Федеральное государственное бюджетное учреждение науки
Институт проблем передачи информации им. А.А. Харкевича
Российской академии наук

Гершгорин Р.А.

На правах рукописи

Гершгорин Роман Александрович

КРАТЧАЙШЕЕ ПРЕОБРАЗОВАНИЕ И РЕКОНСТРУКЦИЯ ХРОМОСОМНЫХ СТРУКТУР

03.01.09 – Математическая биология, биоинформатика

Диссертация на соискание ученой степени
кандидата физико-математических наук

Научный руководитель:
д.ф.-м.н. профессор В.А. Любецкий

Москва – 2018

ВВЕДЕНИЕ

1. Постановки задач и полученные результаты

Актуальность темы

Большое число полно секвенированных хромосом, включая геномы митохондрий и пластид, открыло ещё один путь для построения эволюционных сценариев; кроме традиционного пути, основанного на расхождении гомологичных генов. А именно, появилась возможность строить модель эволюции на основе взаиморасположения генов, иными словами: макроструктуры генома, на расхождении макроструктур. Исследование макроструктуры или, как ещё говорят, геномной или *хромосомной структуры* началось около 1992 года, и к настоящему времени известно много результатов, как по эволюции хромосомных структур отдельных видов, так и по алгоритмам и компьютерным программам, которые работают с такими структурами. Однако эти алгоритмы не позволяют работать с хромосомными структурами общего вида и назначать желаемые цены операций. Модели эволюции рассматривают хромосомные структуры ядерных геномов и геномов органелл, как и виды в целом. Сравнение эволюционных деревьев, получаемых на основе разных моделей эволюции, приводит к важным биологическим выводам. Это – модели, основанные на гомологичных белках (суперматрице выравниваний), рРНК, ультраконсервативных и высококонсервативных элементах, хромосомных структурах, расстояниях между генами и т.д.

Алгоритм называется *точным* (в противоположность – *эвристическому*), если он сопровождается доказательством того, что для любых данных на его входе выдаётся глобальный минимум функционала из соответствующей задачи. Точные алгоритмы имеют очевидное преимущество перед эвристическими; доказательство точности всегда требует условий на данные, поэтому «хороший эвристический» алгоритм – точный алгоритм, успешно применяемый и вне области выполнения таких условий; конечно, условия должны быть достаточно широкими. Алгоритм может точно или эвристически решать исходную задачу и тогда называется *прямым*. Однако алгоритм может состоять в сведении одной задачи к другой и тогда называется *алгоритмом сведения*, в этом случае утверждение о точности относится к алгоритму сведения, а не к решению задачи, которая получается в результате сведения. Представляют интерес *алгоритмы сведения* к каноническим задачам, которые уже прошли широкую апробацию и для которых разработаны широко применяемые пакеты компьютерных решений, а иногда и

специализированные вычислительные устройства. Среди таких канонических задач находится линейное программирование и его варианты, включая целочисленное и булево. Важно, чтобы задача минимизации, к которой сводится исходная, имела квадратичный или близкий к нему размер (от размера исходной задачи), а алгоритм сведения имел столь же низкую сложность вычисления. Например, задача о графах с n вершинами и рёбрами сводилась бы к линейной задаче минимизации с n^2 переменными, равенствами и неравенствами.

Хромосомная структура (часто говорят: *структура*) по сравнению со многими предшествующими исследованиями понимается в нашей работе наиболее общим образом. А именно, как граф, состоящий из любого множества непересекающихся цепей и циклов, в которых каждому ребру приписано направление и имя. Такой граф соответствует множеству линейных и кольцевых хромосом, если пренебречь межгенными участками, длиной и составом генов. Тогда ген с его направлением транскрипции и именем изображается ребром указанного графа. В этом смысле ребро вместе с его именем называют далее *геном*, а цепь или цикл – *хромосомой* (или на математическом языке – *компонентой*). *Край* гена понимается как позиция хромосомы, в которой начинается или заканчивается его транскрипция; численное значение этой позиции также не учитывается в хромосомной структуре. Термины *конец* и *начало* (направленного ребра) используются в контексте работы с графами. В хромосомной структуре края соседних генов *отождествляются* (или, как говорят, *склеиваются*) в вершине графа.

Долгое время основной задачей в этой области было вычисление расстояния между двумя структурами a и b, точнее, нахождение *кратчайшей* последовательности операций из их фиксированного списка, которые преобразуют a в b, и цены кратчайшей последовательности, которую будем называть *кратчайшим расстоянием* между a и b. «Кратчайшая» означает здесь: «с минимальной суммарной ценой операций в последовательности с точностью до некоторой фиксированной аддитивной константы». Точнее, пусть $c(a,b)$ – минимум функционала кратчайшего расстояния, $T(a,b)$ – суммарная цена последовательности, которую выдаёт наш алгоритм; тогда должно выполняться: $|c(a,b)-T(a,b)| \le k$, где константа k не зависит от a и b, а зависит только от цен операций. Заметим, что гораздо чаще предлагаются алгоритмы «с точностью до некоторой фиксированной мультипликативной константы k», что, конечно, является

гораздо более слабым утверждением; оно означает: $\frac{T(a,b)}{c(a,b)} \le k$. В случае равных цен всех операций кратчайшее расстояние иногда называют (обычным) *расстоянием*.

Искомая минимальная цена (=кратчайшее расстояние) не является обычной метрикой. Нахождение кратчайшего расстояния и кратчайшей последовательности называется **задачей о *кратчайшем преобразовании* a в b**. Эта задача рассматривается в Главе 1 (публикация [39]), где предполагается, что имена в структуре не повторяются (биологически это значит, что *отсутствуют паралоги*). В качестве операций фиксированы *стандартные* – двойная, полуторная и одинарная переклейки, и *дополнительные* – удаление и вставка связного участка рёбер (генов).

В предшествующих результатах, относящихся к задаче преобразования, обычно использовались два очень существенных ограничения: множества имён в a и b совпадают («равный генный состав») и цены всех операций равны, т.е. минимизируется всего лишь число операций в кратчайшей последовательности или, иными словами, использовалось обычное расстояние. Часто использовались и более сильные ограничения. В диссертации (раздел 4 Введения) приведён обзор публикаций, тесно связанных с нашими результатами. В диссертации **нигде не предполагается равный генный состав, а цены при отсутствии паралогов могут быть разными**; это принципиально усложняет задачу.

При произвольных ценах поставленная задача NP-трудная, т.е. заведомо не поддаётся обоснованному решению; поэтому её можно решать только при тех или иных условиях на цены операций; впервые в работах диссертанта или его руководителей получены условия, при которых задача решается точным алгоритмом [39,42,76,77].

Решение задачи преобразования ищется в виде *линейного* по сложности алгоритма, т.е. по времени работы и по используемой памяти. Для характерных данных современной биоинформатики, такое или близкое к нему требование (квадратичности или, в крайнем случае, кубичности алгоритма) кажется обязательным. В диссертации предлагаются как *прямые* алгоритмы, так и их альтернатива – алгоритмы *сведения*. Алгоритмы сведения имеют у нас сложность, которая не превосходит размера, т.е. числа переменных и ограничений в соответствующей линейной задаче, и, таким образом, сведения не сложнее, чем сложность выписывания этих переменных и ограничений. *Квадратичного* размера называется целочисленное линейное программирование (ЦЛП) с квадратичным от размера исходных данных числом переменных и ограничений, например, от суммарного числа рёбер в исходных графах. В аналогичном смысле

употребляется термин *линейное* и *кубическое* ЦЛП; прилагательное везде указывает на размер ЦЛП относительно размера исходной задачи. Аналогичные термины используются для булева линейного программирования (БЛП).

Вторая решаемая в диссертации задача – ***задача реконструкции*** структур вдоль дерева, которая, насколько диссертанту известно, ранее не рассматривалась в присутствии паралогов. Она состоит в следующем. Дано корневое (не обязательно бинарное) дерево, каждому его листу приписана структура, в которой имена могут повторяться (тем самым, *допускаются паралоги*). Найти расстановку структур (без паралогов или допуская их) по всем внутренним вершинам дерева, для которой расстояние между структурами на концах любого ребра (суммированное по всем рёбрам и называемое *суммарным расстоянием*) минимально. Расстояние между структурами a и b, приписанными концам ребра, может определяться как число пар различных краев генов, которые в одной структуре склеены, а в другой – не склеены или отсутствуют, сложенное с числом генов, присутствующих в одной структуре и отсутствующих в другой. Это расстояние может вычисляться также с учётом цен: за каждую пару краёв склеенных в одной структуре и расклеенных в другой начисляется фиксированная цена; аналогично за каждый ген, присутствующий в одной структуре и отсутствующий в другой, начисляется своя фиксированная цена. И тогда минимизируется сумма таких цен по всем событиям на ребре (a,b) и по всем рёбрам. Для ребра (a,b) начало (ближе к корню) a и приписанная ему структура a, как и конец ребра (дальше от корня) b и приписанная ему структура b, обозначаются одинаково. Такое расстояние называется *специальным* (или: *брейкпоинтовым*). Различаются случаи: без цен (т.е. равные цены) и с ценами, которые должны быть заданы. Для случая равных генных составов и без цен такое расстояние предложено в работах [1,2]. Задача реконструкции со специальным расстоянием рассматривается и для случая, когда цены от a к b и от b к a могут различаться. В главе 2 задача реконструкции рассматривается только вместе со специальным расстоянием, а в Главе 3 – вместе с кратчайшим расстоянием и равными ценами.

Допущение паралогов существенно усложняет задачу реконструкции: проблема в том, что специальное и кратчайшее расстояния требуют решения трудного и самого по себе биологически важного вопроса, что значит «тот же самый ген» в a и b. Например, в a имеются два гена с именем n, а в b – три гена с тем же именем; заранее неясно, какой из паралогов в a соответствует, какому из паралогов в b. Поэтому необходимо следующее уточнение в постановке задачи. Множество рёбер с именем n в a и b

обозначим соответственно $X(n,a)$ и $X(n,b)$. Нужно для каждого n найти частично определённое инъективное отображение меньшего по числу элементов из множеств $X(n,a)$ и $X(n,b)$ в большее («*соответствие* паралогов»), для которого графы a' и b' уже с уникальными именами рёбер имеют минимальное суммарное расстояние. Точнее, в a' рёбра, одноимённые в a, получают уникальные имена (и аналогично для b' и b), так чтобы уникальные имена сохранялись при этих отображениях, которые различны для разных рёбер. Уникальные имена можно получить, например, просто добавляя вторую позицию к исходным именам, т.е. уникальное имя будет иметь вид $n.k$ одновременно в a и b, если эти $n.k$ *соответствуют* друг другу при отображении, соответствующем ребру (a,b). Иными словами, отображения сохраняют уникальные имена. Гены, которые не имеют паралогов, могут получить на второй позиции значение 0, которое не используется при нумерации паралогов.

На той же основе, что и две указанные задачи, в диссертации рассматривается задача согласования двух произвольных множеств цепей (двух «линейных» структур). При секвенировании возникает ситуация: для генома найдены контиги, составленные каждый из нескольких генов, которые имеют направления транскрипции. В нашей терминологии *контиг* – цепь, в которой гены имеют направления и имена, может быть, повторяющиеся («паралоги»). Контиги из данного множества соединяют в цепь или цикл; эти варианты, по сути, эквивалентны, и мы рассмотрим второй из них, как это сделано в работе [3]. Итак, пусть даны два множества a и b цепей (множества «контигов»). Множество a (как и b) и соответствующий a (как и b) цикл – хромосомные структуры. Нужно соединить контиги из a в цикл (и аналогично – контиги из b в цикл), так чтобы расстояние между этими циклами было минимальным с учётом выбора соответствия паралогов в циклах. Не ограничивая общность, считаем: каждый контиг оканчивается концом своего гена. Эту задачу назовём ***согласованием контигов***. Её биологическое содержание подробно обсуждается в работе [3].

Развитые нами алгоритмы реализованы соответствующими компьютерными программами, тестированы на искусственных данных и затем применены для построения филогенетических деревьев хромосомных структур *митохондрий* инфузорий и споровиков видов класса Aconoidasida, а также – *пластид* родофитной ветви и бактерий рода *Rhizobium*. Напомним: пластиды – полуавтономные органеллы, происходящие от цианобактерий; в родофитной ветви они представлены у красных водорослей (Rhodophyta) и у видов с пластидами вторичного и третичного

происхождения от пластид Rhodophyta. Среди них находятся фотосинтезирующие и нефотосинтезирующие виды.

Итак, нами рассмотрена *тема* – разработка эффективных алгоритмов для работы с хромосомными структурами и применение алгоритмов и соответствующих компьютерных программ для построения филогенетических деревьев хромосомных структур. Тема представляется актуальной, за последние 30 лет по ней опубликовано сотни статей и появляются всё новые.

Цели работы

Найти линейные или близкие к ним по сложности (не выше кубических) точные алгоритмы решения задач преобразования и реконструкции хромосомных структур, согласования множеств контигов; и реализовать их соответствующими компьютерными программами. Применить полученные алгоритмы и программы для построения филогенетических деревьев митохондрий инфузорий и споровиков видов класса Aconoidasida, пластид родофитной ветви и бактерий рода *Rhizobium*.

Методы исследования

В работе использованы методы теории алгоритмов и организации вычислений с использованием известных и оригинальных программ, в том числе для параллельных вычислений на суперкомпьютерах, методы математической биологии и биоинформатики. Оригинальный подход, предложенный автором, состоял также в сведении наиболее сложных случаев рассмотренных задач эффективными алгоритмами к целочисленному или булеву линейному программированию с линейным, квадратичным или (самое большое) кубическим числом переменных и ограничений.

Научная новизна

Полученные алгоритмы и компьютерные программы, как и их применения для построения филогенетических деревьев хромосомных структур митохондрий, пластид и бактерий, являются новыми.

Более подробно. Получено эффективное решение задачи преобразования структур общего вида *без паралогов*; кроме того, оно является точным, если цена вставки находится в интервале от c – равных цен всех других операций до $2c$. Соответствующий алгоритм линейный по сложности.

Получено эффективное решение задачи реконструкции структур общего вида *без паралогов* для специального расстояния, равных и неравных цен. Соответствующий

алгоритм является прямым, точным и квадратичным по сложности. Доказывается его точность. Для этой задачи ранее предлагались лишь эвристические алгоритмы.

Получено эффективное решение задачи преобразования произвольных структур *с паралогами* и равными ценами сведением её квадратичным точным алгоритмом к задаче целочисленного линейного программирования (ЦЛП) квадратичного размера. В общей постановке эта задача не рассматривалась. Для циклических хромосомных структур *с паралогами* и равными ценами задача преобразования решена сведением её линейным точным алгоритмом к задаче ЦЛП линейного размера.

Получено эффективное решение задачи реконструкции *с паралогами* и любыми ценами для специального расстояния сведением ее квадратичным точным алгоритмом к задаче булевого линейного программирования квадратичного размера. Получено эффективное решение задачи реконструкции *с паралогами* и равными ценами для произвольных структур общего вида и кратчайшего расстояния сведением её кубическим точным алгоритмом к задаче ЦЛП кубического размера. Ранее задача реконструкции для структур с паралогами не рассматривалась. Везде выше допускается неравный генный состав.

Получено эффективное решение задачи согласования множеств контигов с неравным генным составом, паралогами и равными ценами сведением линейным точным алгоритмом к ЦЛП линейного размера. Ранее эта задача рассматривалась лишь для случая равного состава и отсутствия паралогов.

Создан комплекс программ для решения задач преобразования и реконструкции, который позволяет эффективно решать задачи для хромосомных структур, содержащих тысячи генов. Это достигается за счёт использования современных методов распараллеливания программ и технологий хранения и обработки больших данных.

На основе оригинальных алгоритмических и программных решений построены разумные филогенетические деревья хромосомных структур митохондрий инфузорий и споровиков видов класса Aconoidasida, пластид родофитной ветви и бактерий рода *Rhizobium*. На их основе выявлены особенности эволюции органелл.

Практическая значимость работы

Работа носит теоретический характер. В то же время, исследование может иметь прикладное значение. Реконструкция хромосомных структур может применяться для анализа хромосомных перестроек, что, в частности, важно при многих заболеваниях: хромосомные перестройки меняют уровни экспрессии генов, что служит одной из

причин заболевания. Рассмотренные методы могут применяться при сборке секвенируемых геномов.

Апробация работы

Компьютерные программы тестировались на искусственных примерах с известными ответами; рассмотрено около 100 таких примеров. Программы снабжены удобным для пользователя интерфейсом. Результаты работы опубликованы в 5 статьях и 3 тезисах и докладывались на следующих конференциях:

- 39-я конференция «Информационные технологии и системы»: ИТиС'15 (Сочи, 7–11 сентября 2015);
- Международная конференция “Moscow Conference on Computational Molecular Biology”: MCCMB'17 (Москва, 27–30 июля 2017);
- 57-я научная конференция МФТИ (Москва, 23-28 ноября 2015).

Работа также докладывалась на научных семинарах механико-математического факультета Московского государственного университета им. М.В. Ломоносова и на семинаре по Математической биологии и биоинформатике Института проблем передачи информации им. А.А. Харкевича РАН.

Публикации

По теме диссертации опубликовано 5 статей и 3 тезисов докладов на конференциях. Все результаты, включённые в диссертацию, получены лично автором.

Структура и объём работы

Работа состоит из введения, четырёх глав и списка литературы. Список литературы содержит 77 наименований. Объём работы составляет 127 страниц, включая 14 таблиц и 75 рисунков.

Основные положения, выносимые на защиту

1) Получен алгоритм решения задачи преобразования структур для случая: цена вставки d находится в интервале от c – равных цен всех других операций до $2c$. Доказательство точности алгоритма не выносится на защиту. Глава 1, [39].

2) Для специального расстояния, отсутствия паралогов, равных и неравных цен получен прямой точный квадратичный алгоритм решения задачи реконструкции; доказана его точность. В случае того же расстояния, присутствия паралогов и любых

цен получен точный алгоритм сведения задачи реконструкции к задаче квадратичного булева линейного программирования. Получены доказательства точности алгоритмов. Глава 2, [42].

3) Получен алгоритм решения задачи преобразования с равными ценами: циклических хромосомных структур (сведением к ЦЛП линейного размера) и произвольных структур (сведением к ЦЛП квадратичного размера). Доказательства точности алгоритмов в пунктах 3–5 не выносятся на защиту. Глава 3, [69].

4) Получен алгоритм решения задачи реконструкции с равными ценами произвольных структур сведением к ЦЛП кубического размера. Глава 3, [69].

5) Получен алгоритм решения задачи согласования с равными ценами двух множеств контигов сведением к ЦЛП линейного размера. Глава 3, [69].

6) На основе алгоритмических и программных решений, предложенных в пунктах 1–5, построены разумные филогенетические деревья хромосомных структур митохондрий инфузорий и споровиков видов класса Aconoidasida, а также – пластид родофитной ветви у водорослей и споровиков и бактерий рода *Rhizobium*. На их основе обсуждаются особенности эволюции этих органелл и видов. Глава 4, [39, 42, 69, 70].

Все результаты диссертации опубликованы.

2. Содержание работы

Во **Введении** приведены постановки задач и формулировки полученных результатов в целом, приведён обзор наиболее близких к нашим результатам публикаций, а затем приведен обзор результатов, наиболее близких к полученным.

В Главе 1, [39] рассматривается задача о преобразования двух данных общего вида структур a и b, первой ко второй, операциями из фиксированного списка. В этой главе предполагается, что имена рёбер в каждой отдельно взятой структуре не повторяются, т.е. *паралоги отсутствуют*. Упомянутые операции над хромосомной структурой следующие, первые четыре из них называются *стандартными*, последние две – *дополнительными*.

1) *Двойная переклейка*: расклейка двух пар склеенных краёв генов и переклейка полученных четырёх краёв по-новому;

2) *Полуторная переклейка*: расклейка пары склеенных краёв генов и склейка одного из полученных краёв с каким-нибудь несклеенным («свободным») краем;

3) *Разрез*: расклейка пары склеенных краёв генов (образуются два свободных края);

4) *Склейка*: склейка пары свободных краёв генов.

5) *Удаление* связного участка *a*-генов (т.е. присутствующих в *a* и отсутствующих в *b*).

6) *Вставка* связного участка *b*-генов (т.е. присутствующих в *b* и отсутствующих в *a*).

В **разделе 1.1** приводится ключевое определение *общего графа* $a+b$ (которое обобщает определение из работы [15]), а задача преобразования переформулируется (теорема 1) как задача приведения неориентированного графа $a+b$ к виду $c+c$ для некоторой структуры c, эта переформулировка обобщает подход, также предложенный в указанной работе. Граф такого вида называется *финальным*.

В **разделе 1.2** описывается оригинальный линейный по сложности алгоритм решения задачи преобразования, точность которого доказана, в частности, для случая: цена вставки d находится в интервале от c – равных цен всех других операций до $2c$. Аддитивная константа k из формулировки задачи равна $k=d–c$ (что при $c=1$ не превышает 1). Из алгоритма сразу следует новый точный линейный алгоритм решения задачи преобразования для случая равных цен операций. Доказательство точности не выносится на защиту.

В **разделе 1.3** приведено тестирование этого алгоритма на искусственных примерах.

В Главе 2, [42, 62, 70] решается задача реконструкции структур вдоль данного дерева. В **разделе 2.1**, [72] приводится постановка задачи с *паралогами* и, возможно, с ценами для *специального* и *кратчайшего* расстояний между структурами на концах рёбер дерева.

