Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики

Факультет информационных технологий и программирования
Кафедра компьютерных технологий

Авдеев Павел Вадимович

Улучшение алгоритма реконструкции родительских геномов и разработка алгоритма построения филогенетических деревьев

Научный руководитель: Алексеев Максим Александрович

Санкт-Петербург
2013

Содержание

Введение

Благодаря развитию и удешевлению технологий секвенирования, появлению разнообразных геномных ассемблеров, а так же разнообразных инициатив [1], призывающих секвенировать как можно больше организмов, количество информации о геномах различных организмов значительно увеличилось. В результате возникла возможность более точно выявлять закономерности организации и эволюции геномов. Так, в области филогенетики, идентифицирующей и проясняющей эволюционные взаимоотношений среди разных видов жизни на Земле, как современных, так и вымерших, появились новые методы наблюдения эволюционных событий, основанные на сравнении геномов, как последовательностей синтени-блоков. Эволюционные изменения генома, представленными последовательностями синтени-блоков, можно охарактеризовать геномными перестройками.
Таким образом на основе набора родственных геномов организмов можно пытаться ответить на следующие вопросы:

- В какое наиболее вероятное дерево организованы организмы, отражающее эволюционные взаимосвязи между ними и их предками.
- Как выглядели геномы их предков.
- Какие события произошли на этапе эволюции, которые привели геномы предков к текущим геномам организмов.

К сожалению, все существующие алгоритмы имеют ограничения и не отвечают наиболее полно на все эти вопросы. Например, GRAPPA [2] алгоритм и MGR [3] алгоритм не различают надежные и ненадежные перестановки и акцентируются на кратчайшем расстояние между геномами. Алгоритм inferCARS [4] предполагает заданное филогенетическое дерево, которое на основе этих данных еще как-то надо получить. EMRAE [5] алгоритм имеет существенные ограничения, так как не пытается реконстру-

ировать филогенетические деревья, а так же ограничен однохромосомными геномами. Но для всех перечисленных алгоритмов наложено ограничение, исключающее наличие дуплицироанных генов, которое приемлимо для большинства вирусов и митохондрий, или для очень близкородственных геномов, где сохраняются целые участки хромосом, а не отдельные гены. Поэтому часто ограничение на отсутстиве дуплицируемых блоков не выполняется. Так как в ходе эволюции какие-то учатски генома предка могут дуплицироваться, геномы-потомки могут содержать несколько экземпляров некоторых синтени-блоков: изначальные экземпляр, сохранившийся от общего предка, и его копии.

В данной работе описывается обобщение MGRA [6] алгоритма для данных, содержащих дуплицируемые синтени-блокиб, описывается разработанный алгоритм реконструкции филогенетических деревьев. Так же дается описание сервера, который был разработан для облегчения работы биологов с данным алгоритмом.

Глава 1. Обзор предметной области

1.1. Биоинформатика

Во многих задачах современной биологии и медицины необходимо работать с большими объемами данных, поэтому для их решения используется вычислительная техника. Таким образом биоинформатика представляет из себя междисциплинарную область науки, которая разрабатывает и усовершенствует методы хранения, поиска, организации и анализа биологических данных.

1.2. Общие сведения о хромосомной эволюции

Перестановки - это геномные "землетрясения изменяющие хромосомную архитектуру. Фундаментальным вопросом в молекулярной эволюции существуют ли "горячие"регионы в хромосомах, где перестановки случаются снова и снова, благодаря высокой вероятности разрыва на этом участке.

1.2.1. Random Breakage модель против Fragile Breakage модель в хромосомной эволюции

В 1970 году Сусуму Оно привел две фундаментальные гипотезы хромосомной эволюции, которые были предметом споров в последние 40 лет. Одна из них Random Breakage Model предложенная Оно [7] и формализованная Nadeau и Taylor [8].

Гипотеза 1.1. *Random Breakage Model(RBM) постулирует возможность перестановок в случайных геномных позициях, таким образом подразумевая низкое переиспользование перестановок в конкретных регионах генома.*

Из-за своей пророческой силы предсказания, RBM стало де-факто теории хромосомной эволюции. Только в 2003 году Певзнер и Теслер [9] опровергли RBM и предложили альтернативную модель хромосомной эволюции Fragile Breakage Model.

Гипотеза 1.2. *Fragile Breakage Model(FBM) постулирует, существование "хрупких"геномных регионов, которые с большей вероятностью подвержены перестановкам, чем остальные части генома, подразумевая высокий уровень повторного использования перестановок в этих "горячих"точках.*

Различные дополнительные исследования подтвердили существование хрупких регионов у огранизмов. Например, Кикута [10] проанализировал связь между хрупкими участками генома необходимость сохранения нетронутыми регуляторных элементов генома и пришел к выводу, что модель RBM ошибочна.

Так как модель FBM выполнена, то существуют консервативные сегменты генома, которые не подвержены перестановкам. Такие регионы называются блоками синтении. Так как геном состоит из одной или более хромосом, которая содержит наследственную информацию в молекуле ДНК. ДНК имеет двухцепочечную структуру, и транскрипция с основной и комплементарной цепей осуществляется в противоположных направлениях, поэтому хромосому можно представить, как знаковую перестановку синтени-блоков. Знак “+” будет соответствовать прямой ориентации, а знак “-” обратной. Существует достаточное количество алгоритмов [11, 12], которые позволяют на основе геномов, состоящих из последовательности нуклеотидов, получать геномы представленные набором блоков синтении.

1.2.2. Операции перестановок

Если геном представляет собой набор из блоков синтени, описанные в разделе 1.2.1, тогда можно определить возможные геномные перестройки как операции над знаковой перестановкой:

1. Инверсия или reversal

$$\pi = (p_1 \ \dots \ p_{i-1}p_i \ \dots \ p_jp_{j+1} \ \dots \ p_n)$$

$$\Downarrow$$

$$\pi' = (p_1 \ \dots \ p_{i-1}p_j \ \dots \ p_ip_{j+1} \ \dots \ p_n)$$

2. Транслокация или translocation

$$\pi = (p_1 \ \dots \ p_{i-1}p_i \ \dots \ p_n) \ \sigma = (s_1 \ \dots \ s_{j-1}s_j \ \dots \ s_m)$$

$$\Downarrow$$

$$\pi' = (p_1 \ \dots \ p_{i-1}s_j \ \dots \ s_n) \ \sigma' = (s_1 \ \dots \ s_{j-1}p_i \ \dots \ p_n)$$

3. Слияние или fission

$$\pi = (p_1 \ \dots \ p_n) \ \sigma = (s_1 \ \dots \ s_m)$$

$$\Downarrow$$

$$\pi' = (p_1 \ \dots \ p_ns_1 \ \dots \ s_m)$$

4. Расщепление или fusion

$$\pi = (p_1 \ \dots \ p_{i-1}p_i \ \dots \ p_n)$$

$$\Downarrow$$

$$\pi' = (p_1 \ \dots \ p_{i-1}) \ \sigma' = (p_i \ \dots \ p_n)$$

Стандартные перестановочные операции приведенные выше, могут быть обобщены введением так называемых 2 - break операций в геноме.

Определение 1.3. 2 - break - это операция которая производит разрыв в двух местах генома и склеивает получившиеся фрагменты в новом порядке.

Можно предположить о существование k - break операции в ходе эволюции.

Определение 1.4. k - break - это операция которая делает k разрывов в геноме и склеивает получившиеся фрагменты в новом порядоке.

Множество биологов верят, что k - break перестановки маловероятны для $k > 3$ и относительно редко для $k = 3$. Действительно, биофизические и селективные ограничения серьезны уже для $k = 2$, не говоря уже о $k > 2$. Однако, 3 - break операция, называмая транспозицией, несомненно случается в эволюции, хотя до сих пор не ясно, как часто она происходила в эволюции разнообразных организмов.

- Транспозиция или transposition

$$\pi = (p_1 \ \dots \ p_{i-1}p_i \ \dots \ p_jp_{j+1} \ \dots \ p_kp_{k+1} \ \dots \ p_n)$$

$$\Downarrow$$

$$\pi' = (p_1 \ \dots \ p_{i-1}p_{j+1} \ \dots \ p_kp_i \ \dots \ p_jp_{k+1} \ \dots \ p_n)$$

Задача о нахождении сценария геномных перестроек эквивалентна поиску последовательности приведенных выше операций, преобразующий один набор знаковых перестановок к другому. Так как крупные геномные перестройки на уровне популяций происходят редко, биологи заинтересованы в нахождении кратчайшей последовательности перестроек между геномами разных видов. Тем не менее, в случае сложных комбинаций перестроек, даже кратчайших сценариев может быть несколько. Поэтому будем считать решением этой задачи нахождение любого возможного кратчайшего сценария.

1.2.3. Whole Genome Duplication модель

Whole Genome Duplication модель это вторая гипотеза предложенная Сусуму Оно, которая постулирует новый тип эволюционных событий, которые дублируют некоторый регион генома. Эта гипотеза была предметом споров в течение долгих лет и только в 2004 году было доказана корректность этого утверждения. Келлис [13] рассмотрел геном дрожжей K.waltii и сравнил с геномом дрожжей S.cerevisiae, и продемонстрировал, что почти каждый регион в R.waltii соотносится с двумя регионами в S.cerevisiae, тем самым доказав, что было целое множество событий дубли-

рования геномов в ходе эволюции дрожжей. За этим открытием быстро последовали открытия дупликации генома у позвоночных и растений. Дехал и Буур [14] нашли доказательство существования двух этапов дупликации генома на эволюционном пути от ранних позвоночных к человеку. Вскоре после этого Майер и Ван Де Пир [15] нашли этап геномных дупликаций у лучеперых рыб.

Эти недавние исследования обеспечивают неопровержимые доказательства, что whole genome duplication представляет новый тип событий, который может объяснить феномены, которые классическое эволюционное учение с трудом объясняет и поэтому очень важно научить существующие алгоритмы работать с такими геномами.

1.3. Парный breakpoint граф

Впервые, определение breakpoint графа было введено Ханхали и Певзнером [16, 17], для разработки полиномиального алгоритма вычисления расстояния реверсиями или reversal distance между двумя знаковыми перестановками.

