ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ И НАУКИ
САНКТ-ПЕТЕРБУРГСКИЙ АКАДЕМИЧЕСКИЙ УНИВЕРСИТЕТ —
НАУЧНО-ОБРАЗОВАТЕЛЬНЫЙ ЦЕНТР НАНОТЕХНОЛОГИЙ
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. 33, рис. 6, 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 рвсширения. Алгоритм реализован как часть программного пакета MGRA2, доступного под лицензией GNU GPL v.2.

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. Восстановление филогенетических деревьев - давно известная задача, решать которую пыталось множество исследователей с переменным успехом.

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 представлены два алгоритма, решающие задачу восстановления деревьев из разделений, информация о которых получается из брейкпоинт-графа, а также 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 восста-

новления 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)
2. Структурные (перестройки на уровне отдельных сегментов молекулы ДНК)

На данный момент науке известно множество структурных изменений молекулы ДНК (инверсия, удаление, вставка, слияние/разделение хромосом и другие). В ходе исследований последних 20 лет ученые пришли к консенсусу, что перестройки не могут происходить в случайных местах генома, так называемой "хрупкой" гипотезе.

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

Зачастую под блоками синтении понимают гены, потому в данной работе слова "ген" и "блок" одинаково обозначают "блок синтении". В данной работе геном будет представляться 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 получения такого представления генома из последовательности нуклеотидов существует целый ряд программных средств, такие как Sibelia [1], DRIMM-Synteny [2], i-ADHoRe3.0 [3] и другие.

Существует две модели перестановок: знаковые и беззнаковые. Описание проблем и задач для беззнаковых перестроек приведены не будут, так как данные знания не важны для понимания дальнейшей работы. Стоит отметить, что использования знаковых перестановок биологически оправданно. Так как молекула ДНК обладает двумя комплиментарными цепями, можно сказать, что если ген лежит на нити 3' - 5', то он имеет положительный знак, а если на 5' - 3', то отрицательный.

Таким образом, на основе введенных определений можно сформулировать решаемую проблему.

Проблема 2. *Проблема восстановления деревьев*
Для данных в виде набора геномов, в котором каждая хромосома представлена в виде перестановки над блоками синтении, восстановить филогенетическое дерево.

Прежде, чем перейти к обзору существующих решений введем понятие брейкпоинт-графа и способы оценки расстояния между геномами.

Рассмотрим одиночную циклическую хромосому, разбитую на уникальные блоки синтении. После данного разделения хромосома может быть представлена в виде графа 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. Данное представление можно обоб-

щить для линейных хромосом, если добавить специальную фиктивную вершину ∞, с которой будут связаны первый и последний синтенные блоки в хромосоме. Далее определим, что для объединения двух хромосом список смежности получается объединением списков смежностей для каждой из них. Таким образом, мы получили описание того как перейти от набора линейных или циклических хромосом, представленного в виде списка блоков синтении к графу смежностей этих блоков, представленному списком смежности вершин.

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. Пусть для объединения геномов вышеописанная структура графа определяется как индексированное объединение списков смежности, где индексами являются любые два несовпадающих цвета из множества цветов. В дальнейшем вместо слова "геном" будет также употребляться "цвет". 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", 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" [4]. Таким образом, получается структура, хранящая в себе информацию о смежностях блоков сразу во многих геномах определенных на этих блоках.

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

Определение 3. *2-брейк (Double Cut and Join, DCJ)*
Назовем 2-брейком следующую операцию: удаление 2 ребер в брейкпоинт-графе и добавление новых двух ребер на "освободившихся" вершинах несовпадающих

с удаленными.

При выполнении 2-брейка над брейкпоинт-графом количество компонент связности в нем может уменьшиться или увеличиться.

Определение 4. *2-брейк расстояние (DCJ-расстояние, d_{DCJ})*
Длина кратчайшей последовательности из 2-брейков приводящей исходный брейкпоинт-граф в состояние с наибольшим числом компонент связности называется 2-брейк расстоянием.

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

Определение 5. *Брейкпоинт-расстояние (BP-расстояние, d_{BP})*
Брейкпоинт-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 ∞.

В данный момент существует широкий набор инструментов, решающий поставленную проблему, далее представлен их обзор.

