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

Никита Карташов
Руководитель:
М.А. Алексеев (Университет Джорджа Вашингтона), доцент, Ph.D.

СПбАУ РАН

16 Июня 2015

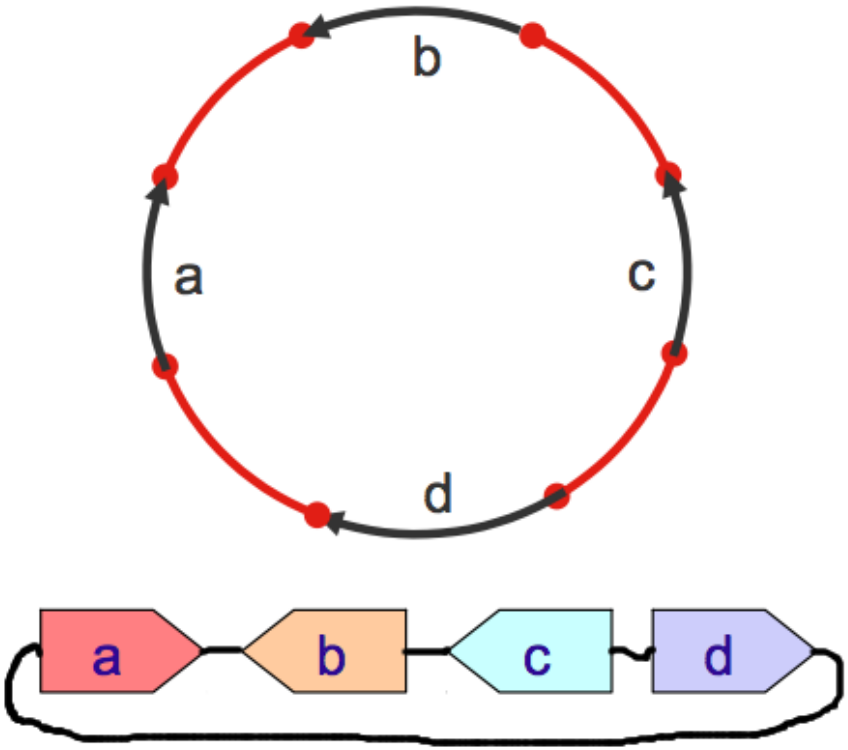
Актуальность работы

- Нет инструментов для восстановления деревьев с возможностью
  - Обрабатывать данные с присутствием вставок и удалений генов
  - Обрабатывать данные с несобранными геномами
  - Использовать информацию об известных поддеревьях
- Нет возможности встроить в существующие решения методы восстановления деревьев по информации полученной из новых структур

Существующие решения

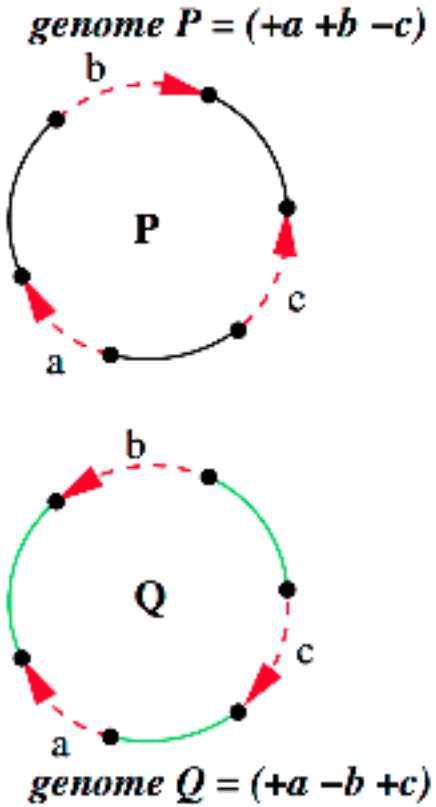
- TreeInferer из Ragout (несобранные данные; BP + NJ)
- MLWD (данные с вставками и удалениями генов, дупликациями генов, разбитые на контиги; ML)
- TIBA (обычные данные; DCJ + NJ/FastME)
- GAS Phylogeny (обычные данные; MP)

