ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ АКАДЕМИЧЕСКИЙ УНИВЕРСИТЕТ РОССИЙСКОЙ АКАДЕМИИ НАУК

Диссертация допущена к защите

Зав. кафедрой

_____ А.В. Омельченко

«____» _____ 2015 г.

ДИССЕРТАЦИЯ
НА СОИСКАНИЕ УЧЕНОЙ СТЕПЕНИ
МАГИСТРА

Тема: **Реконструкция филогенетических деревьев на основе данных о перестройках и событиях вставок и удалений генов**

Направление: 03.04.01 – Прикладные математика и физика

Выполнил студент Н.М. Карташов

(подпись)

Руководитель:

PhD, доцент М. А. Алексеев

(подпись)

Рецензент:

м.н.с. А. В. Банкевич

(подпись)

Санкт-Петербург

2015

Благодарности

Спаибо Павлу Авдееву, Антону Банкевичу (СПбАУ), Максиму Алексееву (GWU) и Yu Lin (EPFL).

Реферат

С. 42, рис. 10, табл. 4.

В данной работе представлены методы извлечения информации из брейкпоинт графа и восстановления филогенетических деревьев на их основе. Разработанный алгоритм основан на поиске паттернов в брейкпоинт графе, их декодирования и дальнейшего восстановления на основе полученной информации. В отличие от большинства аналогичных методов, он способен работать с данными со вставками и удалениями синтенных блоков, несобранными данными и использовать информацию об известных поддеревьях. Также алгоритм позволяет заменять способы извлечения информации из брейкпоинт графа, что оставляет возможности для его дальнейшего расширения. Алгоритм реализован как часть программного пакета MGRA2, доступного под лицензией GNU GPL v2.0.

Ключевые слова: брейкпоинт-граф, филогенетические паттерны, восстановление филогенетических деревьев.

Содержание

Введение

Все биологические дисциплины согласны в том, что биологические виды имеют общую историю. Для выбранной группы организмов данная история может быть представлена в виде филогенетического дерева. Анализ филогенетических деревьев и предковых геномов является одним из главных инструментов эволюционной биологии. Восстановление филогенетических деревьев на сегодняшний день может проводиться на основе данных секвенирования. Новые технологии секвенирования позволяют получать большое количество данных секвенирования относительно дешево и быстро, потому особенно важно уметь восстанавливать филогенетические деревья на их основе быстро и точно. Знание филогенетического дерева полезно в ряде приложений, его получение делает возможным восстановление предковых геномов с высокой точностью, облегчает точную сборку секвенированных геномов, дает понимание о том, как шел процесс эволюции.

На данный момент существует множество программных средств решающих задачу восстановления филогенетических деревьев на основе данных секвенирования, отличающихся как в подходах к решению задачи, так и в информации используемой для восстановления. В данной работе рассматривается восстановление деревьев из брейкпоинт графа - специального вида графа, получаемого из геномов, лежащих в листьях результирующего дерева. Восстановление на основе брейкпоинт графа происходит через выделение разделений геномов - единиц информации о взаимном расположении геномов в результирующем дереве. В работе представлены два алгоритма, решающие задачу восстановления деревьев из разделений, и рассмотрен метод поиска способов извлечения филогенетической информации из брейкпоинт графа. Алгоритмы реализованы как часть программного пакета MGRA2 на языке C++ и доступен под лицензией GNU GPL v2.0.

Диссертация состоит из трёх глав. В первой главе описывается постановка

задачи и дается обзор существующих методов. Вторая глава посвящена описанию методов получения филогенетической информации и алгоритмов восстановления деревьев на ее основе. Третья глава содержит сравнение реализованных методов с известными на тестовых данных.

Глава 1

Обзор предметной области, существующих решений и постановка проблемы

1.1. Обзор предметной области

Общеизвестно, что молекула ДНК в процессе эволюции может меняться. Изменения происходящие в молекуле ДНК можно разделить на две группы:

1. Точечные (замены, вставки и удаления на уровне отдельных нуклеотидов)
    - Замена - замена одного нуклеотида в цепи на другой
    - Вставка - добавление нового нуклеотида в цепь
    - Удаление - удаление нуклеотида из цепи
2. Структурные (перестройки на уровне отдельных сегментов молекулы ДНК)
    - Инверсия - разворот части цепи и встраивание его на то же самое место
    - Вставка - вставка нового фрагмента цепи между существующими
    - Удаление - удаление фрагмента цепи
    - Слияние - слияние двух разных хромосом в одну
    - Разделение - разбиение хромосомы на две новых
    - другие

В ходе исследования эволюции выдвигалось две гипотезы о местах в которых происходят структурные перестройки: “случайная” и “хрупкая”. Первая утверждает, что границы блоков, над которыми происходят структурные перестройки, расположены по геному случайным образом [3], когда последняя

утверждает, что перестройки не могут происходить в случайных местах генома [4] [5] [6]. В ходе исследований последних 20 лет ученые пришли к консенсусу, что перестройки происходят не в случайных местах генома, но в процессе эволюции в геноме появляются и исчезают хрупкие регионы [7]. На основе этого факта можно ввести понятие *блоков синтении* [8].

Определение 1. *Блоки синтении*
В молекуле ДНК существуют консервативные регионы, называемые блоками синтении (синтенными блоками), геномные перестройки в которых маловероятны.

В данной работе геном будет рассматриваться в виде набора хромосом, где каждая хромосома состоит из набора блоков синтении и молекула ДНК подвержена только структурным изменениям.

Хромосомы в геноме могут быть циклическими и линейными. Другими словами, каждая хромосома представляется в виде перестановки над синтенными блоками. Для того чтобы определить понятие *перестановки над блоками синтении* введем понятие *знаковой перестановки.*

Определение 2. *Знаковая перестановка*
Знаковая перестановка над множеством элементов $\mathbb{B}$ *- перестановка над элементами из* $\mathbb{B}$*, в которой каждый элемент имеет знак.*

Имея определения знаковой перестановки необходимо определить множество ее элементов - *множество блоков синтении.*

Определение 3. *Множество блоков синтении*
Множество блоков синтении над множеством хромосом $\mathbb{C}$ *- упорядоченное множество блоков синтении из всех хромосом из* $\mathbb{C}$*.*

Теперь необходимо расширить определение знаковой перестановки так, чтобы возможно было отразить вставки и удаления блоков (присутствие в геноме блока, которого нет в других геномах и его отсутствие соответственно).

Определение 4. *Перестановка над блоками синтении*

Перестановка над блоками синтении - непустая знаковая перестановка над множеством блоков синтении, в которой часть блоков синтении может быть не задействована.

Обычно при рассмотрении молекулы ДНК вводится направление движения по ней, что дает возможность имея набор блоков синтении выделенный на хромосоме задать каждому из них знак: положительный, если направление движение совпадает с направлением блока, иначе - отрицательный. Теперь становится возможным привести набор хромосом к перестановкам над блоками синтении: для этого необходимо выделить все блоки синтении, а затем расставить для блоков каждой хромосомы знаки. Для получения такого представления генома из последовательности нуклеотидов существует целый ряд программных средств, таких как Sibelia [9], DRIMM-Synteny [10], i-ADHoRe3.0 [11] и другие.

Теперь на основе введенных определений можно сформулировать проблему [12], которая будет решаться в данной работе.

Проблема 5. *Проблема восстановления деревьев*

Для данных в виде набора геномов, в котором каждая хромосома представлена в виде перестановки над блоками синтении, восстановить филогенетическое дерево.

Филогенетические деревья содержат в себе информацию о взаимном родстве живых организмов. Эта информация может быть представлена в виде неких расстояний между геномами организмов: предполагается, что организмы, геномы которых расположены на меньших эволюционных расстояниях, находятся ближе в филогенетическом дереве [3]. Введем понятие брейкпоинт графа и рассмотрим, как он может помочь в оценке этих расстояний.

