536_uid58531
- Escherichia_coli_55989_uid59383
- Escherichia_coli_ABU_83972_uid161975
- Escherichia_coli_APEC_O1_uid58623
- Escherichia_coli_APEC_O78_uid187277
- Escherichia_coli_ATCC_8739_uid58783
- Escherichia_coli_BL21_DE3__uid161947
- Escherichia_coli_BL21_DE3__uid161949
- Escherichia_coli__BL21_Gold_DE3_pLysS_AG__uid59245
- Escherichia_coli_B_REL606_uid58803
- Escherichia_coli_BW2952_uid59391
- Escherichia_coli_CFT073_uid57915
- Escherichia_coli__clone_D_i14__uid162049
- Escherichia_coli__clone_D_i2__uid162047
- Escherichia_coli_DH1_uid161951
- Escherichia_coli_DH1_uid162051
- Escherichia_coli_E24377A_uid58395
- Escherichia_coli_ED1a_uid59379
- Escherichia_coli_ETEC_H10407_uid161993
- Escherichia_coli_HS_uid58393
- Escherichia_coli_IAI1_uid59377
- Escherichia_coli_IAI39_uid59381
- Escherichia_coli_IHE3034_uid162007
- Escherichia_coli_JJ1886_uid226103
- Escherichia_coli_K_12_substr__DH10B_uid58979
- Escherichia_coli_K_12_substr__MDS42_uid193705
- Escherichia_coli_K_12_substr__MG1655_uid57779
- Escherichia_coli_K_12_substr__W3110_uid161931
- Escherichia_coli_KO11FL_uid162099
- Escherichia_coli_KO11FL_uid52593
- Escherichia_coli_LF82_uid161965

— Escherichia_coli_LY180_uid219461
— Escherichia_coli_NA114_uid162139
— Escherichia_coli_O103_H2_12009_uid41013
— Escherichia_coli_O104_H4_2009EL_2050_uid175905
— Escherichia_coli_O104_H4_2009EL_2071_uid176128
— Escherichia_coli_O104_H4_2011C_3493_uid176127
— Escherichia_coli_O111_H__11128_uid41023
— Escherichia_coli_O127_H6_E2348_69_uid59343
— Escherichia_coli_O157_H7_EC4115_uid59091
— Escherichia_coli_O157_H7_EDL933_uid57831
— Escherichia_coli_O157_H7_TW14359_uid59235
— Escherichia_coli_O157_H7_uid57781
— Escherichia_coli_O26_H11_11368_uid41021
— Escherichia_coli_O55_H7_CB9615_uid46655
— Escherichia_coli_O55_H7_RM12579_uid162153
— Escherichia_coli_O7_K1_CE10_uid162115
— Escherichia_coli_O83_H1_NRG_857C_uid161987
— Escherichia_coli_P12b_uid162061
— Escherichia_coli_PMV_1_uid219679
— Escherichia_coli_S88_uid62979
— Escherichia_coli_SE11_uid59425
— Escherichia_coli_SE15_uid161939
— Escherichia_coli_SMS_3_5_uid58919
— Escherichia_coli_UM146_uid162043
— Escherichia_coli_UMN026_uid62981
— Escherichia_coli_UMNK88_uid161991
— Escherichia_coli_UTI89_uid58541
— Escherichia_coli_W_uid162101
— Escherichia_coli_Xuzhou21_uid163995

ПРИЛОЖЕНИЕ В. СИМУЛЯЦИЯ НА СГЕНЕРИРОВАННЫХ МЕТАГЕНОМАХ С МОДЕЛЬЮ ОШИБОК SANGER В ПРОЧТЕНИЯХ

Для данной симуляции использовалась стратегия генерации метагеномов с использованием встроенной в программу MetaSim модели генерации ошибок Sanger. В данной модели вероятность ошибки основания зависит от его местоположения в прочтении и линейно растет от начала к концу прочтения. В качестве параметров были зафиксированы errStart $=$ $0{,}0001$ и errEnd $=$ $0{,}001$ – вероятности ошибок первого и последнего основания в прочтении соответственно. Соотношение типов ошибок *инсерция : делеция : замена* было установлено равным $1 : 1 : 3$. Остальные параметры генерации метагеномов были такие же, как и на первом этапе тестирования.