Определение 1.5. reversal distance — минимальное количество реверсий, необходимых для преобразования одной перестановки в другую.

Так как геном, согласно разделу 1.2.1, является знаковой перестановкой, то примение представления генома ввиде граф возможно. Для простоты понимания, начнем изучения графа в предположение, что геном состоит из одной циклической хромосомы, позже распостранив представление графа для линейных хромосом и мультихромосомных геномов.

1.3.1. В случае циклической хромосомы в унихромосомном геноме

Будем представлять геном P состоящий из одной циклической хромосомы, сформированный блоками синтении $x_1 \dots x_n$, как цикл с n направленными ребрами(соответствующие блокам) и с n ненаправленными

непомеченными ребрами(соответсвующие соединением блоков). Пример такого представления генома проиллюстрирован на рис. 1.1. Направление ребер соответсвует знакам блоков. Мы маркируем начало и конец каждого ребра x_i, как x_i^h и x_i^t соответственно. Вершины в хромосоме соединенные ненаправленными ребрами называются прилегающими.

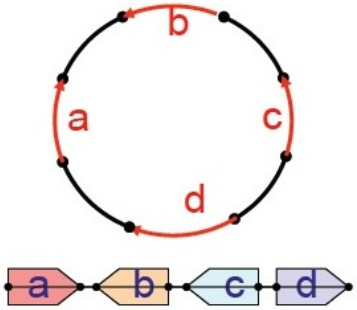

Рис. 1.1: Представление генома ввиде цикла

Пусть P и Q это циклические знаковые перестановки(унихромосомные геномы) над одним и тем же набором блоков синтении Δ.

Определение 1.6. Breakpoint граф $G(P, Q)$ - это граф на наборе вершин $V = \{x^t, x^h | x \in \Delta\}$ с ребрами трех цветов: пунктиные(соединяющие x_i^h и x_i^t), черные(соединяющие прилегающие блоки в геноме P) и серые или зеленые(соединяющие прилегающие блоки в геноме Q).

Пример графа показан на рис. 1.2. Можно заметить, что каждая пара ребер задают переменные циклы(цвет ребер чередуется) в графе G. Черные и серые(зеленые) ребра формируют переменный черно - серый(черно - зеленый) цикл, который играет важную роль в анализе перестановок.

1.3.2. В случае линейной хромосомы в унихромосомном геноме

Теперь, пусть геном P состоит из одной линейной хромосомы, сформированной знаковыми блоками синтении $x_1 \dots x_n$. Будем представлять геном, как путь из n – пунктирных ребер(кодирующие блок и его направле-

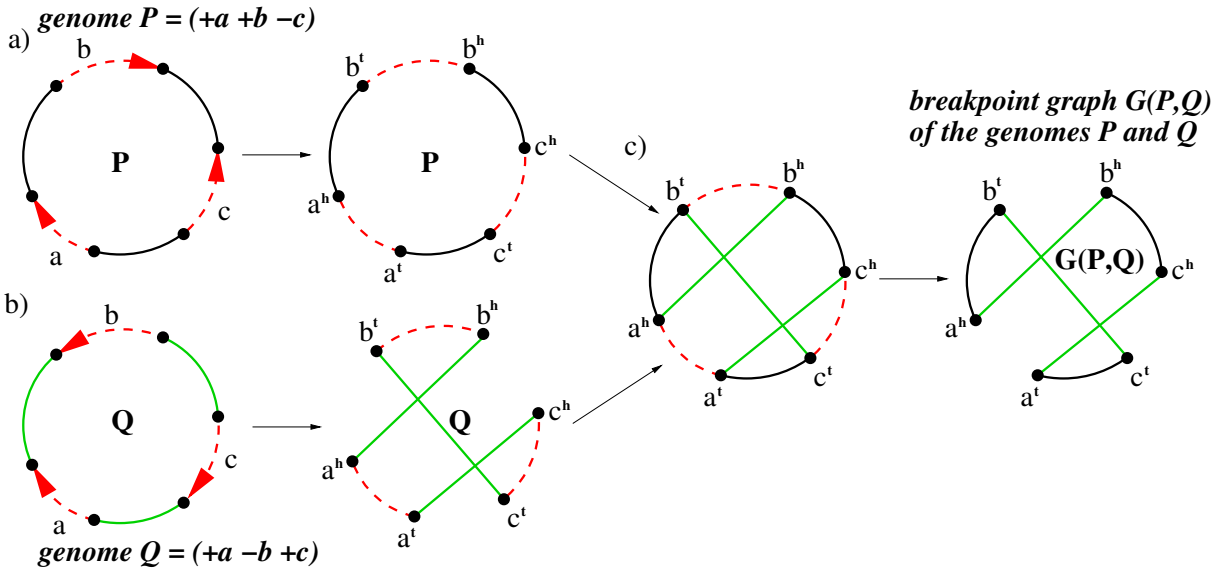

Рис. 1.2: Пример breakpoint графа для двух геномов

ние), а так же с $n-1$ ненаправленными черными ребрами(соединяющие прилегающих блоки). Так же введем новую вершину ∞ и соединим ненаправленными (нерегулярными) черными ребрами с каждыми вершинами представляющими концы хромосомы.

Определение 1.7. Точка в breakpoint графе называется регулярной, если она не является вершиной ∞.

Определение 1.8. Ребро называется регулярным, если обе вершины инцидентные этому ребру регулярные (Ребро называется нерегулярным в остальных случаях).

В таком представление геном состоящий из одной линейной хромосомы - это путь из черных и пунктирных ребер, начинающихся и кончающихся в вершине ∞.

1.3.3. В случае мультихромосомного генома

Расширим наше определение breakpoint графа для генома, состоящего из любого количества хромосов. Breakpoint граф для мультихромосомного генома, отличается от графа унихромосомного генома только тем, что теперь содержит коллекцию непересекающихся циклов (хромосом) с двумя чередующимися цветами: черный, для ненаправленных ребер

и пунктирные цвет для направленных ребер. Мы не будем явно показывать направление ребер, так как они определяются индексом t или h. Отдельно стоит отметить, что в случае линейных хромосом, степень вершины ∞ в breakpoint графе в два раза больше числа хромосом.

1.3.4. Операции над парным breakpoint графом

Для того чтобы проводить дальнейшие филогенетические исследования, нам необхожимо определить геномные перестройки введеные в разделе 1.2.2 для breakpoint графа. Дадим определение, введеной 2-break операции в терминах breakpoint графа:

Определение 1.9. Для любых двух черных ребер графа (u, v) и (x, y) в графе (геноме) P мы определим 2 - break операцию, как замену этих ребер, либо на пару ребер (u, x) и (v, y), либо пару (u, y) и (v, x).

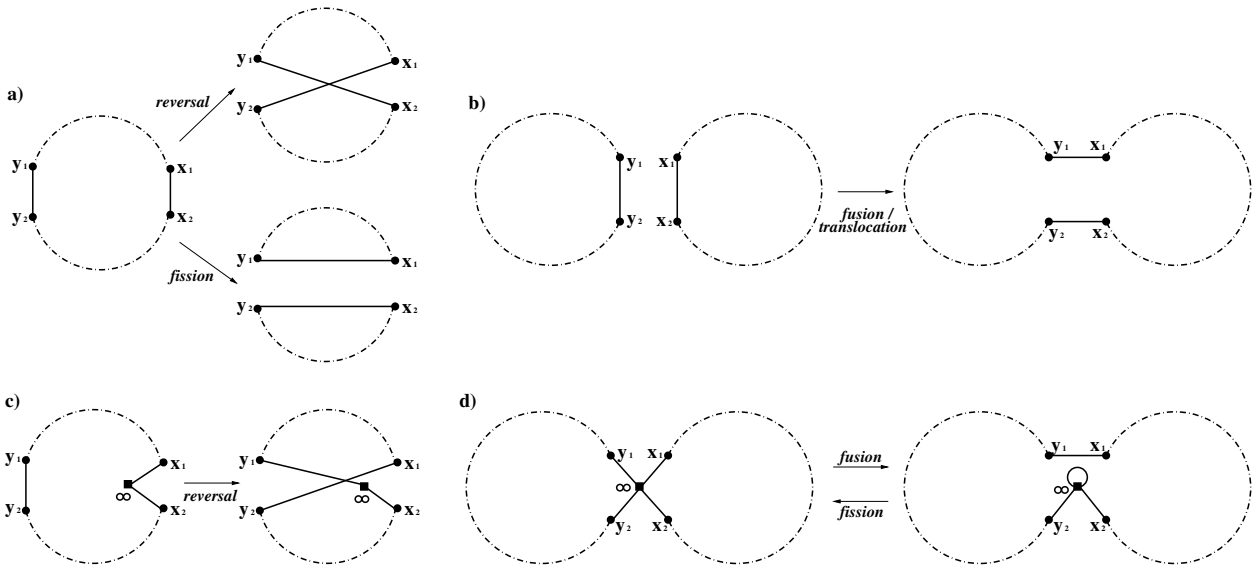

Рис. 1.3: Примеры 2 - break операций на breakpoint графе

Примеры всех таких 2 - break, соответствующие стандартным перестановочным операциям: инверсия, транслокация, деление, слияние, приведены на рис. 1.3. Ключевым наблюдением в исследование парных геномных перестановок является то, что каждая 2 - break трансформация из “черного” генома P в “зеленый“("серый") геном Q, соответствует трансформации

парного breakpoint графа G(P, Q) в идентичный breakpoint граф G(Q, Q) 2 - breakами(см рис. 1.4).