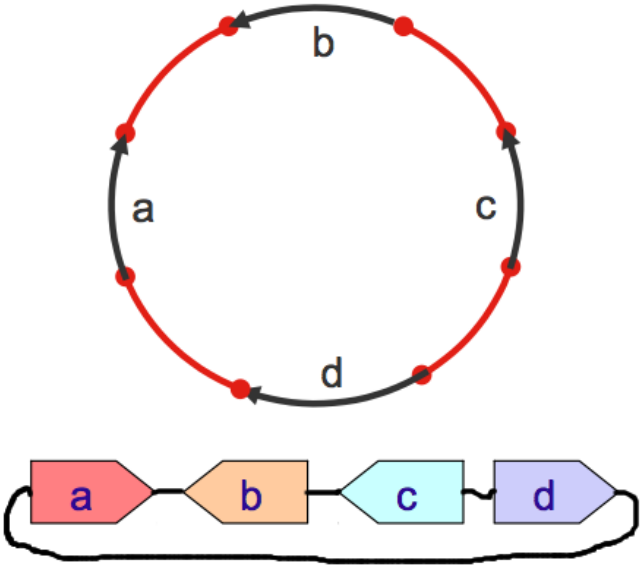

Рис. 1.1. Граф блоков циклической хромосомы

Рассмотрим одиночную циклическую хромосому, разбитую на уникальные блоки синтении. Хромосома в виде перестановки над блоками синтении может быть представлена как граф с двумя типами ребер: направленными, обозначающими блоки синтении, и ненаправленными, обозначающими связи между блоками. В таком представлении для блока a назовем вершину в которую входит обозначающее его ребро a_h, а вершину из которой это ребро исходит - a_t. После такого переименования можно заметить, что даже при удалении направленных ребер информация о них не потеряется. Таким образом, можно перейти к представлению графа в виде списка смежности вершин, где вершины считаются смежными, если они соединены ненаправленным ребром. Например, на рисунке 1.1 граф может быть представлен в виде списка ребер $[(a_h, b_h), (b_t, c_h), (c_t, d_t), (d_h, a_t)]$, из которого исходный граф может быть восстановлен. Описанное выше представление не дает возможности работать с линейными хромосомами, первый и последний синтенные блоки которых связаны с предыдущим и следующим за ними синтенным блоком соответственно, потому данное представление можно обобщить, добавив специальную фиктивную вершину ∞, с которой будут связаны первый и последний синтенные блоки в хромосоме [13]. Далее определим, что для объединения двух хромосом список смежности получается

объединением списков смежностей для каждой из них. Используя описание выше, становится возможным превратить биологический объект (набор хромосом) в математический объект (граф смежностей), далее будет рассмотрено как нас основе геномов в таком виде возможно оценивать расстояния между ними.

Теперь перейдем к рассмотрению нескольких геномов. Для начала примем, что геномы определены на одних и тех же блоках синтении и в каждом из геномов они все присутствуют в единичном экземпляре. Для описания графового представления сразу нескольких геномов нам понадобится понятие *индексированного объединения*.

Определение 6. *Индексированное объединение*
Пусть имеется два множества A *и* B *и множество индексов* $\mathbb{I}$*, тогда их индексированное объединение* $A \cup_{\mathbb{I}} B$ *имеет вид* $\{(i, a)|\forall a \in A, i \in \mathbb{I}\} \cup \{(j, b)|\forall b \in B, j \in \mathbb{I}\}, i \neq j$.

Для объединения геномов вышеописанная структура графа определяется как индексированное объединение списков смежности, где индексами является множество цветов. Так как в данном представлении каждый геном имеет свой уникальный цвет, в дальнейшем слова “геном” и “цвет” будут использоваться взаимозаменяемо. Смысл предыдущей операции состоит в том, чтобы объединить графы смежности для обоих геномов не теряя информации, к какому геному какое ребро принадлежит. Структура графа выше имеет название “брейкпоинт граф” [14].

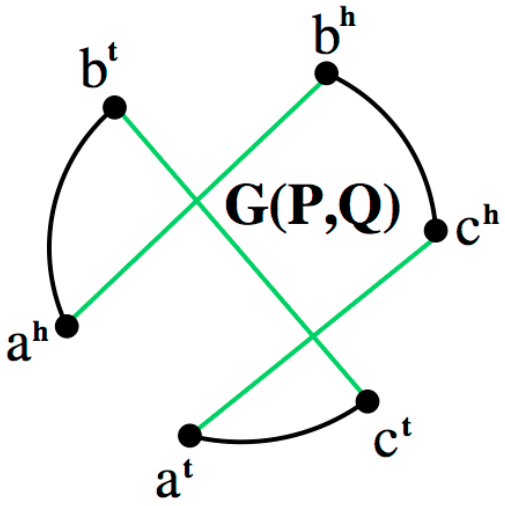

Рис. 1.2. Брейкпоинт граф для пары геномов со списками ребер $[(a_h, b_t), (b_h, c_h), (c_t, a_t)]$ и $[(a_h, b_h), (b_t, c_t), (c_h, a_h)]$.

Пример брейкпоинт графа для пары геномов приведен на рисунке 1.2. Представление геномов в виде брейкпоинт графа позволяет сосредоточиться рассмотрении связей между синтенными блоками и применять для этого существующие методы работы с графами. Также можно определить такую структуру для количества геномов больше двух: для этого для каждого следующего генома выбирается новый цвет и его ребра добавляются к имеющимся. Брейкпоинт граф для множества геномов также называется “множественный брейкпоинт граф” [15]. Таким образом, получается структура, хранящая в себе информацию о смежностях блоков сразу во многих геномах определенных на этих блоках. Полученный множественный брейкпоинт граф для N геномов при условии, что каждый из геномов определен на одном и том же множестве синтенных блоков будет N-регулярным, вставка или удаление блоков в какие-либо из геномов приведет к тому, что N-регулярность потеряется, но в остальном не изменит устройства графа.

Имея брейкпоинт граф для геномов становится возможным ввести оценку филогенетического расстояния между ними. Для этого введем операцию выполняемую над брейкпоинт графом, называемую 2-брейк.

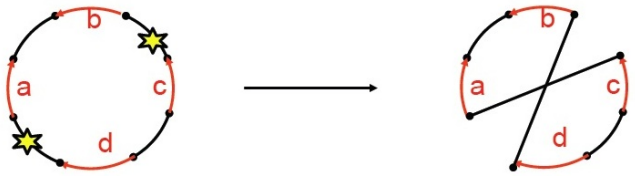

Рис. 1.3. 2-брейк

Определение 7. *2-брейк (Double Cut and Join, DCJ) [16]*
Назовем 2-брейком следующую операцию: удаление 2 ребер в брейкпоинт графе и добавление новых двух ребер на "освободившихся" вершинах несовпадающих с удаленными (рисунок 1.3).

При выполнении 2-брейка над брейкпоинт графом количество компонент связности в нем может уменьшиться или увеличиться. Когда количество компонент связности в брейкпоинт графе достигнет максимума (граф разобьется на циклы длины 2), для двух геномов это будет обозначать, что они трансформировались в один, то есть пришли к состоянию генома их общего предка. Так как последовательность 2-брейков, которая приведет к такому состоянию графа может быть любой длины, если выбрать кратчайшую такую последовательность, то ее длина будет равна оценке эволюционного расстояния между геномами при условии того, что в процессе эволюции происходили только 2-брейки.

Определение 8. *2-брейк расстояние (DCJ-расстояние, d_{DCJ})*
2-брейк расстояние - длина кратчайшей последовательности из 2-брейков приводящей исходный брейкпоинт граф в состояние с наибольшим числом компонент связности.

Данное расстояние может быть расширено для учета различных структурных изменений строения молекулы ДНК [17]. На практике также используется оценка расстояния, называемая брейкпоинт расстояние [18].

Определение 9. *Брейкпоинт расстояние (BP-расстояние, d_{BP})*
Брейкпоинт расстояние - расстояние, вычисляемое по формуле $d_{BP} = n - a - \frac{t}{2}$,

где n - количество генов в каждом геноме, a - количество общих связностей для двух геномов, t - количество общих связностей, где один конец связности является вершиной ∞.

Метрика брейкпоинт расстояния наиболее простая в вычислении и не требует выбора модели эволюции (определения, какие операции могли происходить в процессе эволюции) [18].

Используя введенные определения расстояний можно перейти к обзору существующих инструментов, решающих поставленную выше проблему.

1.2. Существующие решения

В решении задачи восстановления филогенетических деревьев есть три основных подхода:

1. На основе матрицы расстояний (Distance Methods, DM)
2. Максимального правдоподобия (Maximum Likelihood, ML)
3. Максимальной бережливости (Maximum Parsimony, MP)

Далее рассматриваются инструменты представляющие все три подхода.