Эволюция и перестройки

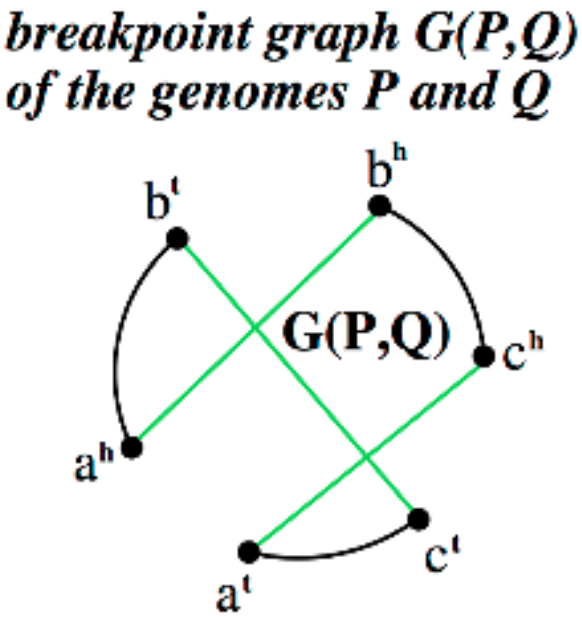


$$[(a_h, b_h), (b_t, c_h), (c_t, d_t), (d_h, a_t)]$$

Эволюция и перестройки 2



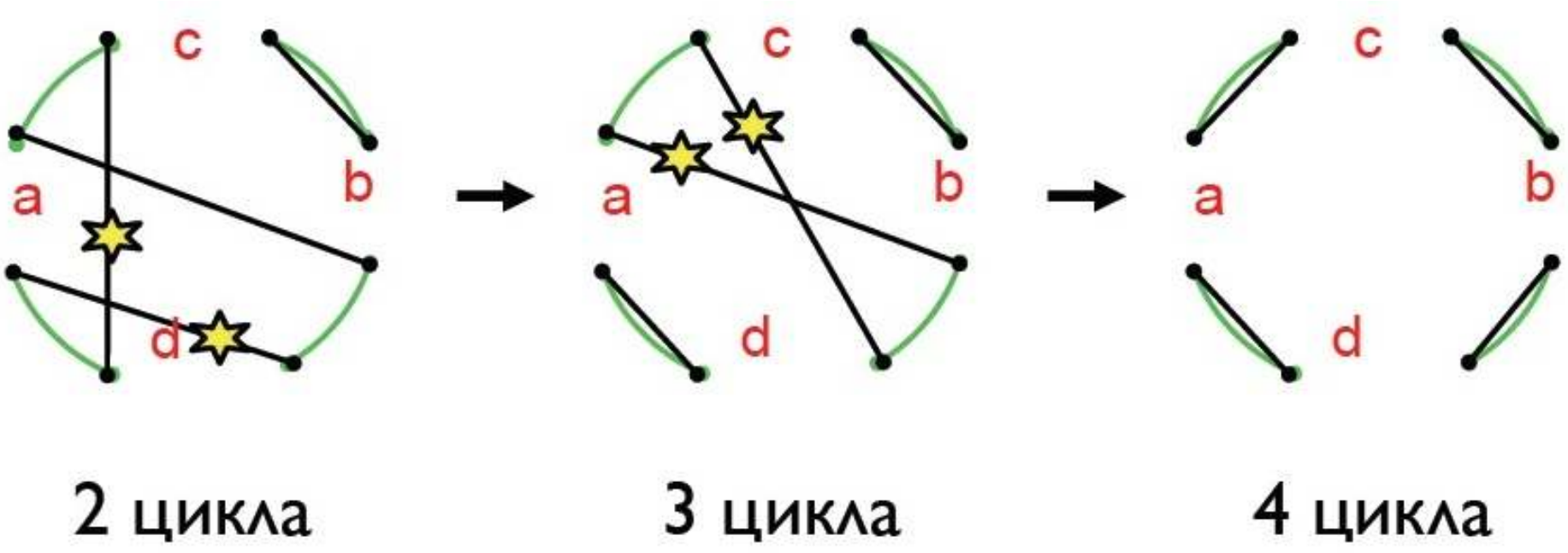
$$[(a_h, b_t), (b_h, c_h), (c_t, a_t)] \cup_i [(a_h, b_h), (b_t, c_t), (c_h, a_h)] =$$
$$[(0, (a_h, b_t)), (0, (b_h, c_h)), (0, (c_t, a_t)), (1, (a_h, b_h)), (1, (b_t, c_t)), (1, (c_h, a_h))]$$



[Alekseyev and Pevzner, 2009]

Преобразуем брейкпоинт-граф

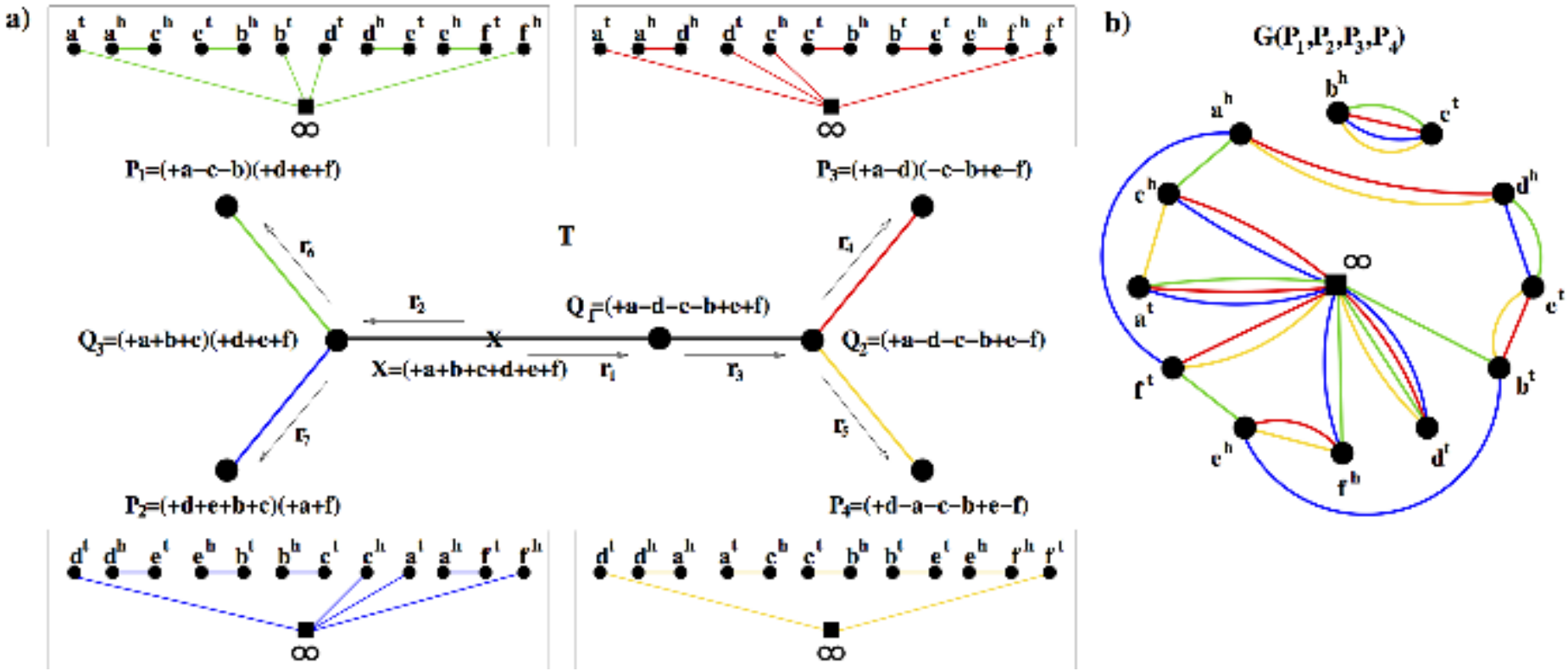
Брейкпоинт-граф задает нам расстояние между парой геномов