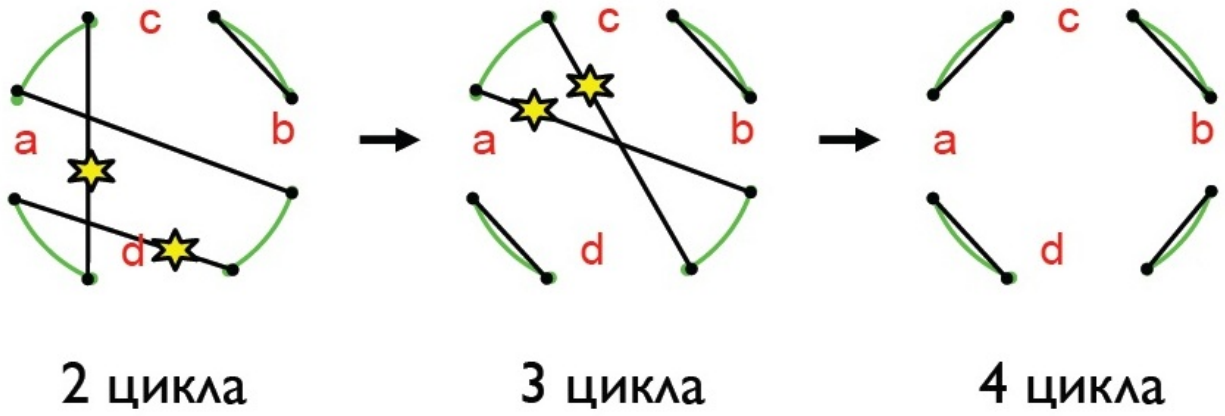

Рис. 1.4: Пример трансформации парного breakpoint графа G(P, Q) в индентичный breakpoint граф G(Q, Q)(самый правый граф).

Логично пытаться посчитать хотя бы минимальное количество 2 - break операций, требующиеся для трансформации одного генома в другой. Введем определения расстояние между двумя геномами:

Определение 1.10. 2 - break расстояние $d_2(P, Q)$ между геномами P и Q определяется, как минимальное число 2 - break, требующихся для трансформации одного генома в другой.

Но как оказалось такое расстояние в терминах парного breakpoint графа можно находить за полиномиальное время [16, 17], так как существует достаточно простая формула дающая это значение:

$$d_2(P, Q) = b(P, Q) - c(P, Q)$$

где $b(P, Q) = |\Delta|$ - это количество блоков синтении в P и Q, и $c(P, Q)$ – это количество черно-зеленых(черно-серых) циклов в G(P, Q).

Стоит отдельно отметить, что в случае линейных хромосом 2 - break операции, включает нерегулярные ребра (см. рис. 1.3), затрагивающие концы хромосом. В таком случае анализ стандартных операций, создает дополнительные алгоритмические проблемы по сравнению с анализом 2 - break в циклических хромосомах. Однако, сценарий перестановок в линейных хромосомах очень хорошо аппроксимируется сценарием в циклических хромо-

сомах. Таким образом, при использование 2 - break операций в циклических геномах нужно учитывать, что это может привести к нарушению линейности хромосом и создание циклических хромосом.

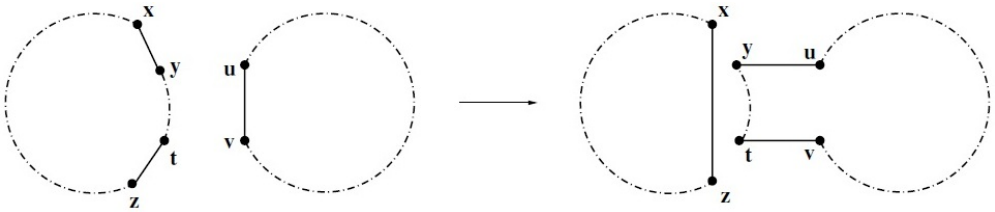

Рис. 1.5: Пример 3 - break операций на breakpoint графе

В разделе 1.2.2 вводилась еще одна операция, называемая транспозицией, которая соответствовала 3 - break операции. Дадим ее определение в терминах breakpoint графа:

Определение 1.11. Для любых трех черных ребер графа (u, v) и (x, y) и (z, t) в графе (геноме) P мы определим операцию транспозиции, как замену этих ребер, либо на тройку ребер (z, x) и (u, y) и (t, v), либо на тройку ребер (z, x) и (y, v) и (u, t).

Пример, такой операции приведен на рис. 1.5. Стоит отметить, что общее определение для 3 - break операций, включает в себя и 2 - break операции. Разные оссобенности 3 - break операции и нахождение расстояния 3 - break расстояния $d_3(P, Q)$ между геномами P и Q здесь рассммотрено не будет, так как в дальнейшем это не используется. Подробную информацию об этом можно прочесть в этих статьях [18–20].

1.4. Breakpoint граф для произвольного количества геномов

В разделе 1.3 breakpoint граф был ограничен двумя геномами. MGRA алгоритм активно использует breakpoint граф построенный на любом количетсве геномов, поэтому ниже приводится обобщение breakpoint графа для n геномов, где $n \geq 2$.

1.4.1. Множественный breakpoint граф

Пусть нам даны произвольные геномы $P_1 \dots P_n$ над одним и тем же множеством блоков синтении Δ. Точно так же, как для парного брейкпоинт графа, множественный breakpoint граф $G(P_1, \dots P_n)$ – является суперпозицией геномов (графов) $P_1, \dots P_n$ над одним и тем же множеством вершин $V = \{b^t, b^h | b \in \Delta\} \cup \{\infty\}$.

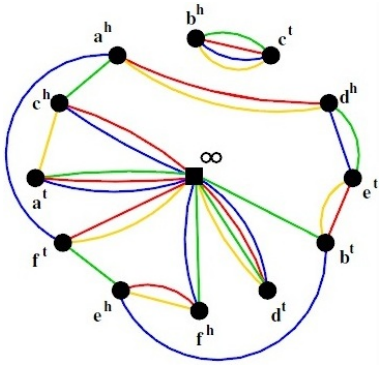

Рис. 1.6: Множественный breakpoint граф $G(P_1, P_2, P_3, P_4)$

Рисунок 1.6 демонстрирует нам множественный breakpoint граф. Введем ряд определений для удобства дальнейшего использования множественного breakpoint графа.

Ребра из $G(P_1, \dots P_n)$ представлены ненаправленными ребрами от геномов $P_1, \dots P_n$ с n различными цветами (следовательно, степень каждой регулярной вершины – k). Для упрощения обозначений, мы используем $P_1, \dots P_n$ для обозначения цветов ребер в множественном breakpoint графе и обозначим множество всех цветов $C = \{P_1 \dots P_n\}$.

Определение 1.12. Любое не пустое подмножество множества C называется мультицветом или multicolor.

Все ребра соединяющие вершины x и y в множественном breakpoint графе формируют мультиребро или multi-edge (x, y) с мультицветом, состоящим из цветов этих ребер. Мультиребра, соответствуют соседним блокам синтении, которые сохраняются на несколько видов существ и представляют ценные филогенетические характеристики. На рис. 1.6 мультиребро

(e^h, f^h) имеет мультицвет (P_3, P_4), представленные красными и желтыми ребрами.

Определение 1.13. Число мультиребер инцидентных вершине (так же равны числу соседних вершин) называется мультистепенью.

Заметим, что мультистепень может быть меньше обычной степени вершины. Например, на рис. 1.6 вершина e^h имеет степень 4, а мультистепень равна 3.

Так же как и разделе 1.3.4 дадим определение множественного индентичного breakpoint графа, к которому мы хотим привести наш изначальный множественный breakpoint граф.

Определение 1.14. Множественный breakpoint граф называется идентичным breakpoint графом $G(X, \ldots X)$ некоторого генома X если он состоит из полных мультиребер, то есть мультиребра из мультицвета (мультистепень каждой веришны равна единице).

1.4.2. Операции над множественным breakpoint графом

В случае $n \geq 2$ геномов $P_1, \ldots P_n$ на множественном breakpoint графе $G(P_1, \ldots P_n)$ существует $(2^n - 2)$ видов 2 - breakов, столько же, сколько различных мультицветов, сформированных собственными подмножествами из множества C. Каждый такой 2 - break может быть применен к мультиребрам. Однако, не каждая серия таких 2 - breakов имеет смысл с точки зрения реконструкции предковых геномов. Базовым свойством для реконструкции предка геномов, является то, что 2-break на мультиребро с мультицветом $Q \in C$, может быть применена только тогда, когда все геномы, соответствующие цветам из Q могут быть объединены в единый предковый геном. Дадим другое определение такой серии 2 - break:

Определение 1.15. Трансформация (серия 2 - breakов) S на множественном breakpoint графе $G(P_1, P_2, \ldots P_n)$ является надежной, если для любой пары 2-break операций (ρ_1 и ρ_2) на мультиреберах из мультицветов Q_1 и Q_2 выполнено $Q_1 \in Q_2$ тогда ρ_1 предшествуют ρ_2 в S.

1.5. Вывод по главе 1

В данной граве были описаны две модели: Random Breakage модель, Fragile Breakage модель. Дано обоснование представления геномов ввиде блоков синтении, исходя из выбора Fragile Breakage модели. Приведена формализация геномных перестановок, как операций над знаковой перестановкой. Так же дано обоснование существования событий дупликации в геномах.

Подробно описан парный breakpoint граф для различных видов генома. Так же переформулированы определения операций над знаковыми перестановками в терминах парного breakpoint графа. Указана основная идея перестановочного анализа в терминах парного breakpoint графа.

Дано подробное обобщение представление breakpoint графа для произвольного количества геномов. Опередедены 2 - break операции над множественным breakpoint графом. Обобщена идея перестановочного анализа в терминах множественного breakpoint графа. Введено опередедение надежной трансформации.

Глава 2. Постановка задачи

2.1. Теоретическая постановка задачи

Как отмечалось в разделе 1.3.4, ключевым анализом геномных перестроек является сведение с помощью 2 - break операций breakpoint графа к индентичному breakpoint графу. Предположим, что в отчилие от стандартного анализа breakpoint графа, базирующегося только на 2 - break операциях на черных ребрах, у нас возможны 2 - break операции либо на черных, либо на зеленых(серых) ребрах. Из этого предположения получаем, что происходит приведение breakpoint графа $G(P, Q)$ не к индентичному breakpoint графу $G(Q, Q)$, а к некому индентичному breakpoint графу $G(X, X)$. Но наша серия 2 - break операций над черными и зелеными(серыми) ребрами, соответсвует трансформации $P \to X \to Q$ с помощью m 2 - break операций на черных ребрах. Переход от 2 - break операций на черных ребрах к смеси 2 - break операций на черных и зеленых(серых) ребрах является простой, но мощной парадигмой, которая оказалась полезной в предыдущих исследованиях [3, 21]. Поэтому, вместо поиска кратчайшей трансформации $G(P, Q) \to G(Q, Q)$ будем искать кратчайшую трансформацию $G(P, Q)$ в любой идентичный breakpoint граф $G(X, X)$ без знаний о графе(геноме) X заранее.

