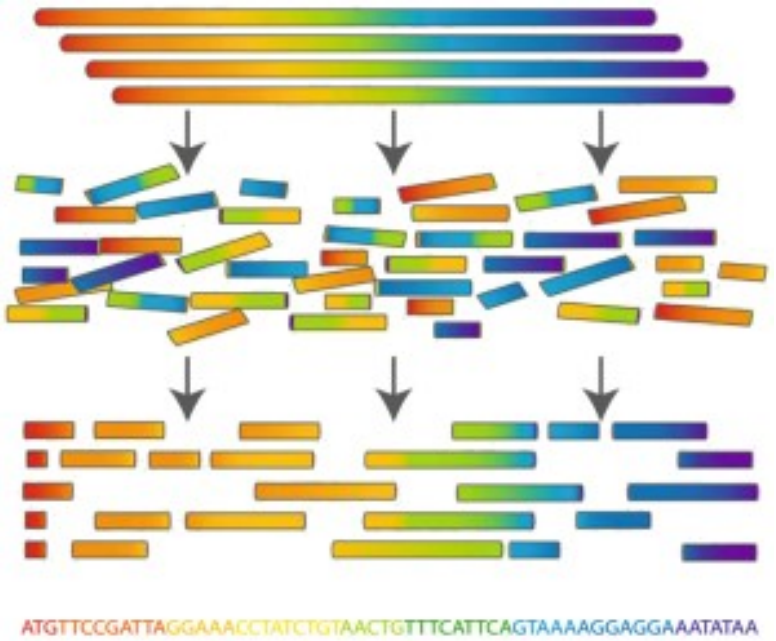
Ragout – алгоритм для сборки генома с использованием нескольких референсных последовательностей

Студент: Михаил Колмогоров
Руководитель: Son Pham, Ph.D.

Академический Университет 2014

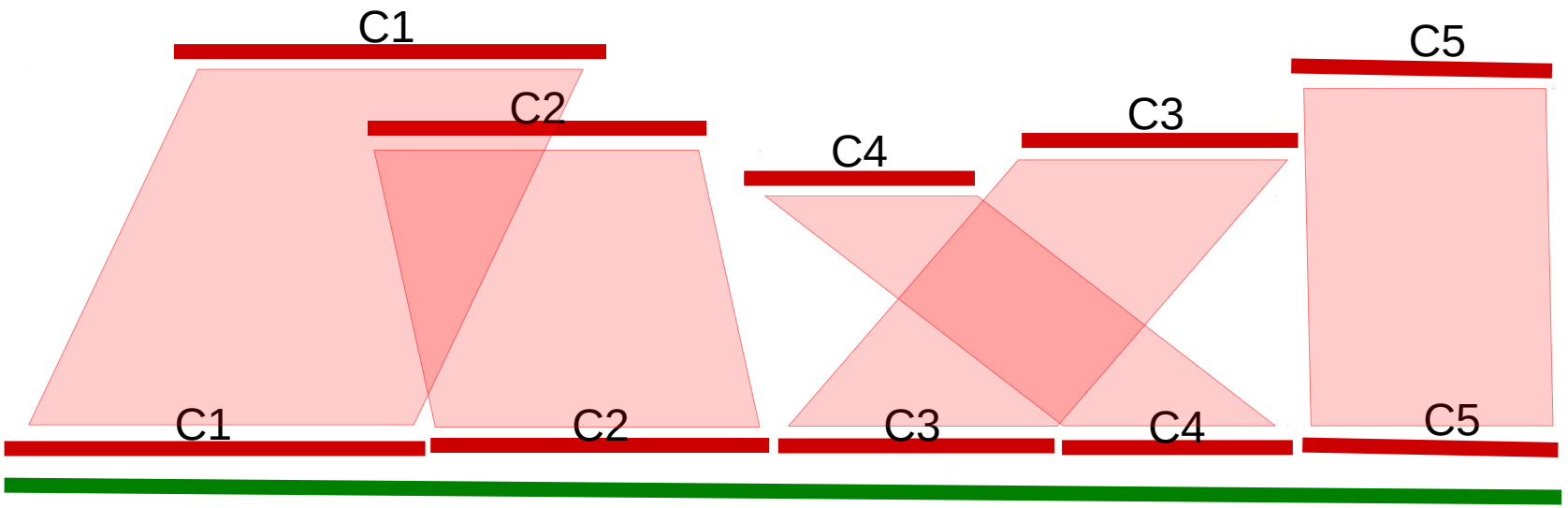
Сборка генома

- Короткие перекрывающиеся фрагменты – *риды* – объединяются в более длинные – *контиги*
- Наша задача – дальнейшее объединение контигов



Референсная сборка

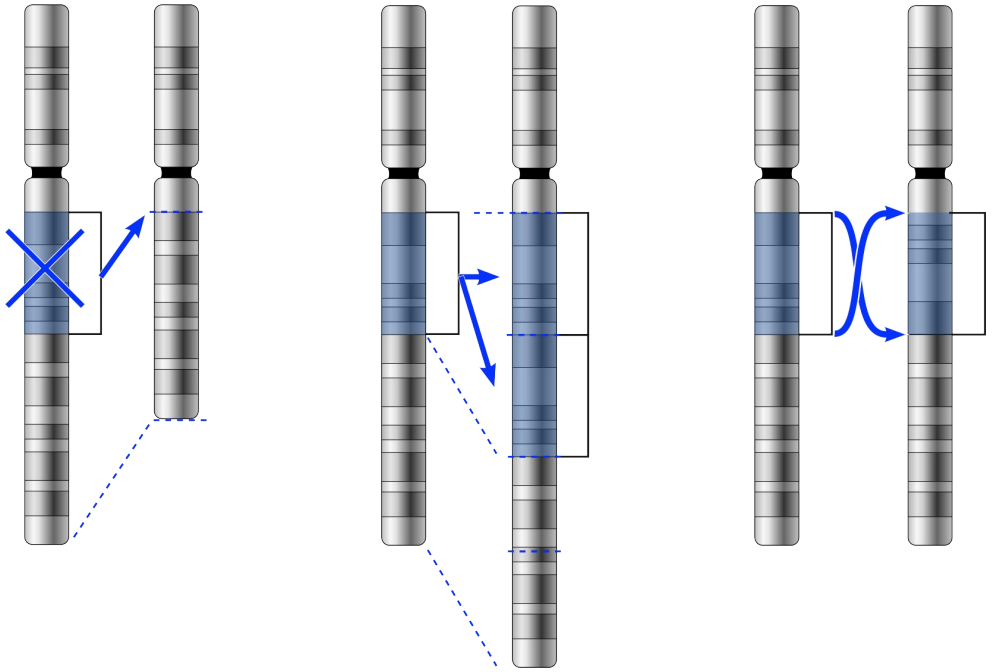
- Используем собранный геном близкородственного организма
- Найдем отображение контигов на позицию в геноме