1.2.1. TreeInferer (Ragout) и TIBA [1]

Оба инструмента восстанавливают деревья с на основе матрицы расстояний. Восстановление деревьев в данном подходе делится на две части: поиск попарных расстояний между геномами и восстановление дерева из матрицы известных расстояний (с возможным пересчетом матрицы на каждом шаге). И подсчет расстояний и восстановление дерева в данном подходе может осуществляться с использованием различных оценок, что дает подходу гибкость. TreeInferer, будучи частью сборщика Ragout [19], может работать с несобранными данными и использует d_{BP} в связке с Neighbour Joining [20]. TIBA работа-

ет только с собранными геномами, использует оценку d_{DCJ} и либо Neighbour Joining, либо еще один механизм восстановления деревьев на основе матрицы расстояний, FastME [21]. Стоит отметить, что оба этих инструмента работают только на геномах без вставок и удалений блоков и не могут учитывать информацию об известных поддеревьях.

1.2.2. MLWD [2]

Данный инструмент использует подход максимального правдоподобия. В данном подходе ключевым моментом является выбор вероятностной модели - математической модели, которая позволяет оценить правдоподобие филогенетического дерева при условии имеющихся листовых геномов и далее выбрать то дерево, что имеет наибольшее правдоподобие. Данный подход делает инструменты с ее использованием труднорасширяемыми, но взамен позволяет строить модели более точно отражающие эволюционные процессы происходящие в действительности. MLWD использует данное преимущество и потому поддерживает работу с блоками, полученными из несобранных геномов, с вставками, удалениями и дупликациями генов, но также как и прошлые инструменты не может использовать информацию об известных поддеревьях. Кроме того, MLWD принимает на вход только блоки, полученные из линейных хромосом.

1.2.3. GAS Phylogeny

Этот инструмент использует подход максимальной бережливости. Суть данного подхода состоит в построении модели, в которой каждое эволюционное событие имеет вклад в оценку восстановленного дерева, и дальнейшего нахождения дерева с наименьшей такой оценкой. Можно сказать, что метод максимального правдоподобия это частный случай метода максимальной бережливости, когда функция оценки является еще и вероятностью получить то или иное дерево в поставленной модели. Задача поиска филогенетического дерева

с наименьшей оценкой принадлежит к классу NP-трудных, потому зачастую при использовании данного подхода применяются эвристические оценки вместо точных и выбирается возможно неоптимальное, но, тем не менее, имеющее близкую к оптимальной оценку. В представленном инструменте [22] используется эвристическая оценка S_{GASTS}, позволяющая ему работать быстро и при этом выдавать точный результат. Данный инструмент не умеет обрабатывать данные, полученные из несобранных геномов, имеющих вставки или удаления и не учитывает информацию об известных поддеревьях.

1.3. Филогенетическая информация в брейкпоинт графе

Опишем основания для выбора предлагаемого в данной работе подхода. Рассмотрим теперь множественный брейкпоинт граф для геномов и некое неизвестное филогенетическое дерево для них. В процессе эволюции геном потомка получается из генома предка с помощью структурных и точечных изменений. Теперь если рассмотреть брейкпоинт граф полученный из генома потомка, то выполняя на нем операции 2-брейков обратные произошедшим можно получить из него геном предка [13]. Имея же брейкпоинт граф, построенный на основе геномов нескольких потомков, можно выполняя 2-брейки на ребрах разных цветов получать из геномов потомков геномы их общих предков, а разные последовательности выполненных 2-брейков будут задавать разные филогенетические деревья на геномах потомков. Таким образом, в брейкпоинт графе “закодирована” информация о филогенетическом дереве в виде преобразований над корневым геномом, ребра различных цветов же, содержат такую информацию для каждого из потомков. Данная информация может быть представлена, например, как требование того, что геномы потомков лежат в разных поддеревьях филогенетического дерева.

Теперь можно описать главную проблему, решаемую в данной работе.

1.4. Задача восстановления деревьев

Главной проблемой в данной работе является проблема восстановления филогенетических деревьев из брейкпоинт графа. На пути к ее решению ставятся две задачи:

1. Найти способы извлечения информации из брейкпоинт графа
2. Найти способы из извлеченной информации построить филогенетическое дерево

Глава 2

Восстановление филогенетических деревьев из брейкпоинт графа

2.1. Получение информации из брейкпоинт графа

Описанные в предыдущей главе методы вычисления редакционного расстояния между геномами можно рассмотреть в ином ключе. Редакционное расстояние в модели методов основанных на расстояниях “стягивает” геномы друг к другу, заставляя их находиться ближе в дереве, то есть, принимается, что геномы с меньшим редакционным расстоянием находятся в одном поддереве с большей вероятностью. Иными словами, расстояние дает нам филогенетическую информацию о ветвях в филогенетическом дереве.

В [23] автор рассматривает несколько оценок редакционного расстояния, включая описанные выше и вводит концепцию филогенетического паттерна позволяющую применять их к восстановлению деревьев из брейкпоинт графа.

Определение 10. *Филогенетический паттерн*
Филогенетический паттерн $P(\mathbb{G}, T)$, где $\mathbb{G}$ - множество геномов, а T - топология филогенетического дерева на них - функция, которая для данных входных геномов сопоставляет каждой из возможных топологий филогенетического дерева эвристическую оценку так, чтобы она была экстремальной только на одной топологии.

Главное свойство данной концепции в том, что имея на руках листовые геномы филогенетического дерева, можно получить всегда только одно возможное дерево. Тогда если найти филогенетический паттерн, который будет давать экстремальное значение на тех же топологиях, что и другая оценка s (например, d_{BP}), то он будет давать ту же филогенетическую информацию, что и s,

несмотря на то, что эту оценку невозможно было бы вычислить, не зная предковых геномов. Получается, что имея на руках паттерн, то становится возможным получать оптимальное дерево с точки зрения использованной оценки. Более того, возможно определить множество паттернов, каждый из которых, в общем случае, может давать экстремальное значение на разных топологиях, для этого для каждого из паттернов необходимо определить тот вес, который он вносит в оценку, для каждого паттерна это будет рассмотрено отдельно. В [23] показано, что филогенетические паттерны удобно строить на основе подграфов брейкпоинт графа, так как зачастую информации (значений оценки) получаемой на основе малых подграфов достаточно для того, чтобы отличить разные топологии филогенетического дерева и восстановить корректное. В дальнейшем филогенетический паттерн и подграф на котором он вычисляется будут использоваться взаимозаменяемо. Рассмотрим предложенный в [23] паттерн “контрастирующая смежность”.

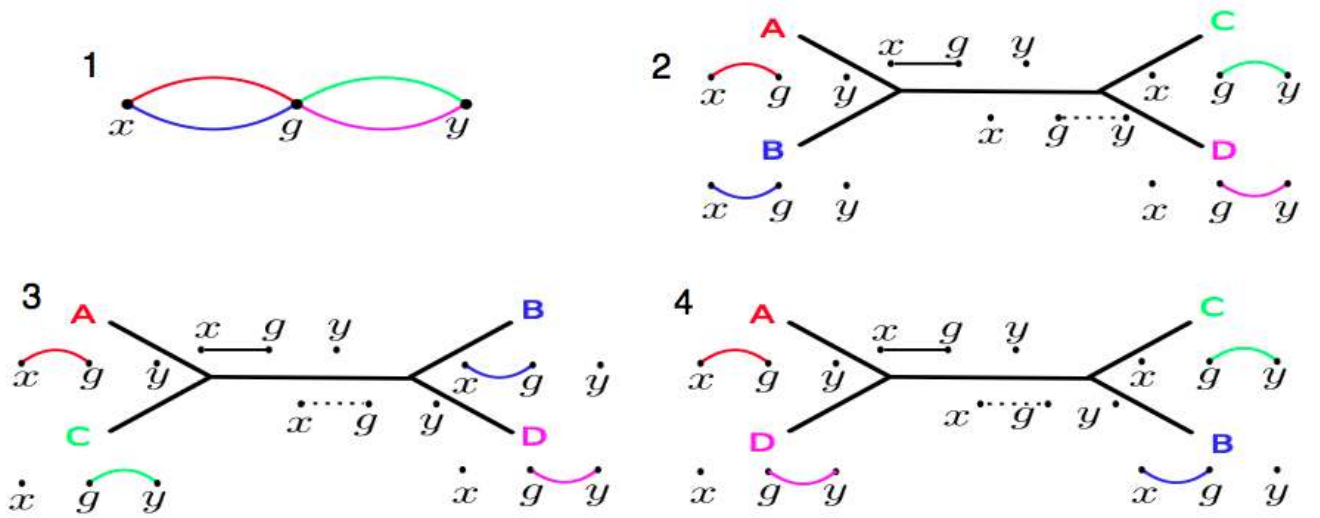

Рис. 2.1. Контрастирующая смежность. 1 - подграф, на котором базируется паттерн; 2 - дерево с топологией $AB|CD$, на котором достигается экстремальное значение d_{DCJ} и d_{BP}; 3, 4 - другие возможные топологии, на которых не достигается экстремальное значение метрик d_{DCJ} и d_{BP}