Аналогично для множественного breakpoint графа $G(P_1, P_2, \ \dots \ P_n)$ сосредоточимся на поиске кратчайшей трансформации в любой идентичный множественый breakpoint граф $G(X, X \ \dots \ X)$ для некоторого неизвестного генома(графа) X. Формализуем множественную геномно перестановочную проблему или Multiple Genome Rearrangement problem(MGRP) в терминах breakpoint графов следующим образом:

Теорема 2.1. *Multiple Genome Rearrangement problem*
Для данных геномов $P_1, P_2 \ \dots \ P_n$ найти кратчайшую надежную серию из 2-break операций, которая трансформирует множественный breakpoint

граф $G(P_1, P_2 \ \ldots \ P_n)$ *в идентичный множественный breakpoint граф.*

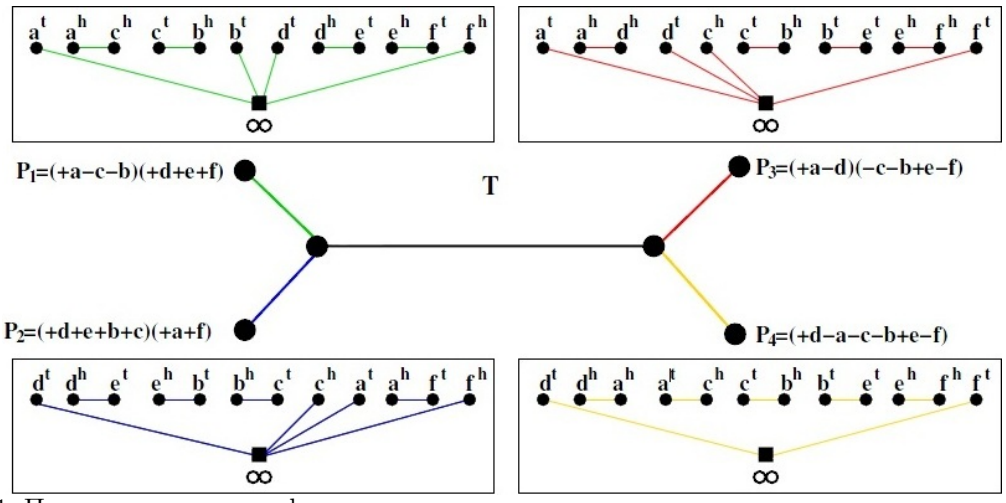

Рис. 2.1: Пример некорневого филогенетического дерева в листьях, которого содержатся геномы

Рассмотрим ситуацию когда есть информация о некорневом филогенетическом дереве T из геномов $P_1, \ldots P_n$ (см. рис. 2.1). Дерево T состоит из n – листьев, $(n-2)$ – внутренних узлов и $(2n-3)$ – ветвей соединяющих пары из узлов. Степень каждого листа равна одному, а степень каждого внутреннего узла равна трем.

Удаление ветви из дерева T разрушает его на два поддерева, каждое из которых индуцировано множеством своих листьев.

Определение 2.2. Мультицвет(multicolor) называется T - согласованным, если он состоит из всех цветов (листьев) любого такого индуцированного поддерева.

Заметим, что если мультицвет Q является T - согласованным, то его дополнение $\overline{Q} = C \setminus Q$ так же T - согласованно. Поэтому, существует взаимно - однозначное соответствие между парами комплементарных T - соглсованных мультицветов и ветвями из T(см. рис. 2.2). Данное наблюдение используется при алгоритме реконструкции деревьев описанного в разделе **??**. Получаем что когда филогенетическое дерево дано, MGRA алгоритм решает урезанную версию MGRP, где 2 - break операции применяются только к мультицветам, соответствующие T - согласованным цветам в филогенетическом дереве. Сформулируем дерево согласованную

множесвенную геномно перестановачную проблему или The Tree-Consistent Multiple Genome Rearrangement problem (TCMGRP):

Теорема 2.3. *The Tree-Consistent Multiple Genome Rearrangement problem Для данных геномов* $P_1, P_2 \ \ldots \ P_n$*, как листьев филогенетического дерева* T*, найти кратчающую надежную серию* T *- согласованных 2 - break операций, трансформирующих множетсвенный breakpoint граф* $G(P_1, \ \ldots \ P_n)$ *в идентичный множественный breakpoint граф.*

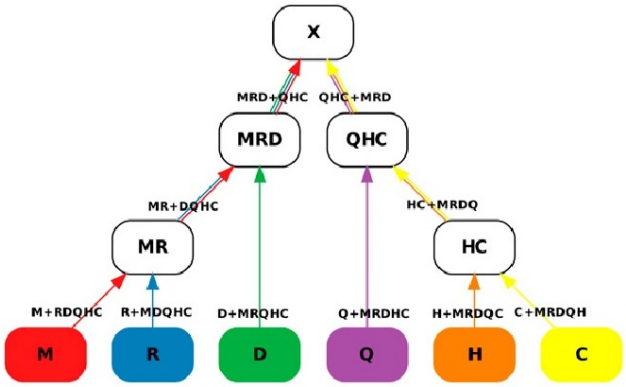

Рис. 2.2: Пример направленного дерева индуцированного цветами геномов

Отметим что MGRP и TCMGRP проблемы, в случае трех унихромосомных геномов, соответствуют проблеме медианы, которая является NP-полной задачей [21]. Поскольку вряд ли существуют точные полиномиальные алгоритмы для решения MGRP и TCMGRP проблем, MGRA использует эвристический подход "уничтожающий"ребра в $G(P_1, P_2 \ \ldots \ P_n)$, используя надежные T - согласованные перестановки.

Для дальнейшего анализа будет удобно находить фиксированные ветви χ в филогенетическом дереве T и предполагать, что это ветвь содержит корень X (рассматривается как еще один узел), точное местоположение которого будет определенно позже. Выбор корня X определяет направление “к” X всех ветвей на филогенетическом дереве T. Для наглядности, пометим каждый лист P_i, содержащий геном P_i, в направленном дереве T единичным мультицветом $\{P_i\}$ и затем рекурсивно помечаем каждый внутренний узел, объединением мультицветов начиная с узлов всех входящих в него ветвей (см. рис. 2.2 общий конец ветвей, выходящих из листьев

M и R, помечаем цветом M, R).

Определение 2.4. Мультицвет, соответсвующий пометке внутреннего узла в дереве T называется $\vec{T}$ - согласованным.

Можно дать альтернативное определение $\vec{T}$ - согласованным мультицветам, которые могут быть определены, как T - согласованные мультицвета, чьи индуцированные поддеревья не содержат ветви χ. Заметим, что именно один из мультицветов в каждой паре комплементарных T - согласованных мультицветов является $\vec{T}$ - согласованным и его пометки начальных узлов, соответствуют направлениям в дереве T (за исключением нескольких цветов соответствующих ветвям χ, что оба $\vec{T}$ - согласованные).

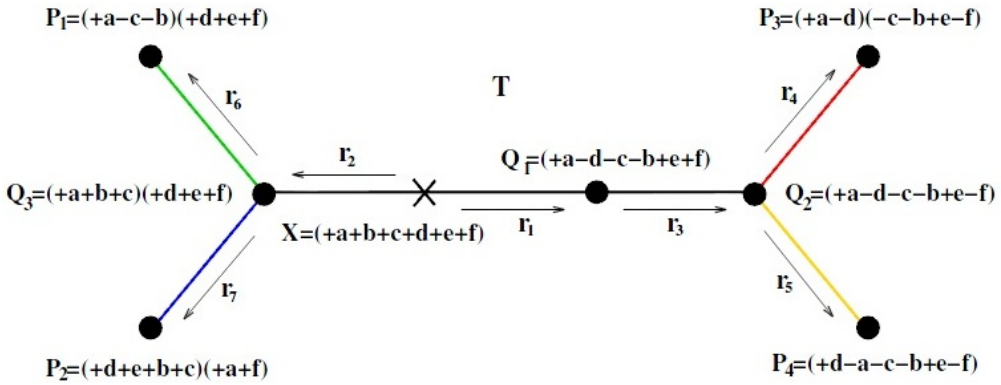

Рис. 2.3: Пример некорневого филогенетического дерева в листьях, которого содержатся геномы

MGRA алгоритм трансформирует геномы $P_1, \ldots P_n$ в X по направленным ветвям из дерева T, используя 2 - break операции на $\vec{T}$ - согласованные мультицвета для мультиребер ($\vec{T}$ - согласованные 2 - break операции). Данное утверждение в терминах breakpoint графов звучит так: MGRA алгоритм устраняет мультиребра в множетсвенном breakpoint графе $G(P_1, \ldots P_n)$ с помощью $\vec{T}$ - согласованных 2 - break операций и трансформирует его в идентичный множественный breakpoint граф $G(X, \ldots X)$. Использование $\vec{T}$ - согласованных 2 - break операций мотивированно важным свойством, что каждое $\vec{T}$ - согласованная трансформация может быть заменена на надежную $\vec{T}$ - согласованную трансформацию генома с изменением порядка в 2 - break операциях. Такие трансформации определим, как реверс трансформации из генома X в геномы $P_1, \ldots P_n$ с $\vec{T}$ - согласованны-

ми 2 - break операциями(результат приведен на рис. 2.3). MGRA алгоритм отслеживает перестановки применяющиеся к множественному breakpoint графу $G(P_1, \ldots P_n)$ во время трансформации в идентичный множественный breakpoint граф $G(X, \ldots X)$. Записанные перестановки (в обратном порядке) определяют обратную трансформацию, которая проходит через каждый внутренний узел в дереве T и таким образом может быть использована, для реконструкции предков генома во внутренних узлах из дерева T.