Референсный геном R

Проблема: геномные перестройки

- В ходе эволюции в геномах могут проходить крупномасштабные перестройки
- Необходим их анализ

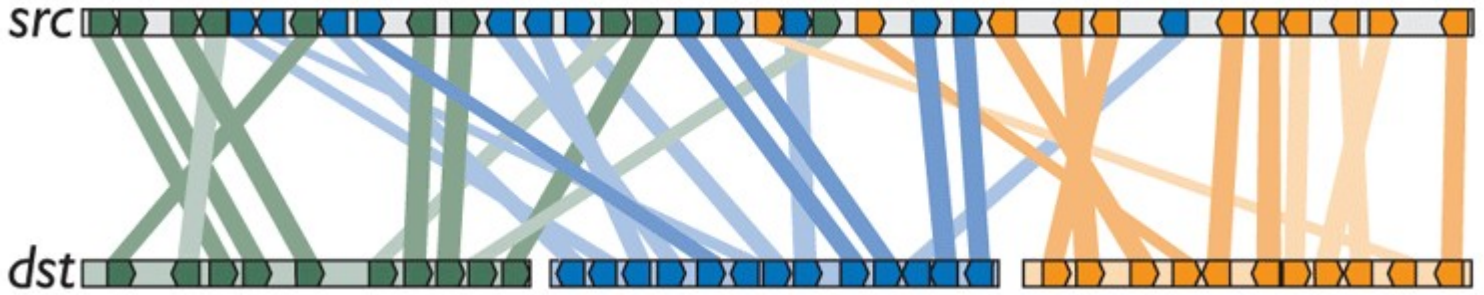


Постановка задачи

- Алгоритм для референсной сборки
- Должен включать в себя анализ геномных перестроек
- Должен использовать информацию из нескольких референсных геномов
- Сборка должна быть правильной и как можно более полной

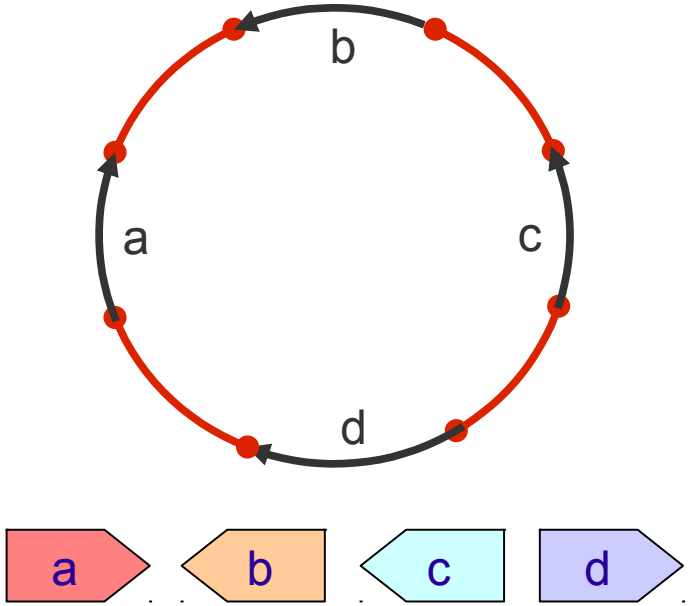
Представление геномов

- Обычно, геном – строка над алфавитом {A, C, G, T}
- Мы рассматриваем его в более крупном масштабе – в виде перестановки из синтенных блоков