[Alexeev]

Breakpoint-граф и филогения

Для филогении имеет смысл рассматривать брейкпоинт-граф для многих геномов, задавая связи в них разными цветами.



[Avdeyev et al, 2015]

MGRA

- MGRA занимается восстановлением предковых геномов на основе брейкпоинт-графа
- Брейкпоинт-граф содержит филогенетическую информацию, которую MGRA мог бы позволить извлечь.

Цель и задачи

Цель: восстанавливать филогенетические деревья из геномов (несобранных, с вставками и удалениями генов) на основе брейкпоинт-графа.
Задачи:

- Найти способы извлечения информации из брейкпоинт-графа
- Научиться восстанавливать деревья по полученным данным

Оценки Wei Xu

В статье [Wei Xu, 2010] вводится понятие *филогенетического паттерна* - эвристики, которая дает экстремального значение на одной топологии, не давая его на других.
Wei Xu использует следующие эвристические оценки:

- S_{BP} = #всех смежностей - #общих смежностей
- S_{DCJ} = #всех смежностей - #циклов
- S_{CA} - учитывает пути в брейкпоинт-графе
- S_{MCA} - учитывает пути и циклы в брейкпоинт-графе

Также Wei Xu вводит паттерн «контрастирующая смежность», обобщением которого служат простые пути в MGRA

Найденные паттерны

Таким образом, были найдены следующие паттерны, не включающиеся в оценки Wei Xu:

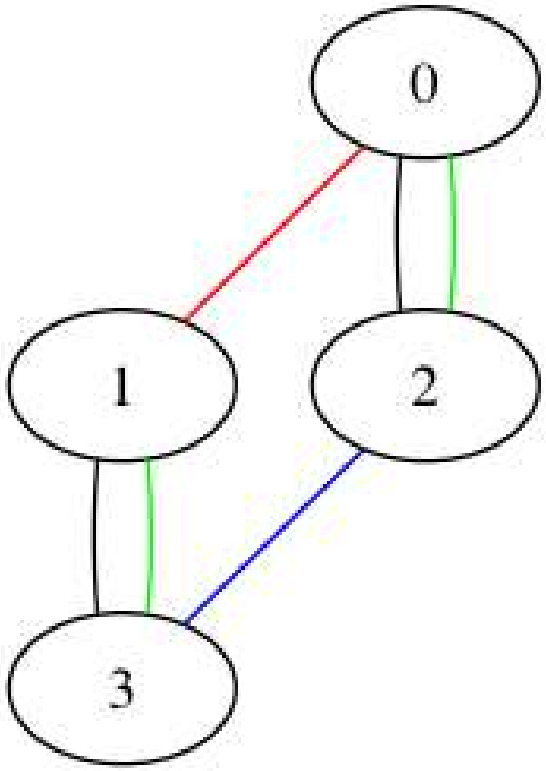


Рис. 1: Паттерн «цилиндр»