2.2. Требования предъявляемые к улучшениям MGRA алгоритма

В разделе 1.4 подробно описан множественный breakpoint граф для данных, где отсутствуют дупликации. Понятно, что определение множественного breakpoint графа для данных с дупликациями остается таким же, но становятся не валидными ряд предположений:

1. Каждый входной геном, содержит одно и тоже множество Δ блоков синтении.

Данное предположение не будет выполнено в общем случае, так как могли произойти дупликация каких-то блоков синтении и тогда общее количество блоков в геноме увеличивается. Так же нельзя забывать и о возможных делециях блоков синтении, поэтому некоторые геномы могут становится подмножеством множества Δ.

2. Каждому исходящему мультиребру соответствовал уникальный мультицвет.

Это предположение так же не будет выполнено из-за дупликации блоков. Приведем простой пример, иллюстрирующий данную ситуаци. Пусть какой-нибудь блок синтении x_k входит в геном P_i m раз, соответственно в множественном breakpoint графе из вершины x_k будут выходить m ребер цвета P_i. Допустив, что в остальных геномах блок x_k был один и имел одни и те же соседние блоки синтении, получаем, что $m - 1$ мультиребер

имели мультицвет $\{P_i\}$.

3. Мультицвет является подмножеством множества $C = \{P_1 \ \dots \ P_n\}$.

Данное определение мультицвета было введено для множественного breakpoint графа на данных без событий дупликаций в разделе 1.4.1. Пусть у нас есть пара соседних блоков синтении x_k, x_{k+1} в геноме P_i. Так же предположим, что эта пара блоков синтении на протяжение эволюции подвергалась только событиям дупликаций, которые произошли m раз. Получаем, что в новом множественном breakpoint графе эти две вершины будут соеденены мультиребром с мультицветом, состоящим, как минимум из m цветов $\{P_i\}$.

НУЖЕН РИСУНОК ГРАФА

Из - за ложности исходных предположений, структуры данных, которые использовал MGRA алгоритм становятся не верными. Необходимо обобощить их, а так же структуры данных соответствующие эти определениям, для корректной обработки данных с дуплицируемыми блоками.

После этого, на биологических данных, состоящих из 59 штаммов бактерии ecoli, был получен граф содержащий 2876 вершин (см. раздел 3.1.1). Из них всего 208 вершин соответствует событиям дупликаций, непосредственно затрагивая еще 526 вершин. Для оставшегося графа из 2142 вершин разумно предположить, что они соответствуют 2 - break операциям в геноме, так как это наиболее распространненые перестановочные операции в геноме, а так же благодаря тому, что частота транспозиций в эволюции пока не установлена (см. раздел 1.2.2).

2.3. Требования предъявляемые к разрабатываемому серверу

Почти все распространенные алгоритмы в биоинформатике имеют возможность работы в режиме онлайн [22–24]. Это делается для того что бы облегчить биологам взаимодействие с алгоритмами, которые требуют знаний в области программного обеспечения. Так же различные сервера выполняют ту или иную логику, помогая биологам обрабатывать данные получаемые от алгоритмом. Именно поэтому создание сервера для MGRA алгоритма является очень актуальной задачей для того что бы этим алгоритмом в дальнейшем было удобно пользоваться. Вот список требований предъявляемых к разрабатываему серверу:

2.3.1. Реализация алгоритма реконструкции филогенетических деревьев

Как было показано в разделе 2.1 MGRA алгоритм в случае заданного филогенетического дерева T решает урезанную версию MGRP, а именно TCMGRP. Поэтому реконструкция наиболее вероятных филогенетических деревьев на основе входных данных представляется очень важной задачей. Нужно учитывать, что из такого набора деревьев биологи смогут с помощью биологических соображений выбрать эволюционно правильное. После этого, скорее всего основной MGRA алгоритм даст лучшие результаты.

2.3.2. Реализация алгоритма отображения филогенетических деревьев

К сожалению, на данный момент в свободном доступе не существует ни одного приложения для отображения некорневых филогенетических деревьев, которые удолетворяли следующим условиям:

1. Все внутренние узлы дерева были бы кликабельны.
2. Все ветви дерева были бы кликабельны.

3. Алгоритм должен генерировать результат, который можно было бы добавить на HTML страницу.

Именно поэтому был разработан достаточно простой алгоритм по отображению филогенетических деревьев, удолетворяющих этим условиям.

2.3.3. Отображение полученных данных после отработки MGRA алгоритма

MGRA алгоритм отвечает на все вопросы эволюционных исследований: выдает предковые геномы, последовательность эволюционных событий, но все эти данные имеют текстовый вид трудно доступные для понимания человека из-за количества этих данных. Поэтому очень важно представить их пользователю в удобном для работы виде.

2.3.4. Достижения интерактивности работы с основным MGRA алгоритмом

MGRA алгоритм представляет собой программное обеспечение с разными сценариями работы. Например, для него явно можно указать какой предковый геном хочется получить, что бы он сконцетрировался на его реконструкции. Аналогично MGRA алгоритм представляет целый ряд этапов по обработке входных данных, которые отличаются надежностью реконструкций. Необходимо предоставить удобный выбор между этими сценариями работы для MGRA алгоритма.

2.4. Вывод по главе 2

В данной главе была приведена теоретическая постановки задачи сведения множественного breakpoint графа к индентичному множественному breakpoint графу с помощью цветных 2 - break операций. Формализованы две проблемы: MGRP и TCMGRP, которые пытается эврестически решить MGRA алгоритм, так как уже для трех геномов данные проблемы

являются NP - полными задачами. Приведено важное обобщение мультицветов на случай, когда заданно филогенетическое дерево.

Так же показано, какие события появляются во множественном breakpoint графе построенном на данных, содержащих дупликации. Дано обоснование, почему представляется логичным обобщить первый шаг алгоритма, заключающийся в поиске хороших путей, и второй шаг алгоритма по обработке справделивый ребер.

Приведены критерии, которым должен удолетворять проектируемый MGRA сервер. Дан список алгоритмов с обоснованием их необходимости. Для каждого алгоритма приведены требования к входных и выходных данных.

Глава 3. Реализация модификаций

3.1. MGRA алгоритм

MGRA алгоритм был реализован на языке C++, поэтому улучшения реализовывались с помощью этого языка. Для графического отображения графов в реализованном алгоритме, используется библиотека graphiz [25]. Текущая версия алгоритма доступна по ссылке [26]

3.1.1. Представление графа

Как говорилось в разделе 2.2, теперь степень каждой вершины может быть больше чем n, где n - количетсво геномов, возможно существование мультицветов, в которые цвета входят по нескольку раз, а так же исходящие из вершины мультиребра с мультицветами могут иметь не пустое пересечение друг с другом. Все эти нововведения делали не валидными старые структуры данных, которые сильно использовали свойства наблюдающиеся в геномах без событий дупликаций.

На данный момент множественный breakpoint граф представлен, как набор breakpoint графов для каждого генома, где в каждый "локальный"breakpoint граф поддерживает возможность хранения кратных ребер. Соответствующий код представлен в файле mpbgraph.h. Мультицвета это теперь не просто множество, а multimap отображающий цвет и его количество и весь код реализован в файле mcolor.h. Так же из-за необходимости обрабатывать исходящие ребра из вершины для них созданна отдельная структура, называемая mularcs, поддерживающая разнообразные операции над ними.

После того как все структуры данных были модернизированны, на основе 59 штаммов бактерий ecoli был получен множественный breakpoint граф, приведенный на рис. **??**, содержащий 2876 вершин.

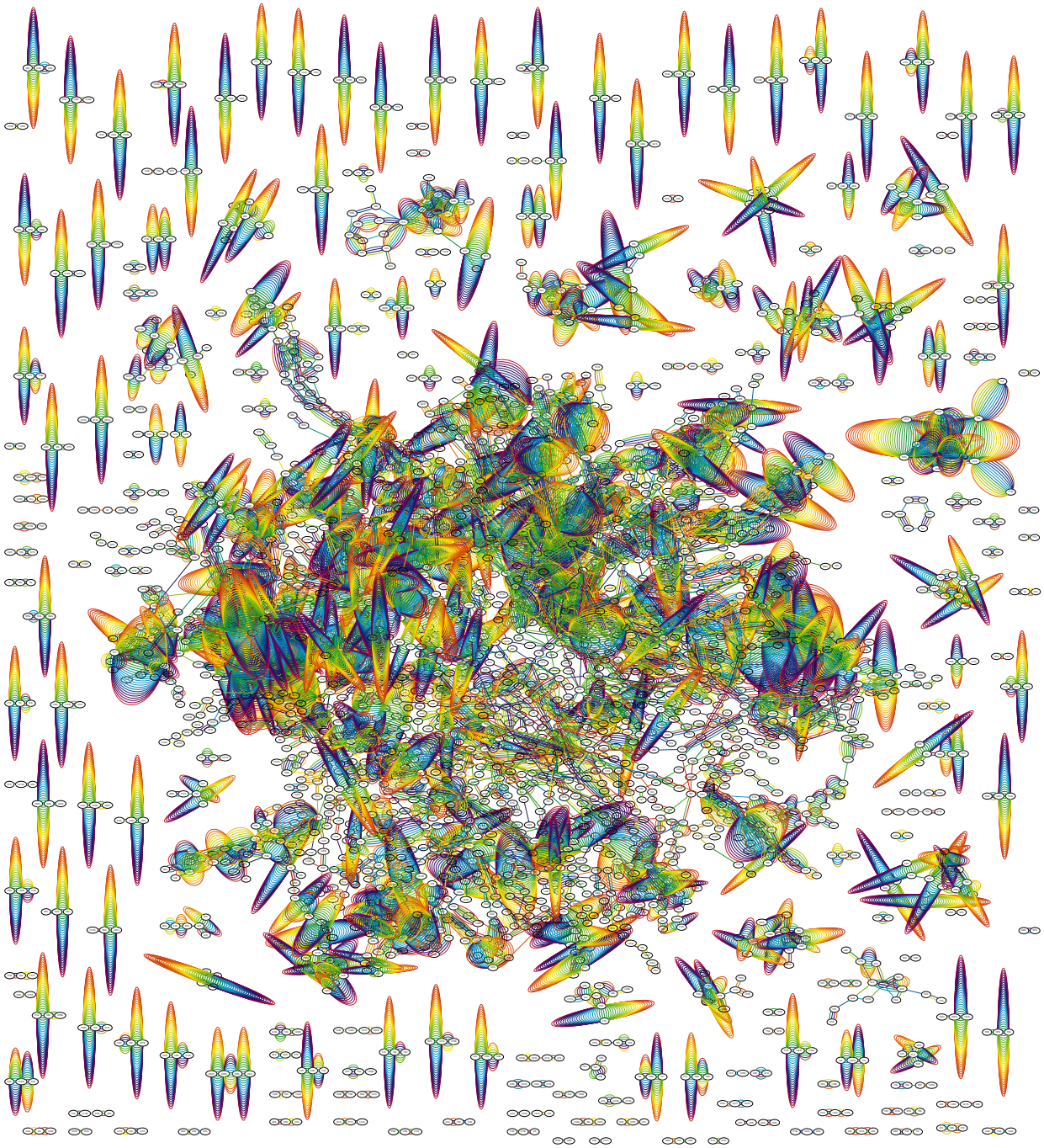

Рис. 3.1: Множественный breakpoint граф построенный на основе 59 штаммов бактерий ecoli