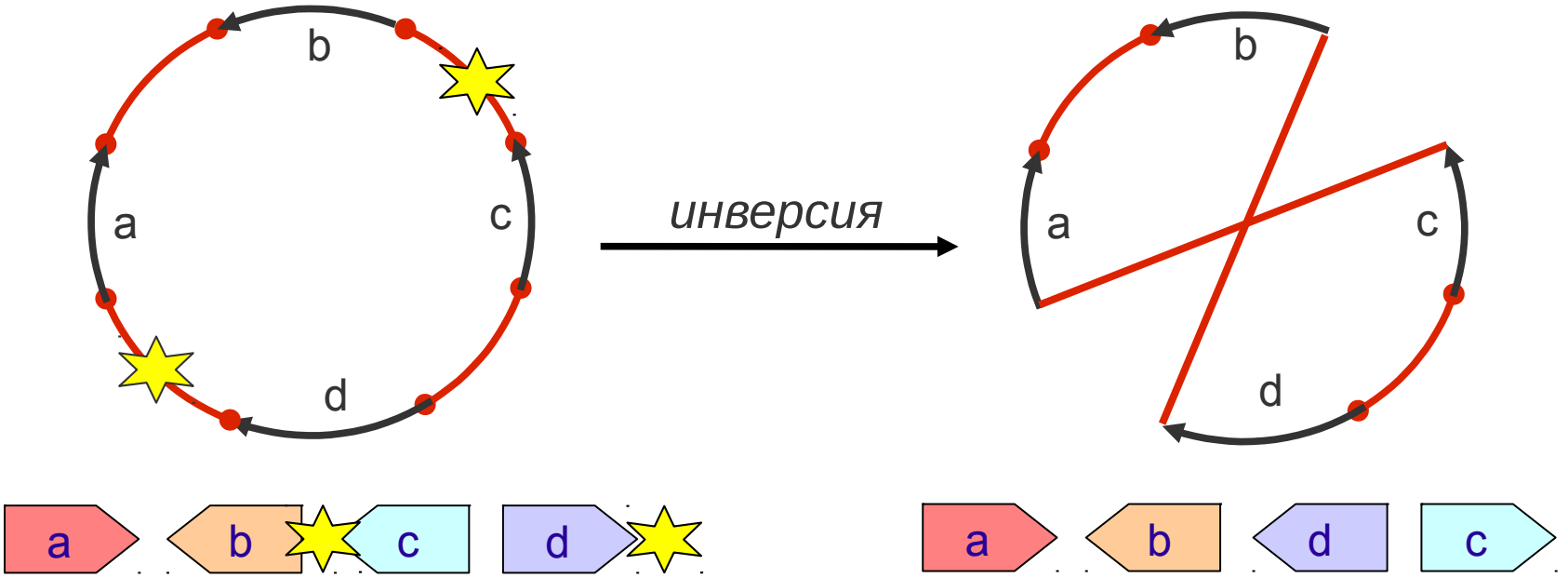
из Meyer et. al., 2009

Геном как набор синтенных блоков и смежностей



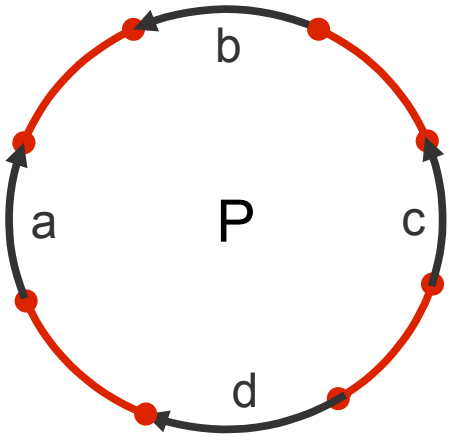
- Хромосома пердставляется как цикл из *ориентированных черных* и *неориентированных красных* ребер
- *Черные* ребра соответсвуют синтенным блокам
- *Красные* ребра соединяют соседние блоки

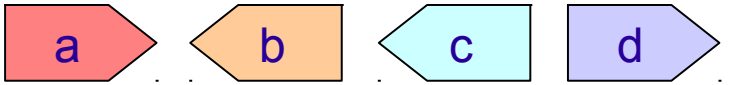
Перестройка как изменение смежностей

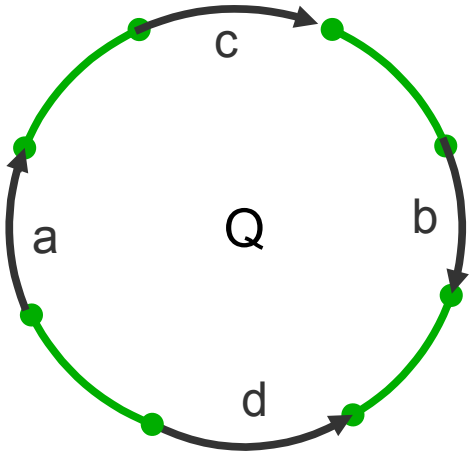


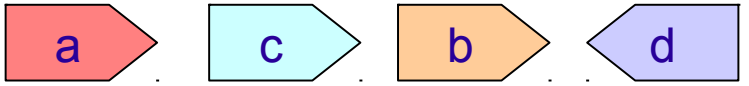
- Инверсия удаляет пару красных ребер, а затем проводит два новых красных ребра

Граф брейкпоинтов – это просто!