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

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 [5], может работать с несобранными данными и использует d_{BP} в связке с Neighbour Joining. TIBA работает только с собранными геномами, использует оценку d_{DCJ} и либо Neighbour Joining, либо еще один механизм восстановления деревьев на основе матрицы расстояний FastME [6]. Стоит отметить, что оба этих инструмента работают только на геномах без вставок и удалений генов и не могут учитывать информацию об известных поддеревьях.

1.2.2. MLWD

Данный инструмент использует подход максимального правдоподобия. В данном подходе ключевым моментом является выбор вероятностной модели - конструкции, которая позволяет оценить филогенетическое дерево при условии имеющихся листовых геномов и далее выбрать то дерево, что имеет наибольшее значение функции правдоподобия. Данный подход делает инструменты с ее использованием труднорасширяемыми, но взамен позволяет строить модели более точно отражающие эволюционные процесс 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.

1.2.3. GAS Phylogeny

Этот инструмент работает 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. 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 [7] используется 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

Рассмотрим теперь множественный брейкпоинт-граф для геномов и некое неизвестное филогенетическое дерево для них. В процессе эволюции молекула ДНК подвергалась структурным изменениям при движении от корня филогенетического дерева к листьям. Имея же на руках брейкпоинт-граф для листовых геномов можно сопоставить операции на нем с поворачиванием вспять истории структурных изменений, иначе говоря, движением по филогенетическому дереву от листьев к корню: при выполнении операций 2-брейк, приводящих к увеличению компонент связности, происходит, по сути, обращение структурных изменений, произошедших на ветвях дерева в процессе эволюции. 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.

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 [8] Wei Xu рассматривает несколько оценок редакционного расстояния, включая описанные выше и вводит концепцию филогенетического паттерна позволяющую применять их к восстановлению деревьев из брейкпоинт-графа.

Определение 6. *Филогенетический паттерн*
Филогенетический паттерн - эвристическая оценка, которая для данных входных геномов из всех возможных топологий филогенетического дерева примет экстремальное (максимальное или минимальное) значение только на одной из них.

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. Такая замена позволит искать в брейкпоинт-графе паттерны и на их основе вы-

полнять восстановление деревьев. Рассмотрим предложенный Wei Xu паттерн “контрастирующая смежность”.

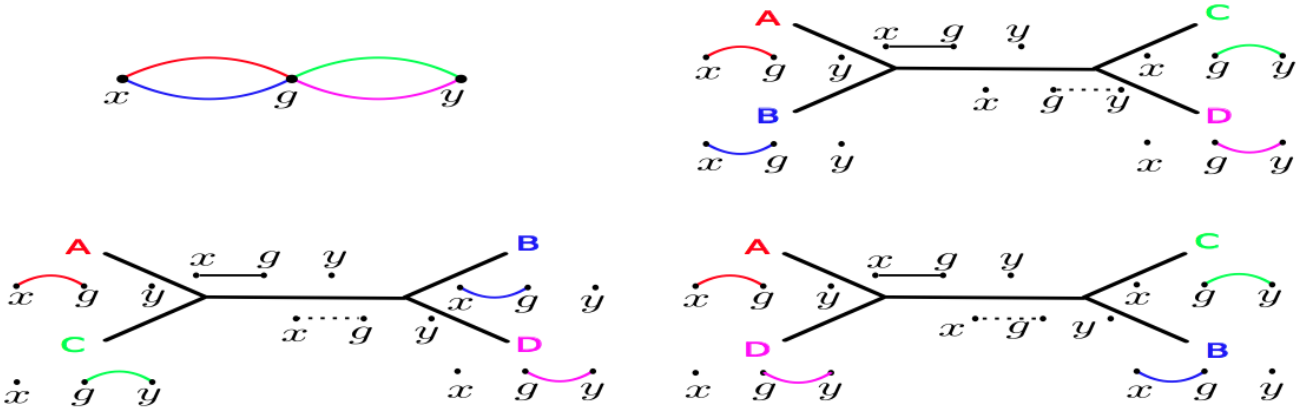

Рис. 2.1. Контрастирующая смежность [8]

Этот паттерн представляет собой подграф брейкпоинт-графа из трех вершин, где одна пара соединена двумя ребрами цветов A и B, а другая - C и D. Из рисунка 2.1 можно увидеть, что на топологии $AB|CD$ оценки d_{BP} и d_{DCJ} достигают минимального значения (0 для d_{BP} и 2 для d_{DCJ}). Вышеописанный паттерн был обобщен для многих геномов в [9] с помощью концепции *простого пути*.