В **разделе 2.2**, [42] в случае специального расстояния и отсутствия паралогов, равных и неравных цен операций, приводится прямой точный квадратичный по времени работы и используемой памяти алгоритм решения задачи реконструкции. Доказывается его точность (теорема 2 для равных цен и теорема 3 для неравных).

В **разделе 2.3**, [72] в случае специального расстояния и присутствия паралогов, любых цен, приводится точный квадратичный по времени работы и используемой памяти *алгоритм сведения* этой задачи к квадратичному булевому линейному программированию. Точнее, число переменных и ограничений зависит биквадратично (т.е. в 4й степени) от суммарного числа паралогов в листьях, а при фиксированном числе паралогов – квадратично от суммарного размера исходных графов в листьях, например, от суммарного числа рёбер в них.

В [71] содержится оригинальный подход к решению задачи ЦЛП, который не включён в диссертацию, вместо него применялся стандартный пакет ЦЛП IBM CPLEX.

В **разделе 2.4** приведено тестирование алгоритмов из этой главы на искусственных примерах.

В Главе 3, [39, 69] рассматриваются четыре задачи, тесно связанные между собой и с задачами из Глав 1–2 как по методу, так и по постановке. Здесь всюду допускаются неравный генный состав и присутствие паралогов, но предполагаются равные цены. Структура называется *циклической*, если она состоит из одних циклов (кольцевых хромосом).

В **разделе 3.1**, [39] содержится точный алгоритм решения задачи преобразования для циклических структур; это – линейный алгоритм, который сводит её к линейной ЦЛП. Точнее, число переменных и ограничений квадратично зависит от суммарного числа паралогов в листьях, а при фиксированном числе паралогов – линейно от суммарного размера исходных графов в листьях.

В **разделе 3.2**, [69] содержится точный алгоритм решения задачи преобразования для произвольных структур; это – квадратичный алгоритм, который сводит её к квадратичному ЦЛП. В **разделе 3.3**, [69] предложен точный (при некотором условии) алгоритм решения задачи реконструкции для произвольных структур; это – кубический алгоритм, который сводит её к кубическому ЦЛП. Для разделов 3.1–3.4 доказательства точности алгоритмов не выносятся на защиту.

Наконец, в **разделе 3.4**, [69] предложен точный линейный алгоритм решения задачи согласования двух произвольных множеств цепей (двух «линейных» структур), который сводит её к линейному ЦЛП. Точнее, число переменных и ограничений квадратично зависит от суммарного числа паралогов в данных структурах и от суммарного числа цепей; если эти значения фиксированы – линейно от суммарного размера структур. В работе [3] предложено точное решение этой задачи при *условии*: равный генный состав двух множеств контигов и отсутствие паралогов в них. В разделе 3.4 предлагается решение задачи *без этого условия*, основанное на её сведении к ЦЛП, сложность её решения определяется сложностью задачи ЦЛП. Эта сложность даже теоретически не может быть улучшена, так как за счёт присутствия паралогов задача становится NP-трудной. В работе [3] решение основано на алгебраической теории перестановок и использует расстояние, которое отличается от расстояния, определённого выше.

В **разделе 3.5** приведено тестирование алгоритмов из этой главы на искусственных примерах.

В **Главе 4**, [39, 69, 70] алгоритмы, развитые в Главах 1 и 3, и соответствующие компьютерные программы применяются для построения филогенетических деревьев хромосомных структур митохондрий инфузорий (Ciliophora) и споровиков видов класса Aconoidasida, пластид родофитной ветви, а также бактерий рода *Rhizobium*. Используются результаты 1й главы без паралогов с разными ценами и результаты 3й главы с паралогами и равными ценами. Для полученных деревьев обсуждаются особенности эволюции соответствующих митохондрий и пластид; в целом полученная реконструкция не расходится с принятыми представлениями об их эволюции. Отметим следующие особенности полученных деревьев.

1) Отличия между деревьями, построенными по белкам, кодируемым в митохондриях Ciliophora, и по хромосомным структурам незначительны и состоят в различном взаимном расположении видов рода *Tetrahymena*; в каждом из этих двух деревьев род *Tetrahymena* образует кладу, [70].

2) Отличие дерева хромосомных структур митохондрий споровиков от общепринятого дерева видов состоит в перемешивании видов близких родов *Leucocytozoon* и *Plasmodium* между собой; вместе виды этих родов образуют кладу, [39]. В этом поддереве виды с линейными и кольцевыми хромосомами собрались в две соответствующие клады.

3) Дерево пластид родофитной ветви, построенное на основе хромосомных структур ([70], рисунок 6), в целом согласуется с деревом, ожидаемым на основе других данных (по белкам, рРНК, высококонсервативным элементам и др.). В некоторых пластидах наблюдается значительная перестройка хромосомной структуры. В пластидах хромерид *Chromera velia*, *Vitrella brassicaformis* (синоним *Chromerida RM11*) и багрянки *Porphyridium purpureum* взаимное расположение генов на хромосоме существенно отличается от такового в любых других пластидах. На том же дереве багрянки *Porphyridium purpureum*, как и *Choreocolax polysiphoniae*, отделились от других видов багрянок и друг от друга. Другие багрянки вместе с криптофитовыми водорослями образовали единую кладу. *Chromera velia* и *Vitrella brassicaformis* расположены близко друг к другу, но далеко от других представителей надтипа Alveolata. Споровик *Babesia bovis* отделился от других споровиков и оказался в кладе, образованной видами родов *Nannochloropsis* и *Trachydiscus*.

Эти особенности можно связать как с особенностями эволюции митохондрий и пластид, так и с относительно малым объёмом данных о них.

3. Выводы

Найдены линейные или близкие к ним по сложности (не выше кубических) точные алгоритмы решения задач преобразования и реконструкции хромосомных структур, согласования множеств контигов, которые реализованы эффективными компьютерными программами. Это – прямые алгоритмы или алгоритмы сведения к вариантам ЦЛП небольшого размера (не выше кубического). Полученные алгоритмы и программы применены для построения филогенетических деревьев митохондрий инфузорий и споровиков видов класса Aconoidasida, пластид родофитной ветви, а также бактерий рода *Rhizobium*. Подробное описание оригинальных алгоритмов и их тестирования приведено выше, в разделе «Основные положения, выносимые на защиту».

Публикации автора по теме диссертации, индексируемые WoS или Scopus

Статьи:

1. Lyubetsky V.A., **Gershgorin** R.A., Seliverstov A.V., Gorbunov K.Yu. Algorithms for Reconstruction of Chromosomal Structures // ***BMC Bioinformatics***. 2016, Vol. 17, № 40, 23 pages. DOI: 10.1186/s12859-016-0878-z. **WoS**
2. Горбунов К.Ю., **Гершгорин** Р.А., Любецкий В.А. Перестройка и реконструкция хромосомных структур //***Молекулярная биология***. 2015, Т. 49, № 3, С. 372–383. DOI: 10.7868/S0026898415030076. Перевод: Gorbunov K.Yu., **Gershgorin** R.A., Lyubetsky V.A. Rearrangement and Inference of Chromosome Structures // *Molecular Biology*. 2015, Vol. 49, № 3, P. 327–338. DOI: 10.1134/S0026893315030073. **WoS**
3. **Gershgorin** R.A., Gorbunov K.Yu., Zverkov O.A., Rubanov L.I., Seliverstov A.V., Lyubetsky V.A. Highly Conserved Elements and Chromosome Structure Evolution in Mitochondrial Genomes in Ciliates // ***Life***. 2017, Vol. 7(1). DOI: 10.3390/life7010009. **Scopus**
4. Lyubetsky V.A., **Gershgorin** R.A., Gorbunov K.Yu. Chromosome structures: reduction of certain problems with unequal gene content and gene paralogs to Integer linear programming // ***BMC Bioinformatics***. 2017, Vol. 18, № 537, 18 pages. **WoS**

5. **Gershgorin** R.A., Rubanov L.I., Seliverstov A.V. Easily Computable Invariants for Hypersurface Recognition // ***Journal of Communications Technology and Electronics***. 2015, Vol. 60, № 12, P. 1429–1431. DOI: 10.1134/S1064226915120074. **WoS**

Тезисы докладов:

6. **Gershgorin** R.A., Gorbunov K.Yu., Seliverstov A.V., Lyubetsky V.A. Evolution of Chromosome Structures //*Proceedings of the 39th IITP RAS Interdisciplinary Conference & School "Information Technology and Systems 2015" (ITaS'15)*, Sochi, Russia, Sep 7–11 2015.
7. Lyubetsky V.A., **Gershgorin** R.A., Rubanov L.I., Seliverstov A.V., Zverkov O.A. Evolution and Systematics of Plastids of Rhodophytic Branch // ***Proceedings of the International Moscow Conference on Computational Molecular Biology*** *(MCCMB'17)*, Moscow, Russia, July 27–30, 2017, 4 стр.
8. **Гершгорин** Р.А., Латкин И.В., Селиверстов А.В. Следы форм высших степеней //*Труды 57-й научной конференции МФТИ*, Москва–Долгопрудный–Жуковский, 24–29 ноября 2014, Управление и прикладная математика. Т. 1, М.: МФТИ, 2014, С. 11–12.

4. Обзор литературы

Рассмотрим публикации, наиболее близкие к нашим результатам.

В [3] предложен почти линейный (точнее, сложности $n \cdot f(n)$, где $f(n)$ – обратная функция Аккермана) алгоритм, который точно решает задачу согласования двух множеств контигов при условиях равного генного состава двух множеств контигов (в множествах из n генов каждое) и отсутствии паралогов в них. Расстояние в [3], хотя и близко к нашему, отличается от него. А именно, в наших обозначениях наше расстояние равно $n–C_1–0.5C_2$, где n – число имён генов, C_1 – число циклов и C_2 – число чётных цепей в общем графе; а расстояние, использованное в их работе, можно вычислить по той же формуле, если считать, что C_2 – число всех цепей.

В [4] изучались геномные перестройки у предков 20 современных митохондриальных геномов. Структуры предполагались циклическими и, по-видимому, из одной хромосомы. Цены всех операций равны 1. В качестве эволюционных событий рассматривались инверсия, транспозиция, удаление и вставка

генов. Использовались следующие величины. Первая из них $D(a,b)$ – число генов, присутствующих в одном геноме и отсутствующих в другом. Вторая $R(a,b)$ – минимальное число инверсий и транспозиций, необходимых для преобразования одной структуры к другой, при вычислении $R(a,b)$ учитывались только гены, общие для структур. Тогда расстояние определялось как $E=D+R$. Для его вычисления предложен эвристический алгоритм и советующая программа Derange, в основе которой (в силу трудности вычисления R) лежит известный метод ветвей и границ.

В [5] рассматривались геномы, состоящие из одной хромосомы и с равным генным составом. Рассматривалась только операция инверсии. Задача преобразования сведена к нахождению кратчайшей последовательности инверсий, которая приводит исходную последовательность целых чисел (ненулевых и разных по модулю) к строго возрастающей последовательности положительных чисел. Полученный алгоритм имеет полиномиальное $O(n^4)$ время работы.

В [6] рассмотрена задача определения расстояния между двумя хромосомными структурами с равным генным составом, в которых хромосома – последовательность ненулевых целых чисел. Предполагались операции – склейка (fusion), разрез (fission), инверсия (reversal), транспозиция (translocation). Получен алгоритм решения, который имеет сложность $O(n^4)$.

В [7] рассмотрена задача преобразования с равным генным составом с учётом блоков синтении. Рассматривалась только предложенная там операция DCJ (double-cut-and-join), которая эквивалентна использованию четырёх стандартных операций с равными ценами. Структуры предполагались состоящими только из линейных хромосом. Получен линейный алгоритм решения.

В [8] описан точный линейный алгоритм решения задачи преобразования для структур с равным генным составом и равными ценами операций.

В перечисленных выше работах паралоги не рассматривались.

В [9] рассматривалась задача преобразования с равным генным составом и одинаковыми числами одноимённых паралогов в структурах, и только с операцией DCJ. Задача сведена к нахождению максимальной декомпозиции графа смежности двух структур, что, в свою очередь, переформулировано как задача ЦЛП.

В [10] рассмотрена задача преобразования со своеобразным условием: неравный генный состав и паралогами, которые после установления инъективных отображений при вычислении расстояния просто игнорируются – операции вставки и удаления отсутствуют. Рассматривается только операция DCJ, эквивалентная использованию стандартных операций. Для вычисления расстояния использовался взвешенный граф смежности. В нём каждый ген получал уникальное имя. Полученная задача решалась сведением к ЦЛП. С самого начала на рёбрах графа вводилась функция похожести, присваивающая ребру, концы которого относятся к структурам, рациональное число от 0 до 1, и для графа смежности специальным образом вводилась функция расстояния между данными структурами.

В [11, 12] исследовался частный случай задачи преобразования с неравным генным составом, но без паралогов и с условием на цены: они равны, или равные цены вставки и удаления не больше равных цен стандартных операций. Предложены ценные идеи и эвристические алгоритмы.

В [13] предложено решение задачи преобразования для случая равных цен операций. Алгоритм принципиально отличается от нашего и, насколько можно судить, не допускает обобщения на случай неравных цен.

В [14], тезисах конференции, предложен план решения задачи преобразования при условии, что все хромосомы в структурах циклические, цены вставки и удаления равны друг другу, остальные цены также равны между собой. Насколько мне известно, до сих пор соответствующее доказательство не было опубликовано; по-видимому, предложенный план не удаётся реализовать.

В [15] предложена важная оригинальная конструкция breakpoint графа, которая обобщена диссертантом на случай неравного генного состава; здесь полученный граф назван *общим* и обозначается $a+b$. В [15] также предложено сведение задачи преобразования к приведению breakpoint графа к финальному виду, что в диссертации делается для общего графа $a+b$.

В [16] разработан эвристический алгоритм реконструкции хромосомных структур с одинаковым генным составом. В [17] предложен эвристический алгоритм решения задачи реконструкции для структур с неравным генным составом, который распространяет алгоритм из [16] на случай неравного генного состава.

ГЛАВА 1. КРАТЧАЙШЕЕ ПРЕОБРАЗОВАНИЕ ХРОМОСОМНЫХ СТРУКТУР БЕЗ ПАРАЛОГОВ: НЕРАВНЫЙ ГЕННЫЙ СОСТАВ И НЕРАВНЫЕ ЦЕНЫ ОПЕРАЦИЙ

1. Общий граф структур

Введём определения. *Особый ген* – ген, присутствующий только в одной из двух хромосомных структур. Особые гены, принадлежащие структуре a, будем называть a-генами, принадлежащие структуре b – b-генами. Гены, не являющиеся особыми, будем называть *общими*.

Напомним рассматриваемые операции над хромосомной структурой:

1) Разрез двух склеек вершин и переклейка четырёх краёв с образованием одного из двух вариантов новых пар склеек – двойная переклейка (рисунок 1a).

2) Разрез склейки вершин и новое склеивание одного ее края с каким-то не склеенным краем – полуторная переклейка (рисунок 1b).

3) Разрез склейки и обратная к ней операция склейки двух не склеенных краев – одинарные переклейки (рисунок 1c).

4) Удаление участка особых a-генов и вставка участка особых b-генов. (рисунок 1d).

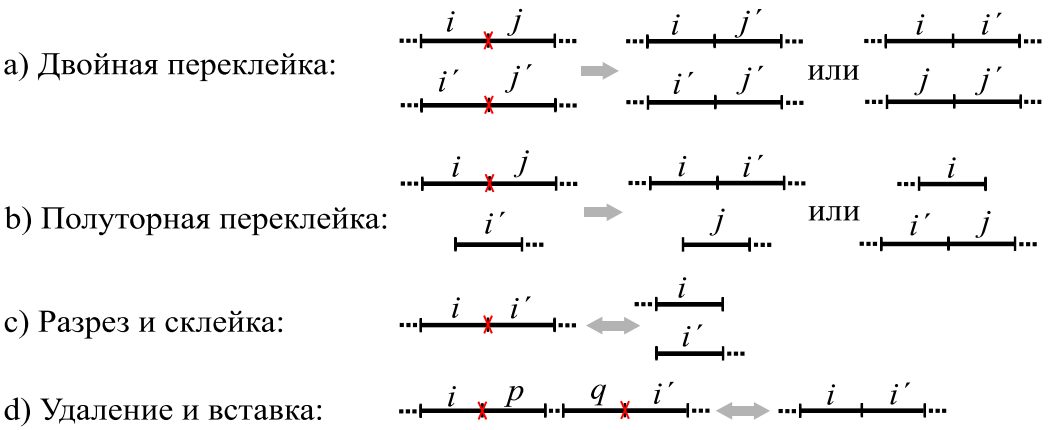


Рисунок 1. Операции над хромосомной структурой
Крестик указывает на разрез (расклейку двух склеенных вершин), двойная стрелка – на результат операции.

Каждой операции приписано положительное рациональное число – *цена*.

Постановка задачи. Даны хромосомные структуры a и b с неравным генным составом и без паралогов. Требуется найти *кратчайшую* последовательность операций, преобразующую структуру a в структуру b. Напомним, под кратчайшей последовательностью понимается последовательность с минимальной суммарной ценой всех входящих в неё операций.

Общий граф двух структур. Для гена с именем i пометкой i_1 будем обозначать 5'-конец гена, пометкой i_2 – 3'-конец. Чтобы определить его, нам понадобится несколько вспомогательных понятий.

Обычная вершина – край i_j общего для структур a и b гена.

Особая вершина (блок) – максимальный по включению связный участок из особых генов. Если он является циклом (включая петлю), будем называть его циклическим, иначе – линейным.

Вершинами общего графа двух структур a и b являются все обычные и особые вершины. Особые вершины помечаются именами особых генов, входящих в соответствующий блок и именем структуры, содержащей этот блок.

Ребра общего графа соединяют следующие пары вершин:

1) Края общих генов, которые отождествлены в одной из структур. Ребро помечается именем соответствующей структуры. Будем называть такое ребро *обычным*.

2) Если линейный блок находится между краями общих генов, то в общем графе данные края соединяются ребрами с особой вершиной, соответствующей блоку, оба ребра помечаются именем соответствующей структуры.

3) Если линейный блок соединяется только с одним краем общего гена, то в общем графе этот край соединяется с особой вершиной, соответствующей данному блоку, ребро помечается именем структуры, которой принадлежит блок. Такое ребро будем называть *висячим*.

4) Если линейный блок не имеет соседних общих генов (т.е. является цепью), то в общем графе ему соответствует изолированная особая вершина.

5) Циклическому блоку в общем графе отвечает петля при соответствующей особой вершине.

Будем обозначать общий граф структур a и b как $a+b$. A-Вершиной назовем особую вершину, помеченную a (аналогично для b-вершины).

Итак, общий граф показывает, какие края генов отождествлены в структурах *a* и *b*. Заметим, так как край любого гена может быть отождествлен максимум с одним краем в каждой из структур, степень каждой вершины в общем графе не выше 2, следовательно, общий граф может включать только циклические и линейные компоненты, а также изолированные вершины.

Общим графом *финального вида* называется общий граф двух совпадающих структур, т.е. граф вида *c*+*c* для некоторой структуры *c*. Он содержит только циклы длины 2 (одно ребро из структуры *a*, второе – из *b*) и изолированные обычные вершины. Такие циклы будем далее называть 2-циклами.

Пример. Рассмотрим в качестве примера две структуры (рисунок 2).

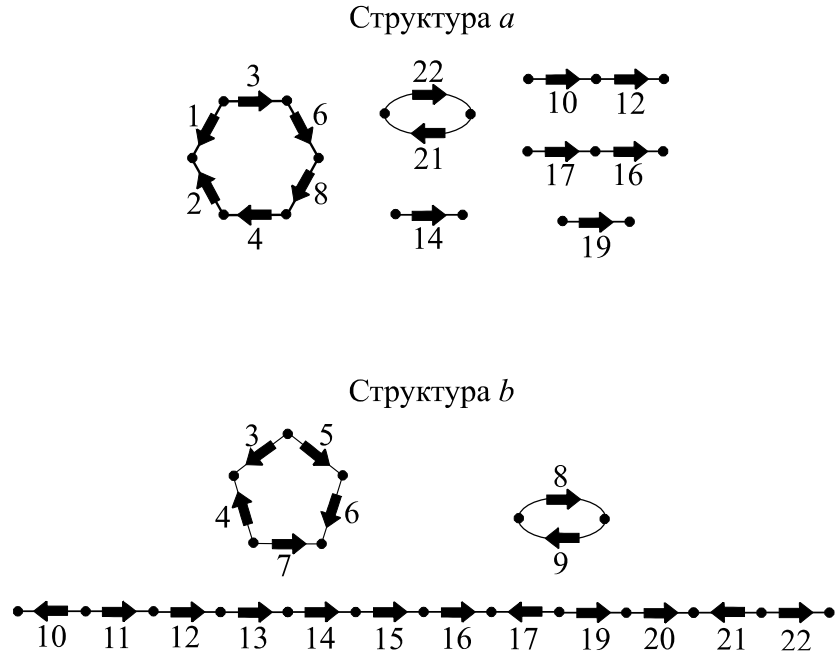


Рисунок 2. Две хромосомные структуры

Общий граф для данных структур состоит из двух циклических и пяти линейных компонент (рисунок 3).