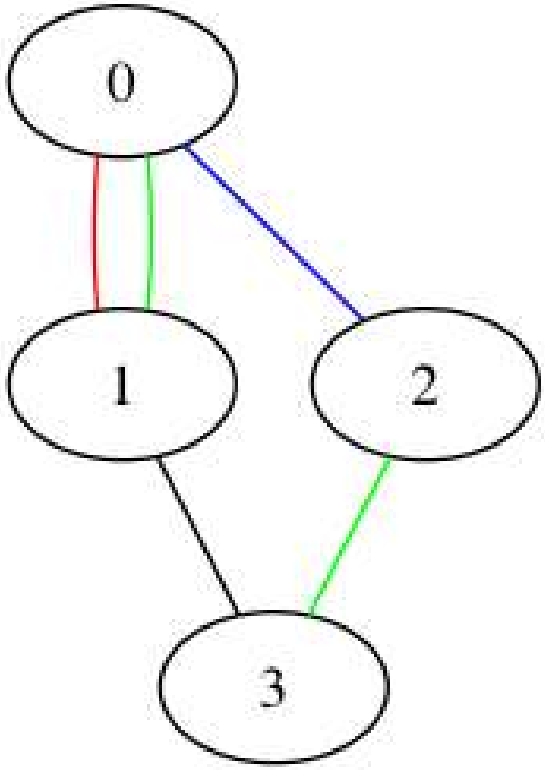
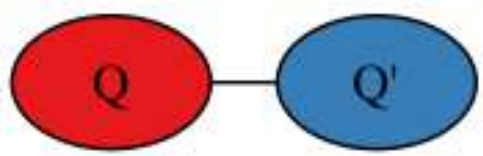


Рис. 2: Паттерн «мешок»

Разделения

- *Разделение* $Q_1|Q_2$ - разбиение множества геномов Q на такие подмножества Q_1 и Q_2, что $Q_1 \cap Q_2 = \varnothing$ и $Q_1 \cup Q_2 = Q$
- Разделение задает взаимное расположение геномов в поддеревьях - $Q|Q' \equiv$ подмножество геномов Q находится в одном поддереве, а подмножество геномов Q' - в другом



Назовем разделение D с эвристической оценкой S - *свидетельством*.

Задача реконструкции из свидетельств

- На входе: набор свидетельств.
- На выходе: собранное дерево (деревья) по данным свидетельствам с максимальной оценкой.