3.1.2. Адаптация первого шага по обработке хороших циклов и путей

Переменный цикл(цвета ребер чередуются) представляет собой хорошо изученный объект в случае парного breakpoint графа. Каждый такой цикл длинной $2 \times m$ формирует $(m-1)$ 2-break [20] операций в худшем сце-

нарии. Заметим, что непосредственным результатом выполнения 2-break операции, проводимой на ветве $Q + \overline{Q}$ в филогенетическом дереве T, является цикл из 4 мультиребер, мультицвета которых чередуются между Q и $\overline{Q}$. Все вершины в этом цикле имеют мультистепень равную двум. Это наблюдение мотивирует нас искать переменные пути или циклы во множественном breakpoint графе. Что бы искать правильные переменные пути или циклы в множественном breakpoint графе, который содержит события дупликации, введем следующие определения:

Определение 3.1. Мультиребро называется нормальным, если в его мультицвете каждый цвет встречается один раз.

Определение 3.2. Вершина называется обычной, если эта регулярная вершина инцидентна двум нормальным мультиребрам, мультицвета которых имеют пустое пересечение.

Определение 3.3. Мультиребро называется обычным, если оно соединяет две обычные вершины.

Определение 3.4. Путь/цикл называется обычным, если в него входят все обычные мультиребра, мультицвета которых чередуются между Q и $\overline{Q}$

Определение 3.5. Путь/цикл называется хорошим, если все мультицвета ребер T-согласованны.

Рисунок 3.2 демонстрирует трансформацию переменного цикла на шести вершинах в три полных мультиребра с помощью двух 2-break операций. Таким образом, разница между переменным циклом в парном breakpoint графе и хорошим циклом во множественном breakpoint графе мала: следовательно, хорошие циклы с чередованием мультицветов Q и $\overline{Q}$ могут быть надежно отнесены к ветке $Q + \overline{Q}$ в филогенетическом дереве T. MGRA алгоритм обрабатывает все надежные перестановки для всех (и для листьевых и для внутренних) ветвей филогенетического дерева T.

Аналогично, хорошие пути так же могут быть соотнесены с ветвями филогенетического дерева T, при превращении их сначала в хороший цикл.

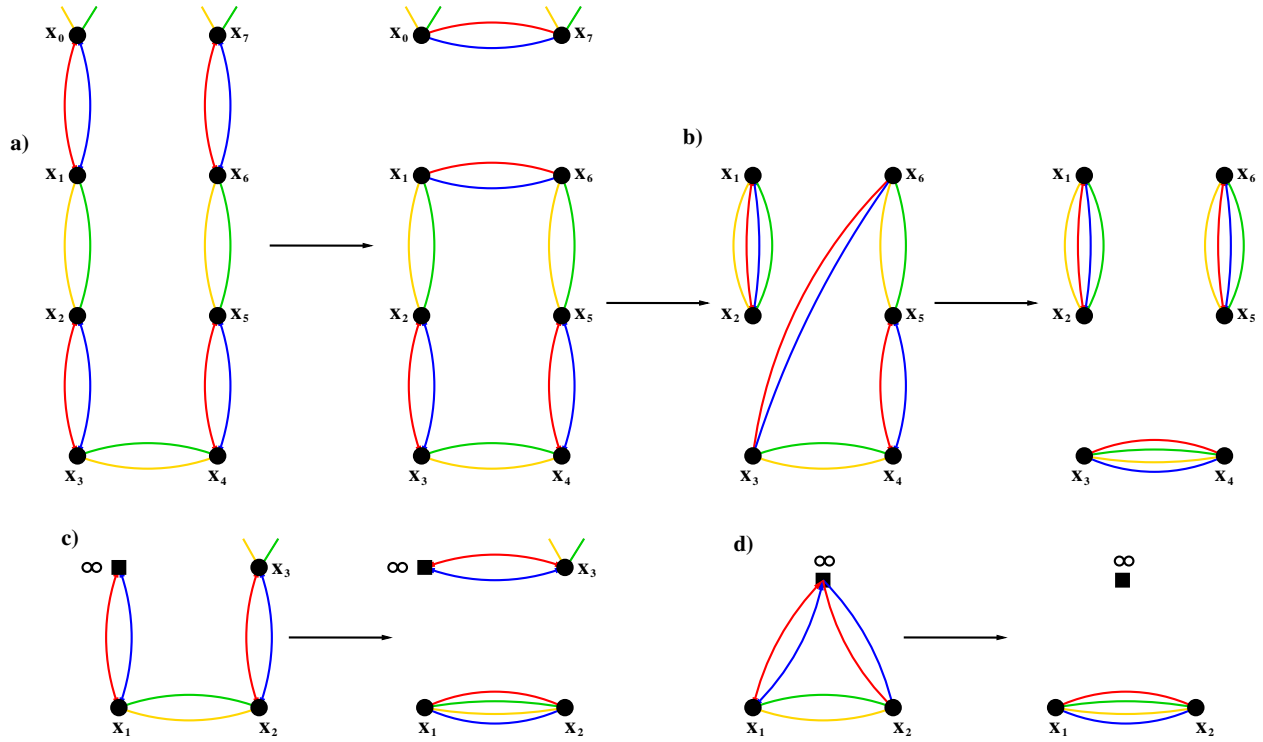

Рис. 3.2: Пример трансформации хорошего пути с помощью $\vec{T}$ - согласованных 2 - break операций

Считаем, что хороший путь $(x_1, \ \dots \ x_m)$ состоит из $(m-1)$ мультиребра с T - согласованными мультицветами, чередующимися между мультицветом Q и $\overline{Q}$. Продолжая этот путь вершинами x_0 и x_{m+1} инцидентных первой и последней вершине соответственно, получаем путь $p = (x_0, x_1 \ \dots \ x_{m+1})$. Если первое и последнее мультиребра в таком пути имеют $\vec{T}$ - согласованные мультицвета, выполним 2 - break операцию над мультиребрами (x_0, x_1) и (x_m, x_{m+1}) трансформирующих p в хороший цикл $c = (x_1, \ \dots \ x_m)$ и мультиребро (x_0, x_{m+1}). Если первая и/или последнее мультиребра является не $\vec{T}$ - согласованными мультицветами, удаляем его/их из нашего пути для получения пути с $\vec{T}$ - согласованными мультицветами по бокам и обрабатываем, как указано выше.

Заметим, что обработка хороших циклов/путей во множественном breakpoint графе может создавать новые хорошие циклы/пути, поэтому обработка хороших циклов/путей происходит интерактивным образом, пока не останется никаких хороших циклов/путей.

3.1.3. Адаптация второго шага по обработке честных ребер

В оставшемся графе после первого шага на данных без дуплицируемых блоков, наблюдалось существование, так называемых справедливых ребер окрашенные в T - согласованные мультицвета. Так как обработка таких ребер приводила к ребрам с хорошим мультицветами с помощью надежных 2 - break операций, то представляется разумным попытаться обобщить поиск таких ребер на данные с дуплицируемыми блоками.

Для начала нам необходимо определить мобильность мультиребра. Данное свойство означает, что мультиребро с данным мультицветом, может участвовать в возможном сценарии, когда происходят 2 - break операции на $\vec{T}$ - согласованном мультицвете Q. То есть, если ребро (x, y) обладает свойством немобильности, то для каждого инцидиентного мультиребра с T - согласованным мультицветом QQ для вершин x, y, выполняется 2 - break операция над этими мультиребрами.

Здесь рисунок.

Рассмотрим конкретный пример определения свойства мобильности справедливого ребра (x, y) T - согласованного цвета Q для полграфа показаного на рис. **??**. Посмотрим на соседей вершины x, исключая y: как видно по рисунку, это вершины p, q, r. Для каждой такой вершины, смотрим можем ли мы составить инцидентное мультиребро мультицвета Q. Видим, что у вершины p могут быть инцидентные мультиребра мультицветов Q_1 и Q_2, такие что $Q1 \cup Q2 = Q$. В этом случае, ребро (x, y) считается мобильным, потому что возможен сценарий, когда происходит $2 - break$ операция на мультицвете Q_1, создавая у вершины p инцидентное мультиребро мультицвета Q и превращая, например, ребро (p, x) в справделивое мультиребро, обладающее свойством мобильности. Тогда появляется неоднозначность в обработке мультиребер, так как возможно два варианта: обработать мультиребро (p, x) или обработать мультиребро (x, y).