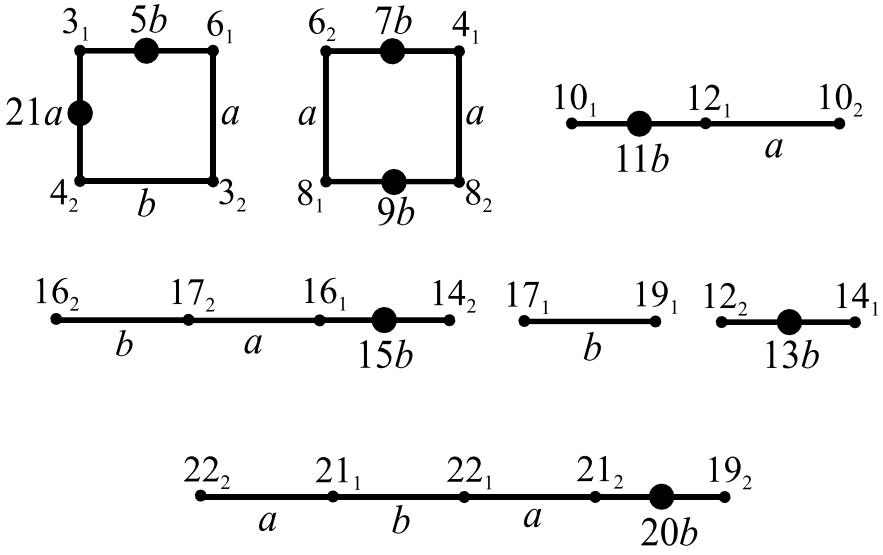


Рисунок 3. Общий граф *a*+*b* структур *a* и *b*

Операции над общим графом. Каждой операции, преобразующей структуру *a*, соответствует операция над общим графом *a*+*b*:

1) Двойная переклейка: удаление двух одинаково помеченных рёбер и соединение четырёх их концов двумя новыми не инцидентными ребрами с той же пометкой (рисунок 4*a*).

2) Полуторная переклейка: удаление ребра и соединение ребром с той же пометкой одного из его концов с обычной вершиной, не инцидентной никакому ребру с этой пометкой, или особой вершиной степени не выше 1 с такой же пометкой (рисунок 4*b*).

3) Склейка: добавление ребра (допустим, с пометкой *a*) между вершинами: обычной не инцидентной ребру с пометкой *a* или особой степени не выше 1 с пометкой *a* (рисунок 4*c*). Добавление *a*-ребра соответствует операции склейки в структуре, добавление *b*-ребра – разрезу.

4) Разрез: удаление любого ребра (рисунок 4*c*) Удаление *a*-ребра соответствует операции разреза в структуре, удаление *b*-ребра – склейке.

5) Удаление особой *a*-вершины (рисунок 4*d*). Соответствует удалению участка.

6) Удаление особой *b*-вершины (рисунок 4*d*). Соответствует вставке участка.

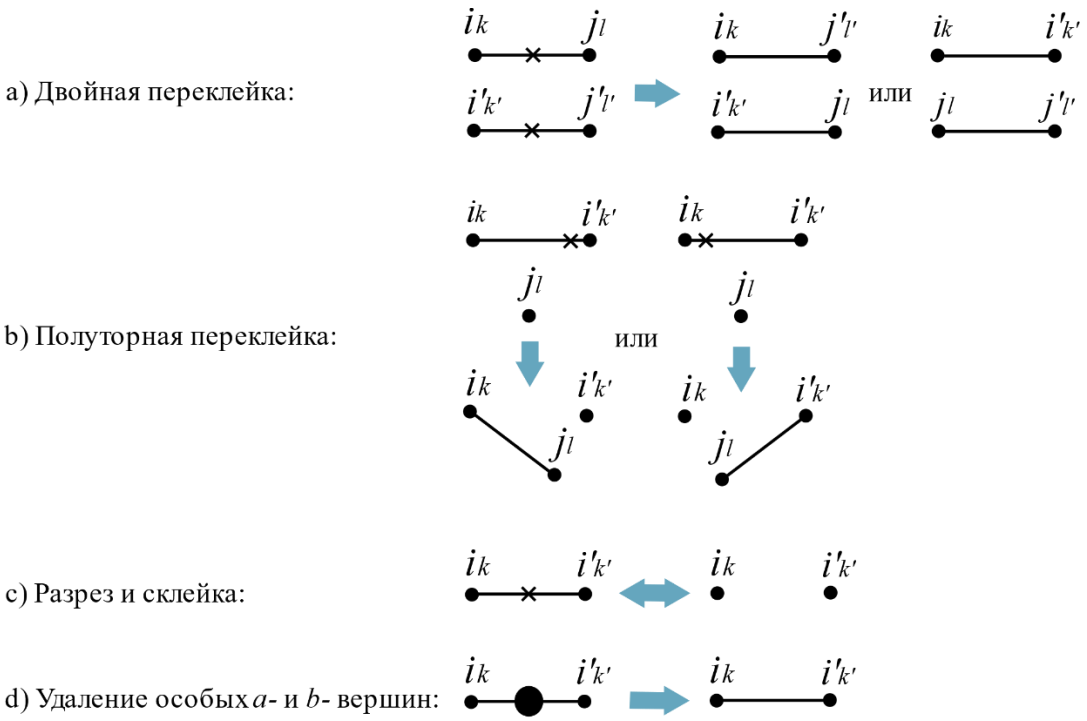


Рисунок 4. Операции над общим графом

Для удобства дальнейшего изложения будем называть операции *a-c стандартными,* операции *d — дополнительными.* Если в результате применения операции *a, b* или *c* образуется ребро, соединяющее две особые вершины с одинаковыми метками, то оно заменяется особой вершиной с именем, равным объединению имен в этих вершинах, число особых вершин таким образом уменьшается на 1. Новая особая вершина соответствует блоку, равному объединению соответствующих исходным вершинам блоков.

Каждой операции над общим графом сопоставляется *цена,* равная цене соответствующей операции над структурой.

Приведение общего графа к финальному виду. Рассмотрим теперь следующую задачу: для заданного общего графа двух структур $a+b$ найти кратчайшую последовательность операций, приводящую общий граф к финальному виду.

Следующая теорема обобщает результат в [43] на случай, когда цены удаления произвольные.

Теорема 1. Пусть цены всех стандартных операций равны между собой, а цены удаления особых *a*- и *b*-вершин произвольные. Тогда задача преобразования структуры *a* к структуре *b*, эквивалентна задаче приведения общего графа $a+b$ к финальному виду.

Точнее, цена решения обеих задач одинакова, и одна задача преобразуется в другую линейным алгоритмом.

Для доказательства рассмотрим третью задачу: преобразовать структуру *a* в структуру *b* кратчайшей последовательностью операций с условием, что каждая операция не разбивает *a*-блок.

Нам понадобятся две леммы.

Лемма 1. Первая задача эквивалентна третьей.

Обозначим $r(a,b)$ цену кратчайшей последовательности операций, преобразующей *a* в *b*. Докажем несколько вспомогательных утверждений.

Утверждение 1. Если структура *d'* получается из структуры *d* удалением из блока одного гена *g*, то $r(d',b) \leqslant r(d,b)$.

Доказательство. Докажем индукцией по величине $r(d,b)$. Заметим, что минимальная цена кратчайшей последовательности не может быть меньше c_{ad}, так как в структуре *d* присутствует *a*-блок. Если $r(d,b) = c_{ad}$, то *b* получается из *d* одной операцией удаления *a*-блока и $r(d',b) = r(d,b)$. Иначе, рассмотрим первую операцию *o* в кратчайшей последовательности преобразований от *d* к *b*. Ей соответствует такая операция *o'* над структурой *d'* (возможно, пустая), что *o'(d')* совпадает с *o(d)* или получается из неё удалением гена *g*. Очевидно: если *o* стандартная, то и *o'* стандартная, если *o* – вставка, то и *o'* – вставка, если *o* – удаление, то *o'* – удаление или пустая операция. Поэтому $c(o) \geqslant c(o')$, где $c(o)$ – цена операции *o*. По предположению индукции $r(o(d),b) \geqslant r(o'(d'),b)$. Отсюда $r(d,b) = r(o(d),b) + c(o) \geqslant r(o'(d'),b) + c(o') = r(d',b)$. □

Утверждение 2. Если структура *d'* получается из структуры *d* добавлением в блок одного гена *g*, то $r(d,b) = r(d',b)$.

Доказательство. Докажем по индукции по величине $r(d,b)$. Если $r(d,b) = c_{ad}$, то *b* получается из *d* одной операцией удаления *a*-блока и утверждение верно. Иначе, рассмотрим первую операцию *o* в кратчайшей последовательности для *d* и *b*. Ей соответствует такая операция *o'* над структурой *d'*, что *o'(d')* совпадает со структурой *o(d)* или получается из неё добавлением гена *g* в некоторый блок. Очевидно: тип операции *o'* совпадает с типом операции *o* и $c(o)=c(o')$. По предположению индукции $r(o(d),b) = r(o'(d'),b)$. Отсюда $r(d,b) = r(o(d),b) + c(o) = r(o'(d'),b) + c(o') = r(d',b)$. □

Операцию, не разбивающую блок, назовём *цельной*. Последовательность из цельных операций будем также называть *цельной*. *Сужением* блока назовём удаление

генов из него, а *расширением* блока – добавление генов, не принадлежащих структуре b, в него. Структуру d' назовём *упрощением* структуры d, если d' получается из d сужением или расширением блоков (т.е. в d' не может появиться блок там, где его не было в d).

Утверждение 3. Существует кратчайшая последовательность операций преобразования a к b, в которой все операции удаления a-блоков цельные.

Доказательство. Рассмотрим произвольную кратчайшую последовательность и первую нецельную оперцию удаления в ней. Обозначим d структуру, полученную после применения данной операции. Заменим эту операцию на цельную – удалим полный блок, получив структуру d'. Тогда из утверждения 1 следует, что $r(d,b) \geqslant r(d',b)$, то есть можно удлинить начало кратчайшей последовательности, не содержащее нецельных удалений. □

Утверждение 4. Существует кратчайшая последовательность операций от a к b, у которой все операции удаления и вставки цельные.

Доказательство. По утверждению 4 существует кратчайшая последовательность S, в которой все операции удаления цельные. Рассмотрим первую нецельную операцию вставки, разбивающую некоторый блок p, полученную после её применения структуру обозначим d. Заменим место вставки: вставим тот же отрезок на любой край блока p. Исходная структура d' получается из полученной удалением нескольких генов из блока, поэтому $r(d,b) \geqslant r(d',b)$. По утверждению 3 существует кратчайшая последовательность операций от d' к b, которая содержит только цельные операции удаления. Таким образом, можно удлинить начало кратчайшей операции вставки, не содержащей нецельные вставки. □

Утверждение 5. Пусть к некоторой структуре d применяется переклейка o, разбивающая блок p. Существует цельная операция переклейки o' (возможно, пустая), для которой структура $o'(d)$ – упрощение структуры $o(d)$.

Доказательство. Жестким назовём блок, ограниченный с обеих сторон общими генами, *полужёстким* – блок, ограниченный общим геном лишь с одной стороны, *свободным* – блок, не имеющий граничных общих генов.

Рассмотрим случаи:

1) o – разрез. Если r – жёсткий или полужёсткий блок, то o' – разрез любого края блока r. Если r – свободный блок, то o' – пустая операция.

2) o – полуторная переклейка, разбивающая блок r. Если r – жёсткий или полужёсткий блок, то o' – модифицированная полуторная переклейка o: расклейка

внутри r заменена на расклейку любого края блока r. Если r – свободный блок, то o' – пустая операция.

3) o – двойная переклейка. Рассмотрим сначала случай, когда одна её расклейка разбивает некоторый блок r, а вторая расклейка, обозначаемая s, не разбивает. Если r – жёсткий или полужёсткий блок, то o' – модифицированная двойная переклейка: расклейка внутри блока r заманена на расклейку любого края r. Если r – свободный блок, то o' – полуторная переклейка, которая заключается в расклейке s и склейке полученного края с любым краем r. Теперь рассмотрим случай, когда одна расклейка переклейки разбивает блок r_1, а другая – блок r_2. Пусть r_1 и r_2 – различные блоки. Если среди них нет свободного блока, то o' – модифицированная переклейка o: расклейки внутри блоков r_1 и r_2 заменены на расклейки любых краёв этих блоков. Иначе, пусть, например, r – свободный блок. Если блок r_2 – жёсткий или полужёсткий блок, то o' – полуторная переклейка, в которой расклейка осуществляется с любого края блока r_2, после чего к любому из двух образовавшихся краёв присоединяется блок r_1. Если блок r_2 – свободный, то o' – пустая операция. Если $r_1 = r_2$, то o' – пустая операция. □

Доказательство леммы 1. По утверждению 4 существует кратчайшая последовательность операций S, в которой все удаления и вставки цельные. Рассмотрим первую нецельную операцию переклейки o. Структуру, полученную после применения o назовём d. По утверждению 5 существует такая цельная переклейка o', что $r(o'(d),b) \leqslant r(o(d),b)$. По утверждению 4 существует кратчайшая последовательность операций, преобразующая $o'(d)$ в b, в которой все удаления и вставки цельные. Преобразуем последовательность S, убирая все нецельные переклейки и сохраняя цельные удаления и вставки. □

Лемма 2. Существует кратчайшая последовательность операций для третьей задачи, где все операции удаления участка идут перед всеми операциями вставки.

Утверждение 6. Пусть d – структура и $f = o_2(o_1(d))$, где операция o_1 обычная, а операция o_2 особая и обе они цельные. Существуют цельные операции o_3 и o_4 (одна из них может быть пустой) такие, что структура $o_4(o_3(d))$ – упрощение структуры f, и операция o_4 обычная.

Доказательство. Доказывается рассмотрением всех возможных операций o_1 и o_2. Так как o_1 – обычная операция, она сохраняет все блоки. Склеим те блоки, которые склеивала o_2, а затем доделаем то, что делала o_1. □

Доказательство теоремы 1. Пусть дано приведение общего графа $a+b$ к графу $c+c$. Тогда структура a преобразуется в структуру c последовательностью операций S_1, структура b преобразуется в структуру c последовательностью операций S_2. Рассмотрим последовательность операций, обратных к операциям из S_2. Эта последовательность S_2^{-1} преобразует структуру c в структуру b. Итак, существует последовательность $S_1 \cdot S_2^{-1}$, преобразующая структуру a в структуру b через c с такой же ценой.

Обратно, пусть дана последовательность преобразований из a в b. Тогда по леммам 1 и 2 существует кратчайшая последовательность S преобразующая a в b, в которой никакая операция не разбивает a-блоки и все операции удаления идут до операций вставки. Рассмотрим структуру c в последовательности S, полученную после всех удалений, но до всех вставок. Обозначим последовательность преобразований из a в c как S_1, последовательность преобразований из c в b как S_2. S_1 содержит только стандартные операции и удаления a-блоков. Рассмотрим последовательность S_2^{-1} операций, обратных к исходным. Она является кратчайшей последовательностью, преобразующей b в c и содержит только стандартные операции и удаления участков особых генов. По лемме 1 существует другая кратчайшая последовательность, преобразующая b в c, содержащая стандартные операции, не разбивающая b-блоки, и удаления b-блоков. Применяя аналоги операций из S_1 и S_2^{-1} для общего графа, получаем преобразование графа $a+b$ в граф $c+c$ с той же ценой. □

Таким образом, умея решать задачу приведения общего графа $a+b$ к финальному виду, можно решить исходную задачу. Рассмотрим алгоритм решения задачи для общего графа. Этот алгоритм является аддитивно точным при условии: цены всех операций, кроме операции b-удаления равны c, а цена операции b-удаления равна d, где $c \leqslant d \leqslant 2c$, аддитивная константа k равна $d–c$. Будем называть это условие *условием точности*. Сам алгоритм можно применять и за пределами нестационарного случая. После описания алгоритма приведены *эвристические рекомендации*, указывающие

варианты алгоритма в зависимости от соотношений цен. При выполнении условий точности алгоритм остаётся аддитивно точным независимо от выбранного варианта.

Заметим: автор участвовал в выработке схемы доказательства аддитивной точности алгоритма [39], но схема и само доказательство не выносится на защиту; оно опубликовано в [18].

2. Алгоритм кратчайшего преобразования структур

Пусть дан общий граф $a+b$ двух структур с неравным генным составом и без паралогов. Требуется найти последовательность операций, приводящих его к финальному виду, цена которой отличается от минимальной цены не более чем на $d–c$.

Размер компоненты общего графа – сумма числа внутренних (т.е. степени 2) особых вершин и числа обычных рёбер.

Нечетной (*четной*) *цепью* назовём цепь нечетного (четного) размера. *a-Цепью* называется нечётная цепь, у которой крайние невисячие ребра помечены *a*, или изолированная *b*-вершина. Аналогично определяется *b-цепь*. Оставшимся после шагов 1-2 цепям и циклам (кроме финальных 2-циклов и изолированных обычных вершин, см. описание алгоритма) *приписываем тип*: 2-циклу, содержащему *a*-вершину, но не *b*-вершину – «*a*-цикл»; симметрично – «*b*-цикл». Циклу, в котором имеются как *a*-вершины, так и *b*-вершины, приписываем тип «цикл». Особой *b*-петле приписываем тип «петля». *a*-Цепи приписываем типы: 1*a*, если в ней одно висячее ребро; 2*a*, если в ней два таких ребра; 2*a′* – если это изолированная *b*-вершина; 3*a*, если у неё нет висячих рёбер, но имеются как *a*-, так и *b*-вершины (тогда размер цепи строго больше 1); 3*a′*, если нет ни висячих ребер, ни *b*-вершин (тогда размер цепи равен 1). *b*-Цепям тип приписывается аналогично. Отметим, что выделение типов «без штриха» и «со штрихом» связано с тем, что важно выделить цепи, в которых нет *b*-вершин или *a*-вершин. Чётной цепи приписывается тип: 1, если в ней одно висячее ребро и имеется *b*-вершина и *a*-вершина; 1′, если она состоит из одной обычной вершины и инцидентной ей *a*-вершины; 1″, если она состоит из одной обычной вершины и инцидентной ей *b*-вершины; 2, если в ней два висячих ребра и есть невисячие ребра; 2′, если в ней два висячих ребра и нет других ребер; 3, если в ней имеется хотя бы одно ребро и нет висячих рёбер. Среди цепей типа 1 выделим цепи типа 1_a (если висячая вершина – *a*-вершина) и 1_b (если она – *b*-вершина).

Рисунок 5. Типы цепей и циклов, содержащих особые вершины

Описание алгоритма. Алгоритм состоит из 5 шагов, схема представлена на рисунке 6, далее следует подробное описание каждого шага.

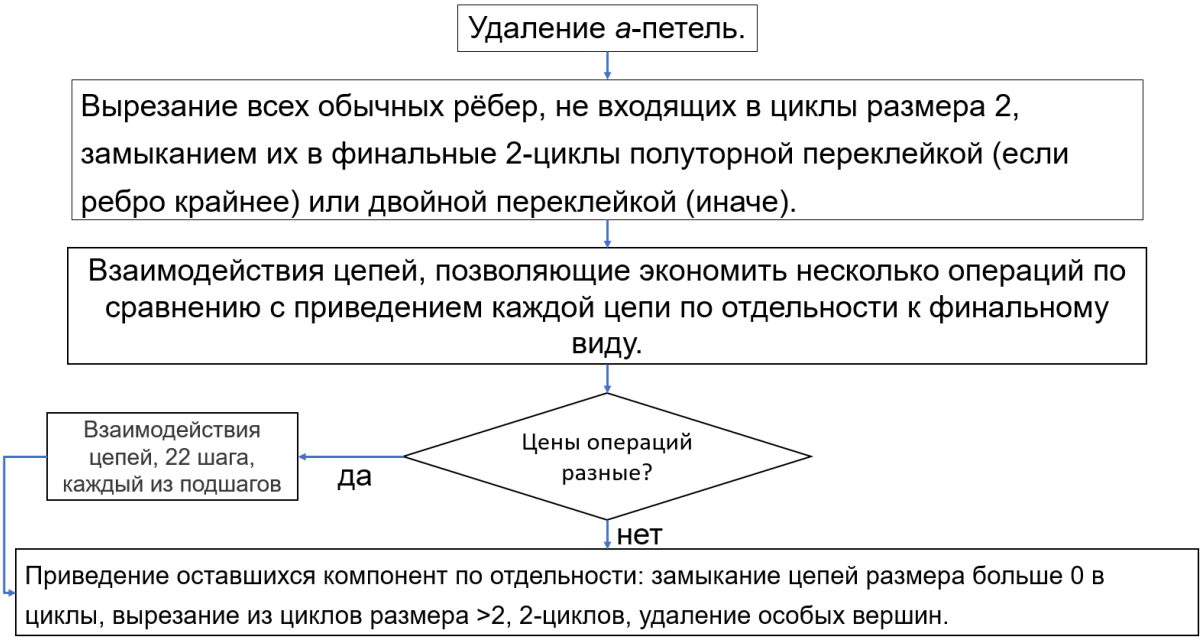


Рисунок 6. Схема алгоритма приведения общего графа к финальному виду

Шаг 1. Удалить особые *a*-петли.

Шаг 2. Вырезать все обычные ребра, не входящие в 2-циклы (т.е. циклы размера 2), замыкая их в финальные 2-циклы двойной (если ребро не крайнее, рисунок 6*a*) или полуторной (если оно крайнее, рисунок 6*b*) переклейками или склейкой (если оно изолированное, рисунок 6*c*).