Этот паттерн представляет собой подграф брейкпоинт графа из трех вер-

шин, где одна пара соединена двумя ребрами цветов A и B, а другая - C и D. Обозначим, что для 4 геномов A, B, C, D, $AB|CD$ обозначает, что геномы A и B находятся в одном поддереве, а C и D - в другом. Тогда для топологии $AB|CD$ можно посчитать расстояние $d(d_{DCJ}, d_{BP})$ как $d(A, B) + d(C, D) + d(I_1, I_2)$, где $d(\_, \_)$ - функция расстояния между двумя геномами, I_1, I_2 - предки в дереве на парах геномов AB и CD соответственно. Из рисунка 2.1 можно увидеть, что на топологии $AB|CD$ оценки d_{BP} и d_{DCJ}, полученные как указано выше, достигают минимального значения. Вышеописанный паттерн был обобщен для многих геномов в [13] с помощью концепции *простого пути*. Для того, чтобы ввести эту концепцию введем понятие *альтернирующих мультицветов* на последовательности ребер.

Определение 11. *Альтернирующие мультицвета*
Альтернирующие мультицвета на последовательности ребер $\{e_i\}$ *- пара мультицветов* Q *и* Q'*, таких что* $\forall i$ *если цвет* e_i Q*, то цвет* e_{i+1} *-* Q' *и наоборот, если цвет* e_i Q'*, то цвет* e_{i+1} *-* Q*.*

Определение 12. *Простой путь*
Простой путь - путь в брейкпоинт графе, состоящий из вершин мультистепени 2 и ребер двух мультицветов Q *и* Q'*, таких что* $Q \cap Q' = \varnothing \wedge Q \cup Q' = \mathbb{Q}$*, где* $\mathbb{Q}$ *- множество всех цветов, если путь состоит из последовательности ребер* $\{e_i\}$*, то цвета* Q *и* Q' *альтернирующие на* $\{e_i\}$*.*

Оба этих паттерна дают более экстремальную оценку для тех деревьев, в которых геномы из каждого из альтернирующих мультицветов находятся в одном поддереве.

В брейкпоинт графе один и тот же филогенетический паттерн может встречаться несколько раз и “свидетельствовать” о разном строении филогенетического дерева (более экстремальная оценка достигается на деревьях с разной топологией), это происходит из-за того, что в паттерн включается только подграф брейкпоинт графа. Основываясь на предыдущем замечании, можно по-

нять, что информация (“свидетельства” о расположении геномов в поддеревьях филогенетического дерева) извлеченная из брейкпоинт графа будет тем точнее, чем больше непересекающихся (не использующих одни и те же основания для различия топологий) паттернов будет найдено. Так становится необходимым искать непересекающиеся филогенетические паттерны в брейкпоинт графах.

Определение паттернов вручную - тяжелый и медленный процесс, потому в данной работе предложен метод их автоматического поиска. Филогенетический паттерн - это подграф в брейкпоинт графе, раскрашенный в несколько цветов. Важно заметить два факта:

1. Каждый цвет в данном случае необязательно соответствует одному геному, но может соответствовать целой их группе.

2. Если взять ребра одного цвета из подграфа образующего филогенетический паттерн, то они не будут вершинно пересекаться и образуют паросочетание.

Первый факт важен для получения информации с помощью филогенетических паттернов из брейкпоинт графов для многих геномов, а второй позволяет перейти к автоматическому поиску филогенетических паттернов.

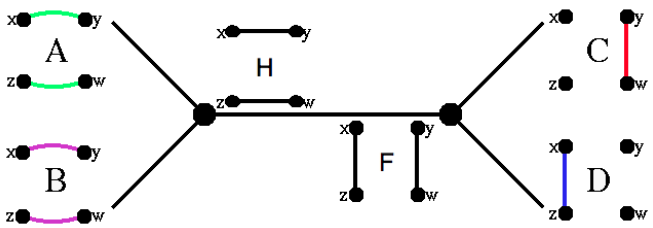

Рис. 2.2. Схема для перебора паттернов. A, B, C, D - конфигурации листовых геномов, каждый из которых является паросочетанием на вершинах брейкпоинт графа. H, F - предковые геномы, каждый из которых также является паросочетанием.

На рисунке 2.2 показана схема для перебора паттернов. Целью перебора является поиск таких 4 конфигураций геномов, на которых обе оценки (d_{BP},

d_{DCJ}) или одна из них достигает экстремального значения на одной топологии дерева, но не достигает его на других. Условие выше выполняется если найдется такая пара конфигураций внутренних вершин, что на одной из топологий достигается более экстремальное значение оценки, когда на других оно не достигается, что говорит о том, что найденный набор конфигураций есть филогенетический паттерн. Ниже представлен алгоритм поиска:

1. Выбрать с повторениями 4 паросочетания, как конфигурации геномов.

2. Для каждой из 3 топологий перебрать паросочетания для внутренних вершин (они тоже являются паросочетаниями).

3. Выбрать те кортежи из 4 геномов, на которых оценка достигает экстремума.

Важно заметить, что алгоритм выше выполняет перебор, без проверки на уникальность найденных графов, потому необходимо удалить "похожие" из них. Для этого можно определить понятие *изоморфизма паттернов*.

Определение 13. *Изоморфизм паттернов*
Паттерны A и B изоморфны тогда и только тогда, когда существует пара биективных отображений (f, h), f - между вершинами паттерна A вершинами паттерна B, h - между ребрами паттерна A и ребрами паттерна B, такая что любые две вершины u и v в A связаны мультимножеством раскрашенных ребер E тогда и только тогда, когда вершины $f(u)$ и $f(v)$ связаны $h(E)$.

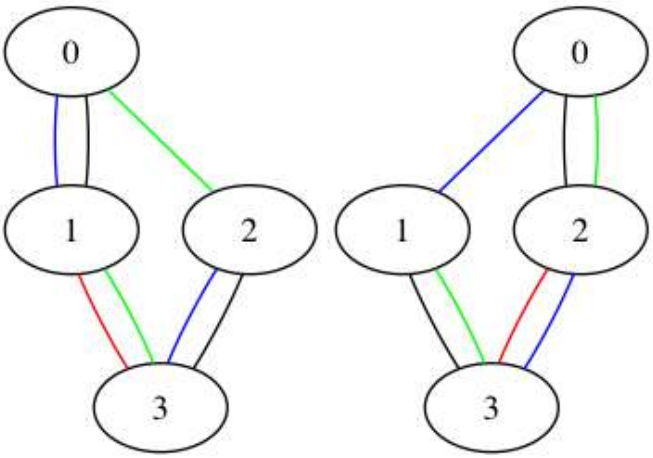

Рис. 2.3. Изоморфные паттерны

Таким образом, могут быть найдены все паттерны на любом количестве вершин. В данной работе удалось провести поиск паттернов на 4 вершинах (перебор на большем числе вершин оказался слишком вычислительно затратным), в результате него после отсеивания изоморфных паттернов и паттернов, которые учитывались бы в оценках, используемых Wei Xu (паттерны, содержащие простые пути длины больше 1 или простые циклы) остались следующие паттерны, показанные на рисунках 2.4 и 2.5. Оба этих паттерна дают более экстремальную оценку для деревьев, в которых цвета, из которых состоят ребра помеченные звездочкой, находятся в одном поддереве.