3.2. Структура MGRA сервера

Сервер был реализован на языке Java(исходный код можно найти по адресу [27]) с использованием библиотеки Jetty [28] для обслуживания запросов пользователя. Стандартный сеанс работы пользователя с сервером следующий:

1. Пользователь выбирает формат входных данных, вводит их и запускает их на обработку.
2. Сервер проверяет валидность входных данных, создает все необходимые файлы для запуска основного mgra алгоритма и запускает его.
3. Если пользователь выбрал опцию по реконструкции филогенентических деревьев или основной алгоритм mgra не смог завершить трансформацию, то запускается алгоритм реконструкции филогенетических деревьев.
4. Получившиеся деревья, добавляются к входным данным и происходит перезапуск основного mgra алгоритма.
5. Вся полученная информация обрабатывается алгоритмами, которые будут описаны ниже, и итоговый отчет представляется ввиде динамической страницы HTML формата.

3.2.1. Алгоритм реконструкции филогенетических деревьев

Таблица 3.1 предоставляет статистику в множественном breakpoint графе, которая показывает, как перестановочный анализ способствует конструированию филогенетических деревьев. Действительно, все три внутренних ветви (правильное разделение деревьев) поддерживает большое количество хороших циклов/путей и хороших мультиребер. Каждая из 32 некорректных ветвей (только восемь из них представлены в таблице 3.1) имеют не более одного обычного пути/цикла и не более шести обычных мультиребер, что на порядок меньше, чем число правильных ветвей. Эти

наблюдения иллюстрируют нам, что реконструкцию филогенетических деревьев правильной топологии можно производить по такой статистике.

Мультицвет	количество мультиребер
R + MDQHC	540 + 633 = 1173
MR + DQHC	246 + 241 = 487
D + MRQHC	184 + 289 = 473
M + RDQHC	150 + 73 = 223
MRD + QHC	82 + 126 = 208
Q + MRDHC	65 + 97 = 162
HC + MRDQ	54 + 86 = 140
C + MRDQH	16 + 29 = 45
H + MRDQC	4 + 11 = 15
QC + MRDH	4 + 5 = 9
MRQ + DHC	4 + 4 = 8
MD + RQHC	4 + 4 = 8
QH + MRDC	1 + 6 = 7
RQ + MDHC	5 + 2 = 7
DC + MRQH	3 + 3 = 6
DQ + MRHC	1 + 4 = 5
$\emptyset$ + MRDQHC	0 + 2 = 2
MRC + DQH	0 + 1 = 1

Таблица 3.1: Статистика мультиребер, имеющих определенные мультицвета

Алгоритм кратко приведен в листинге 1, являющийся алгоритмом полного перебора. Входными данными для него являются ветви дерева, представленные в таблице 3.1, а так же ветви сгенерированные по уже заданными поддеревьям. Результатом работы алгоритма являются все возможные деревья отсортированные по качеству их биологической корректности, которые могут быть получены на таких входных данных.

Листинг 1 Алгоритм реконструкции филогенетических деревьев

Вход: *input_set* — входное множество ветвей, которые надо обработать; *current_branchs* — текущее множество ветвей на основе которых можно реконструировать дерево; *trees* — текущий набор реконструированных деревьев;

```
1: Function reconstruction (input_set, current_branchs, trees)
2: for x in input_set do
3:    if x ∉ current_branchs и compatibilityAll(current_branchs, x) then
4:       current_branchs.add(x)
5:       reconstruction(input_set, current_branchs, trees)
6:       current_branchs.remove(x)
7:    end if
8: end for
9: trees.add(reconstruction_tree(current_branches))
10: EndFunction
```

Временная сложность работы этого алгоритма, состовляет $O(2^n)$,

где n - это количество ветвей во входных данных.

3.2.2. Алгоритм отображения филогенентических деревьев

В разделе 2.3 отмечалось, что на данный момент не существует каких - либо удобных средств по отображению филогенетических деревьев, поэтому был написан достаточно простой алгоритм, отображающий такие деревья. Происходит это в несколько этапов:

1. Генерируется XML файл, содержащий филогенетическое дерево.
2. С помощью XSLT - преобразования, соответсвующий XML документ переводится в html документ.
3. На основе размера области, в которой должно быть отображено дерево, и самого дерева, область разбивается на квадраты в которых происходит отрисовка узлов дерева и ребер с помощью HTML5 Canvas, используя библиотеку KineticJs [29].

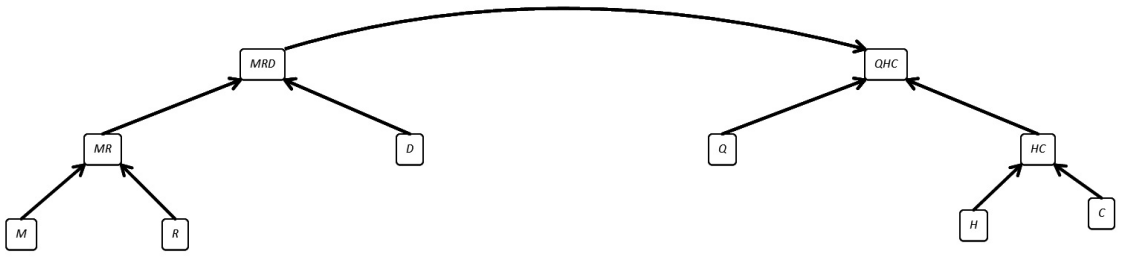

Рис. 3.3: Пример филогенетического дерева, построенного алгоритмом на сервере

Пример дерева приведен на рис. 3.3 . Узлы дерева можно перемещать в пределах области, где отрисовано дерево. Если узел дерева окрашен в темно-голубой цвет, то по двойному клику на него пользователь вызовет AJAX - запрос серверу для генерации информации о предковом геноме в этой вершине. Если же ветвь дерева окрашена в черный цвет, то по двойному клику на нее пользователь вызовет AJAX - запрос серверу для генерации серии 2 - break операций, трасформирующих один геном в другой.

3.2.3. Представление предкового генома

Так как основной MGRA алгоритм выдает предковый геном в текстовом формате, необходимо как-то визуализировать его. MGRA сервер на основе текстовых данных, пытается сгенерировать изображение, но если возникает какая-либо проблема (картинка большая, занимает много памяти), то сервер пытается сгененировать XML документ, а потом с помощью XSLT - преобразования создать HTML - представление. Стоит отметить, что MGRA алгоритм умеет обрабатывать геном представленный в формате inferCARS, но предковый геном выдает в формате GRIMM, теряя информацию о длине блоков синтении. Поэтому MGRA сервер в случае информации о длине блоков синтении генерирует изображения предковых геномов, учитывая эту длину.

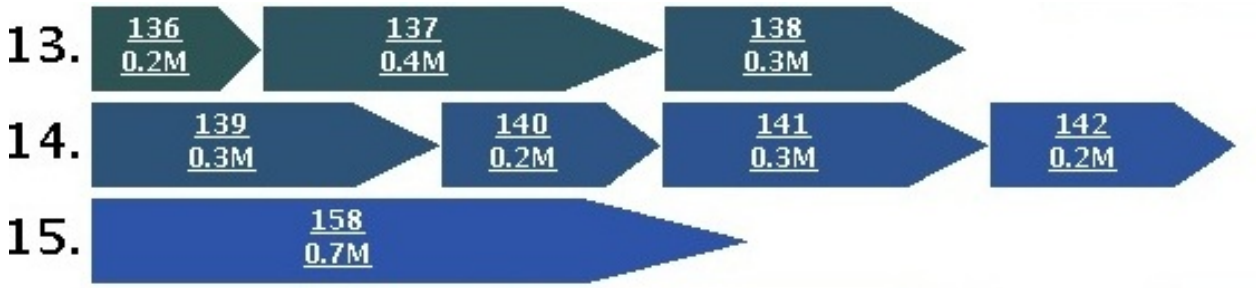

Рис. 3.4: Часть изображения генома, сгенерированного на сервере на основе inferCARs данных

Пример изображения, сгенерированный из входных данных inferCARS, приведен на рис. 3.4. Направление блоков синтении, определяется направлением пятиугольников. Так же по рисунку видно, что блоки содержат информацию, касаемо длины блока измеряемое в мего- килобазовых пар, а так же номер этого блока.

3.2.4. Восстановление геномных перестроек и их отображение

Как говорилось ранее, у пользователься есть возможность запросить генерацию серии геномных перестроек, трансформирующих один геном в другой. К сожалению, у нас есть только информация о 2 - break операциях, которые показывают блоки по которым происходил разрез, поэтому

возникает проблема восстановления корректных геномных перестроек. Эта проблема решается алгоритмом воспроизводящий эволюцию в обратном порядке (см. листинг 2). Входными параметрами этого алгоритма является начальный геном, с которого начинают происходить 2 - break операции, а так же серия 2 - break операций, которые трансформируют один геном в другой.