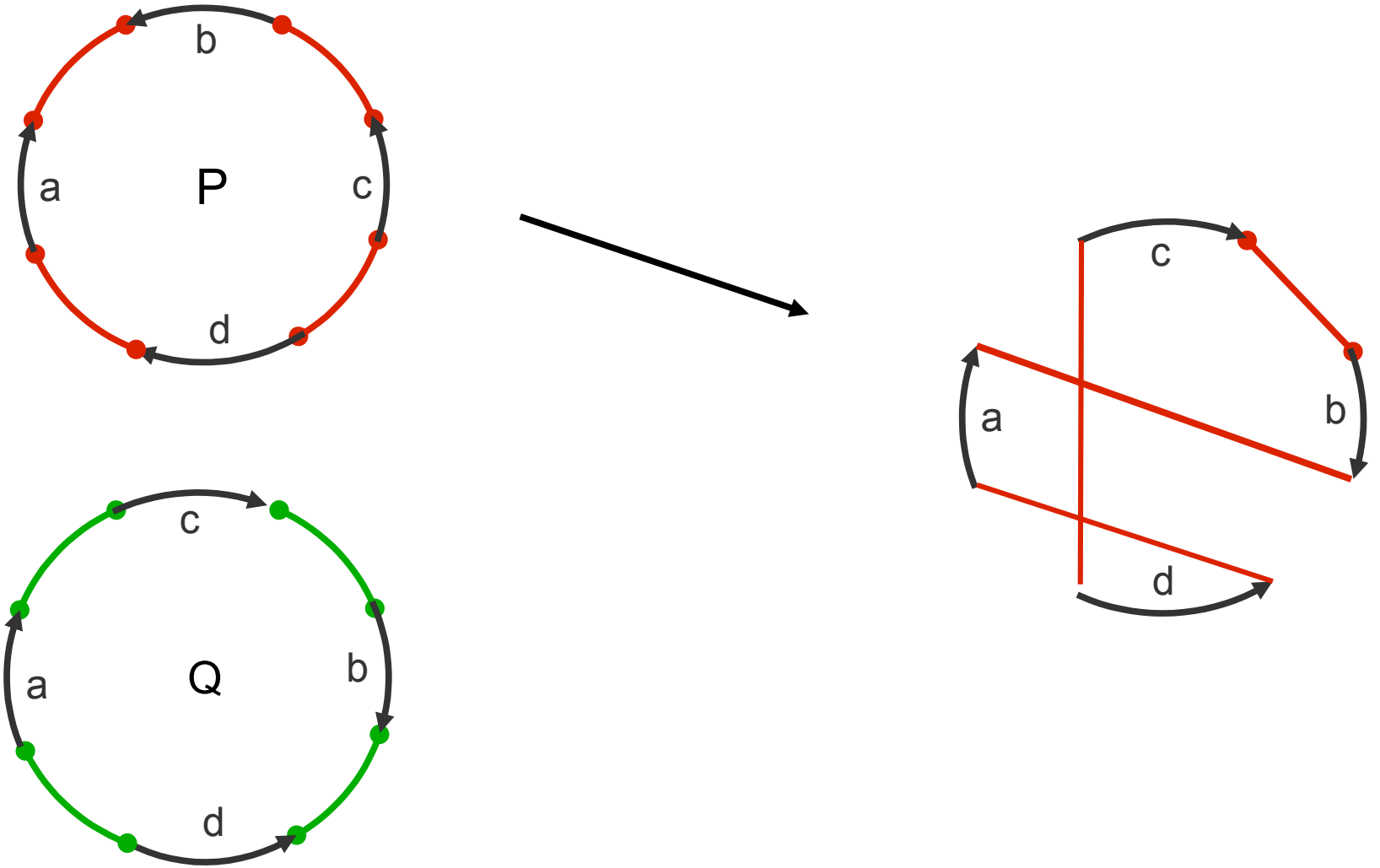




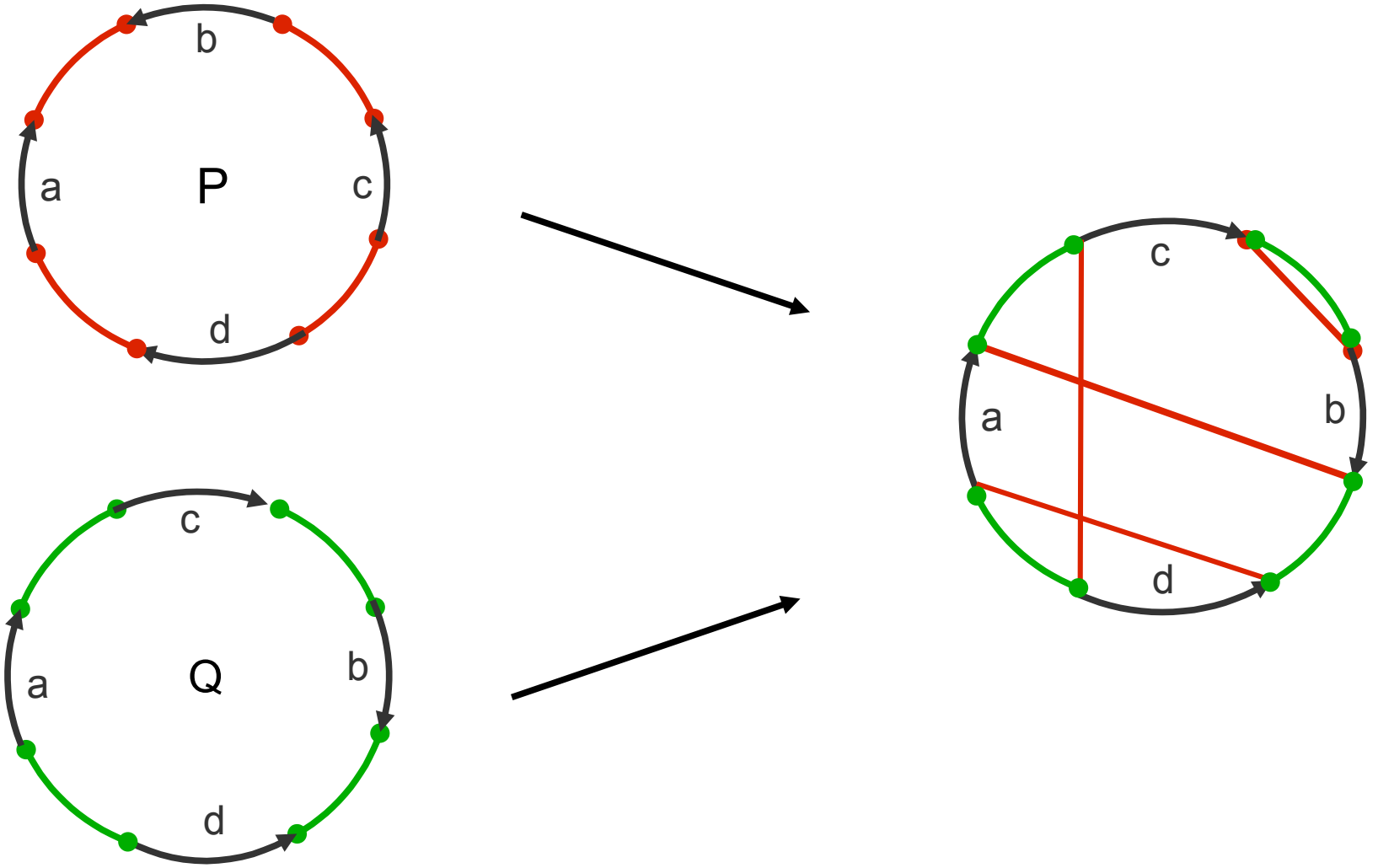




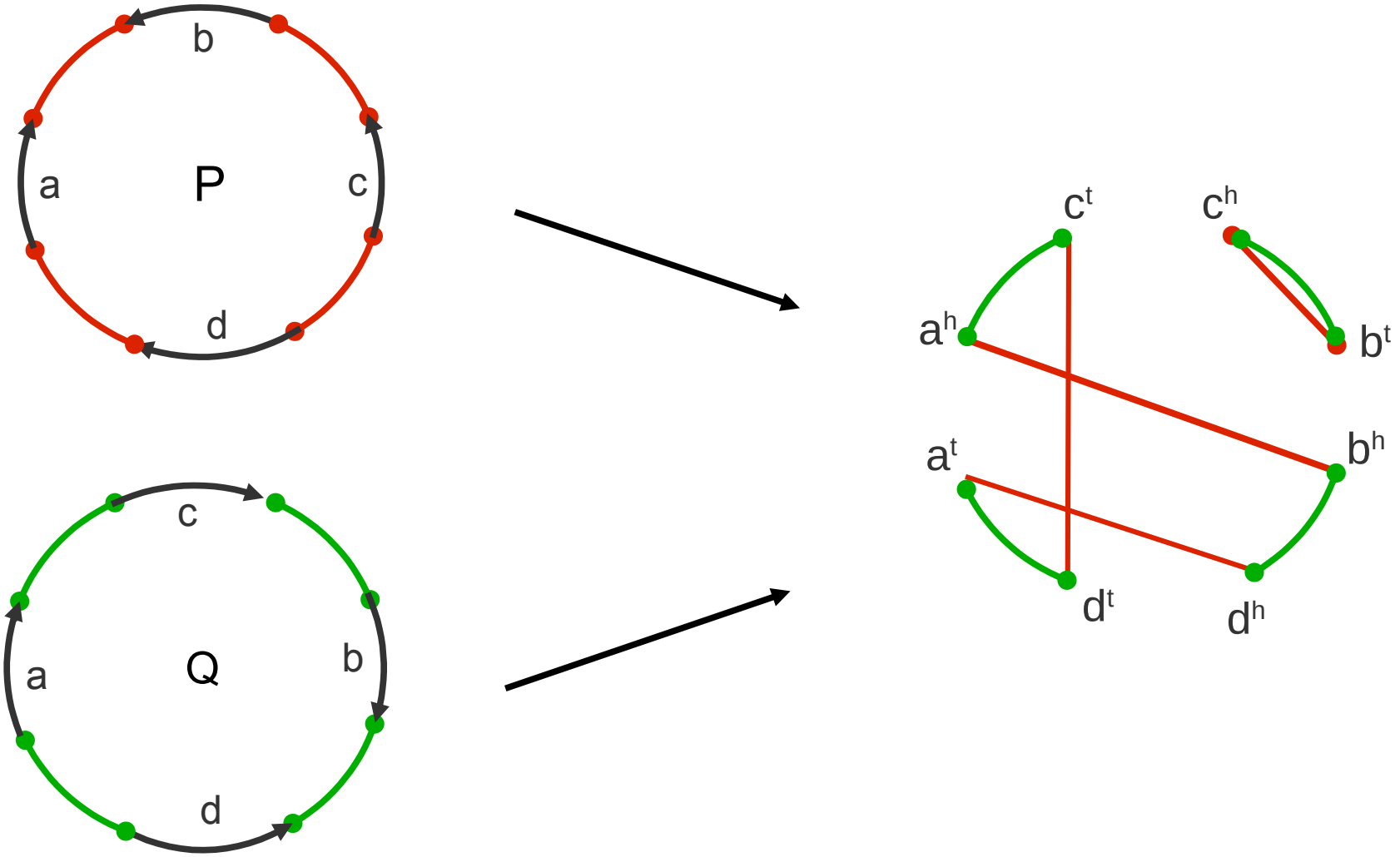