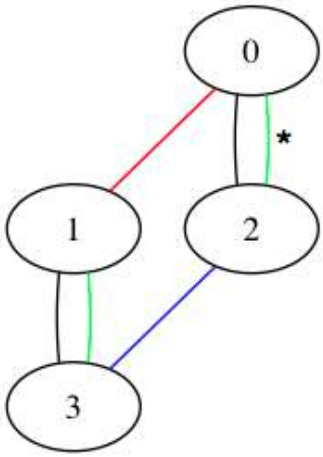

Рис. 2.4. Паттерн “цилиндр”

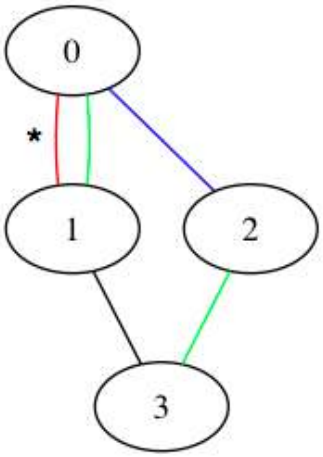

Рис. 2.5. Паттерн “мешок”

2.2. Сборка деревьев из разделений

Смысл использования филогенетических паттернов для восстановления деревьев заключается в том, чтобы извлечь информацию о устройстве филогенетического дерева из брейкпоинт графа в удобном для восстановления виде. Так каждая копия филогенетического паттерна найденная в брейкпоинт графе дает информацию о том, что одна часть геномов находятся ближе друг к другу, а другие - дальше друг от друга в филогенетическом дереве. Например, если в графе встречается паттерн “простой путь” на двух альтернирующих цветах Q и Q', то он “свидетельствует”, что геномы из множества Q находятся в одном поддереве, а геномы из множества Q' - в другом, как показано на рисунке 2.6.

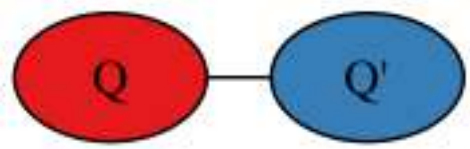

Рис. 2.6. Дерево, о котором “свидетельствует” простой путь на альтернирующих цветах Q и Q'.

Таким образом, механизм филогенетических паттернов позволяет получить из брейкпоинт графа информацию о ветвях филогенетического дерева. Сформулируем задачу восстановления филогенетического дерева из полученной информации. Для этого введем понятие *разделения* как единицы информации об устройстве филогенетического дерева.

Определение 14. *Разделение*
Пусть $\mathbb{Q}$ *- множество всех листовых геномов. Разделение - пара множеств вида* $Q_1|Q_2$*, таких, что* $Q_1 \subset \mathbb{Q}$ *и* $Q_2 \subset \mathbb{Q}$ *и при этом* $Q_1 \cap Q_2 = \varnothing$ *и* $Q_1 \cup Q_2 = \mathbb{Q}$.

Введем обозначения что для разделения вида $D = Q_1|Q_2$, $left(D) = Q_1$, $right(D) = Q_2$. Также заметим, что для дальнейшего использования разделения $Q_1|Q_2$ и $Q_2|Q_1$, где Q_1, Q_2 - подмножества множества всех геномов $\mathbb{Q}$,

идентичны.

Представим каждое вхождение паттерна P в брейкпоинт граф как разделение $Q_1|Q_2$, если согласно вхождению паттерна P геномы Q_1 должны находиться в одном поддереве результирующего филогенетического дерева, а геномы Q_2 - в другом. Таким образом, если $search(G, P)$ дает набор всех вхождений паттерна P в брейкпоинт граф G, а $toDivision_P(I)$ преобразует вхождение I паттерна P в разделение, то набор разделений $\mathbb{D}$ полученный из брейкпоинт графа G и множества паттернов $\mathbb{P}$ можно выразить как $\bigcup_{P\in\mathbb{P}}\bigcup_{I\in search(G,P)}\{toDivision_P(I)\}$. В таком случае все разделения считаются одинаково ценными и неясно, как восстановить дерево из них, так как непонятно, как выбирать из двух противоречащих друг другу разделений то, которое должно войти в результирующее дерево, потому введем понятие *свидетельства.*

Определение 15. *Свидетельство*
Свидетельство - пара разделения и оценки, указывающей, насколько весома информация о наличии данного разделения в результирующем филогенетического дерева.

Например, простой путь длины l на альтернирующих цветах Q и Q' дает свидетельство в $\lfloor\frac{l}{2}\rfloor$ единиц и свидетельство $(Q|Q',\ \lfloor\frac{l}{2}\rfloor)$. Далее, сгруппируем все свидетельства с одинаковыми разделениями и сложим их оценки. Тогда мы получим множество свидетельств $\mathbb{S} = \{(D, V)\}$, где каждое разделение D встречается не больше одного раза. Обозначим, что для свидетельства $S = (D, V)$, $D = div(S)$ и $V = value(S)$, а $left(S) = left(div(S))$ и $right(S) = right(div(S))$. Сформулируем теперь задачу, решение которой даст нам лучшее восстановленное дерево из возможных на основе выделенной филогенетической информации.

Задача 16. *Восстановление филогенетического дерева из свидетельств*
Восстановить филогенетическое дерево с наилучшей оценкой имея на входе набор свидетельств, полученных из брейкпоинт графа.

В данной работе будет описано два алгоритма восстановления деревьев из полученных свидетельств: наивный алгоритм и реализация динамическим программированием. Для описания алгоритмов введем отношение *непересечения* между двумя разделениями.

Определение 17. *Непересекающиеся разделения*
Два разделения $Q_1|Q_2$ *и* $R_1|R_2$ *не пересекаются, если* $Q_1 \cap R_1 = \varnothing \vee Q_1 \cap R_2 = \varnothing$.

Отношение непересечения симметрично.

2.2.1. Наивный алгоритм

Суть данного алгоритма в том, чтобы перебрать все деревья, о которых "свидетельствует" брейкпоинт граф и выбрать из них лучшее. Отличие данного алгоритма от полного перебора всех деревьев с их оцениванием в том, что основываясь на свидетельствах полученных из имеющегося брейкпоинт графа часть из них построить невозможно, а поэтому перебор уменьшится. Введем понятие *класса непересекающихся разделений*, которое описывает возможное дерево в виде разделений.

Определение 18. *Класс непересекающихся разделений*
Класс непересекающихся разделений F *над множеством разделений* $\mathbb{D}$ *- подмножество* $\mathbb{D}$*, в котором любые два разделения не пересекаются между собой.*

Таким же образом определим понятие *класса непересекающихся свидетельств.*

Определение 19. *Класс непересекающихся свидетельств*
Класс непересекающихся свидетельств H *над множеством свидетельств* $\mathbb{S}$ *- подмножество* $\mathbb{S}$*, такое что* $\forall S \in H : div(S) \in F$*, где* F *- класс непересекающихся разделений над множеством разделений.*

Для класса непересекающихся свидетельств H определим его оценку $value(H) = \sum_{S \in H} value(S)$.

Определение 20. *Максимальный класс непересекающихся разделений*
Максимальный класс непересекающихся разделений F на множеством разделений $\mathbb{D}$ - класс непересекающихся разделений над $\mathbb{D}$, такой что него невозможно добавить новое разделение из $\mathbb{D}$, так чтобы он сохранил свойство непересечения.

Определение 21. *Максимальный класс непересекающихся свидетельств*
Максимальный класс непересекающихся свидетельств H над множеством свидетельств $\mathbb{S}$ - класс непересекающихся свидетельств H, такой что $\forall S \in H : div(S) \in F$, где F - максимальный класс непересекающихся разделений над множеством разделений.

Заметим, что из исходного множества разделений, в общем случае, можно получить несколько максимальных классов непересекающихся разделений.

Пример 1. *Рассмотрим случай, что $\mathbb{D} = \{AB|CD, AC|BD, A|BCD\}$, где A, B, C, D - геномы. В данном случае можно получить максимальные классы непересекающихся разделений $H_0 = \{AB|CD, A|BCD\}$ и $H_1 = \{AC|BD, A|BCD\}$.*

Из этого примера также видно, что разделение может входить в несколько непересекающихся классов разделений одновременно, что соответствует тому случаю, когда одна ветвь присутствует в разных деревьях. В [12] показано, что из класса непересекающихся разделений можно построить дерево единственным образом: для этого необходимо выписать разделения в таблицу $M \times N$, где M - количество разделений в классе, N - количество геномов, и для каждого генома G и разделения $D = Q_1|Q_2$ записать на их пересечении 1, если $G \in Q_1$ и ноль иначе.