Результаты проверки качества полученных классов приведены в табл. В.1 и В.2 При выборе параметре threshold $=$ $0{,}0001$ результаты классификации хуже, чем при работе на прочтениях без ошибок. При выборе параметре threshold $=$ $0{,}001$ результаты совпадают с результатами классификации на прочтениях без ошибок. Из этого можно сделать выводы, что алгоритм ожидаемо работает хуже при наличии ошибок в прочтениях, однако он правильно классифицирует геномы с достаточным уровнем точности при выборе относительно малого параметра отсечения.

Таблица В.1 – Результаты тестирования классификатора на сгенерированных метагеномах с моделью ошибок Sanger в прочтениях. Параметр threshold $= 0{,}0001$

	Вероятность ошибки первого рода, %	Вероятность ошибки второго рода, %
Виды с равномерным распределением	$21{,}22$	$< 0{,}01$
Виды с экспоненциальным распределением	$21{,}98$	$< 0{,}01$
Штаммы с равномерным распределением	$34{,}90$	$0{,}32$
Штаммы с экспоненциальным распределением	$34{,}43$	$0{,}16$

Таблица В.2 – Результаты тестирования классификатора на сгенерированных метагеномах с моделью ошибок Sanger в прочтениях. Параметр threshold $= 0{,}001$

	Вероятность ошибки первого рода, %	Вероятность ошибки второго рода, %
Виды с равномерным распределением	$9{,}83$	$< 0{,}01$
Виды с экспоненциальным распределением	$9{,}37$	$< 0{,}01$
Штаммы с равномерным распределением	$24{,}10$	$0{,}32$
Штаммы с экспоненциальным распределением	$19{,}92$	$0{,}16$

ПРИЛОЖЕНИЕ Г. ТАКСОНОМИЧЕСКАЯ АННОТАЦИЯ КОРЗИН НАБОРОВ ПРОЧТЕНИЙ

Таблица Г.1 – Результаты таксономической классификации корзин прочтений, собранных из класса *пришли от донора* для реципиента R01

bin_1	Bacteria; Terrabacteria group; Firmicutes; Clostridia; Clostridiales
bin_2	Archaea; Euryarchaeota; Methanomada group; Methanobacteria; Methanobacteriales; Methanobacteriaceae; Methanobrevibacter; Methanobrevibacter smithii; Methanobrevibacter smithii ATCC 35061
bin_3	Bacteria; FCB group; Bacteroidetes/Chlorobi group; Bacteroidetes; Bacteroidia; Bacteroidales; Rikenellaceae; Alistipes; Alistipes finegoldii; Alistipes finegoldii DSM 17242
bin_4	Bacteria; Terrabacteria group; Firmicutes; Clostridia; Clostridiales; Lachnospiraceae
bin_5	Bacteria; Terrabacteria group; Firmicutes; Clostridia; Clostridiales; Oscillospiraceae; Oscillibacter; Oscillibacter valericigenes; Oscillibacter valericigenes Sjm18-20
bin_6	Bacteria; Terrabacteria group; Actinobacteria; Coriobacteriia; Eggerthellales; Eggerthellaceae; Slackia; Slackia heliotrinireducens; Slackia heliotrinireducens DSM 20476
bin_7	Bacteria; FCB group; Bacteroidetes/Chlorobi group; Bacteroidetes; Bacteroidia; Bacteroidales; Bacteroidaceae; Bacteroides; Bacteroides vulgatus; Bacteroides vulgatus ATCC 8482
bin_8	Bacteria; Terrabacteria group; Firmicutes; Clostridia; Clostridiales
bin_9	Bacteria
bin_10	Bacteria; FCB group; Bacteroidetes/Chlorobi group; Bacteroidetes; Bacteroidia; Bacteroidales; Bacteroidaceae; Bacteroides; Bacteroides helcogenes; Bacteroides helcogenes P 36-108
bin_11	Bacteria; Terrabacteria group; Firmicutes; Clostridia; Clostridiales
bin_12	Bacteria; Terrabacteria group; Firmicutes; Clostridia; Clostridiales; Clostridiaceae; Clostridium