Граф брейкпоинтов – это просто!



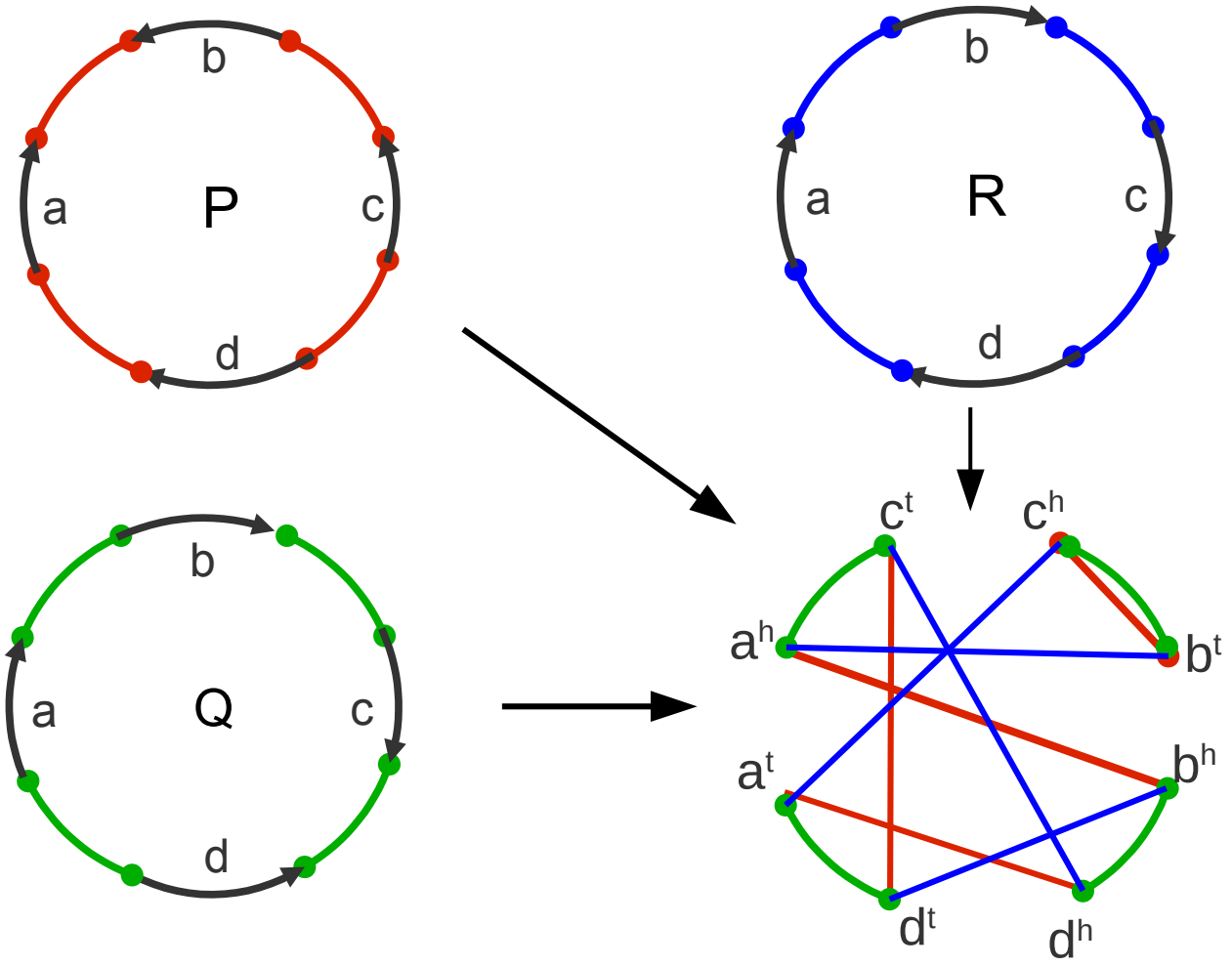
Граф брейкпоинтов – это просто!