На основе вышеизложенного сформулируем наивный алгоритм:

1. Выделить среди свидетельств множество максимальных классов непересекающихся $\mathbb{C}$

2. Выбрать лучший по оценке класс из $\mathbb{C}$

3. Выделить разделения из свидетельств выбранного класса и собрать из них дерево

2.2.2. Реализация динамическим программированием

Проблемой предыдущего алгоритма являлось то, что он выполнял перебор всех деревьев, о которых "свидетельствует" брейкпоинт граф, в случае многих геномов таких деревьев может быть много, что приведет к тому, что работа алгоритма может занять большое время. Каждое разделение вида $Q_1|Q_2$ задает ветку в некорневом бинарном дереве с помеченными листьями, иными словами, делит все множество геномов на два подмножества Q_1 и Q_2, топология дерева на которых (расположение геномов из подмножества в филогенетическом дереве) может быть уточнена в дальнейшем другими разделениями. Из этого следует, что поиск топологий для подмножеств может проводится независимо. Поиск топологий на подмножестве геномов G' - такая же задача как исходная, но при этом для каждого разделения D из множества разделений (и каждого свидетельства с разделением D) из D удаляются все геномы, не присутствующие в G' Теперь, если для любого свидетельства S с разделением вида $Q_1|Q_2$ из некоторого набора $\mathbb{S}$ известны топологии с наилучшей оценкой, построенные на множествах геномов Q_1 и Q_2, то чтобы найти дерево с максимальной оценкой нужно перебрать все свидетельства из $\mathbb{S}$ и найти то, оценка которого в сумме с оценками поддеревьев даст наибольшее значение. Данную логику можно применить также для того, чтобы построить оптимальные деревья для вышеупомянутых пар множеств Q_1 и Q_2, составляющих разделения свидетельств. Таким образом, приходим к задаче динамического программирования для решения которой сформулируем следующий алгоритм:

1. Обозначим, что для любого множества F размера f, $size(F) = f$. Пусть даны множества $\mathbb{G}$ геномов и $\mathbb{S}$ имеющихся свидетельств. Выпишем построчно набор множеств $\{SCLevel_i | i = 1..N\}$ в порядке убывания i, где

$N = size(\mathbb{G})$, а $\forall i, SCLevel_i$ - множество четверок вида $(C, V, V, (null, null))$, таких что $size(C) = i$, $\exists S \in \mathbb{S} : (C = left(S) \vee C = right(S)) \wedge value(S) = V$ ($null$ обозначает отсутствие значения). Компоненты четверки обозначают соответственно: множество геномов, для которого известна топология филогенетического дерева; оценку этого множества (оценку свидетельства, в разделении которого это множество присутствует); суммарную оценку построенного дерева; пару множеств геномов, на которые делится данное множество геномов.

2. Начиная с $i = 2$, $\forall c = (C, V, CS, SS) \in SCLevel_i$, $(C1, V1, CS1, SS1) \in SCLevel_j$, $(C2, V2, CS2, SS2) \in SCLevel_k$ обозначим, что $c = (C, V, V + CS1 + CS2, (C1, C2))$, если $V + CS1 + CS2$ - максимальное значение при условии $j = 1..i-1$, $k = i - j$ и $C1 \cup C2 = C \wedge C1 \cap C2 = \varnothing$

3. Пусть единственный элемент $SCLevel_N$ имеет вид $(C, V, CS, (C1, C2))$ тогда результирующее дерево можно построить сверху вниз для множеств $C1$ и $C2$, после чего объединить результаты корнем. Для множеств $C1$ и $C2$ можно выполнить сборку тем же образом, базой для индукции послужит случай, когда $C1$ и $C2$ - множества с одним элементом.

2.2.3. Примечания к практической реализации

В первой главе были описаны недостатки существующих решений, которые не позволяли бы им работать с блоками, полученными из геномов со вставками и удалениями, несобранными геномами или использовать для восстановления информацию об известных поддеревьях восстанавливаемого дерева. Опишем, как предложенный подход борется с данными проблемами.

Рассмотрим, что происходит с брейкпоинт графом при работе с данными в которых происходили вставки или удаления блоков, рассмотрим только случай циклических хромосом (в линейных различия будут случаться только в конечных блоках): при удалении или вставке блока 2 вершины брейкпоинт графа

теряют степень N (N - количество геномов), при вставке степень растет, при удалении - падает. Для того, чтобы "не потерять" структуры графа, зависящие от данных вершин в случае удаления блока выполняется добавление "протезного" ребра необходимого цвета. Далее свидетельства на графе считаются так же как и раньше, но к результирующим свидетельствам добавляется информация (в форме оценки) о том, что было добавлено ребро такого цвета. Данные преобразования приводят к тому, что даже в условиях большого количества вставок или удалений в графе все равно возможно найти структуры, которые исчезли в процессе эволюции, но при этом также не теряется информация о том, что вставки или удаления произошли.

При работе с несобранными данными возникает проблема того, что при разбиении хромосомы собранного генома теряется одна смежность на каждые два контига. Данная потеря не наносит большого вреда, так как обычно при работе с несобранными геномами блоков много больше (зачастую на несколько порядков), чем контигов, таким образом, потеря связностей из-за несобранности геномов мало влияет на процесс восстановления деревьев.

При работе с реальными данными, возможна ситуация когда из биологических источников доступна информация о расположении геномов в филогенетическом дереве. Для того, чтобы поддержать работу с известными поддеревьями используется идея того, что любое разделение из известного поддерева должно присутствовать в результирующем. Для того, чтобы обеспечить выполнение этого требования известное поддерево разбивается на разделения следующим образом:

1. Пусть есть известное поддерево $(T, Q) = ((T_1, Q_1), (T_2, Q_2))$, где T, T_1, T_2 - бинарные деревья, а Q, Q_1, Q_2 - множества геномов в содержащиеся, $\mathbb{Q}$ - множество всех геномов, тогда для такого дерева набор разделений $Divisions(T)$ будет выглядеть как $\{Q|\mathbb{Q} \setminus Q,\ Q_1|\mathbb{Q} \setminus Q_1,\ Q_2|\mathbb{Q} \setminus Q_2\} \cup Divisions(T_1) \cup Divisions(T_2)$, базой для данной индукции будет случай,

когда Q - множество из одного генома.

2. Для каждого из полученных разделений назначаем оценку $\mathbb{V} = \sum_{S \in \mathbb{S}} value(S)$, где $\mathbb{S}$ - множество свидетельств, полученных из брейкпоинт графа.

После таких операций каждое разделение из известного поддерева окажется в результате, так как при сборке любое пересекающееся с ним разделение принесет меньше веса в оценку результирующего дерева и потому не будет взято.

Глава 3

Тестирование

Тестирование результатов выполнялось в два этапа: тестирование найденных филогенетических паттернов и тестирование восстановления деревьев с помощью MGRA2.

3.1. Тестирование найденных паттернов

В тестировании на симуляционных данных используется консервативная проверка правильности восстановления - если восстановленное дерево не совпадает с верным (RF-метрика [24] не дает расстояние 0) или, в случае тестирования паттернов, другое дерево из возможных имеет ту же оценку, то дерево считается неправильно восстановленным.

3.1.1. Тестирование на геномах со случайными брейкпоинт графами

Для тестирования найденных паттернов использовался подход описанный в [23]. Генерировались случайные геномы, состоящие из одной циклической хромосомы с 200 генами.