Определение 7. *Простой путь*

Назовем простым путем путь в брейкпоинт-графе, состоящий из вершин мультистепени 2 и ребер двух альтернирующих (меняющих друг друга по ходу движения по ребрам) мультицветов.

Заметим что, каждый филогенетический паттерн по сути дает часть филогенетической информации, закодированной в брейкпоинт-графе. Основываясь на предыдущем замечании, можно понять, что информация 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 филогенетические паттерны в графах, для чего необходимо определить, какие паттерны искать.

Определение паттернов вручную - тяжелый и медленный процесс, потому в данной работе предложен метод их автоматического поиска. Для начала, стоит разобрать, из чего состоит 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. Ниже представлен алгоритм такого поиска.

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

2. Для каждой из 3 топологий перебрать паросочетания для внутренних вершин (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 выполнить проверку на изоморфизм найденных паттернов. Для этого можно определить *изоморфизм паттернов* как изоморфизм мультиграфов с окрашенными ребрами.

Определение 8. *Изоморфизм паттернов*
Паттерны A и B изоморфны тогда и только тогда, когда существует биекция f между вершинами паттерна A вершинами паттерна B, определенных

над одним множеством цветов, 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 E'.

Также будем считать изоморфными паттернами те, которые подпадают под определение выше под действием любой перестановки на цветах паттерна.

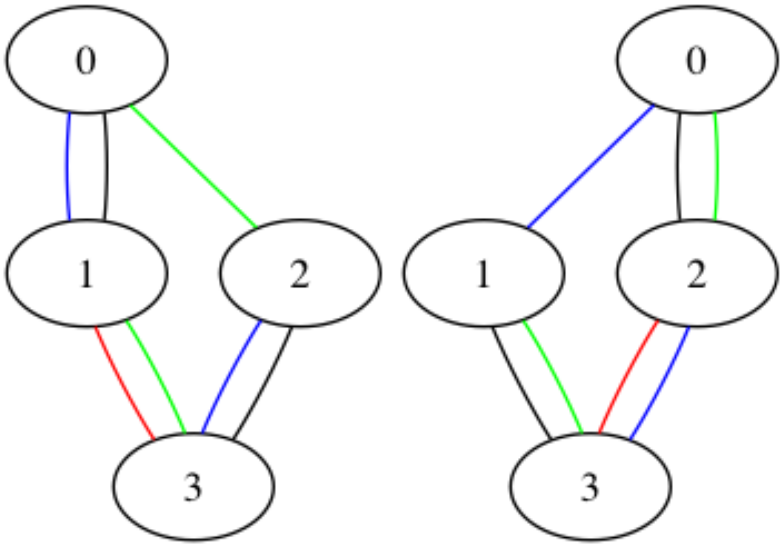

Рис. 2.2. Изоморфные паттерны

Таким образом, могут быть найдены все паттерны на любом количестве вершин. В данной работе были найдены паттерны, показанные на рисунках 2.3 и 2.4.

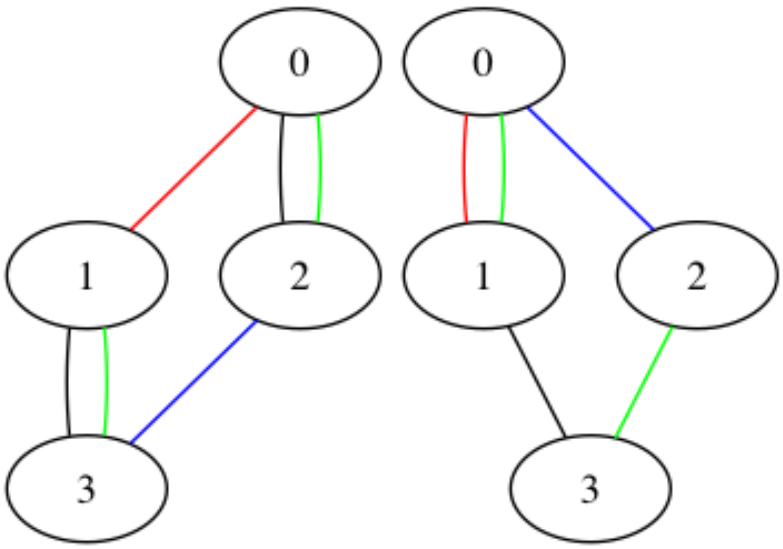

Рис. 2.3. Паттерн “цилиндр” Рис. 2.4. Паттерн “мешок”