Граф брейкпоинтов – это просто!

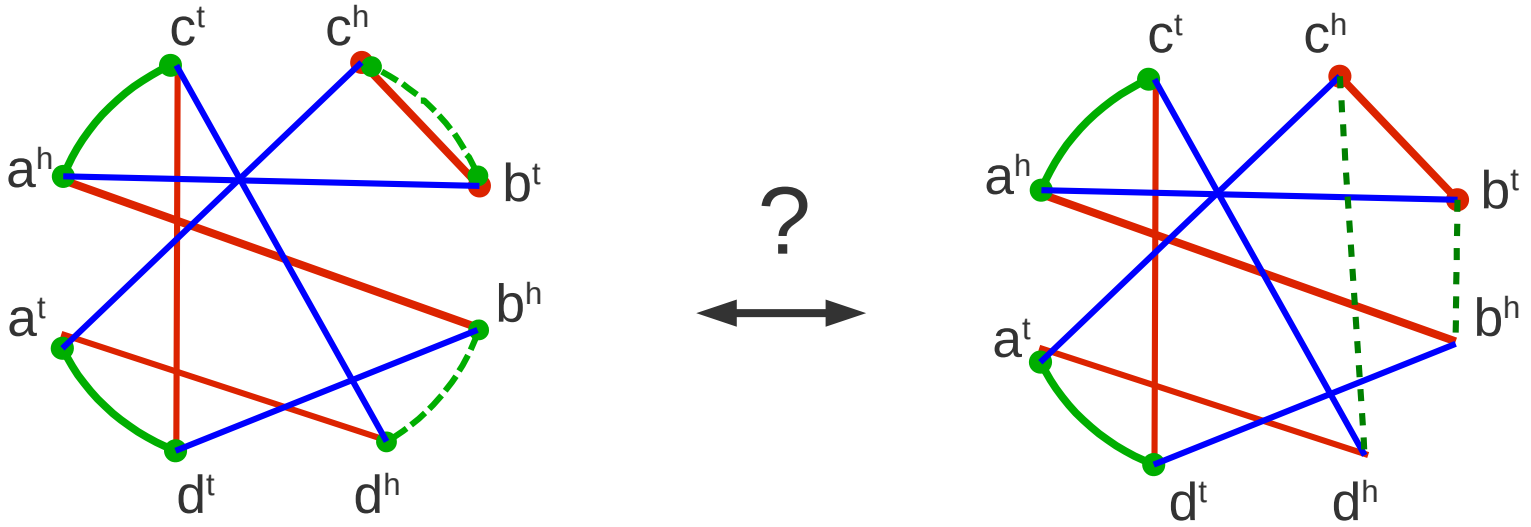


Граф для нескольких геномов



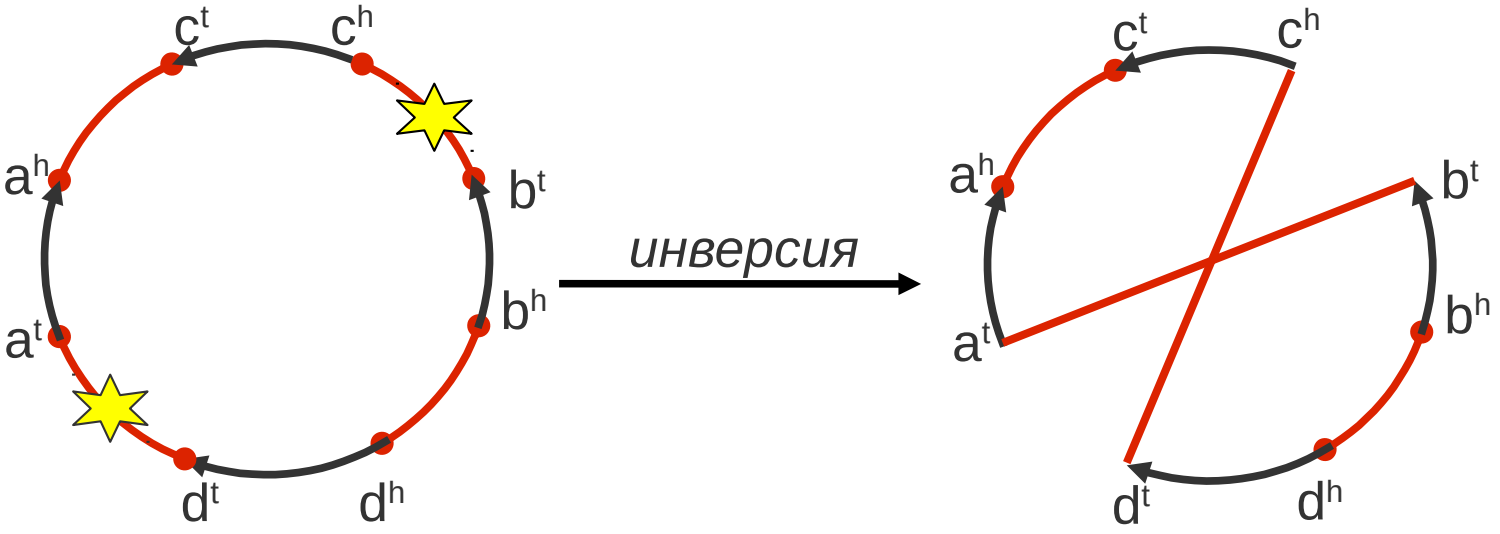
- Ребра каждого из цветов образуют совершенное паросочетание

Контиги: невидимые смежности



- Из-за фрагментации на контиги некоторые смежности скрыты
- Будем их искать как ребра совершенного паросочетания
- Проблема: совершенных паросочетаний несколько