Таблица Г.2 – Результаты таксономической классификации корзин прочтений, собранных из класса *пришли от донора* для реципиента R02

bin_1	Bacteria; Terrabacteria group; Firmicutes; Clostridia; Clostridiales; Lachnospiraceae; Roseburia; Roseburia hominis; Roseburia hominis A2-183
bin_2	Bacteria; FCB group; Bacteroidetes/Chlorobi group; Bacteroidetes; Bacteroidia; Bacteroidales; Tannerellaceae; Parabacteroides; Parabacteroides distasonis; Parabacteroides distasonis ATCC 8503
bin_3	Bacteria; FCB group; Bacteroidetes/Chlorobi group; Bacteroidetes; Bacteroidia; Bacteroidales; Prevotellaceae; Prevotella
bin_4	Bacteria; FCB group; Bacteroidetes/Chlorobi group; Bacteroidetes; Bacteroidia; Bacteroidales; Bacteroidaceae; Bacteroides; Bacteroides helcogenes; Bacteroides helcogenes P 36-108
bin_5	Bacteria; FCB group; Bacteroidetes/Chlorobi group; Bacteroidetes; Bacteroidia; Bacteroidales; Bacteroidaceae; Bacteroides
bin_6	Bacteria; FCB group; Bacteroidetes/Chlorobi group; Bacteroidetes; Bacteroidia; Bacteroidales; Bacteroidaceae; Bacteroides; Bacteroides helcogenes; Bacteroides helcogenes P 36-108
bin_7	Bacteria; Proteobacteria; Betaproteobacteria; Burkholderiales; Burkholderiaceae; Burkholderia
bin_8	Bacteria; FCB group; Bacteroidetes/Chlorobi group; Bacteroidetes; Bacteroidia; Bacteroidales; Bacteroidaceae; Bacteroides; Bacteroides thetaiotaomicron; Bacteroides thetaiotaomicron VPI-5482
bin_9	Bacteria; Terrabacteria group; Firmicutes; Clostridia; Clostridiales
bin_10	Bacteria; FCB group; Bacteroidetes/Chlorobi group; Bacteroidetes; Bacteroidia; Bacteroidales; Rikenellaceae; Alistipes; Alistipes finegoldii; Alistipes finegoldii DSM 17242
bin_11	Bacteria; FCB group; Bacteroidetes/Chlorobi group; Bacteroidetes; Bacteroidia; Bacteroidales; Tannerellaceae; Parabacteroides; Parabacteroides distasonis; Parabacteroides distasonis ATCC 8503
bin_12	Bacteria; Proteobacteria
bin_13	Bacteria; Terrabacteria group; Firmicutes; Clostridia; Clostridiales

Продолжение таблицы Г.2

bin_14	Bacteria; FCB group; Bacteroidetes/Chlorobi group; Bacteroidetes; Bacteroidia; Bacteroidales; Bacteroidaceae; Bacteroides; Bacteroides helcogenes; Bacteroides helcogenes P 36-108
bin_15	Bacteria
bin_16	Bacteria; Terrabacteria group; Firmicutes; Clostridia; Clostridiales; Lachnospiraceae; unclassified Lachnospiraceae; [Eubacterium] rectale; [Eubacterium] rectale ATCC 33656
bin_17	Bacteria; FCB group; Bacteroidetes/Chlorobi group; Bacteroidetes; Bacteroidia; Bacteroidales; Bacteroidaceae; Bacteroides; Bacteroides helcogenes; Bacteroides helcogenes P 36-108
bin_18	Bacteria; FCB group; Bacteroidetes/Chlorobi group; Bacteroidetes; Bacteroidia; Bacteroidales; Bacteroidaceae; Bacteroides; Bacteroides vulgatus; Bacteroides vulgatus ATCC 8482
bin_19	Bacteria; FCB group; Bacteroidetes/Chlorobi group; Bacteroidetes; Bacteroidia; Bacteroidales; Bacteroidaceae; Bacteroides; Bacteroides vulgatus; Bacteroides vulgatus ATCC 8482
bin_20	Bacteria; Terrabacteria group; Firmicutes; Clostridia; Clostridiales; Oscillospiraceae; Oscillibacter; Oscillibacter valericigenes; Oscillibacter valericigenes Sjm18-20