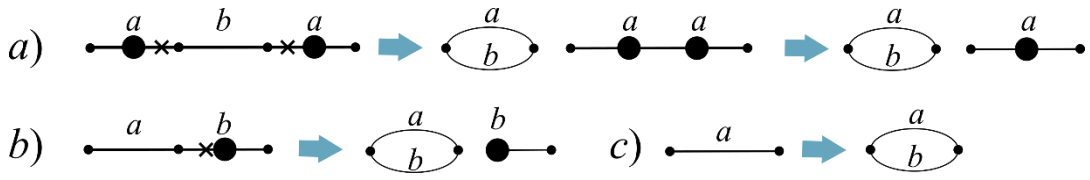


Рисунок 6. Шаг 2 алгоритма

Шаг 3. Один пункт на шагах 3–4 может включать несколько последовательных преобразований, которые отделяются друг от друга знаком равенства. В описании одного пункта перед каждым знаком равенства указываются (разделяемые знаком +) типы цепей до выполнения операции, а после равенства – тип цепи, полученной после её выполнения. Изолированные обычные вершины и финальные 2-циклы не указываются. Для краткости приводится описание только *первого равенства*; описания последующих равенств аналогичны. Типом $2a$ называем объединение типов $2a$ и $2a'$, типом $3b$ – типов $3b$ и $3b'$, типом 1_b – типов 1_b и $1''$ и типом 2 – типов 2 и $2'$ (это позволяет сократить число преобразований).

Итак, продолжим описание алгоритма. На примере пункта 3.2 поясним содержания пунктов: последовательно применяем к парам цепей типов $2a$ и $3b$ указанную операцию, получаем цепь типа 1_b, затем аналогично последовательно преобразуем цепи типа $2b$ и $3a$, $2b'$ и $3a$, $2b$ и $3a'$, $2b'$ и $3a'$ в цепи соответственно типов 1_a, 1_a, 1_a и $1'$. Таким образом, последовательно выполняем все пункты.

3.1) $1a+1b=1_b$. Расклеим крайнее невисячее ребро (назовём его *внешним*) в цепи типа $1a$ и соответствующую особую вершину склеим с крайней особой вершиной цепи $1b$ (полуторная переклейка, рисунок 9).

Рисунок 9. Шаг 3.1 алгоритма

3.2) $2a+3b=1_b$, $2b+3a=1_a$, $2b'+3a=1_a$, $2b+3a'=1_a$, $2b'+3a'=1'$. В $3b$-цепи расклеим внешнее ребро и особую вершину склеим с крайней особой вершиной $2a$-цепи (рисунок 10).

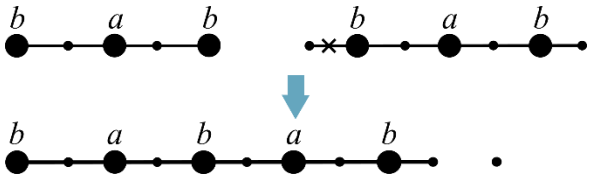


Рисунок 10. Шаг 3.2 алгоритма

3.3) $2+3=1_b$. В 3-цепи расклеим внешнее ребро и особую вершину склеим с крайней особой вершиной 2-цепи (рисунок 11).

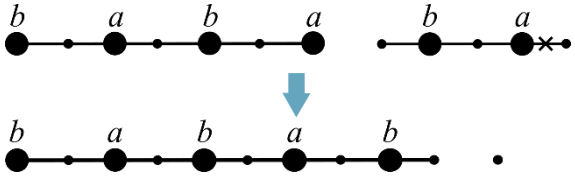


Рисунок 11. Шаг 3.3 алгоритма

3.4) $1b+2a+3=2+3=1_b$, $1a+2b+3=2+3=1_b$, $1a+2b'+3=2+3=1_b$. Сначала выполняем $1b+2a=2$ (описание ниже); затем $2+3=1_b$.

3.5) $1a+3b+2=3+2=1_b$, $1b+3a+2=3+2=1_b$, $1b+3a'+2=3+2=1_b$. Сначала $1a+3b=3$ (описание ниже); затем $2+3=1_b$.

3.6) $1a+2=2a$, $1b+2=2b$. В $1a$-цепи расклеим внешнее ребро и особую вершину склеим с крайней особой вершиной 2-цепи (рисунок 12).

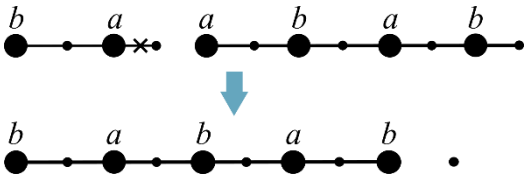


Рисунок 12. Шаг 3.6 алгоритма

3.7) $1a+3=3a$, $1b+3=3b$. В 3-цепи расклеим крайнее b-ребро и особую вершину склеим с крайней особой вершиной $1a$-цепи (рисунок 13).

Рисунок 13. Шаг 3.7 алгоритма

3.8) $1a+1a+2b+3b=2+3=1_b$, $1a+1a+2b'+3b=2+3=1_b$, $1b+1b+2a+3a=2+3=1_b$, $1b+1b+2a+3a'=2+3=1_b$. Сначала выполняем $1a+2b=2$ и $1a+3b=3$; затем $2+3=1_b$.

3.9) $1a+1a+2b=3a+2b=1_a$, $1a+1a+2b'=3a+2b'=1_a$, $1b+1b+2a=3b+2a=1_b$. Сначала $1a+1a=3a$ (описание ниже); затем $2b+3a=1_a$.

3.10) $1a+1a+3b=1a+3=3a$, $1b+1b+3a=1b+3=3b$, $1b+1b+3a'=1b+3=3b$. Сначала $1a+3b=3$; затем $1a+3=3a$.

3.11) $1a+1a=3a$, $1b+1b=3b$. Склеим крайние особые вершины двух $1a$-цепей (рисунок 14).

Рисунок 14. Шаг 3.11 алгоритма

3.12)$1a+2b=2$, $1a+2b'=2$, $1b+2a=2$. В $1a$-цепи расклеим внешнее ребро и особую вершину склеим с крайней особой вершиной $2b$-цепи (рисунок 15).

Рисунок 15. Шаг 3.12 алгоритма

3.13) $1a+3b=3$, $1b+3a=3$, $1b+3a'=3$. В $3b$-цепи расклеим внешнее ребро и особую вершину склеим с крайней особой вершиной $1a$-цепи (рисунок 16).

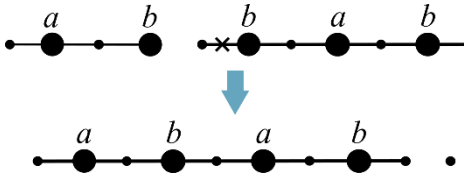


Рисунок 16. Шаг 3.13 алгоритма

3.14) $2a+2b+3+3=2+3=1_b$, $2a+2b'+3+3=2+3=1_b$. Сначала $2a+2b+3=2$, затем $2+3=1_b$.

3.15) $3a+3b+2+2=3+2=1_b$, $3a'+3b+2+2=3+2=1_b$. Сначала $3a+3b+2=3$, затем $2+3=1_b$. Упомянутые переходы описаны ниже.

3.16) $2a+3+3=1a+3=3a$, $2b+3+3=1b+3=3b$, $2b'+3+3=1b+3=3b$. Сначала $2a+3=1a$, затем $1a+3=3a$.

3.17) $3b+2+2=1b+2=2b$, $3a+2+2=1a+2=2a$, $3a'+2+2=1a+2=2a$. Сначала $3b+2=1b$; затем $1b+2=2b$. Упомянутые описания приведены в шаге 4.

3.18) $2a+2b+3=2a+1b=2$, $2a+2b'+3=2a+1b=2$. Сначала $2b+3=1b$; затем $1b+2a=2$.

3.19) $3a+3b+2=3a+1b=3$, $3a'+3b+2=3a'+1b=3$. Сначала $3b+2=1b$; затем $1b+3a=3$.

Шаг 4. Данный шаг применяется при следующем условии: удаление *b*-вершин больше цен других операций.

Каждый пункт применяем пока возможно, затем переходим к следующему.

4.1) «петля»+любой тип *t* с *b*-вершиной = тип *t*. Если *t* не равен 2*a′*, двойной переклейкой «вставить» петлю в компоненту типа *t* с отождествлением *b*-вершин. Иначе, сделать то же полуторной переклейкой (рисунок 17).

Рисунок 17. Шаг 4.1 алгоритма

4.2. «цикл»+любой тип *t* с *b*-вершиной и *a*-вершиной = тип *t*. Вставить цикл (двойной переклейкой, отождествляющей две *b*-вершины) рядом с *b*-вершиной из компоненты типа *t* с той стороны, в которой находится *a*-вершина; образовавшееся обычное ребро вырезать (рисунок 18).

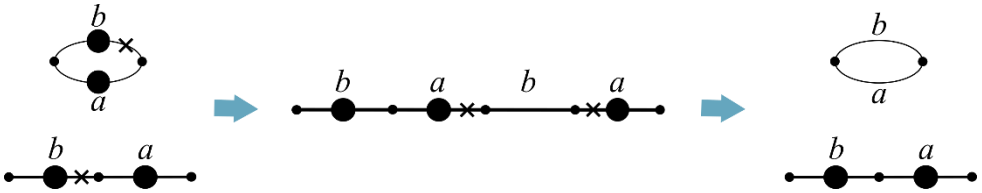


Рисунок 18. Шаг 4.2 алгоритма

4.3) 2*a*+2*b*=2+1′. Полуторная переклейка с отрезанием двух вершин 2*b*-цепи (крайней *a*-вершины и соседней обычной вершины) и склейкой образовавшегося края с крайней *b*-вершиной 2*a*-цепи (рисунок 19).

Рисунок 19. Шаг 4.3 алгоритма

4.4) 3*a*+3*b*=3. В 3*a*-цепи расклеить внешнее ребро и особую вершину склеить с крайней обычной вершиной 3*b*-цепи (рисунок 20).

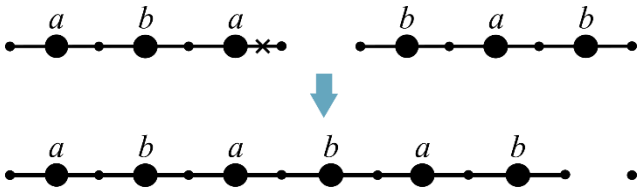


Рисунок 20. Шаг 4.4 алгоритма

4.5) 2*a*+3=1*a*, 2*b*+3=1*b*. В 3-цепи расклеить внешнее *b*-ребро и особую вершину склеить с крайней особой вершиной 2*a*-цепи (рисунок 21).

Рисунок 21. Шаг 4.5 аллгоритма

4.6) 3*a*+2=1*a*, 3*b*+2=1*b*. В 3*a*-цепи расклеить внешнее ребро и особую вершину склеить с крайней особой вершиной 2-цепи (рисунок 22).

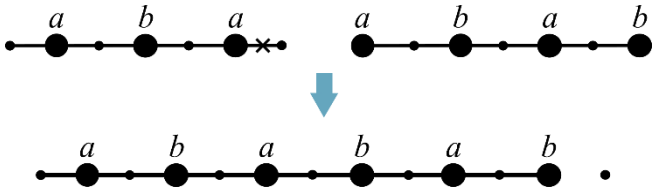


Рисунок 22. Шаг 4.6 алгоритма

4.7) 2*a*+2*a*=2*a*, 2*b*+2*b*=2*b*. Склеить крайние особые вершины двух цепей (рисунок 23).

Рисунок 23. Шаг 4.7 алгоритма

4.8) 3*a*+3*a*=3*a*, 3*b*+3*b*=3*b*. Две крайние обычные вершины цепей соединить обычным ребром с последующим его вырезанием (рисунок 24).

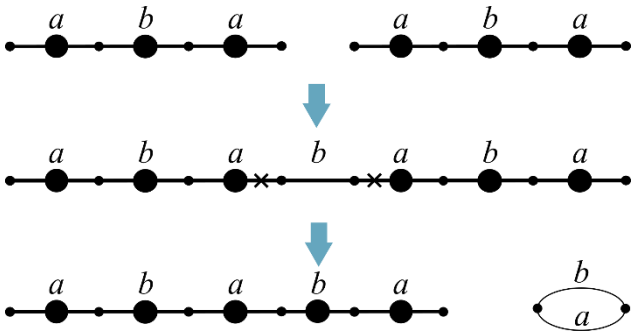


Рисунок 24. Шаг 4.8 алгоритма

4.9) 1*a*+2*a*=1*a*, 1*b*+2*b*=1*b*. Склеить крайние особые вершины двух цепей (рисунок 25).

Рисунок 25. Шаг 4.9 алгоритма

4.10) 1*a*+3*a*=1*a*, 1*b*+3*b*=1*b*. Две крайние обычные вершины цепей соединить обычным ребром с последующим его вырезанием (рисунок 26).

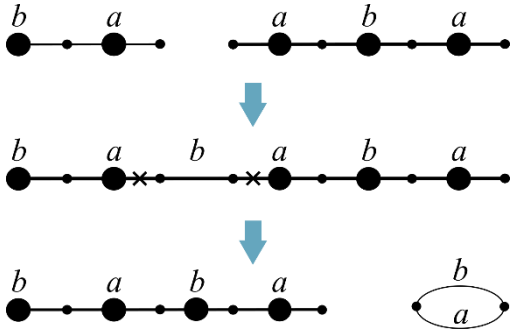


Рисунок 26. Шаг 4.10 алгоритма

4.11) 2*a*+2=2, 2*b*+2=2. Склеить крайние особые вершины двух цепей (рисунок 27).

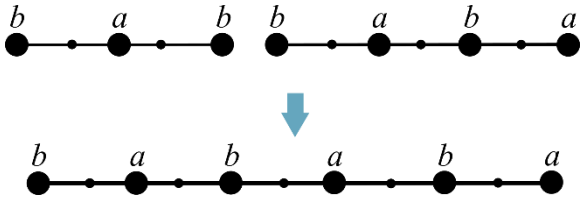


Рисунок 27. Шаг 4.11 алгоритма

4.12) 3*a*+3=3, 3*b*+3=3. Две крайние обычные вершины цепей соединить обычным ребром с последующим его вырезанием (рисунок 28).

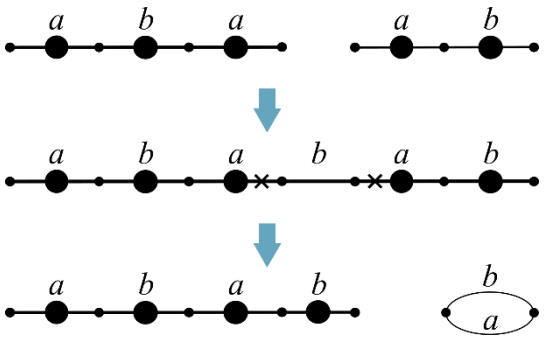


Рисунок 28. Шаг 4.12 алгоритма

4.13) 2+2=2+1′. Полуторная переклейка с отрезанием двух вершин 2-цепи (крайней *a*-вершины и соседней обычной вершины) и склейкой образовавшегося края с крайней *b*-вершиной другой 2-цепи (рисунок 29).

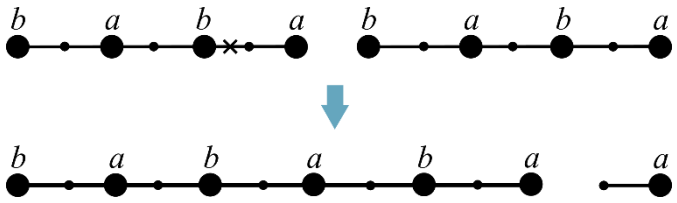


Рисунок 29. Шаг 4.13 алгоритма

4.14) 3+3=3. В 3-цепи расклеить внешнее *a*-ребро и образовавшийся край этой цепи склеить с *b*-краем другой 3-цепи (рисунок 30).

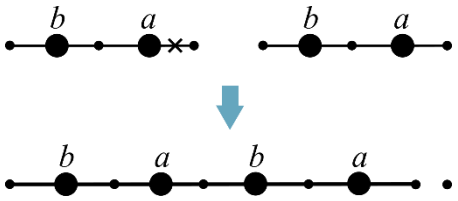


Рисунок 30. Шаг 4.14 алгоритма.

4.15) $1_a+1_a=1_a$, $1_b+1_b=1_b$. В 1_a-цепи расклеить внешнее ребро и особую вершину склеить с крайней особой вершиной другой 1_a-цепи (рисунок 31).

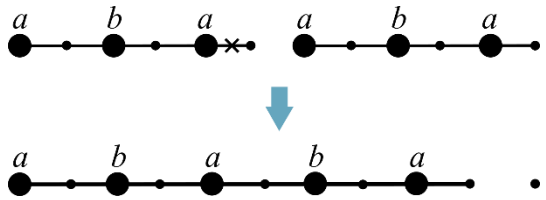


Рисунок 31. Шаг 4.15 алгоритма

4.16) $1a+1_b=1a$, $1b+1_a=1b$. В 1_b-цепи расклеить внешнее ребро и особую вершину склеить с крайней особой вершиной $1a$-цепи (рисунок 32).

Рисунок 32. Шаг 4.16 алгоритма

4.17) $1a+1_a=1a$, $1b+1_b=1b$. В $1a$-цепи расклеить внешнее ребро и особую вершину склеить с крайней особой вершиной $1a$-цепи (рисунок 33).

Рисунок 33. Шаг 4.17 алгоритма

4.18) $2a+1_b=2a$, $2b+1_a=2b$. В 1_b-цепи расклеить внешнее ребро и особую вершину склеить с крайней особой вершиной $2a$-цепи (рисунок 34).

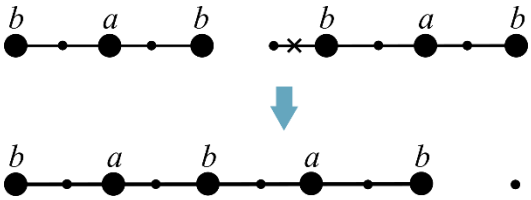


Рисунок 34. Шаг 4.18 алгоритма

4.19) $3a+1_a=3a$, $3b+1_b=3b$. В $3a$-цепи расклеить внешнее ребро и особую вершину склеить с крайней особой вершиной 1_a-цепи (рисунок 35).

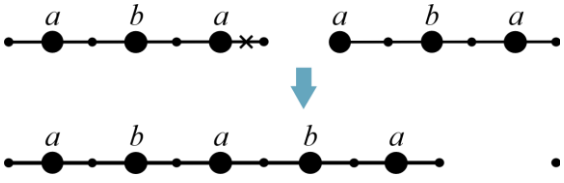


Рисунок 35. Шаг 4.19 алгоритма

4.20) $2+1_a=2$, $2+1_b=2$. В 1_a-цепи расклеить внешнее ребро и особую вершину склеить с крайней особой вершиной 2-цепи (рисунок 36).

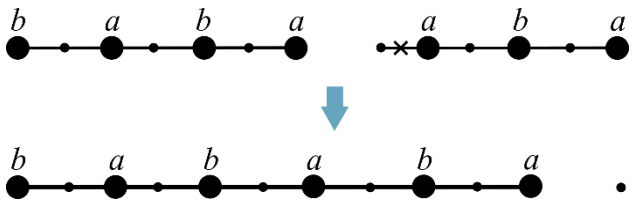


Рисунок 36. Шаг 4.20 алгоритма

4.21) $3+1_a=3$, $3+1_b=3$. В 3-цепи расклеить внешнее ребро и особую вершину склеить с крайней особой вершиной 1_a-цепи (рисунок 37).

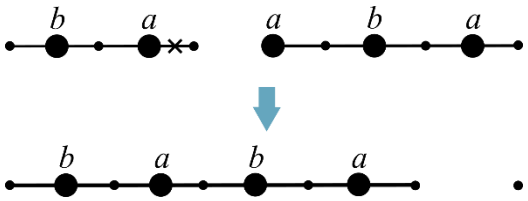


Рисунок 37. Шаг 4.21 алгоритма

4.22) Цепи, имеющие невисячее ребро, замыкаем в циклы склейкой (цепи типа $2a$, $2b$, $3a$, $3b$), полуторной переклейкой с отождествлением особых вершин (цепи типа 1_a, 1_b, 1_c, 2) или без отождествления (цепи типа $1a$, $1b$, 3). При замыкании цепи типа 1_c определяем $c=b$. При замыкании цепи типа 2 выбираем вариант с отождествлением двух b-вершин (рисунок 38), из образовавшейся цепи типа $1'$ удаляем a-вершину. Из циклов, получившихся при замыкании цепей типа $3a$ или $3b$, вырезаем обычные ребра. Затем снова применяем шаг 4.2.

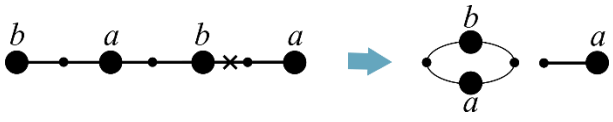


Рисунок 38. Шаг 4.22 алгоритма

Шаг 5. Удаляем висячие рёбра, петли и изолированные особые вершины. Из циклов размера большего 2 вырезаем 2-циклы так, чтобы происходило отождествление двух b-вершин (соответственно, в 2-цикл включается a-вершина, рисунок 39). Из 2-циклов удаляем особые вершины.