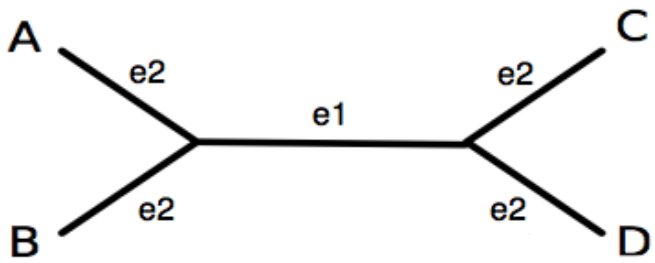

Рис. 3.1. Филогенетическое дерево симулированных геномов

На 3.1 показано филогенетическое дерево по которому генерировались тестовые четверки геномов. На ветвях филогенетического дерева происходило различное число случайных инверсий $e_1 = 1..5$, $e_2 = 5, 10, 20, 30, 40, 50, 60$. Для

каждой пары значений (e_1, e_2) генерировалось 1000 четверок геномов с случайно выбранной топологией. Результаты тестирования (доли правильно восстановленных деревьев для разных оценок) приведены в таблице ниже. Оценки S_{CA} и S_{MCA} взяты из [23], они учитывают простые пути и циклы на альтернирующих цветах. оценка S_{MCA+} получается включением информации полученной из филогенетических паттернов в оценку S_{MCA}.

e1	e2	CA	MCA	MCA+	e1	e2	CA	MCA	MCA+
1	5	0.768	0.92	0.925	4	5	1.0	1.0	1.0
1	10	0.572	0.721	0.728	4	10	0.993	0.999	0.999
1	20	0.434	0.515	0.518	4	20	0.877	0.959	0.96
1	30	0.39	0.432	0.44	4	30	0.749	0.832	0.836
1	40	0.39	0.421	0.426	4	40	0.659	0.734	0.738
1	50	0.384	0.407	0.413	4	50	0.599	0.656	0.667
1	60	0.346	0.36	0.368	4	60	0.516	0.559	0.572
2	5	0.984	1.0	1.0	5	5	1.0	1.0	1.0
2	10	0.849	0.944	0.944	5	10	0.999	1.0	1.0
2	20	0.624	0.756	0.758	5	20	0.957	0.988	0.989
2	30	0.542	0.611	0.619	5	30	0.847	0.927	0.927
2	40	0.486	0.55	0.562	5	40	0.721	0.792	0.798
2	50	0.446	0.48	0.488	5	50	0.657	0.709	0.714
2	60	0.413	0.439	0.449	5	60	0.601	0.651	0.656
3	5	1.0	1.0	1.0					
3	10	0.963	0.997	0.997					
3	20	0.777	0.874	0.878					
3	30	0.659	0.754	0.758					
3	40	0.625	0.688	0.693					
3	50	0.526	0.585	0.601					
3	60	0.481	0.528	0.542					

Таблица 3.1. Результаты тестирования на геномах со случайными брейкпоинт-графами

В таблице 3.1 приведены результаты тестирования эффективности восстановления деревьев с помощью филогенетических паттернов. Стобцы "$e1$" и "$e2$" содержат количество случайных инверсий происходящих как указано на схеме 3.1, столбцы "CA", "MCA", "$MCA+$" содержат эффективность вос-

становления для каждой конфигурации выраженную в виде доли правильно восстановленных деревьев. Как видно из таблицы 3.1, использование информации, получаемой с помощью филогенетических паттернов дает улучшение эффективности восстановления относительно оценки S_{MCA} от 0 до 1.5%. На основе того, что в некоторых случаях улучшений от использования информации полученной из филогенетических паттернов не происходило, было выдвинуто предположение, что паттернов в брейкпоинт графе в данных случаях не встречалось и проведено тестирование на случайных геномах, в брейкпоинт графах которых гарантированно были искомые паттерны.

3.1.2. Тестирование на геномах с отфильтрованными брейкпоинт графами

Генерация геномов производилась тем же образом, что и в предыдущем случае, но на параметрах $e_1 = 4, e_2 = 60$. Было сгенерировано 20434 четверки геномов, из которых 1001 четверка содержала паттерны “мешок” и “цилиндр”, и 10001 хотя бы паттерн “мешок”.

Паттерн	Доля	CA	MCA	MCA+
“мешок”	0.489	0.5386	0.5848	0.602
“мешок” и “цилиндр”	0.049	0.586	0.619	0.644

Таблица 3.2. Тестирование на геномах с отфильтрованными брейкпоинт-графами

Результаты представлены в таблице 3.2, столбец “Паттерн” содержит описание того, какие паттерны присутствовали в каждом из случайно сгененрированных графов; столбцы “CA”, “MCA”, “$MCA+$” содержат ту же информацию, что и в таблице 3.1; столбец “Доля” обозначает долю брейкпоинт графов с указанной конфигурацией от общего числа сгенерированных. Почти в половине случаев брейкпоинт граф случайно сгенерированной четверки геномов содержал хотя бы паттерн “мешок” и в этом случае наблюдалось улучшение эффек-

тивности восстановления относительно оценки S_{MCA} на 1.72%. В случае присутствия и паттерна “мешок” и паттерна “цилиндр” (а это наблюдалось в примерно 5% случаев) улучшение составило 2.5%.

3.2. Тестирование восстановления с помощью MGRA2

3.2.1. Тестирование на симуляционных данных

Для тестирования восстановления деревьев с помощью MGRA2 генерировались случайные деревья на $N = 6, 10, 12$ геномах. Каждый геном состоял из $G = 1000, 1500$ блоков. На наборе из N геномов строилось случайное полное бинарное дерево с N листьями, далее проводился симуляционный процесс вдоль каждой ветки: при прохождении по ветке геном претерпевал случайное число эволюционных событий из диапазона $[\frac{E}{2}, E]$, где $E = 100, 200$; $ID = 0.2, 0.4, 0.6$ из этих событий были вставками или удалениями одного блока, вставки и удаления происходили равновероятно. Для сравнения в тестировании были взяты инструменты TIBA, GAS Phylogeny, MLWD и выделен компонент TreeInferer из инструмента Ragout. MGRA2 запускался на двух оценках - “распределение” и “простые пути”. Первая оценка использует как “свидетельства” ребра: если $\mathbb{Q}$ - все множество цветов, то каждое ребро цвета Q добавляет 1 к оценке разделения $Q|\mathbb{Q} \setminus Q$. Вторая оценка использует идею простых путей и циклов в брейкпоинт графе - путей и циклов, состоящих из вершин мультистепени 2, а также ищет паттерны “мешок” и “цилиндр” (присутствующие в них двойные ребра используются как свидельства того, что геномы их составляющие должны быть вместе), где ребра имеют цвета попеременно Q и $\mathbb{Q} \setminus Q$. Тестирование разделено на 2 части - с вставками и удалениями и без них. В первой части участвуют все инструменты, во второй - только MGRA2 и MLWD, так как другие не могут работать с данными в которых были вставки и удаления.

Тестирование без вставок и удалений

G	N	E	GAS P	TIBA	TreeInferer	MLWD	MGRA2_D	MGRA2_SP
1000	6	100	1.0	1.0	1.0	1.0	1.0	1.0
1000	6	200	1.0	1.0	1.0	1.0	1.0	1.0
1000	10	100	1.0	1.0	1.0	1.0	1.0	1.0
1000	10	200	1.0	1.0	1.0	1.0	0.8	0.0
1000	12	100	1.0	1.0	1.0	1.0	1.0	0.5
1000	12	200	1.0	1.0	0.9	1.0	0.7	0.0
1500	6	100	1.0	1.0	1.0	1.0	1.0	1.0
1500	6	200	1.0	1.0	1.0	1.0	1.0	1.0
1500	10	100	1.0	1.0	1.0	1.0	1.0	1.0
1500	10	200	1.0	1.0	1.0	1.0	1.0	0.9
1500	12	100	1.0	1.0	1.0	1.0	1.0	1.0
1500	12	200	1.0	1.0	1.0	1.0	1.0	0.5

Таблица 3.3. Результы тестирования на симуляционных данных без вставок и удалений

В таблице 3.3 приведены результаты сравнения инструментов по восстановлению филогенетических деревьев из симулированных геномов без вставок и удалений блоков. Столбцы G, N, E содержат значения параметров описанных выше; столбцы $GASP$, $TIBA$, $TreeInferer$, $MLWD$, $MGRA2\_D$, $MGRA2\_SP$ содержат эффективность восстановления деревьев соответствующими инструментами а в случае $MGRA2$ - на оценках “распределение” и “простые пути”. Как видно из таблицы 3.3, вторая оценка, обозначенная как MGRA2_SP не показала ожидаемой эффективности (учитывая то, что она использует обобщенный паттерн, введенный Wei Xu), когда первая оценка, обозначенная как MGRA2_D, либо сравнивалась, либо незначительно проигрывала в эффективности восстановления существующим инструментам.