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_1 и Q_2, то он "свидетельствует", что геномы из множества Q_1 находятся в одном поддереве, а геномы из множества Q_2 - в другом. 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 9. *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* $Q_1 \cup Q_2 = \mathbb{Q}$

Обозначим, что для разделения вида $D = Q_1|Q_2, left(D) = Q_1, right(D) = Q_2$.

Тогда представим каждое вхождение паттерна P в брейкпоинт-граф как, если согласно паттерну P геномы Q_1 должны находиться в одном поддереве результирующего филогенетического дерева, а геномы Q_2 - в другом. Таким образом, имея набор искомых паттернов и брейкпоинт-граф, можно преобразовать последний в список вхождений паттернов в него, после чего получить набор разделений $\mathbb{D}$. 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 *статистики.*

Определение 10. *Статистика*
Статистика - разделение с оценкой

Под оценкой в данном случае подразумевается то, насколько весомо "свидетельство" о данном разделении: например, простой путь длины l дает свидетельство в l единиц. Далее, сгруппируем все статистики с одинаковыми разделениями и сложим их оценки. Тогда мы получим множество статистик $\mathbb{S} = (D, V)$, где каждое разделение D встречается не больше одного раза. Обозначим оценку статистики S как $V = value(S)$, разделение как $D = div(S)$. Из этого можно сформировать задачу восстановления из статистик.

Задача 11. *Восстановление из статистик*
Восстановить филогенетическое дерево с наилучшей оценкой имея на входе набор статистик, полученных из брейкпоинт-графа.

В данной работе будет описано два алгоритма сборки деревьев из полученных статистик: наивный алгоритм и реализация динамическим программированием.

Для описания алгоритмов опишем отношение *непересечения* между двумя разделениями.

Определение 12. *Непересекающиеся разделения*
Два разделения $Q_1|Q_2$ *и* $R_1|R_2$ *не пересекаются, если* $Q_1 \cap R_1 = \varnothing \vee Q_1 \cap R_2 = \varnothing$.

2.2.1. Наивный алгоритм

Суть данного алгоритма в том, чтобы перебрать все деревья, о которых "свидетельствует" брейкпоинт-граф и выбрать из них лучшее. Стоит также заметить, что в данном алгоритме перебираются не все *возможные* деревья, но только те, о которых "свидетельствует" брейкпоинт-граф, что, в зависимости

от способа извлечения филогенетической информации из брейкпоинт-графа, может значительно уменьшить перебор. Введем понятие *класса непересекающихся разделений*, которое описывает возможное дерево в виде разделений.

Определение 13. *Класс непересекающихся разделений*
Класс непересекающихся разделений D_i - подмножество множества всех разделений $\mathbb{D}$, в котором любые два разделения не пересекаются между собой.

Таким же образом определим понятие *класса непересекающихся статистик.*

Определение 14. *Класс непересекающихся статистик*
Класс непересекающихся статистик S_i - подмножество множества всех статистик $\mathbb{S}$, такое что $\forall S \in S_i : div(S) \in D_i$, где D_i - класс непересекающихся разделений.

Для класса непересекающихся статистик S_i определим, его оценку $V_i = \sum_{S=(D,V)\in S_i} V$

Определение 15. *Максимальный класс непересекающихся разделений*
Будем называть класс непересекающихся разделений D_i максимальным, если в него невозможно добавить новое разделение из множества разделений $\mathbb{D}$, так чтобы он сохранил свойство непересечения.

Определение 16. *Максимальный класс непересекающихся статистик*
Максимальный класс непересекающихся статистик - класс непересекающихся статистик S_i, такое что $\forall S \in S_i : div(S) \in D_i$, где D_i - максимальный класс непересекающихся 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 $D_0 = \{AB|CD, A|BCD\}$ и $D_1 = \{AC|BD, A|BCD\}$.*

Из этого примера также видно, что разделение может входить в различные классы, что соответствует тому случаю, когда одна ветвь присутствует в разных деревьях.