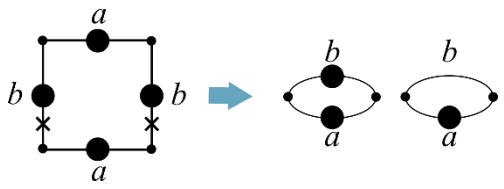


Рисунок 39. Шаг 5 алгоритма

Конец описания алгоритма.

Использование алгоритма за пределами нестационарного случая. Как мы отмечали, данный алгоритм можно применять и при ценах, не удовлетворяющих условию точности. На практике, оказалось удобным применять его при следующих ценах операций (здесь c_2 – цена двойной переклейки, $c_{1.5}$ – цена полуторной переклейки, c_1 – цена разреза, c_1' – цена склейки, c_{ad} – удаление a-вершины, c_{bd} – удаление b-вершины):

$c_2 = 0.9, c_1 = 1, c_1' = 1.1, c_{1.5} = 1.2, c_{ad} = 0.8, c_{bd} = 1.5$ – циклический вариант и

$c_2 = 1.2, c_1 = 0.9, c_1' = 1, c_{1.5} = 1.1, c_{ad} = 0.8, c_{bd} = 1.5$ – линейный вариант.

Циклический вариант даёт хорошие результаты, если в структурах преобладают циклические хромосомы (например, в пластидах), а линейный вариант – если преобладают линейные хромосомы (например, в митохондриях споровиков).

Рассмотрим несколько эвристических дополнений к алгоритму, позволяющих в зависимости от использования линейных и циклических наборов цен получать сценарии с меньшей ценой.

Если цена двойной переклейки не больше цены полуторной, то сначала на шаге 2 выполняются всевозможные двойные переклейки, иначе – всевозможные полуторные.

На примере преобразований 3.3 заметим следующее: в зависимости от того, какое из двух внешних рёбер расклеивается в 3-цепи, получается цепь типа 1_a или 1_b (рисунок 40*a* и 40*b*). Аналогичное верно для 3.1, 3.4, 3.5, 3.8, 3.14, 3.15.

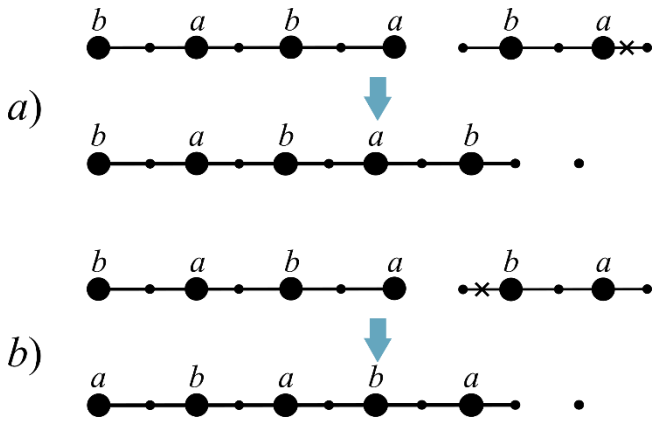


Рисунок 40. Два варианта реализации шага 3.3

Обозначим отложенный выбор между цепями типов 1_a и 1_b – двумя возможными результатами соответствующей операции, как тип 1_c. Теперь в преобразованиях из 4.15-4.21 будем определять $c=b$ и включать их в множество 1_b цепей. Если после шага 4.21 остаются неиспользованные 1_c цепи, проведём следующие преобразования: $1a+1_c=1_a$, $1b+1_c=1b$, $1a+1_c=1a$, $2b+1_c=2b$, $3a+1_c=3a$, определив $c=a$. После этого шага оставшиеся 1_c цепи преобразуем с помощью $1_b+1_b=1_b$, определив $c=b$. На шаге 4.22 для цепей 1_c определяем $c=b$. Такая эвристика позволяет отождествить как можно больше b-вершин друг с другом, что может уменьшить аддитивную ошибку алгоритма. Для шага 4 также можно применить следующую эвристику: если цена двойной переклейки меньше цены полуторной, вместо преобразований 4.3 – 4.21 применяем преобразования 4.3′ – 4.21′.

4.3′) $2a'+2b=2+1'$.

4.4′) $3a+3b'=3$.

4.5′) $2a'+3=1a$.

4.6′) $3a+2'=1a$, $3b'+2=1b$.

4.7′) $2a'+2a=2a$.

4.8′) $3b'+3b=3b$.

4.9′) $1a+2a'=1a$.

4.10′) $1b+3b'=1b$.

4.11′) $2a'+2=2$, $2a+2'=2$, $2b+2'=2$.

4.12′) $3b'+3=3$.

4.13′) $2'+2=2+1'$.

4.14′) Пустое действие.

4.15′) $1''+1_b=1_b$, $1''+1_c=1_b$ (определяется $c=b$).

4.16′) $1a+1''=1a$.

4.17′) $1b+1''=1b$.

4.18′) $2a'+1_b=2a$, $2a+1''=2a$, $2a'+1_c=2a$ (определяется $c=b$).

4.19′) $3b'+1_b=3b$, $3b+1''=3b$, $3b'+1_c=3b$ (определяется $c=b$).

4.20′) $2'+1_a=2$, $2'+1_b=2$, $2+1''=2$, $2'+1_c=2$ (определяется $c=b$).

4.21′) $3+1''=3$.

3. Тестирование на искусственных примерах

Данный алгоритм был реализован в виде утилиты командной строки и протестирован на искусственных и биологических данных. Результаты работы на биологических данных обсуждаются в Главе 4. Приведём примеры работы для искусственных структур.

Пример 1. Рассмотрим структуры *a* и *b*, показанные на рисунке 41.

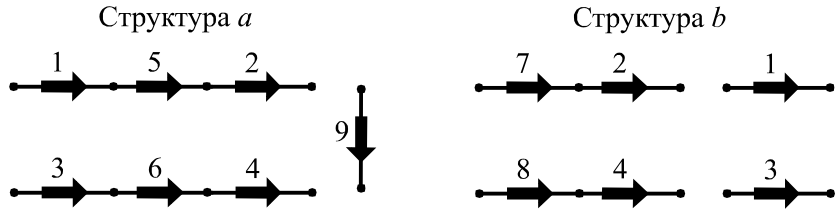


Рисунок 41. Две хромосомные структуры

На рисунке 42 показана последовательность структур и операций, которую выдает алгоритм при преобразовании к финальному виду общего графа *a*+*b* структур *a* и *b*, показанных на рисунке 41.

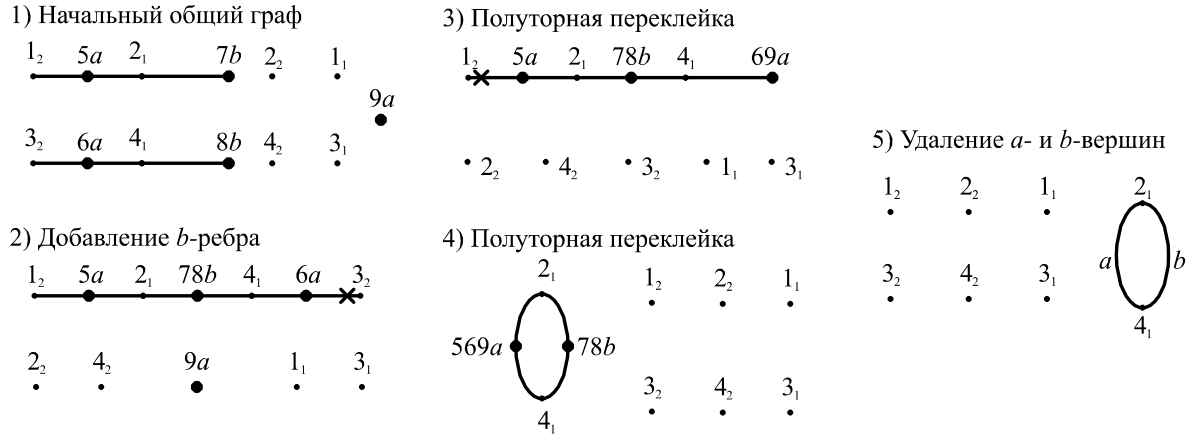


Рисунок 42. Последовательность структур и операций, выдаваемая алгоритмом для примера, показанного на рисунке 41

Приведенная на рисунке 42 последовательность является кратчайшей при линейных ценах операций, суммарная цена равна 5.4. При циклических ценах эта последовательность не является кратчайшей, так как тогда ее цена становится равной 5.7, а последовательность операций, показанная на рисунке 43, имеет суммарную цену 5.4.

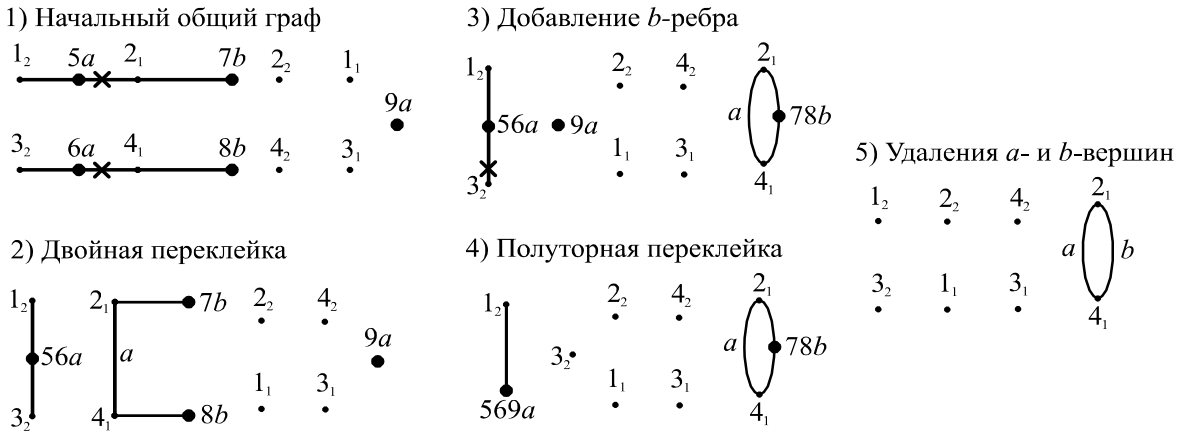


Рисунок 43
Последовательность структур и операций, кратчайшая при циклических ценах для примера, показанного на рисунке 41

Пример 2.

Рассмотрим алгоритм приведения общего графа для следующих структур.

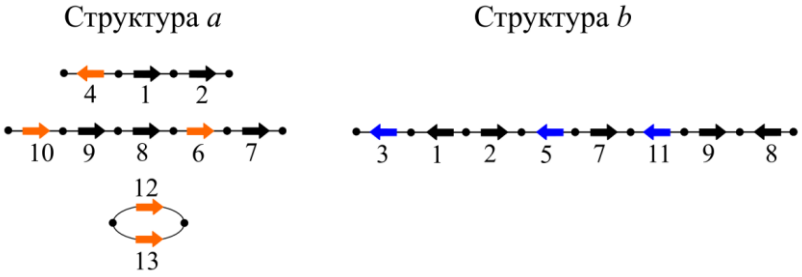


Рисунок 44. Исходные структуры примера 2

Общий граф имеет вид:

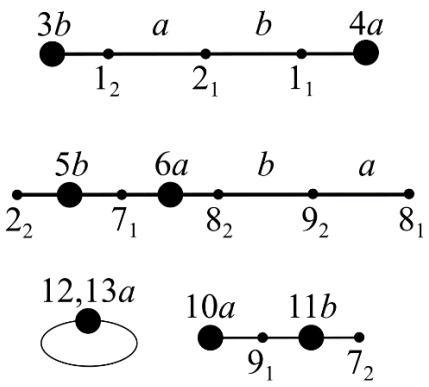


Рисунок 45. Общий граф структур с рисунка 44

Алгоритм строит следующую последовательность преобразований:

1) Удалим *a*-петлю:

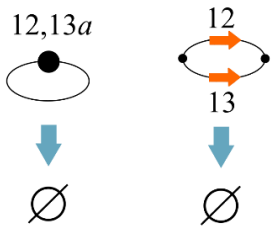


Рисунок 46. Удаление a-петли в общем графе
Данному удалению отвечает удаление 2-цикла из структуры *a*

2) Вырежем обычные ребра двойной (рисунок 47) и полуторной (рисунок 48) переклейкой.

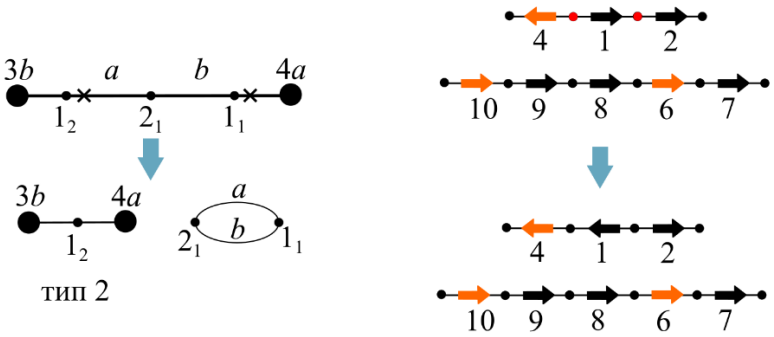


Рисунок 47. Вырезание обычного *b* ребра двойной переклейкой
Соответствует инверсии ребра 1 в структуре *a*

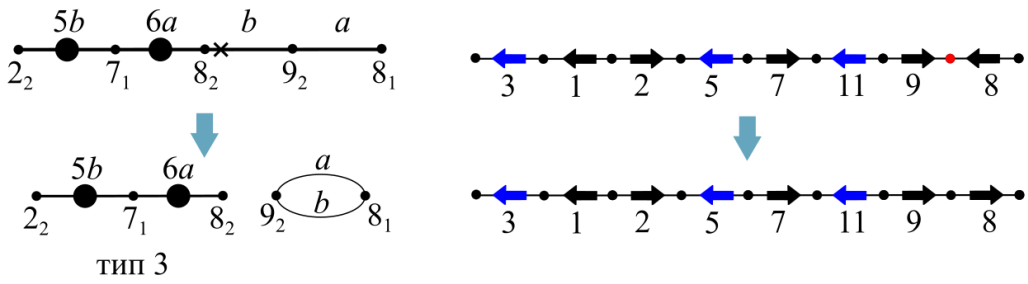


Рисунок 48. Вырезание обычного *a* ребра полуторной переклейкой
Соответствует инверсии ребра 8 в структуре *b*

3) Проведём взаимодействие цепей $2+3=1_b$:

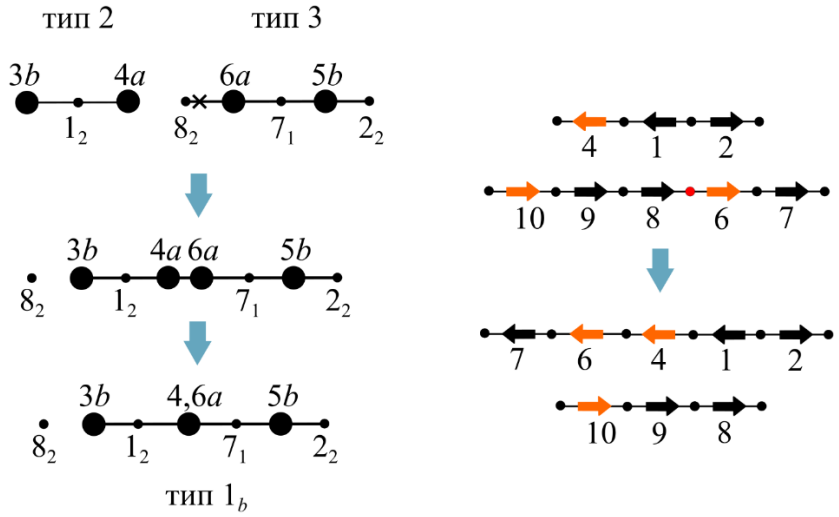


Рисунок 49. Взаимодействие цепей $2+3=1_b$
Полуторной переклейкой расклеиваем ребро 8_2 – 6*a* и склеиваем 6*a* с 4*a*. Соответствует переклейке участка рёбер 6, 7 к концу другой цепи в структуре *a*

4) Проведем взаимодействие цепей $1b+1_b=1_b$:

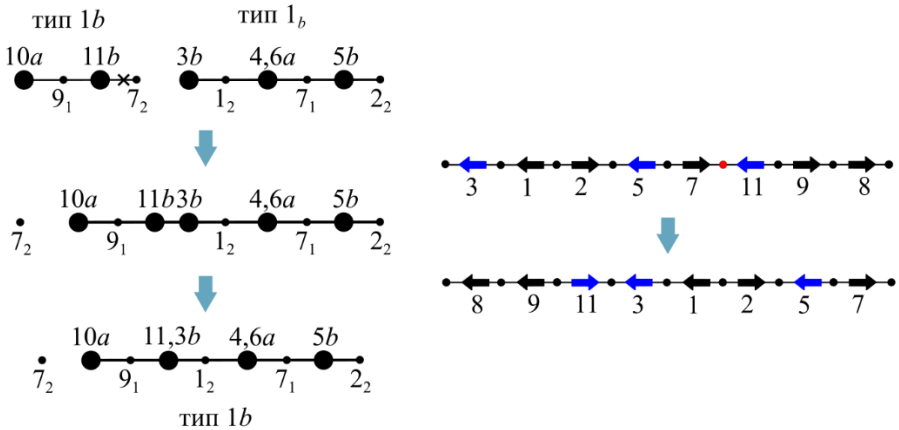


Рисунок 50. Взаимодействие цепей $1b+1_b=1_b$

Расклеим ребро 11*b* – 7_2 и склеим 11*b* с 3*b*. Соответствует переклейке участка рёбер 11, 9, 8.

5) Замкнём цепь в цикл:

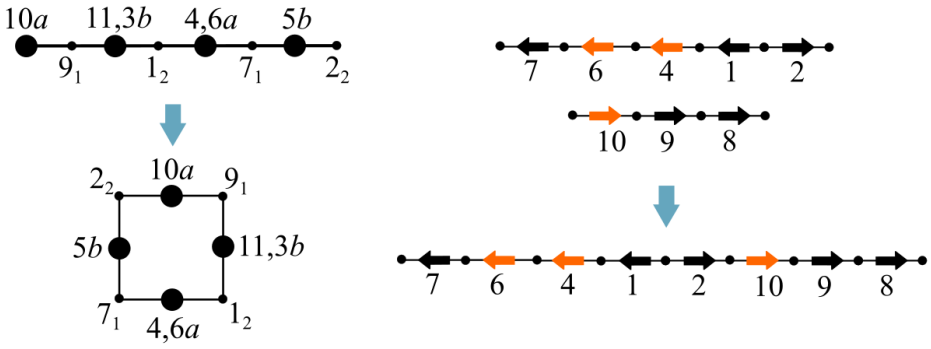


Рисунок 51. Замыкание цепи в цикл
Соответствует склейке двух цепей в одну в структуре *a*

6) Вырежем 2-цикл из 4-цикла:

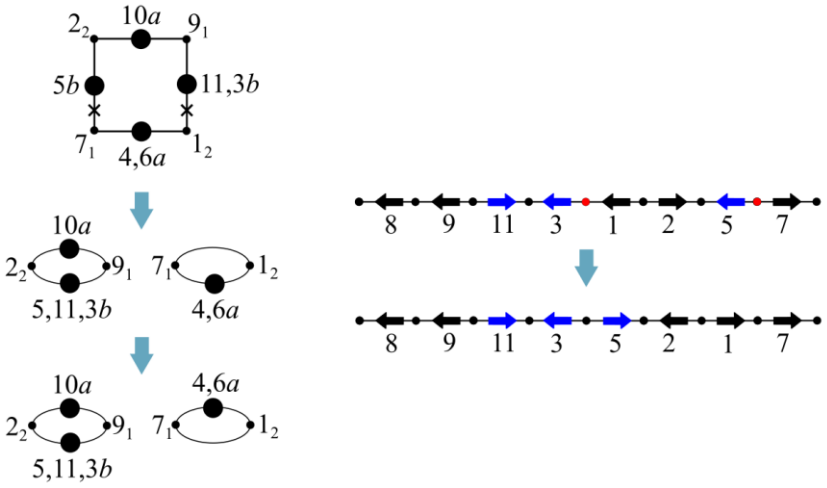


Рисунок 52. Разрезание 4 цикла на два 2-цикла.
Соответствует инверсии участка рёбер 1, 2, 5 в структуре *b*

7) Удалим особые вершины:

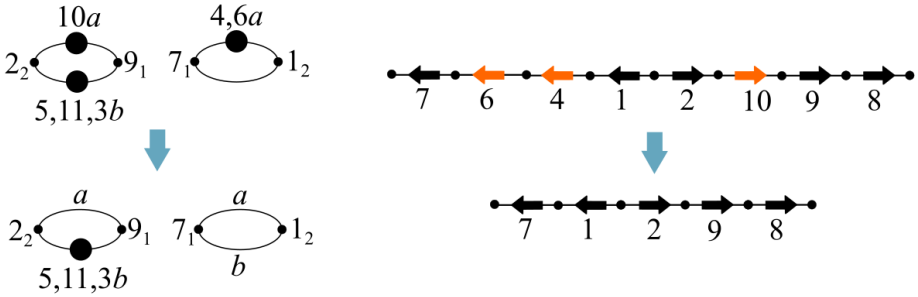


Рисунок 53. Удаление особых *a*-вершин
Соответсвует удалению участков особых рёбер 6, 4 и 10 из структуры *a*.