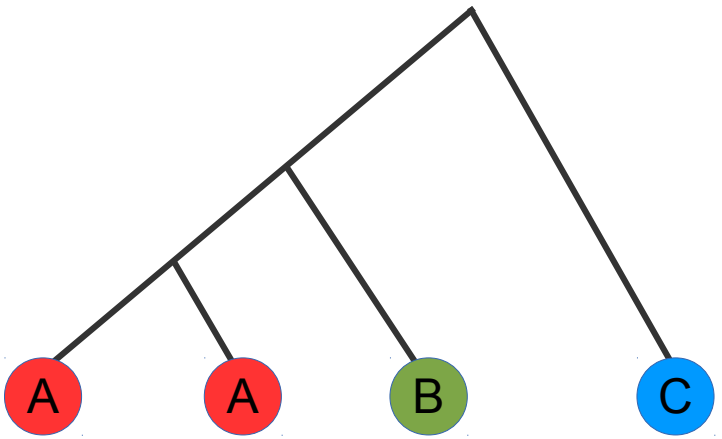
Состояния смежностей



$State(a^t)=d^t$ $State(a^t)=b^t$

- Перестройки меняют *состояния* смежностей
- Будем оценивать *стоимость* каждой перестройки

Процедура парсимонии



Ничто в биологии не имеет смысла,
кроме как в свете эволюции
Ф. Добржанский

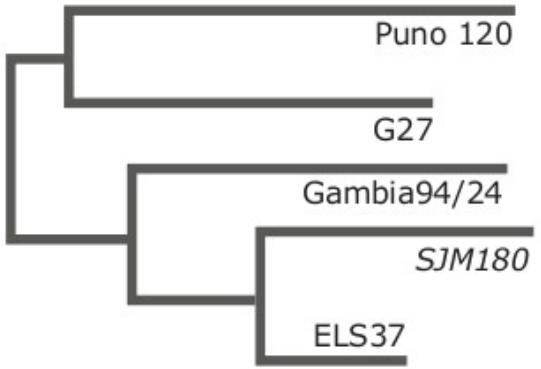
- Каждое мультиребро соответсвует некоторой конфигурации смежностей
- Оценим “эволюционную стоимость” такой конфигурации
- Вес мультиребра равен полученной стоимости

Результаты: один референс *E.coli*

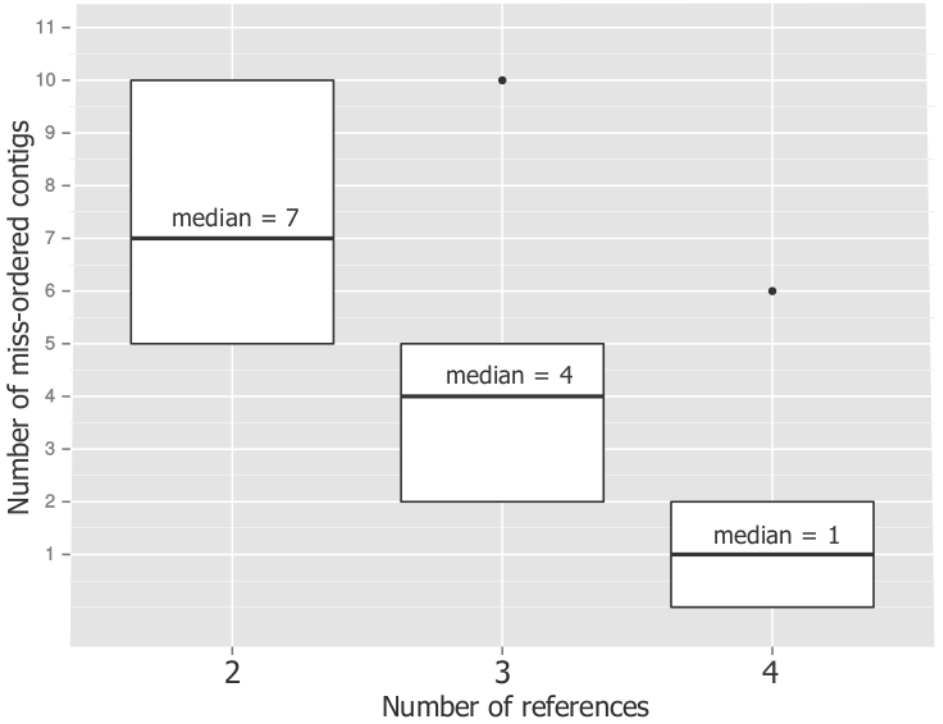
	Ragout	MCM	OSLay
Скэффолды	1	1	8
Покрытие	97.7	97.6	96.7
Контиги	129	77	80
Пропуски	52	73	61
Ошибки	0	0	1

Результаты: четыре референса *H.Pylori*

Референсы	Скэффолды	Покрытие	Контиги	Пропуски	Ошибки
Ragout					
G27, ELS37	2	97.8	95	22	1
G27, Puno120, ELS37	1	97.8	95	21	1
G27, Puno120, ELS37, Gambia94/27	1	97.6	93	22	0
RACA					
G27, ELS37	3	83.6	35	29	2
G27, Puno120, ELS37	2	83.6	35	30	2
G27, Puno120, ELS37, Gambia94/27	2	83.8	35	31	1