Заметим также, что из класса непересекающихся разделений можно собрать дерево единственным образом. Здесь и далее под деревьями подразумеваются некорневые бинарные деревья с помеченными листьями, для их представления используется формат Newick [10] с фиктивным корнем, поддеревья с неизвестной внутренней топологией обозначаются как $\{A, B, C\}$, где A, B, C - геномы, топология дерева для которых неизвестна.

Пример 2. *Пусть есть класс непересекающихся разделений* $D_0 = \{AB|CDEF, ABC|DEF\}$, *где* A, B, C, D, E, F *- геномы. Используя трактовку разделений, что разделение вида* $Q_1|Q_2$ *обозначает то, что геномы из множества* Q_1 *лежат в левом поддереве, а геномы из множества* Q_2 *- в правом, можно получить два дерева, удовлетворяющих требованиям разделений из* D_0*:*

- $((\{AB\}, C), \{DEF\})$;
- $(\{AB\}, (C, \{DEF\}))$;

Если рассматривать эти деревья в постановке описанной выше, то становится ясно, что они одинаковы.

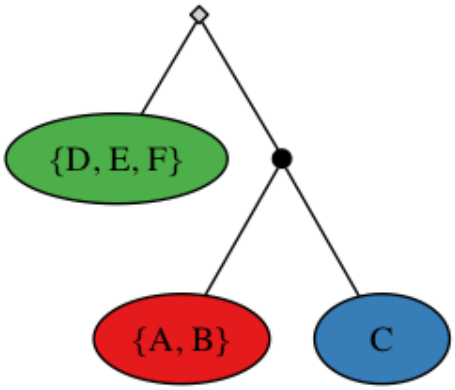

Рис. 2.5. $((\{AB\}, C), \{DEF\})$;

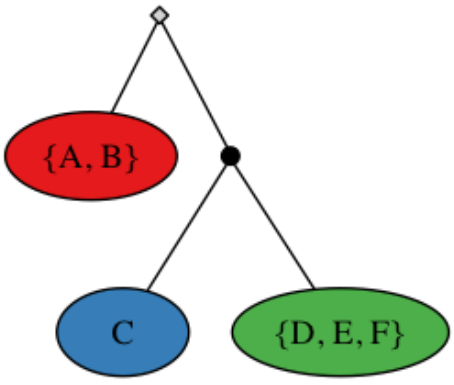

Рис. 2.6. $(\{AB\}, (C, \{DEF\}))$;

Закрашенный кружок обозначает предка, ромб обозначает фиктивный корень.

На основе вышеизложенного сформулируем наивный алгоритм:

1. Выделить среди статистик множество максимальных классов непересекающихся $\mathbb{C}$
2. Отсортировать множество $\mathbb{C}$ по убыванию оценки
3. Выбрать лучший по оценке класс
4. Выделить разделения из статистик 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. Предположим, что для любой статистики S с разделением вида $Q_1|Q_2$ из некоторого набора $\mathbb{S}$ известны поддеревья с наилучшей оценкой, построенные на множествах геномов Q_1 и Q_2, тогда чтобы найти дерево с максимальной оценкой нужно перебрать все статистики из $\mathbb{S}$ и найти ту, оценка которой 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 статистик. Таким образом, приходим к задаче динамического программирования для решения которой сформулируем следующий алгоритм:

1. Пусть для множества F размера f, $size(F) = f$. Пусть даны множества $\mathbb{G}$ геномов и $\mathbb{S}$ имеющихся статистик. Введем структуру множеств $\{SCLevel_i | i = 1..N\}$, где $N = size(\mathbb{G})$, при $\forall i, SCLevel_i$ - множество четверок вида $(C, V, V, ())$, таких что $size(C) = i$, $\exists S \in \mathbb{S} : (C = left(S) \vee C = right(S)) \wedge value(S) = V$, компоненты четверки обозначают соответствен-

но: множество геномов, на котором построено дерево; оценку этого множества; суммарную оценку построенного дерева; пару множеств геномов, лежащих в левом и правом поддеревьях.

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

3. Пусть единственный элемент $SCLevel_N$ имеет вид $(\_, \_, \_, (C1, C2))$ 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 индуктивной сборки 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 статистики на графе считаются так же как и раньше, но к результирующим статистикам добавляется информация (в

форме оценки) о том, что было добавлено ребро такого цвета.