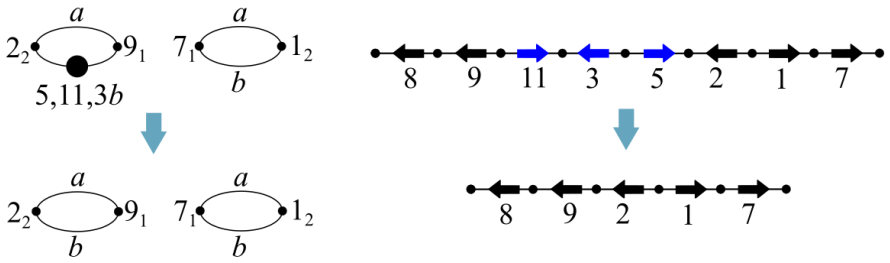


Рисунок 54. Удаление особой b-вершины
Соответствует удалению участка особых рёбер 11, 3, 5 из структуры *b*

Пример 3. Даны структуры, содержащие как линейные, так и кольцевые хромосомы.

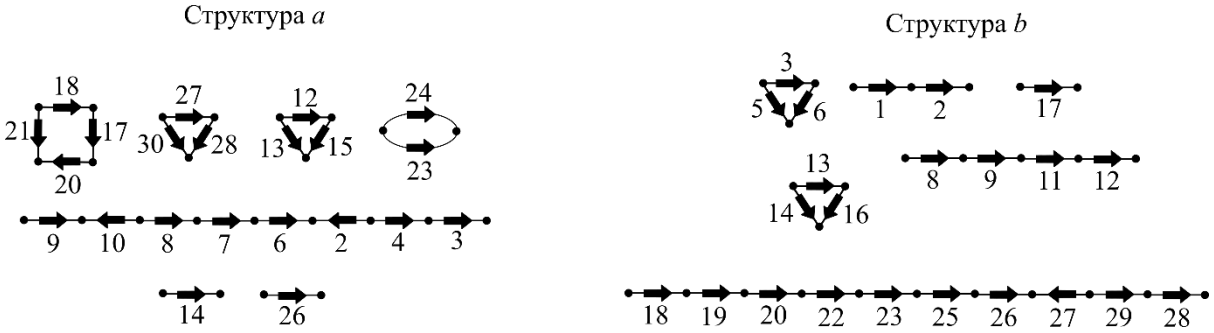


Рисунок 55. Исходные структуры

Общий граф этих структур имеет следующий вид:

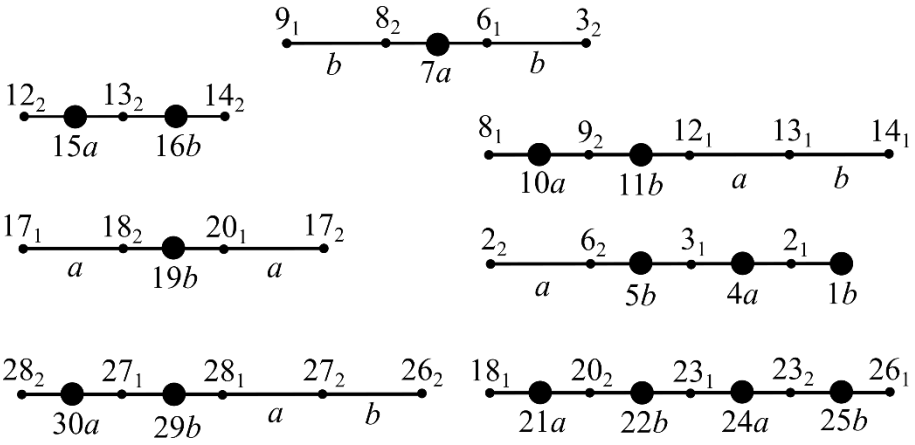


Рисунок 56. Общий граф

Для краткости введём следующие обозначения операций: DP – двойная переклейка, SP – полуторная переклейка, J – склейка, C – разрез, aD – удаление *a*-вершины, bD – удаление *b*-вершины. Циклы и цепи будем описывать в текстовом формате следующим образом: общие вершины описываются именем гена и индексом, указывающим на его начало (1) или конец (2), особые вершины описываются последовательностью входящих в соответствующий блок генов, в конце в зависимости от принадлежности блока одной из структур ставится соответствующая метка (*a* или *b*). Ребра обозначаются нижним подчёркиванием, также сопровождаемым меткой структуры. Тип компоненты обозначается буквами L для цепей и C для циклов.

В таблице 1.1 приведена последовательность операций, приводящих общий граф к финальному виду для циклического варианта цен. На рисунке 57 далее приведена соответствующая последовательность преобразования структуры *a* в *b*.

Таблица 1.1. Последовательность преобразований общего графа из примера 3 к финальному виду, циклический вариант цен

Исходные компоненты	Получившиеся компоненты	Операция
8.1_10a_9.2_11b_12.1_a13.1_b14.1(L)	8.1_10a_9.2_11b_14.1(L) 12.1_a13.1_b12.1(C)	DP
28.2_30a_27.1_29b_28.1_a27.2_b26.2(L)	28.2_30a_27.1_29b_26.2(L) 28.1_a27.2_b28.1(C)	DP
9.1_b8.2_7a_6.1_b3.2(L)	7a_6.1_b3.2(L) 9.1_b8.2_a9.1(C)	SP
7a_6.1_b3.2(L)	7a(L) 6.1_b3.2_a6.1(C)	SP
2.2_a6.2_5b_3.1_4a_2.1_1b(L)	5b_3.1_4a_2.1_1b(L) 2.2_a6.2_b2.2(C)	SP
17.1_a18.2_19b_20.1_a17.2(L)	19b_20.1_a17.2(L) 17.1_a18.2_b17.1(C)	SP
19b_20.1_a17.2(L)	19b(L) 20.1_a17.2_b20.1(C)	SP
28.2_30a_27.1_29b_26.2(L) 18.1_21a_20.2_22b_23.1_24a_23.2_25b_26.1(L) 19b(L) 7a(L)	28.2(L) 26.2(L) 26.1_25b_23.2_24a_23.1_22b_20.2_21307a_27.1_2919b(L) 18.1(L)	SP, SP, SP
12.2_15a_13.2_16b_14.2(L) 8.1_10a_9.2_11b_14.1(L), 5b_3.1_4a_2.1_1b(L)	8.1_10a_9.2_111b_2.1_4a_3.1_165b_13.2_15a_12.2(L) 14.2(L) 14.1(L)	SP, SP
8.1_10a_9.2_111b_2.1_4a_3.1_165b_13.2_15a_12.2(L)	8.1_10a_9.2_111b_2.1_4a_3.1_165b_13.2_15a_12.2_b8.1(C)	C
8.1_10a_9.2_111b_2.1_4a_3.1_165b_13.2_15a_12.2_b8.1(C)	12.2_b8.1_a12.2(C), 9.2_111b_2.1_4a_3.1_165b_13.2_1510a_9.2(C)	DP
26.1_25b_23.2_24a_23.1_22b_20.2_21307a_27.1_2919b(L)	27.1_252919b_23.2_24a_23.1_22b_20.2_21307a_27.1(C), 26.1(L)	SP
27.1_252919b_23.2_24a_23.1_22b_20.2_21307a_27.1(C), 9.2_111b_2.1_4a_3.1_165b_13.2_1510a_9.2(C)	20.2_22b_23.1_24a_23.2_252919111b_2.1_4a_3.1_165b_13.2_151021307a_20.2(C), 9.2_b27.1_a9.2(C)	DP, DP
20.2_22b_23.1_24a_23.2_252919111b_2.1_4a_3.1_165b_13.2_151021307a_20.2(C)	20.2_22252919111b_2.1_4a_3.1_165b_13.2_151021307a_20.2(C), 23.1_24a_23.2_b23.1(C)	DP
20.2_22252919111b_2.1_4a_3.1_165b_13.2_151021307a_20.2(C)	20.2_22252919111165b_13.2_151021307a_20.2(C), 2.1_4a_3.1_b2.1(C)	DP
23.1_24a_23.2_b23.1(C)	23.1_a23.2_b23.1(C)	aD
2.1_4a_3.1_b2.1(C)	2.1_a3.1_b2.1(C)	aD
20.2_22252919111165b_13.2_151021307a_20.2(C)	20.2_b13.2_151021307a_20.2(C)	bD
20.2_b13.2_151021307a_20.2(C)	20.2_b13.2_a20.2(C)	aD

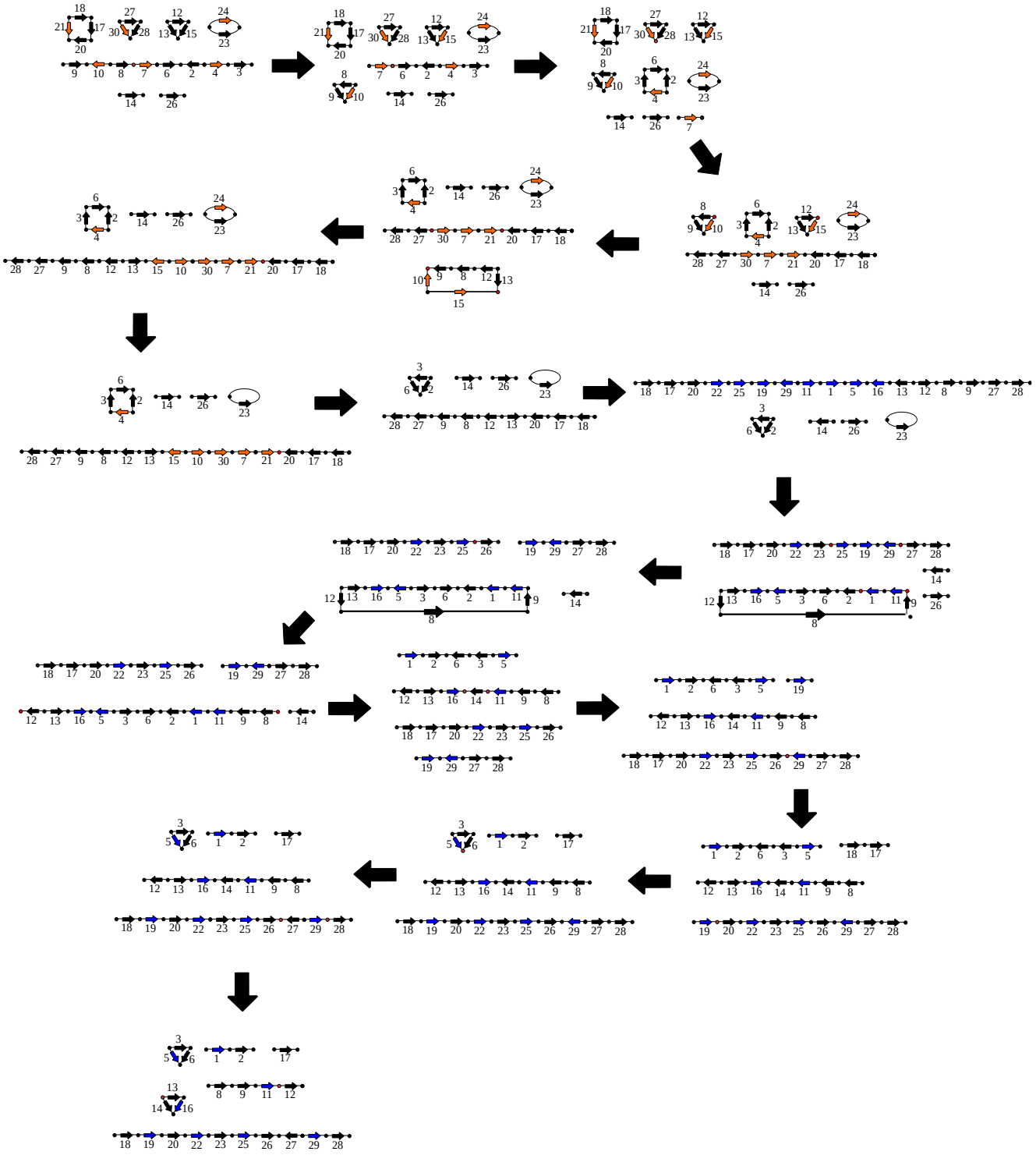


Рисунок 57. Последовательность операций, преобразующих исходную структуру *a* в структуру *b* для циклического варианта цен

Количество операций в полученной последовательности – 23, кратчайшее расстояние равно 24.4.

Построенная для линейного варианта цен последовательность немного отличается от циклического и приведена в таблице 1.2 и на рисунке 58.

Таблица 1.2. Последовательность преобразований общего графа из примера 1 к финальному виду, линейный вариант цен

Исходные компоненты	Получившиеся компоненты	Операция
9.1_b8.2_7a_6.1_b3.2(L)	7a_6.1_b3.2(L), 9.1_b8.2_a9.1(C)	SP
7a_6.1_b3.2(L)	7a(L), 6.1_b3.2_a6.1(C)	SP
8.1_10a_9.2_11b_12.1_a13.1_b14.1(L)	8.1_10a_9.2_11b_12.1(L), 13.1_b14.1_a13.1(C)	SP
2.2_a6.2_5b_3.1_4a_2.1_1b(L)	5b_3.1_4a_2.1_1b(L), 2.2_a6.2_b2.2(C)	SP
17.1_a18.2_19b_20.1_a17.2(L)	19b_20.1_a17.2(L), 17.1_a18.2_b17.1(C)	SP
19b_20.1_a17.2(L)	19b(L), 20.1_a17.2_b20.1(C)	SP
28.2_30a_27.1_29b_28.1_a27.2_b26.2(L)	28.2_30a_27.1_29b_28.1(L), 27.2_b26.2_a27.2(C)	SP
28.2_30a_27.1_29b_28.1(L), 18.1_21a_20.2_22b_23.1_24a_23.2_25b_26.1(L), 19b(L), 7a(L)	28.2(L), 28.1(L), 18.1_21a_20.2_22b_23.1_24a_23.2_252919b_27.1_307a(L), 26.1(L)	SP, SP, SP
12.2_15a_13.2_16b_14.2(L), 8.1_10a_9.2_11b_12.1(L), 5b_3.1_4a_2.1_1b(L)	8.1_10a_9.2_111b_2.1_4a_3.1_165b_13.2_15a_12.2(L), 14.2(L), 12.1(L)	SP, SP
8.1_10a_9.2_111b_2.1_4a_3.1_165b_13.2_15a_12.2(L), 18.1_21a_20.2_22b_23.1_24a_23.2_252919b_27.1_307a(L)	8.1_10a_9.2_111b_2.1_4a_3.1_165b_13.2_15307a_27.1_252919b_23.2_24a_23.1_22b_20.2_21a_18.1(L), 12.2(L)	SP
8.1_10a_9.2_111b_2.1_4a_3.1_165b_13.2_15307a_27.1_252919b_23.2_24a_23.1_22b_20.2_21a_18.1(L)	8.1_10a_9.2_111b_2.1_4a_3.1_165b_13.2_15307a_27.1_252919b_23.2_24a_23.1_22b_20.2_21a_18.1_b8.1(C)	C
8.1_10a_9.2_111b_2.1_4a_3.1_165b_13.2_15307a_27.1_252919b_23.2_24a_23.1_22b_20.2_21a_18.1_b8.1(C)	18.1_b8.1_a18.1(C), 9.2_111b_2.1_4a_3.1_165b_13.2_15307a_27.1_252919b_23.2_24a_23.1_22b_20.2_2110a_9.2(C)	DP
9.2_111b_2.1_4a_3.1_165b_13.2_15307a_27.1_252919b_23.2_24a_23.1_22b_20.2_2110a_9.2(C)	9.2_111165b_13.2_15307a_27.1_252919b_23.2_24a_23.1_22b_20.2_2110a_9.2(C), 2.1_4a_3.1_b2.1(C)	DP
9.2_111165b_13.2_15307a_27.1_252919b_23.2_24a_23.1_22b_20.2_2110a_9.2(C)	9.2_111165252919b_23.2_24a_23.1_22b_20.2_2110a_9.2(C), 13.2_15307a_27.1_b13.2(C)	DP
9.2_111165252919b_23.2_24a_23.1_22b_20.2_2110a_9.2(C)	9.2_11116525291922b_20.2_2110a_9.2(C), 23.2_24a_23.1_b23.2(C)	DP
2.1_4a_3.1_b2.1(C)	2.1_a3.1_b2.1(C)	aD
13.2_15307a_27.1_b13.2(C)	13.2_a27.1_b13.2(C)	aD

23.2_24a_23.1_b23.2(C)	23.2_a23.1_b23.2(C)	aD
9.2_11116525291922b_20.2_2110a_9.2(C)	9.2_b20.2_2110a_9.2(C)	bD
9.2_b20.2_2110a_9.2(C)	9.2_b20.2_a9.2(C)	aD

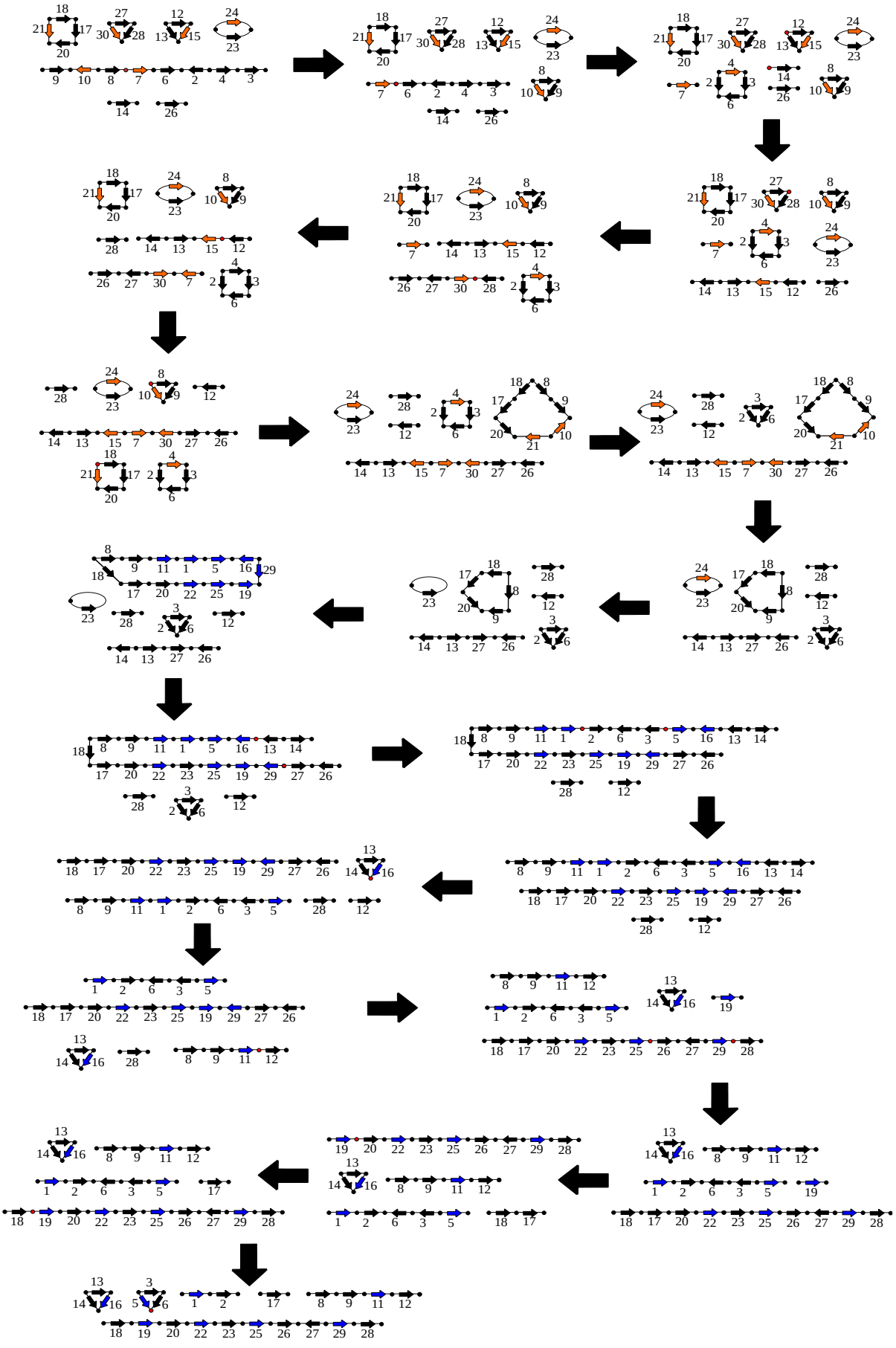


Рисунок 58. Последовательность операций, приводящих исходную структуру *a* к структуре *b* для линейного варианта цен

Количество операций в полученной последовательности – 23, кратчайшее расстояние равно 24.7.

Нетрудно видеть, что в обоих случаях число удалений *b*-вершин сведено к минимуму –одной операции, также для линейного варианта цен чаще применяется полуторная переклейка, а для циклического – двойная.

Больше примеров доступны в [40].