Результаты: симуляция перестроек



- Четыре референса с большим количеством перестроек

Выводы

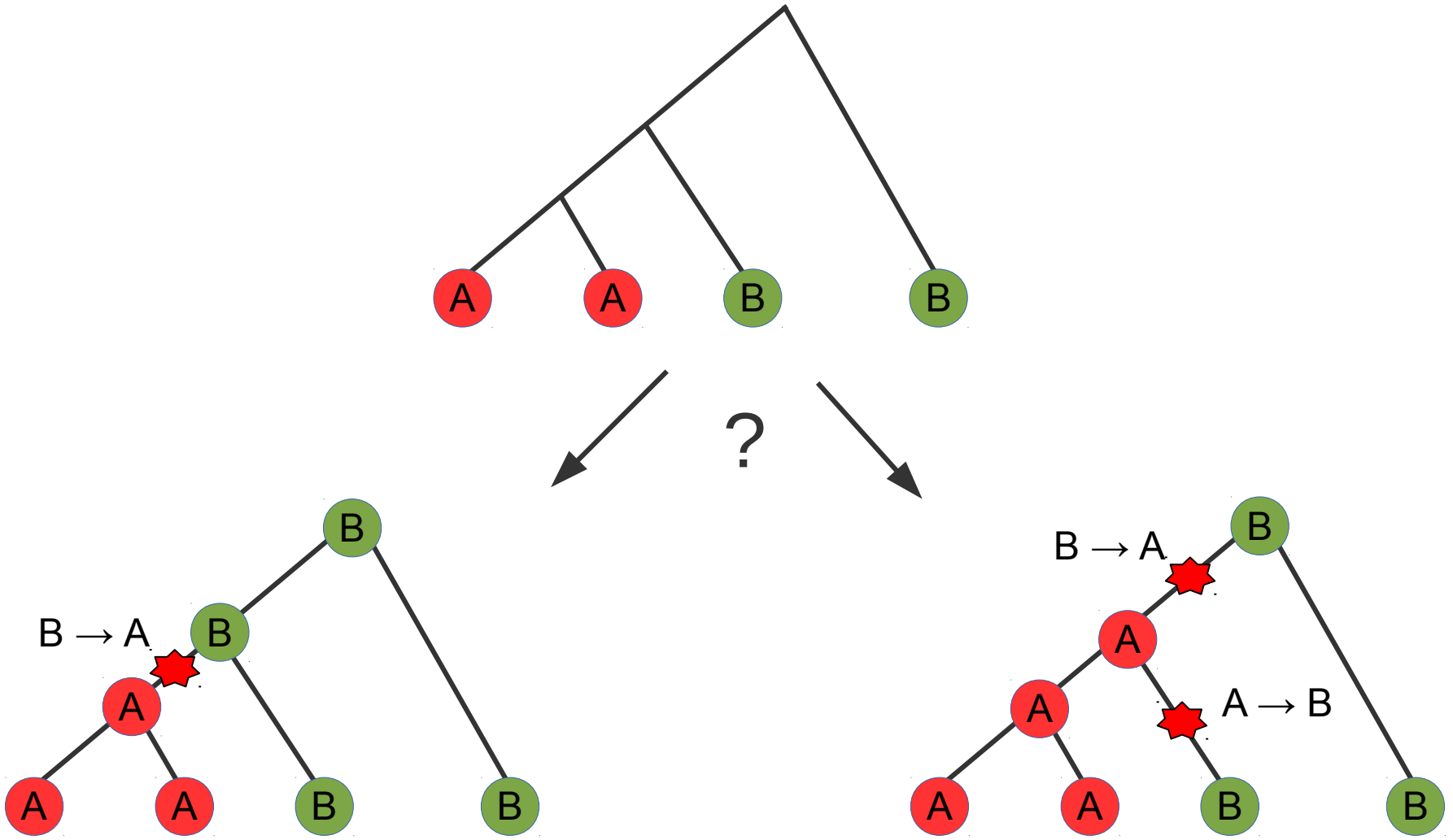
- Разработан алгоритм для референсной сборки, включающий в себя анализ геномных перестроек
- Алгоритм превосходит существующие решения по корректности/качеству сборки
- Алгоритм реализован в виде программы Ragout, написанной на Python/C++
- Принята статья на конференцию ISMB 2014:

  Mikhail Kolmogorov, Brian Raney, Benedict Paten, and Son Pham. *Ragout: a reference-assisted assembly tool for bacterial genomes.*

Благодарности

- Son Pham
- Алла Лапидус
- Павел Авдеев
- Дмитрий Мелешко
- Тамара Панеш
- Николай Вяххи

Вес смежностей: процедура парсимонии

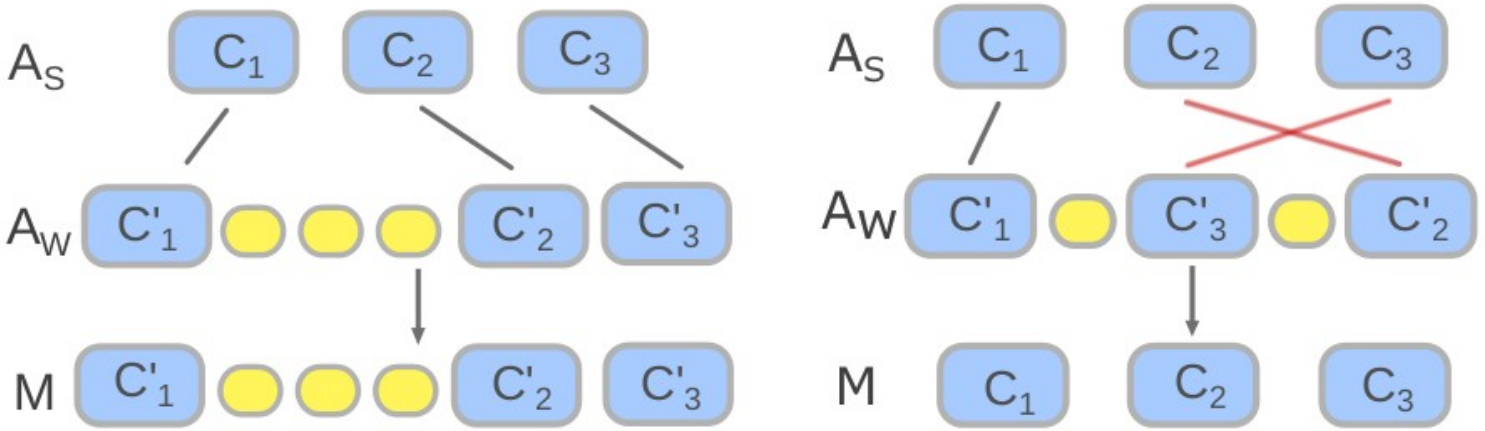


1 изменение

2 изменения

Итеративная сборка

- Вопрос: какой размер синтенного блока выбрать?
- Большие – более надежны. Но некоторые контиги слишком короткие, чтобы их содержать
- Сначала мы строим “скелет” из больших блоков, а затем заполняем его более малыми, если нет противоречий



(a) *Локально согласованные* (b) *Локально несогласованные*

Уточнение с помощью графа перекрытий

- Маленькие/повторные контиги трудно разбить на синтенные блоки
- Можно использовать информацию из графа перекрытий для их упорядочивания
- Аналогично разрешению повторов с помощью парных ридов