При работе с несобранными данными возникает проблема того, что при разбиении хромосомы собранного генома теряется одна смежность на каждые 2 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.

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 поддерево $(T, Q) = ((T_1, Q_1), (T_2, Q_2))$, где T, T_1, T_2 - деревья, а Q, Q_1, Q_2 - их цвета, а $\mathbb{Q}$ - 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 статистик, полученных из брейкпоинт-графа

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

Глава 3

Тестирование

Тестирование результатов выполнялось в два этапа: тестирование для 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 [11] не дает расстояние 0) или есть другое дерево имеющее 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 Wei Xu [8]. Генерировались случайные геномы, состоящие из одной циклической хромосомы с 200 генами.

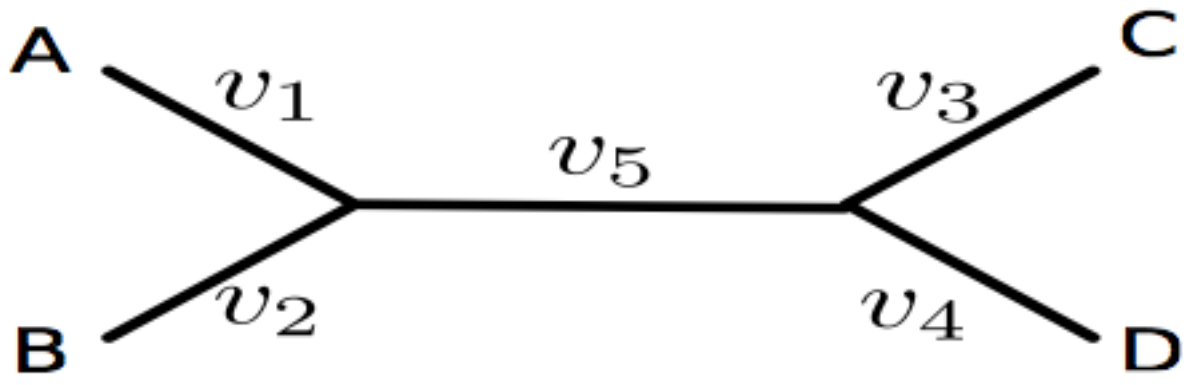

Рис. 3.1. Филогенетическое дерево симулированных геномов [8]

На 3.1 показано филогенетическое дерево по которому генерировались тестовые четверки геномов. Для генерации использовались значения $e_1 = v5, e_2 =$

$v1 = v2 = v3 = v4$, $e_1 = 1..5$, $e_2 = 5, 10, 20, 30, 40, 50, 60$. Для каждой пары значений (e_1, e_2) генерировалось 1000 четверок геномов с случайно выбранной топологией. Результаты тестирования приведены в таблице ниже. Оценки S_{CA} и S_{MCA} взяты из [8], оценка S_{MCA+} получается включением информации полученной из филогенетических паттернов в оценку S_{MCA}.

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

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					

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 проведено тестирования 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”.

Таблица 3.2. Тестирование на геномах с отфильтрованными брейкпоинт-графами

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

Как видно из таблицы 3.2, почти в половине случаев брейкпоинт-граф случайно сгенерированной четверки геномов содержал хотя бы паттерн “мешок” и в этом случае наблюдалось улучшение эффективности 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: на 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, 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

Таблица 3.3. Результы тестирования на симуляционных данных 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

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 восстановлении.

Тестирование со вставками и удалениями

Таблица 3.4. Результы тестирования на симуляционных данных 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

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 и рода Yersinia.

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.

Заключение

В данной работе представлен механизм 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.

У предложенного механизма есть ряд незначительных недостатков. Самым главным недостатком использования филогенетических паттернов для получения информации из брейкпоинт-графа является факт, что данные паттерны встречаются не во всех брейкпоинт-графах, что может быть причиной относительно низкого качества работы оценки на основе “простых путей”.

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

Литература

1. 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.

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

3. 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.

4. 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.

5. 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.

6. 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.

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

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

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

10. Archie J., Day W. H., Maddison W. et al. 1986. URL: http://evolution.genetics.washington.edu/phylip/newicktree.html (дата обращения: 2015-06-03).

11. 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.