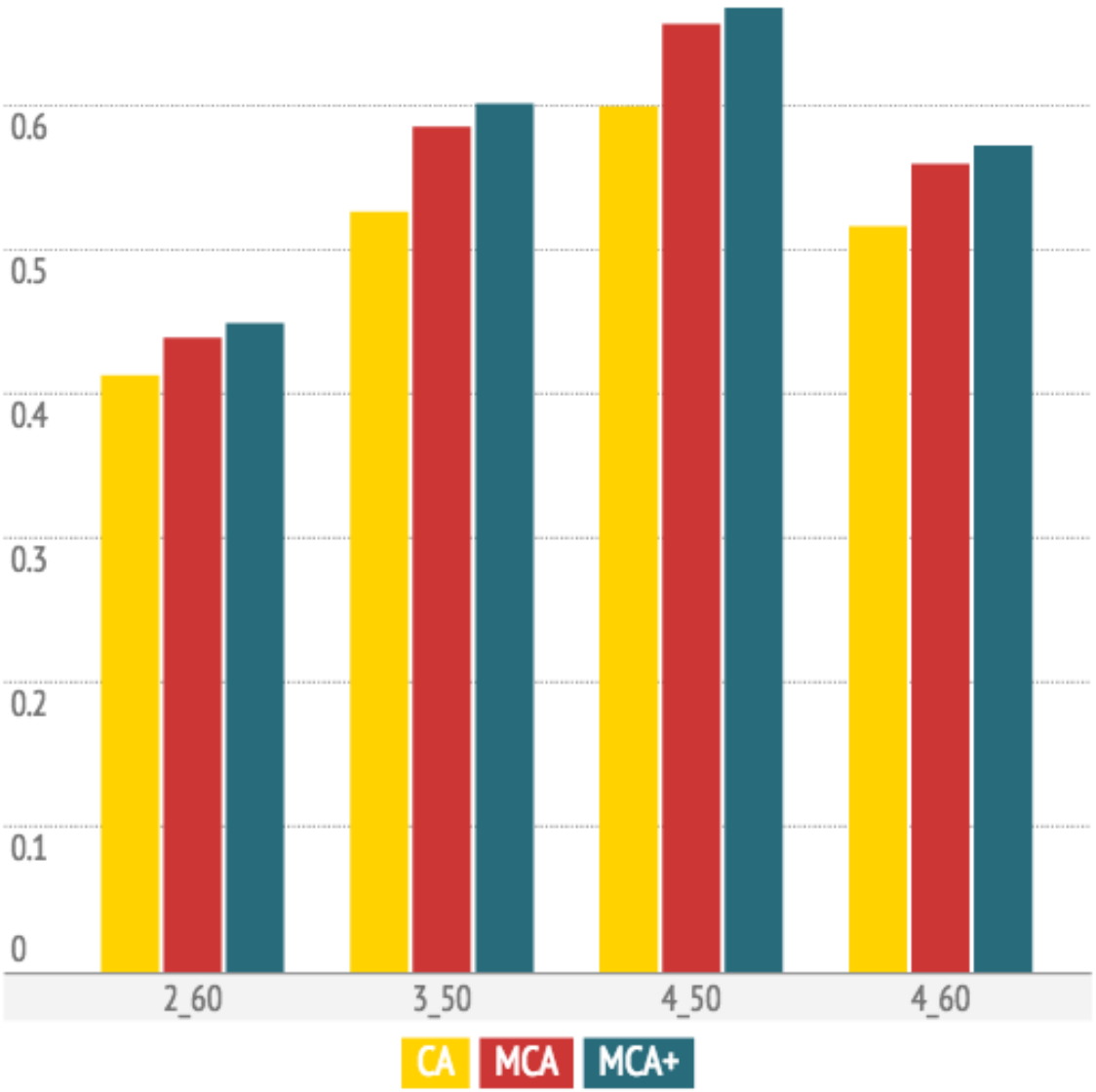
Решение динамическим программированием

1. Введем многоуровневую структуру, такую что для каждого разделения вида $Q_1|Q_2$, где Q_1 имеет размер i, а Q_2 - j, запишем Q_1 на i-тый уровень, а Q_2 на j-тый уровень
2. Снизу вверх, будем проверять возможно ли выразить множество Q с i-того уровня как объединение непересекающихся множеств с уровней j и k, так что $i = j + k$, если да, то припишем их оценку к оценке Q
3. Выберем подмножество на котором достигается самая высокая оценка и построим на нем дерево

Решение проблем существующих иструментов