ГЛАВА 2. РЕКОНСТРУКЦИЯ ХРОМОСОМНЫХ СТРУКТУР вдоль ДЕРЕВА: СПЕЦИАЛЬНОЕ РАССТОЯНИЕ и НЕРАВНЫЙ ГЕННЫЙ СОСТАВ

1. Постановка задачи

Дано эволюционное дерево видов (не обязательно бинарное), и в каждом его листе задана своя структура со своим набором генов. Нужно реконструировать структуры во внутренних вершинах дерева так, чтобы минимизировать функционал: сумму по всем ребрам «расстояний» между структурами на концах ребра. Аргументом функционала является *расстановка* структур по всем внутренним вершинам дерева. Поскольку в листьях структуры заданы, любая расстановка приписывает каждой вершине дерева одну структуру. Значение функционала на данной расстановке назовем её *ценой*.

Обозначим множество допустимых имён генов (то есть множество имён, которые встречаются в листовых структурах), как *S*. Структуры во внутренних вершинах могут включать только гены с именами из *S*. Таким образом, реконструкция структур, заданных в листьях, ищется на основе принципа парсимонии.

Разработаны алгоритмы для решения задачи реконструкции, в основе которых лежат различные определения расстояния между хромосомными структурами. Также представлено сведение к булеву линейному программированию (в дальнейшем БЛП) задачи реконструкции хромосомных структур общего вида с *паралогами* вдоль филогенетического дерева с использованием специального расстояния.

Специальным (брейкпоинтовым) расстоянием между структурами a и b (приписанным концам ребра дерева) называется число пар различных краёв генов, которые в одной структуре склеены, а в другой – отсутствуют (один или оба) или не склеены, сложенное с числом генов, присутствующих в одной структуре и отсутствующих в другой. В разделах 2.2 и 2.3 будет рассматриваться задача реконструкции для специального расстояния.

2. В отсутствии паралогов

Равные цены операций разреза, склейки, удаления и вставки. Для данной расстановки и каждой вершины дерева v, и каждой пары различных краёв a_i и b_j генов из S введём переменную x_{vp}, где $p=(a_i,b_j)$, равную 1, если эти края склеены в структуре, соответствующей вершине, и 0 – в противном случае.

Для данной расстановки, каждого гена *k* из *S* каждой вершины v дерева введём переменную y_{kv}, равную 1, если данный ген отсутствует в структуре, соответствующей данной вершине, и 0 – иначе.

Считаем рёбра дерева ориентированными от корня к листьям. Будем называть пару значений переменных $(x_{v(a_i,b_j)}, x_{u(a_i,b_j)})$ на ребре (u,v) разрезом, если значение этой пары равно (1,0) и склейкой, если (0,1). Аналогично, пару переменных (y_{kv}, y_{ku}) будем называть удалением, если значение равно (0,1) и вставкой, если (1,0). Нетрудно заметить, что подсчет специального расстояния можно выразить в терминах переменных *x* и *y*, а именно как сумму разностей соответствующих переменных на концах ребёр. Введём цены операций удаления, вставки, разреза и склейки. Это будет соответствовать появлению коэффициентов при соответствующих разностях. В данном разделе рассматривается случай равных цен операций, будем считать их равными 1.

Алгоритм. В листьях дерева значения всех переменных заданы. Для каждой переменной найдём значения во внутренних вершинах так, чтобы минимизировать указанный функционал, который можно определить, как число ребер, на концах которых значения какой-то переменной отличаются; каждое ребро считается столько раз, сколько имеется таких переменных.

1) Найдем минимум функционала отдельно по каждой переменной. Для этого воспользуемся методом динамического программирования. Приведем алгоритм для нахождения оптимальной разметки для переменных *x*, для *y* процедура аналогична.

Пусть $p = (a_i, b_j)$ – фиксированные края генов. Обозначим оптимальную цену разметки поддерева, висящего на вершине v для пары краёв *p* при условии, что в вершине v значение переменной равно *i*, как C_{vpi}. Поскольку для различных пар краёв *p* значения переменных вычисляются независимо, будем далее опускать индекс *p*. Для листьев значения переменных x_v заданы, цена разметки

$$C_{vi} = \begin{cases} 0, & x_v = i, \\ \infty, & x_v \neq i \end{cases}.$$

Пусть внутренняя вершина v имеет *n* потомков $\eta_1, \ldots, \eta_n$. Для неё значения C_{v0} и C_{v1} будем пересчитывать по следующей формуле:

$$C_{vi} = \min_{(j_1,\ldots,j_n)\in[0,1]^n} \sum_{\eta_k} \left(C_{\eta_k j_k} + |\, j_k - i \,| \right)$$

Для каждого C_{vi} будем хранить набор $(j_1,\ldots,j_{n_v})$, на котором достигнут минимум. Величина оптимальной разметки в результате работы алгоритма будет равна $\min_{i\in[0,1]} C_{ri}$, где r – корень дерева. Сама разметка может быть восстановлена по сохраненным наборам $(j_1,\ldots,j_{n_v})$ рекурсивно, начиная с корня дерева. Полученный набор значений назовем *разметкой* дерева. Данный шаг имеет линейную вычислительную сложность от размера дерева.

Полученная на шаге 1 разметка может указывать на склейку одного края с двумя различными краями, будем называть это *противоречием первого рода.* Также разметка может указывать на склейку края гена, отсутствующего в вершине, назовем это *противоречием второго рода*.

2) Следующий шаг алгоритма состоит в устранении всех противоречий, он также имеет линейную вычислительную сложность от произведения числа переменных на размер дерева; важно, что при этом цена разметки не меняется. Так алгоритм находит решение – разметку, на которой достигается минимум цены. Шаг алгоритма по устранению противоречий состоит в следующем. Упорядочим все края генов из S в каком-то линейном порядке; например, лексикографически. Перебираем вершины дерева в произвольном порядке, а для каждой вершины – края генов согласно этому порядку. Для каждого края удаляем все его склейки в данной вершине, если их чётное число, или иначе оставляем одну склейку с краем, наибольшим в этом порядке. В результате устраняются все противоречия первого рода. Затем перебираем гены в порядке возрастания их имён и для каждого края отсутствующего гена, если он с кем-то склеен (очевидно, он может быть склеен не более чем с одним краем), удаляем эту склейку и считаем этот ген присутствующим в вершине. В результате устранены противоречия второго рода.

Теорема 2. Алгоритм строит разметку с минимальной ценой.

Доказательство. Шаг 1 находит разметку с минимальной ценой. Очевидно, шаг 2 устраняет все противоречия. Покажем, что при этом не увеличивается цена разметки. Противоречие первого рода в вершине v – пара принимающих единичное значение переменных, которые соответствуют двум парам краёв генов (a_i, b_j) и (a_i, c_k) из S.

Такие пары краёв назовем *инцидентными*. Порядок на краях генов индуцирует порядок на инцидентных парах краёв. Шаг алгоритма по устранению противоречий первого рода эквивалентен следующей процедуре. Перебираем инцидентные пары краёв согласно этому порядку. Для каждой пары перебираем вершины дерева и в каждой вершине, если обе соответствующих переменных принимают значения 1, заменяем оба значения на 0. При этом не возникают новые противоречия по другим парам краёв. Таким образом, достаточно показать сохранение цены разметки при выполнении данной процедуры для фиксированной пары $(x_{v(a_i,b_j)}, x_{v(a_i,c_k)})$. Назовём дефектом разметки разность её цены и минимальной цены разметки. Ребро дерева будем считать противоречивым, если в одном его конце разметка (1,1), т.е. $x_{v(a_i,b_j)}=1,\ x_{v(a_i,c_k)}=1$, а в другом конце – (0,0). В листьях дерева противоречий нет по определению. Пусть дефект текущей разметки равен d. Покажем, что всегда выполнены два свойства:

1) d – чётное число (считаем 0 четным)

2) в дереве существует не менее $\frac{d}{2}$ противоречивых рёбер

Действительно, сначала дефект разметки нулевой и указанные свойства выполнены. Рассмотрим очередную вершину v. Если в ней имеется противоречие, т.е. разметка равна (1,1), то по алгоритму она заменяется на (0,0). Рассмотрим вершину u, инцидентную вершине v. Возможны три случая.

Разметка в u, а именно значение $(x_{v(a_i,b_j)}, x_{v(a_i,c_k)})$ равно (0,1) или (1,0). Из определения расстояния видно, что при такой замене цена разметки на ребре (v,u) не изменится, это ребро было и осталось непротиворечивым, рисунок 59.

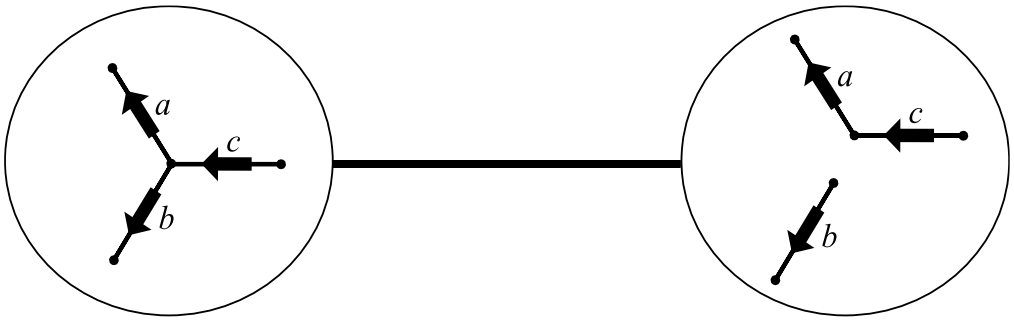
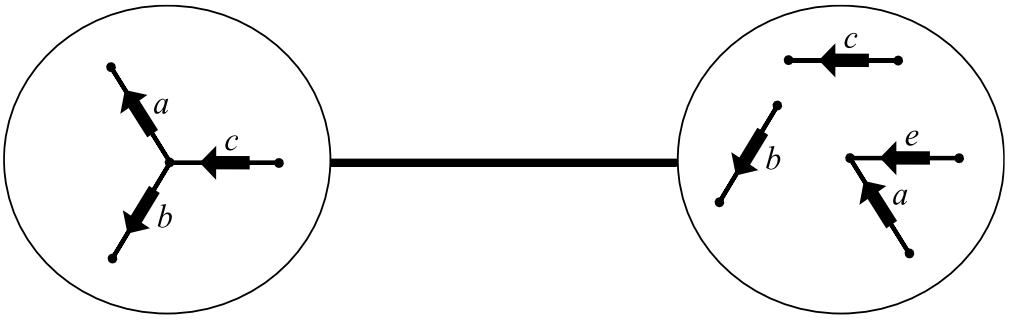


Рисунок 59. Разметка вида (1,0) на конце противоречивого ребра.

В вершине v противоречивая разметка, в инцидентной ей вершине u $x_{u(a_i,b_j)}=0$, $x_{u(a_i,c_k)}=1$

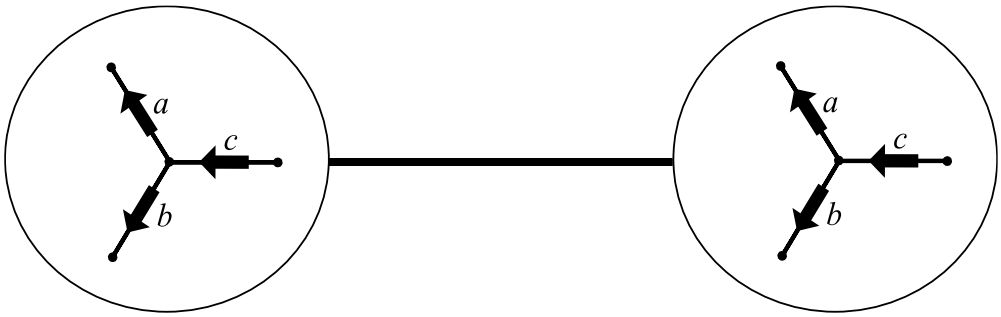
Разметка в u равна (0,0), тогда цена разметки на ребре (v,u) уменьшается на 2, это ребро было противоречивым, а стало непротиворечивым (рисунок 60).



Вершина v Вершина u

Рисунок 60. Разметка вида (0, 0) на конце противоречивого ребра.
В вершине v противоречие, в инцидентной ей вершине u $x_{u(a_i,b_j)}=0$, $x_{u(a_i,c_k)}=0$

Разметка в u равна (1,1), тогда цена разметки на ребре увеличивается на 2, ребро было непротиворечивым, а стало противоречивым, рисунок 61.



Вершина v Вершина u

Рисунок 61. Два противоречия на концах v и u одного ребра

Таким образом, число непротиворечивых рёбер увеличивается (уменьшается) на c, если, и только если, цена разметки увеличивается (уменьшается) на $2c$. В конце процедуры противоречий и противоречивых рёбер не останется, по второму свойству дефект разметки станет нулевым. Поскольку мы всегда меняли значения переменных с

1 на 0, новых противоречий по другим парам переменных не возникнет. Заметим, что описанная процедура устранения противоречий годится и для произвольного графа.

Противоречием второго рода в вершине v является пара принимающих значения 1 переменных, первая из которых – индикатор отсутствия гена, а вторая соответствует склейке края этого гена. Такие пары назовем *инцидентными*. Порядок на краях генов индуцирует порядок на инцидентных парах переменных. Описанная процедура устранения противоречий второго рода эквивалентна следующей процедуре. Перебираем инцидентные пары переменных в этом порядке. Для каждой пары перебираем вершины дерева и в каждой вершине, если обе соответствующих переменных принимают значения 1, заменяем оба значения на 0. Дословно повторяется рассуждение для противоречий первого рода.

Разные цены операций разреза, склейки, удаления и вставки. Для разных цен операций разреза и склейки можно использовать алгоритм из предыдущего раздела. Для нахождения оптимальной разметки нужно в формуле пересчёта использовать цены изменения значений *x* с 0 на 1 (аналогично для *y*):

$$C_{vi} = \min_{(j_1,\ldots,j_n)\in[0,1]^n} \sum_{\eta_k} \left(C_{\eta_k j_k} + \rho(i, j_k)\cdot |\, i - j_k \,| \right),$$

где $\rho(x_1, x_2) = \begin{cases} 0, & x_1 = x_2 \\ c_{01}, & x_1 = 0, x_2 = 1 \\ c_{10}, & x_1 = 1, x_2 = 0 \end{cases}$ и c_{01} – цена склейки, c_{10} – цена разреза.

При устранении противоречий теперь цена разметки может увеличиваться. Однако при соотношении цен, которое кажется биологически возможным (цена склейки двух краев не больше цены их расклейки, цена потери гена не больше цены возникновения) – цена разметки также не увеличивается.

Теорема 3. Если цена изменения значения переменных $0 \to 1$ строго меньше цены изменения $1 \to 0$, то противоречий не возникнет. Таким образом, в этом случае алгоритм является точным.

Доказательство. Пусть d – разность цен этих переходов. Фиксируем инцидентную пару переменных. Индукцией по высоте дерева покажем: для любой (не обязательно минимальной) разметки, если в корне дерева есть противоречие, то при его устранении цена разметки уменьшится не более чем на $2d$; иначе цена не увеличится. Начальный шаг индукции очевиден, опишем индуктивный шаг.

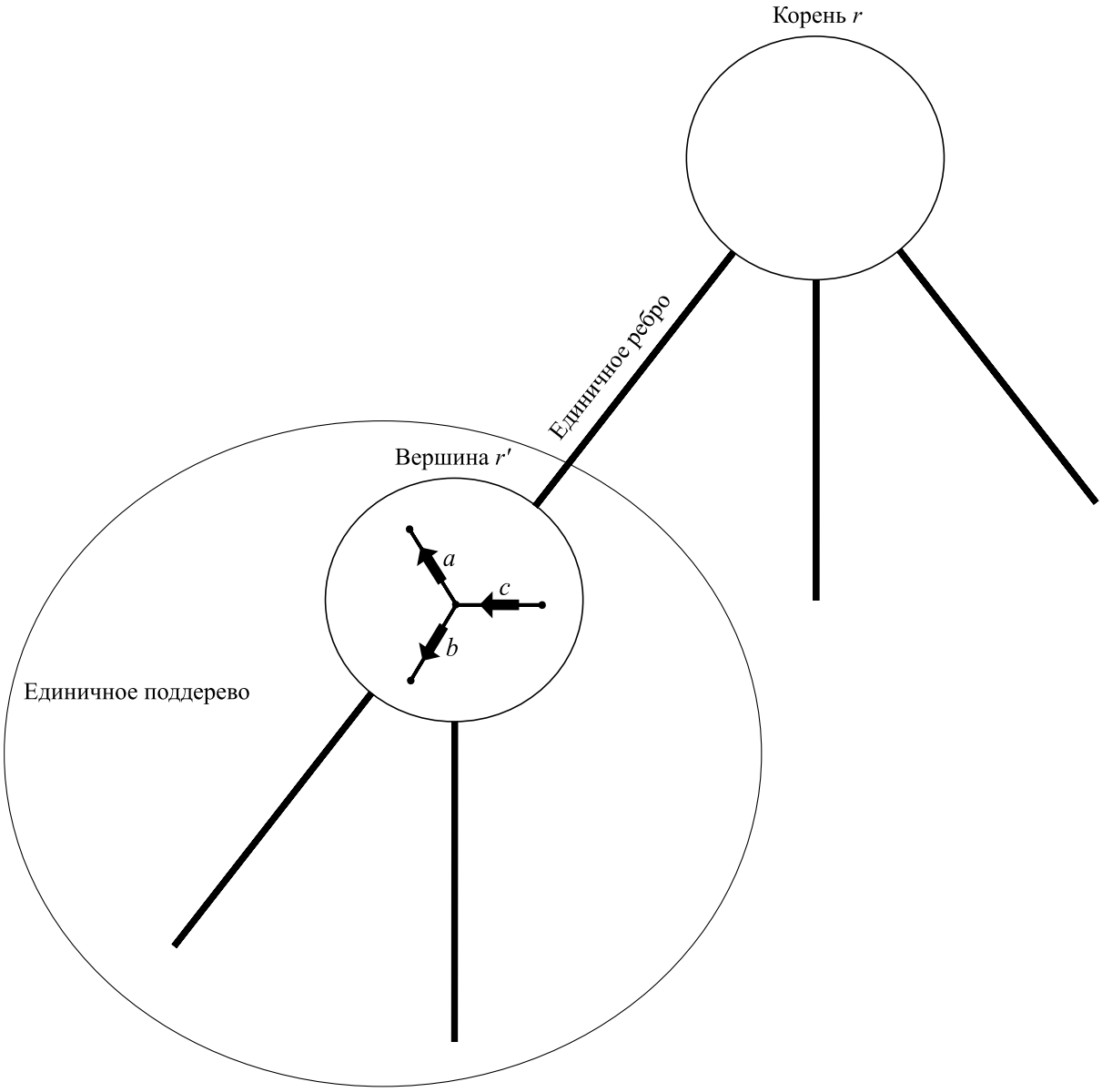


Рисунок 62. Противоречие в потомке r' корня r дерева

Рассмотрим дерево с корнем r и предположим, что для всех поддеревьев, висящих на вершинах, инцидентных корню, утверждение проверено. Если в какой-то инцидентной корню вершине r' разметка равна (1,1) (рисунок 62), будем называть ребро (r,r') и дерево с корнем в r' *единичными*. Переберём случаи разметки (x,y) вершины r. Пусть $(x,y)=(0,0)$. Очевидно, при устранении противоречий цена разметки не увеличится.

Пусть $(x,y)=(1,0)$ или $(x,y)=(0,1)$. На каждом единичном ребре цена разметки увеличится на d, но в каждом единичном дереве цена уменьшится на $2d$. Цена всей разметки не увеличится. Пусть $(x,y)=(1,1)$. На каждом неединичном ребре цена уменьшается как минимум на d, на единичном ребре цена не меняется, но в единичном

дереве цена уменьшается на $2d$. Поскольку у вершины не менее двух сыновей, цена всей разметки уменьшается как минимум $2d$.

Докажем от противного, что противоречия в разметке отсутствуют. Рассмотрим максимальное по включению дерево T с корнем v, помеченным парой (1,1). Если в разметке есть противоречия, то их устранение в T (вне T разметку не меняем) уменьшило бы цену разметки в T на $2d$, а на родительском ребре v цена могла бы увеличиться не более чем на d (так как выше v нет вершины (1,1)). Цена всей разметки уменьшается, а это противоречит ее минимальности.

3. В присутствии паралогов

Рассмотрим задачу реконструкции хромосомных структур, которые допускают присутствие паралогов. Каждому гену с именем k, которому отвечают ортологи в листьях, припишем натуральное число n_k – максимальное допустимое число паралогов для данного имени. Паралоги гена с именем k будем обозначать как $k.1,\ldots,k.n_k$ и называть k-паралогами, таким образом, у паралога $k.i$ имя k и номер i. Каждой вершине дерева отвечает структура, которая может включать гены, ортологичные листовым генам.

Началом ребра дерева будем называть ближайший к корню конец данного ребра. Второй конец ребра будем называть *концом*. Нумерация паралогов задается для каждого ребра, каждый ген k задан в обоих концах i и j любого ребра посредством биекции f между двумя подмножествами множества всех k-паралогов в i и j. Если $i=f(j)$, мы полагаем, что паралог $k.i$, соответствующий концу ребра, происходит от паралога $k.j$, расположенного в начале ребра. Если для i не существует соответствующего j, будем говорить, что ген $k.i$ возникает в конце ребра. Каждому паралогу приписывается имя вида $k.l$, где $1 \leqslant l \leqslant n_k$. Если зафиксировать структуры и нумерацию вдоль всего дерева, можно переименовать все паралоги, начиная с корня, так, чтобы на каждом ребре и для каждого паралога $k.i$ (для которого существует соответствующий $f(k.i)$) было выполнено $k.i=f(k.i)$. Заметим, что имена генов в корне дерева задаются случайно, имена генов в листьях обычно перенумеровываются соответствующим образом. Также для удобства будем считать, что в каждой вершине сначала «виртуально присутствует» каждый паралог каждого гена.