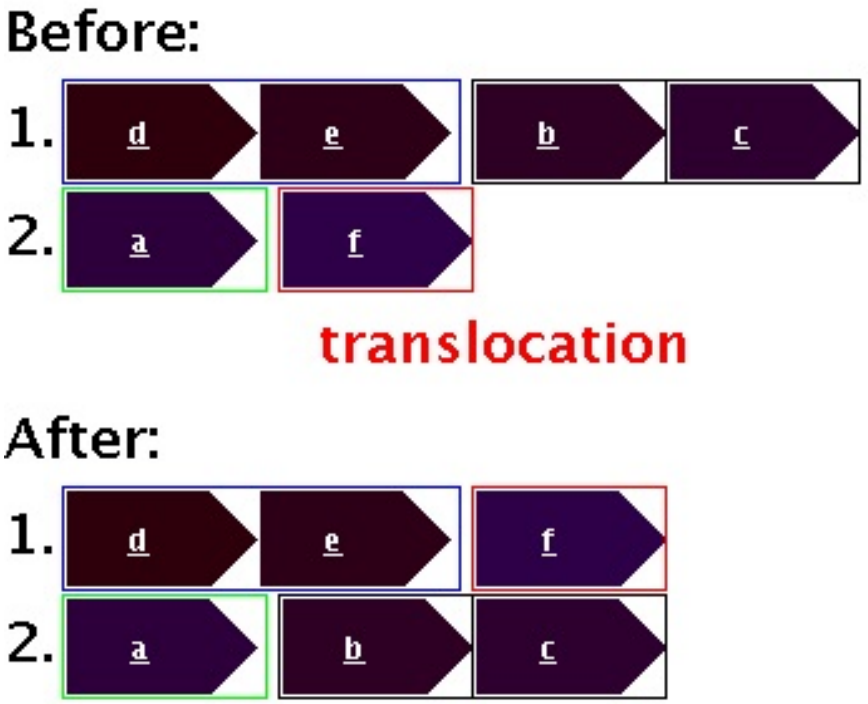

Рис. 3.5: Вид геномной перестройки, сгенерированный сервером

Листинг 2 Алгоритм воспроизводящий эволюцию на основе 2 - break операций

Вход: *start* — начальный геном, от которого начинают происходить трансформации; Δ — набор 2 - break операций, трасформирующих геном *start* в другой геном;

1: $current \leftarrow start$
2: **for** $2break$ in Δ **do**
3: Разрезаем хромосомы *current* в соответствие текущей $2break$
4: Объединяем части хромосом в местах, где у них совпадают последние блоки синтении
5: Определяем какой геномной перестройке, соответствует $2break$
6: Обновляем геном *current*
7: **end for**

После восстановления всех геномных перестроек происходит генерация информации по ней в HTML. Пример представления геномной перестройки изображен на рисунке 3.5. Есть две секции, "Before"и "After отвечающие за геном до и после трансформации. Последовательности блоков

синтении, отображаются с помощью метода изложенного в разделе . Куски генома, участвующие в геномной перестройке, обводятся прямоугольником определенного цвета, и это позволяет нам показать, где изначальные части генома оказались после трансформации.

3.3. Вывод по главе 3

Приведено краткое описание новых стурктур данных, используемых в MGRA алгоритме. Дано описание модифицированных классов, а так же приведен получившийся множественный breakpoint граф на реальных биологических данных. Приведено подробное описание модификации первого шага по поиску хороших путей/циклов и дано теоретическое обоснование поиска, таких хороших циклов/путей. Описано обобщение поиска и обработки справделивых ребер на втором шаге алгоритма. Дано определение свойства мобильности мультиребра, которое открывает большие возможности по обработке мультиребер с T - согласованными мультицветами.

Дано обоснование верности реконструкции филогенетических деревьев на основе статистики, посчитанной по множетвенному breakpoint графу. Приведен алгоритм, который на основе данной статистики ветвей дерева реконструирует все возможные филогенетические деревья, выдывая их отсортированным списоком по суммарной весовой характеристики.

Описана созданная структура MGRA сервера. Кратко приведен алгоритм отображения геномов в виде изображения и HTML текста. Описан алгоритм, отрисовывающий филогенетические деревья, которые удолетворяют свойствам, описанным в разделе 2.3. Так же приведен алгоритм превращения серии 2 - break операций в серию геномных перестроек. описан алгоритм отображения этих геномных перестроекв в виде изображения и HTML текста.

Глава 4. Результаты

4.1. Результаты обобщения MGRA алгоритма

Как было показано ранее в разделе 3.1.1, после модернизации структур данных на основе 59 штаммов бактерий E.coli был получен корректный множественный breakpoint граф, содержащий всю информацию из входных геномах. После дальнейшего обобщения шагов алгоритма были получены результаты, приведенные в таблице 4.1.

	Было	1 шаг	2 шаг
ecoli59	3	40	43

Таблица 4.1: Результат отработки MGRA алгоритма. Цифры в таблице, обозначают количество полных мультиребер мультицвета C

Из таблицы видно, что существующие шаги алгоритма на данных с событиями дупликаций и делеций, ведут себя гораздо хуже, чем на данных без таких событий. Это является удивительным фактом, так как ранее упоминалось, что в множественном breakpoint графе вершин, соответствующих дупликациям всего 208, затрагивающих еще 605 вершин. Из полученных данных следует, что на оставщихся 2000 вершин, 2 - break операции были не единственными событиями эволюции. Дальнейшие поверхностные исследование показали существование событий делеции блоков. Возможно стоит учесть, что над дуплицирвоанными данными могло произойти большое количество 2 - break операций, которые превратили некоторые вершины и мультицвета, не соответсвующие дупликациям. Так же представляется реальным, существование некоторого числа операций транспозиций в ходе эволюции. Основываясь на полученных результатах, предполагается разумным, дальнейшее детальное изучение эврестического решения MGRP и TCMGRP проблем, описанных в главе 2, на данных, содержащих события дупликации и делеции.

4.2. MGRA сервер

Данный сервер доступен по адресу [], а его исходный код доступен по адресу []. На данный момент, MGRA сервером уже воспользовалось 3 коллектива биологов в течение года, что является прекрасным результатом с учетом, количества проводимых исследований в этой области на данный момент. Стоит отметить, что при накопление информации о геномах эта цифра будет только расти. Биологи, которые воспользовались сервером, остались довольны добавленным функционалом, реализацией, а так же высказали ряд полезных замечаний, которые будут устраннены в будущем и предложений, которые будут добавлены в будущем.

ДОБАВИТЬ СРАВНЕНИЕ ЗАПУСКА MGRA алгоритма с ДЕРЕВЬЯМИ и без НЕГО.

4.3. Вывод к главе 4

Заключение

Список литературы

1. Genome 10K Project. https://genome10k.soe.ucsc.edu.
2. *Moret B. and Wyman S. and Bader D.A. and Warnow T. and Yan M.* A new implementation and detailed study of breakpoint analysis // Proc. 6th Pacific Symp. on Biocomputing. 2001. Pp. 583–594.
3. *Bourque G. and Pevzner P. A.* Genome-scale evolution: reconstructing gene orders in the ancestral species // Genome Res. 2002. Pp. 26–36.
4. *Ma J. and Zhang L. and Suh B. B. and Raney B. J. and Burhans R. C. and Kent J. W. and Blanchette M. and Haussler D. and and Miller W.* Reconstructing contiguous regions of an ancestral genome // Genome Res. 2006. Pp. 1557–1565.
5. *Zhao H. and Bourque G.* Recovering genome rearrangements in the mammalian phylogeny // Genome Res. 2009.
6. *Max A. Alekseyev and Pavel A. Pevzner.* Breakpoint Graphs and Ancestral Genome Reconstructions // Genome Research. 2009. Pp. 943–957.
7. *Ohno S.* Evolution by gene duplication // Spriger Brlin. 1970.
8. *Nadeau J. H. and Taylor B. A.* Lengths of Chromosomal Segments Conserved since Divergence of Man and Mouse // Proceedings of the National Academy of Sciences. 1984. Pp. 814–818.
9. *Pevzner P. A. and Tesler G.* Human and mouse genomic sequences reveal extensive breakpoint reuse in mammalian evolutions // Proceedings of the National Academy of Sciences. 2003. Pp. 7672–7677.
10. *Kikuta H and Laplante M and Navratilova P and Komisarczuk A.Z. and Engstorm P. G. and Fredman D. and Akalin A. and Caccamo M. and Sealy I. and Howe K. and Ghislain J. and Pezeron G. and Mourrain P. and Ellingsen S. and Oates A. C. and Thisse C. and Thisse B. and Foucher I. and Adolf B. and Geling A. and Lenhard B. and Becker T.S.* Genomic regulatory blocks encompass multiple neighboring genes and maintain conversed synteny in vertebrates // Genome Research. 2007. Pp. 545–555.
11. *Son K. Pham and Pavel A. Pevzner.* DRIMM-Synteny: Decomposing Genomes into Evolutionary Conserved Segments // Bioinformatics. 2010. Pp. 1367–4803.
12. *Ilya Minkin and Nikolay Vyahhi and Son Pham.* Sibelia: A Synteny Blocks Generation and Genome Comparison Tool // Poster in WABI2012. 2012.
13. *Kellis M. and Birren B. W. and Lander E.S.* Proof and evolutionary analysis of ancient genome duplication in yest Saccharomyces cerevisiae // Nature. 2004. Pp. 617–624.
14. *Dehal P. and Boore J. L.* Two rounds of genome duplication in the ancestral vertebrate genome // PLoS Biology 3. 2005.
15. *Meyer A. and De Peer Y, V.* From 2R to 3R: evidence for a fish-specific genome duplication // BioEssays 27. 2005. Pp. 937–945.
16. *Hannenhalli S. and Pevzner P.* Transforming men into mouse (polynomial algorithm for genomic distance problem) // Proceesings of the 36th Annual Symposium on Foundations of Computer Science. 1995. Pp. 581–592.
17. *Hannenhalli S. and Pevzner P.* Transforming cabbage into turnip (polynomial lgorithm for sorting signed permutations by reversal) // Journal of the ACM. 1999. Pp. 1 –27.
18. *Max A. Alekseyev.* Multi-Break Rearrangements: from Circular to Linear Genomes // Lecture Notes in Computer Science. 2007. Pp. 1–15.
19. *Max A. Alekseyev and Pavel A. Pevzner.* Whole Genome Duplications, Multi-Break Rearrangements, and Genome Halving Theorem // Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA). 2007.
20. *Max A. Alekseyev.* Multi-Break Rearrangements and Breakpoint Re-uses: from Circular to Linear Genomes // Journal of Computational Biology. 2008. Pp. 1117–1131.

21. *Caprara A.* Formulations and hardness of multiple sorting by reversals // Conference on Computational Molecular Biology. 1999. Pp. 84 –93.
22. *Tesler G.* GRIMM: Genome rearrangements web server // Bioinformatics. 2002. 492–493.
23. Blast web server. http://blast.ncbi.nlm.nih.gov/.
24. *Boc A and Diallo Alpha B. and Makarenkov V.* T-REX: a web server for inferring, validating and visualizing phylogenetic trees and networks // Nucleic Acids Research. 2012. Pp. 573–579.
25. Graphiz project. http://www.graphviz.org/.
26. Sources MGRA algorithm. https://github.com/ablab/mgra.
27. Sources MGRA server. https://github.com/AvdeevPavel/MGRA.
28. Jetty. http://www.eclipse.org/jetty/.
29. Library kinetic.js. http://kineticjs.com/.