- Для борьбы с вставками и удалениями на брейкпоинт-графе выполняется балансировка
- В несобранных геномах теряется одна связь на каждое разделение на 2 контига, учитывая, что блоков, как правило, много, а контигов на несколько порядков меньше, несобранность оказывает небольшое влияние
- Для сборки с известными поддеревьями оцениваем каждое разделение из известного поддерева в большое значение и восстанавливаем как обычно

Эффективность восстановления с помощью паттернов



Выигрыш от 0 до 1.5%

Эффективность восстановления с помощью паттернов. Отфильтрованные графы с паттернами.

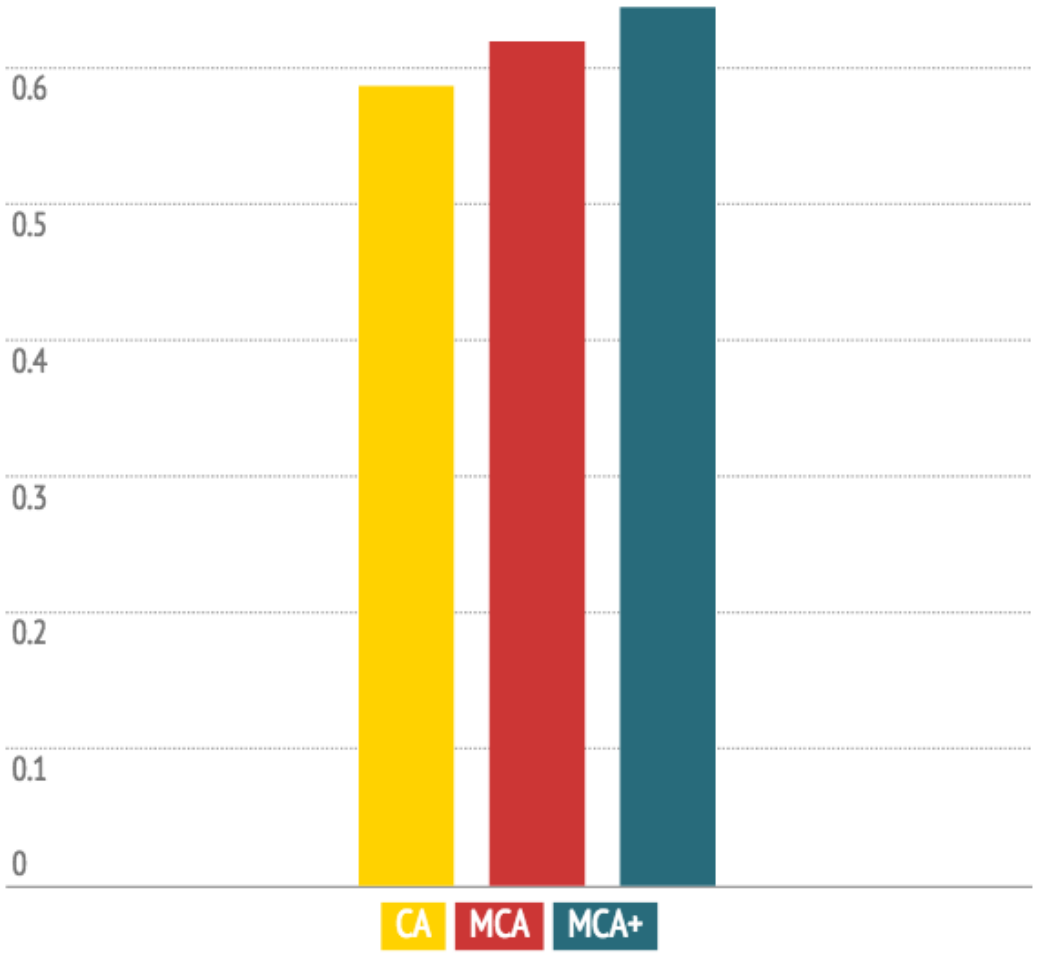
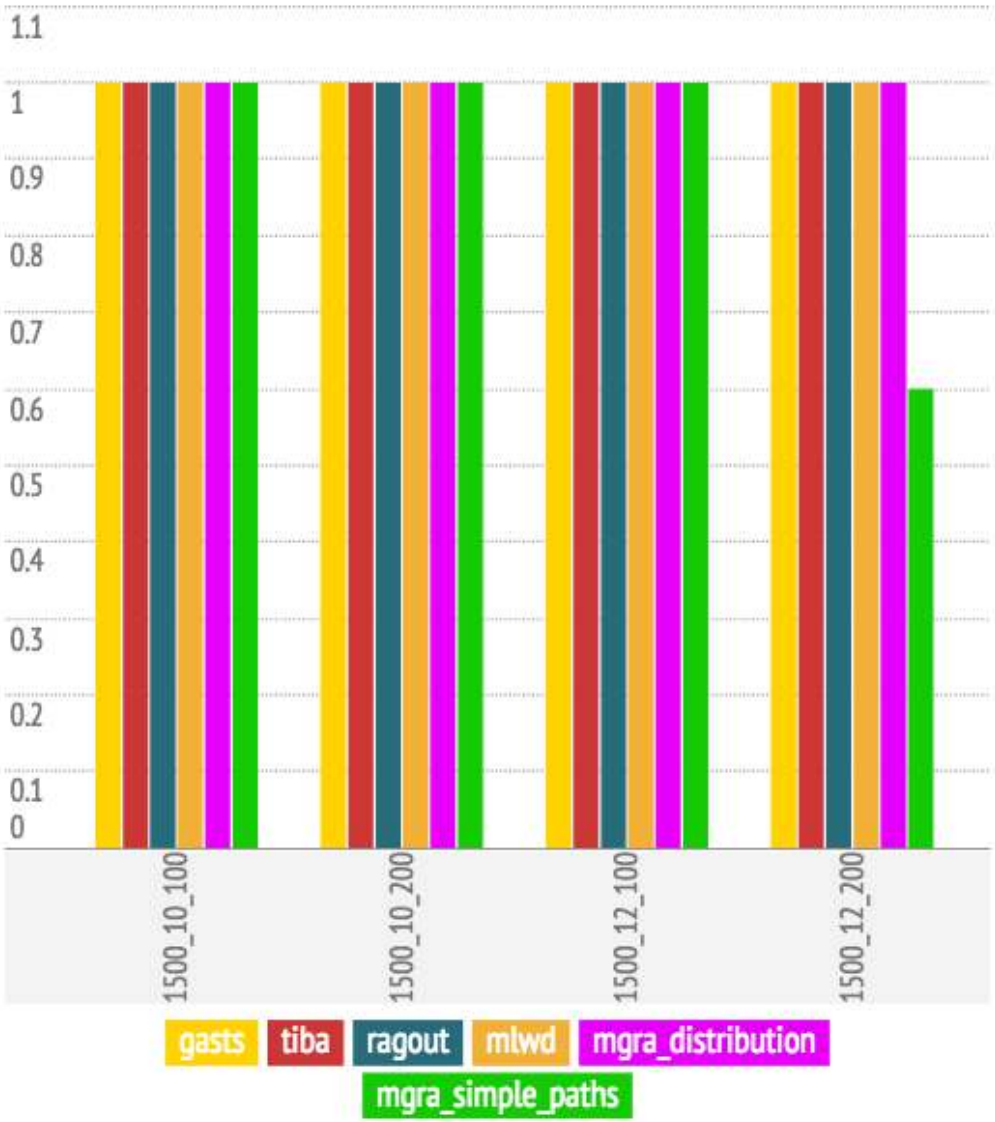