Тестирование со вставками и удалениями

G	N	E	ID	MLWD	D	SP	G	N	E	ID	MLWD	D	SP
1000	6	100	20	1.0	1.0	1.0	1500	6	100	20	1.0	1.0	1.0
1000	6	100	40	1.0	1.0	1.0	1500	6	100	40	1.0	1.0	1.0
1000	6	100	60	1.0	1.0	1.0	1500	6	100	60	1.0	1.0	1.0
1000	6	200	20	1.0	1.0	1.0	1500	6	200	20	1.0	1.0	1.0
1000	6	200	40	1.0	1.0	1.0	1500	6	200	40	1.0	1.0	1.0
1000	6	200	60	1.0	1.0	1.0	1500	6	200	60	1.0	1.0	1.0
1000	10	100	20	1.0	1.0	1.0	1500	10	100	20	1.0	1.0	1.0
1000	10	100	40	1.0	1.0	1.0	1500	10	100	40	1.0	1.0	1.0
1000	10	100	60	1.0	1.0	1.0	1500	10	100	60	1.0	1.0	1.0
1000	10	200	20	1.0	1.0	0.4	1500	10	200	20	1.0	1.0	1.0
1000	10	200	40	1.0	1.0	0.9	1500	10	200	40	1.0	1.0	1.0
1000	10	200	60	1.0	1.0	0.9	1500	10	200	60	1.0	1.0	1.0
1000	12	100	20	1.0	1.0	1.0	1500	12	100	20	1.0	1.0	1.0
1000	12	100	40	1.0	1.0	1.0	1500	12	100	40	1.0	1.0	1.0
1000	12	100	60	1.0	1.0	1.0	1500	12	100	60	1.0	1.0	1.0
1000	12	200	20	1.0	0.6	0.0	1500	12	200	20	1.0	1.0	0.7
1000	12	200	40	1.0	0.9	0.2	1500	12	200	40	1.0	1.0	0.8
1000	12	200	60	1.0	1.0	0.4	1500	12	200	60	1.0	1.0	1.0

Таблица 3.4. Результы тестирования на симуляционных данных со вставками и удалениями

В таблице 3.4 приведены результаты сравнения инструментов по восстановлению филогенетических деревьев из симулированных геномов с вставками и удалениями блоков. Столбцы G, N, E, ID содержат значения параметров выше; столбцы $MLWD$, D, SP содержат эффективность восстановления деревьев инструментами $MLWD$ и $MGRA2$ на метриках “распределение” и “простые пути” соответственно. Как видно из таблицы 3.4, вторая оценка, обозначенная как

SP, опять показала меньшую по сравнению с первой оценкой, обозначенной как D, эффективность, которая в этом тесте лишь в двух случаях не восстановила все деревья идеально.

3.2.2. Тестирование на реальных данных

Для тестирования на реальных данных использовались данные предоставленные московской лабораторией М. С. Гельфанда. Для тестирования использовались два набора данных: из рода Burkholderia, штаммы видов mallei, pseudomallei (возбудители сапа и мелиоидоза) и штаммы Yersinia pestis (чумная палочка).

В первом случае было восстановленное MGRA2 дерево не совпало с полученным биологическими методами. Далее было замечено, что количество перестроек для полученной биологически и восстановленной топологий отличается на 1 из 95, после этого в лаборатории было построено дерево на основе матрицы наличия/отсутствия гена. Это дерево совпало с восстановленным MGRA2.

Восстановленное MGRA2 из второго набора данных дерево совпало с полученным биологическими методами в делении на три основные пандемии чумы. Несовпадения произошли в поддеревьях с сомнительной топологией полученного биологически дерева.

По результатам можно сказать, что MGRA2 корректно восстанавливает филогенетические деревья для геномов отличающихся на значительные расстояния, либо на основе вставок и удалений блоков.

Заключение

В данной работе представлен алгоритм восстановления филогенетических деревьев из брейкпоинт графа. Представленный механизм поддерживает расширение на каждом из двух своих шагов, позволяет восстанавливать филогенетические деревья из наборов блоков со вставками и удалениями, полученными из несобранных геномов, использовать при восстановлении информацию об известных поддеревьях результирующего дерева.

Самым главным недостатком предложенного подхода для получения информации из брейкпоинт графа является факт, что найденные филогенетические паттерны встречаются не во всех брейкпоинт графах, что может быть причиной относительно низкого качества работы оценки “простые пути”.

Дальнейшее развитие данной работы может включать поиск филогенетических паттернов большей размерности или поиск новых способов извлечения информации из брейкпоинт графа.

Литература

1. Lin Y., Rajan V., Moret B. M. TIBA: a tool for phylogeny inference from rearrangement data with bootstrap analysis // Bioinformatics. 2012. Vol. 28, no. 24. P. 3324–3325.

2. Hu F., Lin Y., Tang J. MLGO: phylogeny reconstruction and ancestral inference from gene-order data // BMC bioinformatics. 2014. Vol. 15, no. 1. P. 354.

3. Ohno S. Ancient linkage groups and frozen accidents // Nature. 1973. Vol. 244. P. 259–262.

4. Pevzner P., Tesler G. Human and mouse genomic sequences reveal extensive breakpoint reuse in mammalian evolution // Proceedings of the National Academy of Sciences. 2003. Vol. 100, no. 13. P. 7672–7677.

5. Webber C., Ponting C. P. Hotspots of mutation and breakage in dog and human chromosomes // Genome research. 2005. Vol. 15, no. 12. P. 1787–1797.

6. Peng Q., Pevzner P. A., Tesler G. The fragile breakage versus random breakage models of chromosome evolution // PLoS Comput Biol. 2006. Vol. 2, no. 2. P. e14.

7. Alekseyev M. A., Pevzner P. A. Comparative genomics reveals birth and death of fragile regions in mammalian evolution // Genome Biol. 2010. Vol. 11, no. 11. P. R117–R117.

8. Renwick J. The mapping of human chromosomes // Annual review of genetics. 1971. Vol. 5, no. 1. P. 81–120.

9. Minkin I., Patel A., Kolmogorov M. et al. Sibelia: a scalable and comprehensive synteny block generation tool for closely related microbial genomes // Algorithms in Bioinformatics. Springer, 2013. P. 215–229.

10. Pham S. K., Pevzner P. A. DRIMM-Synteny: decomposing genomes into evolutionary conserved segments // Bioinformatics. 2010. Vol. 26, no. 20. P. 2509–2516.

11. Proost S., Fostier J., De Witte D. et al. i-ADHoRe 3.0—fast and sensitive detection of genomic homology in extremely large data sets // Nucleic acids research. 2012. Vol. 40, no. 2. P. e11–e11.

12. Gusfield D. Efficient algorithms for inferring evolutionary trees // Networks. 1991. Vol. 21, no. 1. P. 19–28.

13. Alekseyev M. A., Pevzner P. A. Breakpoint Graphs and Ancestral Genome Reconstructions // Genome Res. 2009, May. Vol. 19(5), pp. 943-57.

14. Bafna V., Pevzner P. A. Genome rearrangements and sorting by reversals // SIAM Journal on Computing. 1996. Vol. 25, no. 2. P. 272–289.

15. Caprara A. On the tightness of the alternating-cycle lower bound for sorting by reversals // Journal of Combinatorial Optimization. 1999. Vol. 3, no. 2-3. P. 149–182.

16. Bergeron A., Mixtacki J., Stoye J. A unifying view of genome rearrangements // Algorithms in Bioinformatics. Springer, 2006. P. 163–173.

17. Yancopoulos S., Friedberg R. DCJ path formulation for genome transformations which include insertions, deletions, and duplications // Journal of Computational Biology. 2009. Vol. 16, no. 10. P. 1311–1338.

18. Blanchette M., Bourque G., Sankoff D. Breakpoint phylogenies // Genome informatics. 1997. Vol. 8. P. 25–34.

19. Kolmogorov M., Raney B., Paten B., Pham S. Ragout—a reference-assisted assembly tool for bacterial genomes // Bioinformatics. 2014. Vol. 30, no. 12. P. i302–i309.

20. Saitou N., Nei M. The neighbor-joining method: a new method for reconstructing phylogenetic trees. // Molecular biology and evolution. 1987. Vol. 4, no. 4. P. 406–425.

21. Desper R., Gascuel O. Fast and accurate phylogeny reconstruction algorithms based on the minimum-evolution principle // Journal of computational biology. 2002. Vol. 9, no. 5. P. 687–705.

22. Xu A. W., Moret B. M. GASTS: parsimony scoring under rearrangements // Algorithms in Bioinformatics. Springer, 2011. P. 351–363.

23. Xu A. W. On exploring genome rearrangement phylogenetic patterns // Comparative Genomics. Springer, 2010. P. 121–136.

24. Robinson D., Foulds L. Comparison of phylogenetic trees // Mathematical Biosciences. 1981. — feb. Vol. 53, no. 1-2. P. 131–147. URL: http://dx.doi.org/10.1016/0025-5564(81)90043-2.