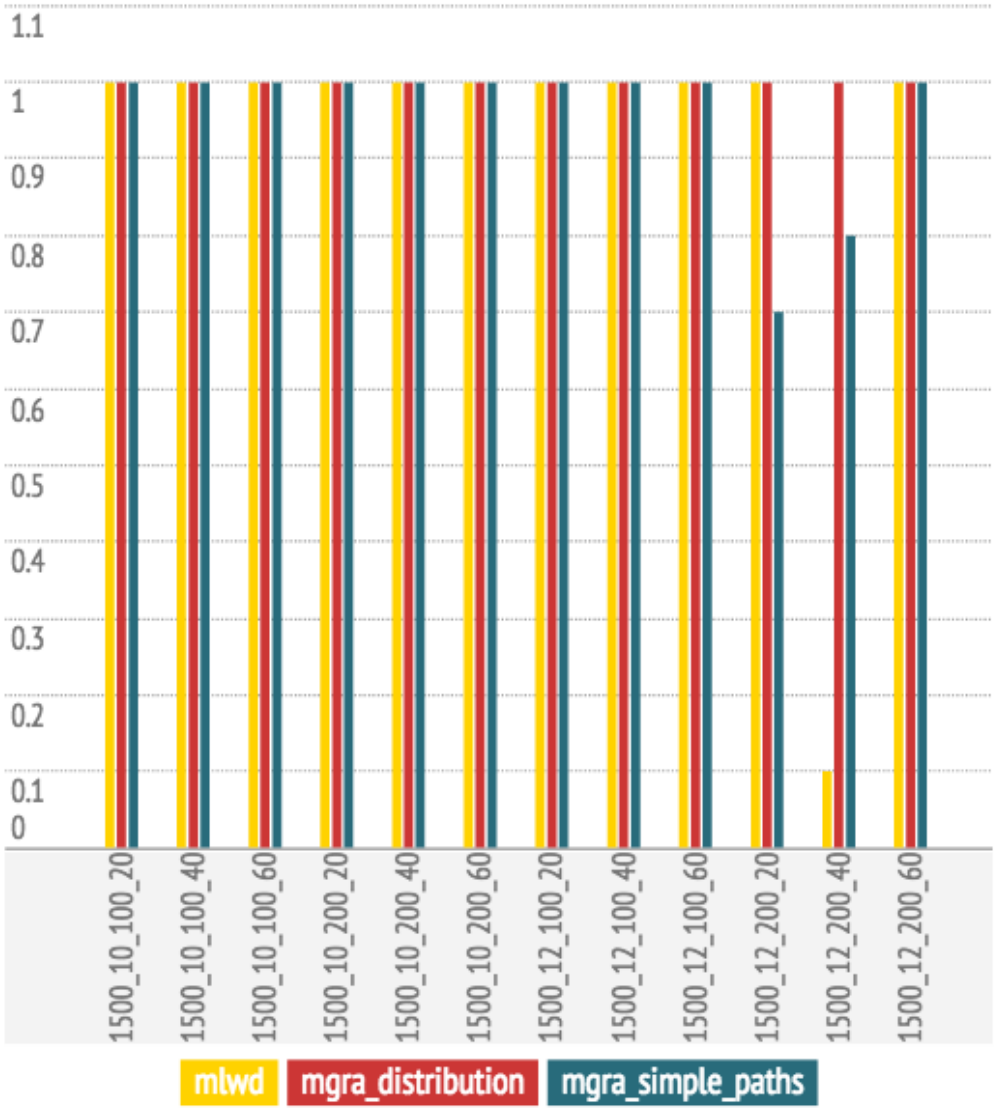
Рис. 4: Эффективность оценок

- Выигрыш 2.5% на 5% графов
- Выигрыш 1.7% на 50% графов

Сравнение инструментов на N геномах. Без вставок и удалений блоков.



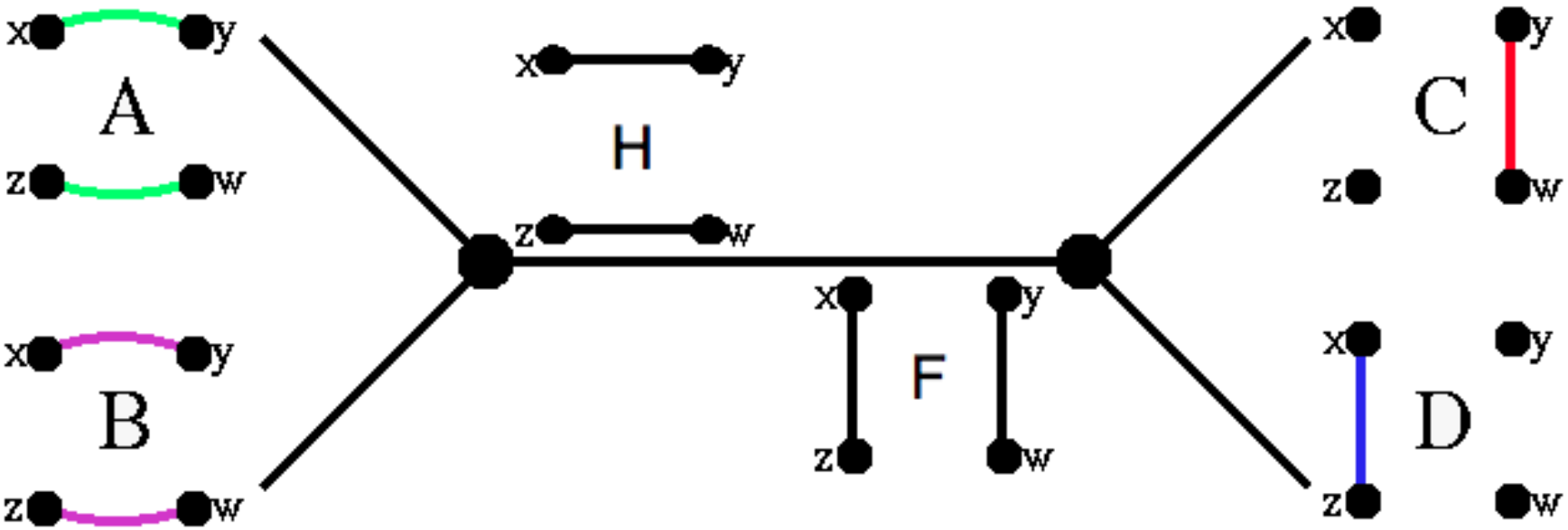
Сравнение инструментов на N геномах. Со вставками и удалениями блоков.



Результаты

- Найдено 2 филогенетических паттерна под S_{DCJ}, которые улучшают S_{MCA}
  `https://github.com/nkartashov/4genome_simulator`
  `https://github.com/nkartashov/matchings-enumeration`
  `https://github.com/nkartashov/4genome_tester`
- Реализовано 2 алгоритма восстановления деревьев из разделений
  `https://github.com/ablab/mgra/tree/recover-tree`
- Помогли разработчикам MLWD исправить их инструмент

Автоматический поиск паттернов



1. Перебор геномов в листьях и внутренних вершинах
2. Перебор для каждой конфигурации 3 топологий
3. Выбор конфигураций геномов с экстремальными значениями
4. Удаление изоморфных паттернов

Автоматический поиск паттернов 2

Пользуясь описанной идеей запускаем перебор на 4 вершинах и находится 108 паттернов, большая часть из которых «похожи» друг на друга.

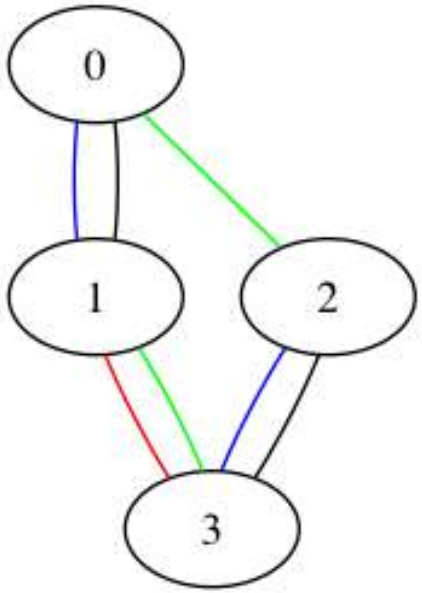


Рис. 5: Паттерн 1

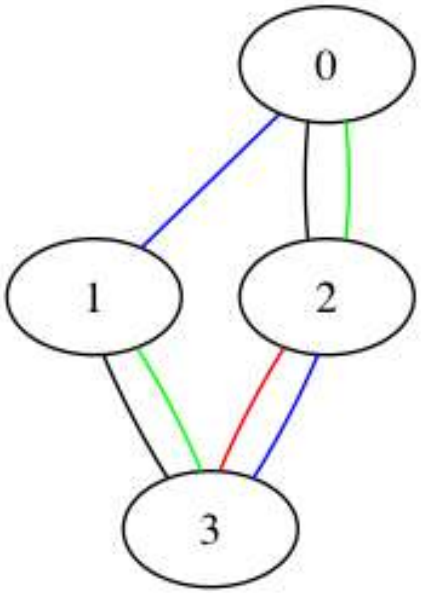


Рис. 6: Паттерн 2

Тогда определим *изоморфизм паттернов*, как изоморфизм мультиграфов с раскрашенными ребрами и добавим в наш алгоритм

Поиск паттернов больших размерностей

Все найденные паттерны были размерностей меньше 6 вершин, что будет если искать такие паттерны для больших размерностей?

- На 6 вершинах есть 75 парасочетаний (не обязательно совершенных)
- Для внутренних вершин существует 5625 конфигураций
- 3 топологии

Итого: перебор на $1426425 \times 5625 \times 3 = \mathbf{24\ 070\ 921\ 875}$, что чересчур много для перебора, но существует идея, как ускорить перебор с помощью жадного алгоритма.

Решение «в лоб»

Разделения $Q_1|Q_2$ и $R_1|R_2$ *не пересекаются*, если $Q_1 \cap R_1 = \varnothing \vee Q_1 \cap R_2 = \varnothing$

- A|BCD и AB|CD - не пересекаются
- AB|CD и AC|BD - пересекаются

1. Выделим подмножества попарно непересекающихся
2. Соберем из каждого подмножества по дереву
3. Имея на руках деревья с их оценками, отсортируем деревья по оценкам, выберем лучшие

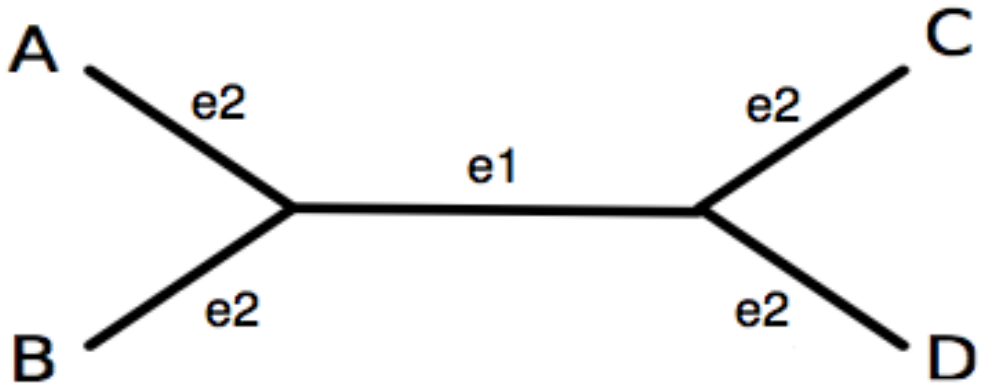
Сборка деревьев

$Q_1|Q_2$ - обозначает, что Q_1 в левом поддереве, Q_2 - в правом. Пусть на руках есть разделения AB|CDEF, ABC|DEF. Тогда в результате получатся деревья:

- (({AB}, {C}), {DEF})
- ({AB}, ({C}, {DEF}))

Так как рассматриваются некорневые деревья с помеченными листьями, то такие деревья одинаковы.

Поиск паттернов



случайные инверсии на геномах из одной циклической хромосомы длиной 200 генов

$e1 = 1, 2, 3, 4, 5$

$e2 = 5, 10, 20, 30, 40, 50, 60$

по тысяче случайно сгенерированных наборов на каждую конфигурацию