Опишем решение задачи с помощью метода булева линейного программирования. Введем следующие переменные.

1) Для каждого ребра *e*, для каждой упорядоченной пары *k.i*, *k.j* паралогов гена *k* определим переменную z_{kije}, равную 1 в случае, если *k.i=f(k.j)*, где *k.i* расположен в конце ребра, *k.j* – в начале, и 0 в противном случае. Для удобства будем рассматривать *i* и *j* лежащими в диапазоне от 1 до *n.k*. Для обеспечения биективного соответствия между паралогами, вводим следующие линейные ограничения между переменными *z*. Для фиксированных *k.i* и *e*, $\sum_j z_{kije} = 1$, аналогично при фиксированных *k, j, e* верно равенство $\sum_i z_{kije} = 1$. Эта переменная определяет нумерацию паралогов.

2) Для каждой вершины ν дерева и каждой неупорядоченной пары различных краёв генов *k.l* введём переменную $x_{kl\nu}$, равную 1, если эти края отождествлены в структуре, соответствующей данной вершине, и 0 в противном случае. Каждый край может быть отождествлен не более, чем с одним другим краем. Отсюда получаем следующие ограничения: для фиксированных ν, *k* имеем $\sum_l x_{kl\nu} \leqslant 1$, для фиксированных ν, *l* имеем $\sum_k x_{kl\nu} \leqslant 1$. Переменная определяет структуру в вершине, т.е. расположение структур вдоль дерева.

3) Для каждой вершины ν дерева и каждого гена *k* введем переменную $y_{k\nu}$, равную 1, если ген *k* отсутствует в структуре, соответствующей вершине ν и 0 иначе. Переменные $x_{kl\nu}$ и $y_{k\nu}$ заданы в листьях. Концы отсутствующих генов не отождествлены, это условие можно выразить соотношением $x_{ij\nu} \leqslant 1 - y_k \nu$, где *i* или *j* – край гена *k*. Для каждого гена *k* и каждого его края *i* можно записать следующее соотношение $\sum_{i \neq j} x_{ij\nu} \leqslant 1 - y_{k\nu}$, где *j* пробегает все края генов в вершине ν. Переменная *y* определяет генный состав всех вершин.

4) Будем называть две пары краёв генов *k,l* и *m,n* (принадлежащих, соответственно, началу и концу одного ребра) *похожими*, если обе пары либо концы генов, либо начала генов, либо начало и конец гена (в последнем случае концы должны принадлежать одному или различным генам в обеих парах). Также, гены i_1 и j_1 с концами *k* и *m* соответственно, а также гены i_2 и j_2 с концами *l* и *n* (или, наоборот, гены с концами *l* и *m* и гены с концами *k* и *n*) должны быть одноименными. Для каждого ребра *e* и для двух похожих пар *k,l* и *m,n* краёв генов введём переменную s_{klmne}

, равную 1, если эти края принадлежат соответствующим (соответствие задано переменными z) паралогам и отождествлены в одном конце ребра *e* и не отождествлены в другом. В противном случае она равна 0. Условия на переменные *s* выражаются следующими неравенствами (для случая двух начал или двух концов генов-паралогов из одной группы): если *e=(u,v)*, то

$s_{klmne} \geqslant x_{klv} - x_{mnu} - (1 - z_{i_1 j_1 e}) - (1 - z_{i_2 j_2 e})$, $s_{klmne} \geqslant x_{mnu} - x_{klv} - (1 - z_{i_1 j_1 e}) - (1 - z_{i_2 j_2 e})$,

$s_{klmne} \geqslant x_{klv} - x_{mnu} - (1 - z_{i_1 j_2 e}) - (1 - z_{i_2 j_1 e})$, $s_{klmne} \geqslant x_{mnu} - x_{klv} - (1 - z_{i_1 j_2 e}) - (1 - z_{i_2 j_1 e})$.

5) Для каждого ребра *e=(u,v)* и двух генов *i* (в конце ребра) и *j* (в начале ребра) введём переменную s_{ije}, равную 1, если гены соответствуют друг другу в терминах переменных *z* и при этом один ген присутствует в соответствующей вершине, а другой – отсутствует.

6) В качестве минимизируемого функционала рассмотрим $\sum s_{klmne} + \sum s_{ije}$. Любое решение задачи минимизации данного функционала при рассмотренных выше ограничениях определяет расстановку структур с минимальным общим весом и соответствующей нумерацией всех паралогов.

Неравные цены операций разреза, склейки, удаления и вставки. Решим задачу с помощью булева линейного программирования. Для этого модифицируем алгоритм сведения из предыдущего раздела. А именно, вместо переменных s_{klmne} введём переменные $s1_{klmne}$ и $s2_{klmne}$. На эти переменные наложим следующие ограничения (для случая двух начал или двух концов генов-паралогов из одной группы):

$s1_{klmne} \geqslant x_{klv} - x_{mnu} - (1 - z_{i_1 j_1 e}) - (1 - z_{i_2 j_2 e})$, $s2_{klmne} \geqslant x_{mnu} - x_{klv} - (1 - z_{i_1 j_1 e}) - (1 - z_{i_2 j_2 e})$,

$s1_{klmne} \geqslant x_{klv} - x_{mnu} - (1 - z_{i_1 j_2 e}) - (1 - z_{i_2 j_1 e})$, $s2_{klmne} \geqslant x_{mnu} - x_{klv} - (1 - z_{i_1 j_2 e}) - (1 - z_{i_2 j_1 e})$.

Аналогично вместо переменных s_{ije} введём переменные $s1_{ije}$ и $s2_{ije}$. Для учёта цен операций теперь можно заменить в минимизируемой функции переменные *s* на сумму переменных *s*1 и *s*2 и умножить их на соответствующие коэффициенты.

Очевидно, набор значений переменных, удовлетворяющий ограничениям, взаимно однозначно соответствует расстановке структур по внутренним вершинам дерева, а минимизируемая функция равна сумме по ребрам дерева специальных расстояний между структурами на концах ребра. Из этого следует корректность описанного сведения.

4. Тестирование на искусственных примерах

Рассмотрим политомическое дерево ((a,b),(c,d),(e,f)). Данные структуры в листьях приведены в таблице 2.1.

Таблица 2.1. Структуры в листьях дерева
Звездочка перед именем гена указывает на его расположение гена на комплементарной цепи

Имя листа	Структура
a	1 (C); 2 3 (C)
b	1 (C); 2 *3 (C)
c	2 (C); 1 3 (C)
d	2 (C); 1 *3 (C)
e	3 (C); 1 2 (C)
f	3 (C); 1 *2 (C)

Рассмотрим результаты работы алгоритмов из раздела 2.2 для двух наборов цен склейки и расклейки для специального расстояния.

На рисунке 63 приведены предковые структуры для случая равных цен склейки и расклейки. Заметим, что в структурах преобладают линейные хромосомы, при том, что в листьях заданы циклические.

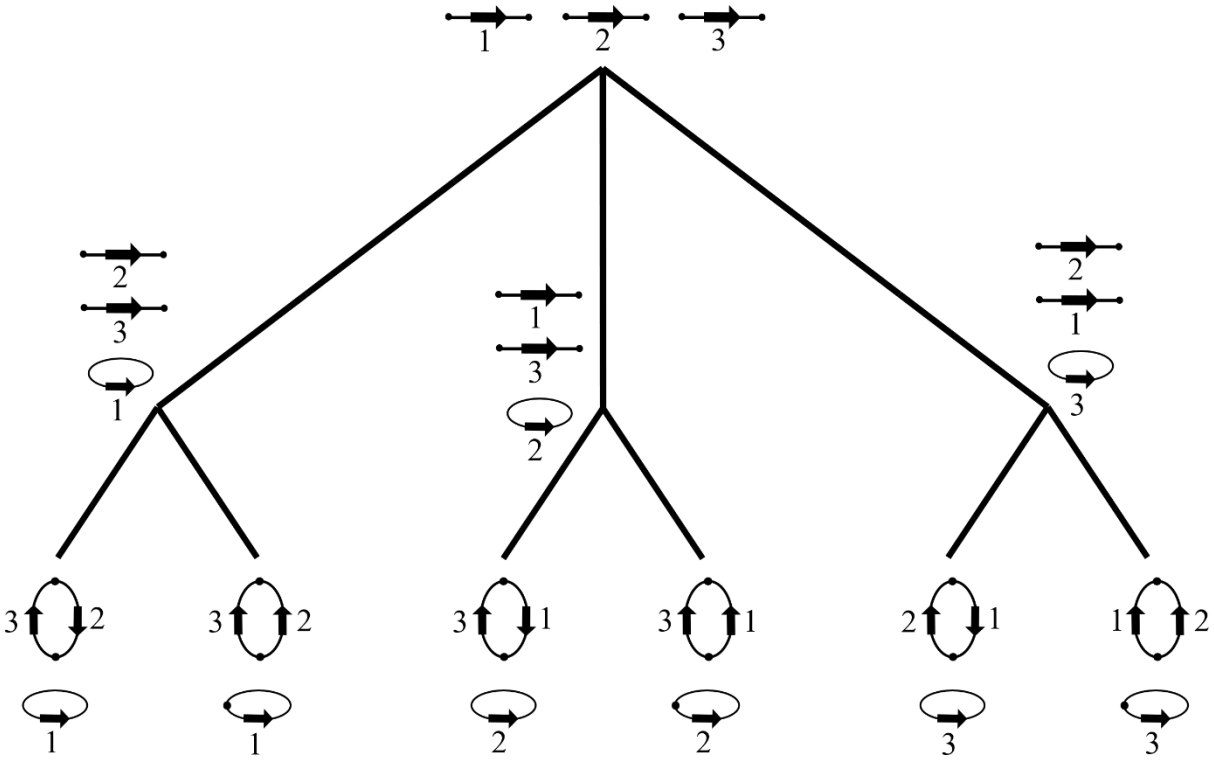


Рисунок 63. Реконструкция хромосомных структур для специального расстояния и равных цен операций

На рисунке 64 приведён результат работы алгоритма для цен расклейки и склейки равных 2 и 3 соответственно. Данная реконструкция является более правдоподобной, так предковые структуры также получаются циклическими.

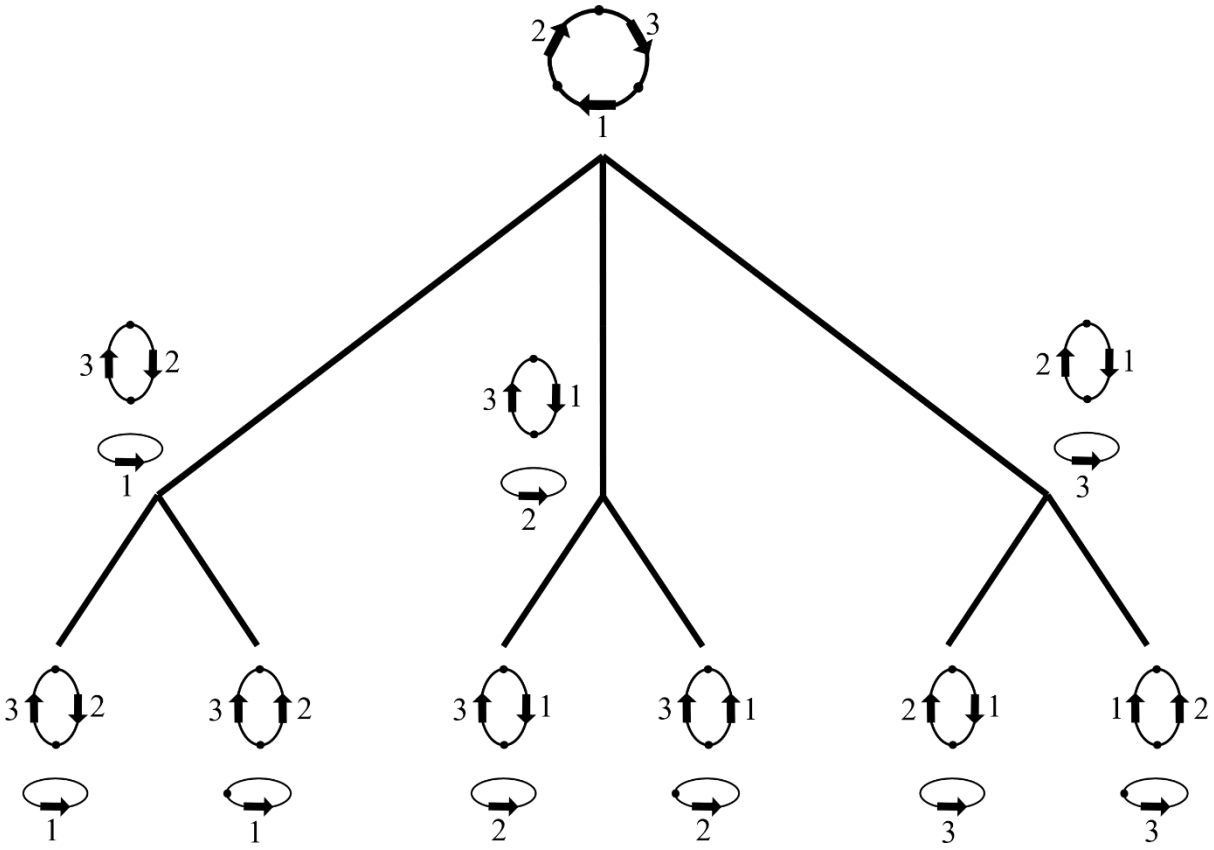


Рисунок 64. Реконструкция хромосомных структур для специального расстояния и различных вариантов цен

ГЛАВА 3. ПРЕОБРАЗОВАНИЕ и РЕКОНСТРУКЦИЯ ХРОМОСОМНЫХ СТРУКТУР, СОГЛАСОВАНИЕ КОНТИГОВ СВЕДЕНИЕМ к ЦЕЛОЧИСЛЕННОМУ ЛИНЕЙНОМУ ПРОГРАММИРОВАНИЮ: с ПАРАЛОГАМИ и РАВНЫМИ ЦЕНАМИ

Везде в этой главе рассматривается биологическое расстояние и единичные цены операций.

Идентификация паралогов гена с именем *k* означает, что им присваиваются новые уникальные имена *k.1, k.2* и так далее. Будем называть множество таких имён *нумерацией* паралогов, сами имена будем называть *полными* именами паралогов гена *k*. С помощью такой нумерации становится возможно задавать частичную биекцию между двумя множествами паралогов. Биекция частичная, так как паралоги могут исчезать и появляться в ходе эволюции структуры *a* в *b*. Заметим, что форма общего графа зависит от выбора нумераций всех паралогов всех генов, также от него зависит приведение графа к финальному виду.

1. Преобразование циклических структур сведением к ЦЛП с линейным числом переменных и ограничений

Рассмотрим сначала структуры, состоящие из кольцевых хромосом. Общий граф таких структур состоит только из циклов.

Следующая теорема следует из [77].

Теорема 4. Длина кратчайшей последовательности от *a* к *b* равна $B+S_1-S_2$, где *B* – число особых вершин в графе *a*+*b*, S_1 – сумма целых частей половин длин максимальных по включению участков из обычных рёбер, S_2 – число циклов, состоящих из обычных рёбер.

Рассмотрим сведение к ЦЛП, позволяющее вычислить слагаемые *B*, S_1, S_2. Будем называть пару отождествлённых концов рёбер в хромосомной структуре склейкой.

Пусть даны две структуры *a* и *b*. Рассмотрим булеву переменную z_{kij}, равную 1, если паралог *i* гена *k* в структуре *a* отвечает паралогу *j* того же гена *k* в структуре *b*; иначе равную 0. Заметим, что значения переменных z_{kij} задают частичную биекцию паралогов в структурах *a* и *b*.

1) Вычисление *B*. Опишем каждую склейку *s* в структуре *a* булевой переменной x_{as}; она равна 1 если эта склейка расположена на границе *a*-блока, иначе она равна 0. Аналогично введём x_{bs} для структуры *b*. Ограничения таковы, что если край паралога i_1 гена *k* отождествляется с краем паралога i_2 гена *l* в *s*, то $x_{as} \geq \sum_j z_{ki_1 j} - \sum_j z_{li_2 j}$ и $x_{as} \geq \sum_j z_{li_2 j} - \sum_j z_{ki_1 j}$; аналогичные ограничения верны для склеек из *b*. Эти ограничения означают, что если $\sum_j z_{ki_1 j}$ и $\sum_j z_{ki_2 j}$ не равны, то есть края склейки *s* принадлежат общему и особому генам, то $x_{as} = 1$. Запишем первую часть минимизируемой функции в виде $F = 0.5 \cdot \sum_s \left(x_{as} + x_{bs} \right) + \ldots$, где остальные слагаемые будут определены позже. Если $\sum_j z_{ki_1 j}$ и $\sum_j z_{ki_2 j}$ равны, то $x_{as} = 0$, так как x_{as} и x_{bs} – слагаемые в *F* с положительным коэффициентом.

Теперь вычислим количество петель, т.е. циклических блоков (будем далее называть их *особыми хромосомами*). Для этого введём для каждой хромосомы *h* в исходных структурах булеву переменную o_h, которая должна быть равна 1, если *h* особая, и 0 иначе. Накладываем ограничения: $o_h \geq 1 - \sum_{k.i} \sum_j z_{kij}$, если *h* лежит в *a* или $o_h \geq 1 - \sum_{k.j} \sum_i z_{kij}$, если *h* лежит в *b*, где первая сумма берётся по всем генам из *h*. Если *h* особая, то двойная сумма равна 0 и o_h=1, а иначе равенство o_h=0 следует из того, что переменные o_h входят в минимизируемую функцию *F* с положительными коэффициентами: определяем её как $F = \ldots + \sum_h o_h + \ldots$, где другие слагаемые приведены выше и ниже.

Каждому нециклическому блоку соответствуют две граничные переменные *x*, а каждому циклическому – одна переменная *o*, что обеспечивает корректность вычисления величины *B*.

2) Вычисление s_1. Каждую склейку *s* в структуре *a* опишем булевой переменной y_{as}; она равна 0, если эта склейка расположена на границе или внутри блока. Аналогично введём переменную y_{bs}. Для склеек общих генов переменные y_{as} и y_{bs} принимают чередующиеся значения 0 и 1 внутри каждого участка, состоящего из обычных рёбер, это чередование начинается с 0 на одной из границ участка. Опишем

ограничения. Две склейки будем называть *потенциально соседними* (как рёбра в общем графе), если они принадлежат различным структурам и обе содержат один и тот же край паралога одного гена. Следующие ограничения накладываются на пару s_1 (из a) и s_2 (из b) потенциально соседних склеек: $y_{as_1} \leqslant 4 - z_{kij} - \sum_j z_{k_1 i_1 j} - \sum_j z_{k_2 j i_2} - y_{bs_2}$ и $y_{bs_2} \geqslant z_{kij} + \sum_j z_{k_1 i_1 j} + \sum_j z_{k_2 j i_2} - 2 - y_{as_1}$, где край паралога i гена k и паралога i_1 гена k_1 являются отождествлёнными в s_1, а также края паралога j гена k и паралога i_2 гена k_2 в s_2. Эти неравенства означают, что значения y_{as} и y_{bs} чередуются на каждом участке обычных рёбер. Продолжим определять минимизируемый функционал: $F = \ldots + \sum_s \left(y_{as} + y_{bs} \right) + \ldots$, где остальные слагаемые будут определены ниже. На границах участков, состоящих из нечётного числа обычных рёбер, а также внутри или на границе блоков, переменные y_{as} и y_{bs}. равны 0, так как они входят в F с положительным коэффициентом. Таким образом, сумма переменных y_{as} и y_{bs} равна s_1.

3) Вычисление s_2. Воспользуемся идеей о подсчёте числа циклов из [9]. Каждую склейку s в структурах a и b опишем целочисленной переменной u_s, ограниченной неравенством $u_s \leqslant m_s$, где m_s принимает значение от 1 до общего числа склеек в a и b. Введём также булеву переменную p_s, ограниченную неравенством $p_s m_s \leqslant u_s$. Данная переменная указывает на то, достигает ли u_s максимально возможное значение m_s. Продолжим определение: $F = \ldots - \sum_s p_s$, где остальные слагаемые были определены ранее. Переменные p_s входят в F с отрицательными коэффициентами, поэтому, если u_s равно m_s, то $p_s = 1$.

Для каждой склейки s, содержащей паралог i гена k в a введём следующее ограничение: $u_s \leqslant m_s \sum_j z_{kij}$. Для b введём аналогичные неравенства. Они обеспечивают равенство $u_s = 0$, если s лежит на границе или внутри блока.

Для двух потенциально соседних склеек s_1 из a и s_2 из b введём следующие ограничения: $u_{s_1} \leqslant u_{s_2} + m_{s_1}(1 - z_{kij})$ и $u_{s_2} \leqslant u_{s_1} + m_{s_2}(1 - z_{kij})$, где s_1 содержит паралог i гена

k и s_2 содержит паралог j гена k. Эти неравенства обеспечивают равенство $u_{s_1} = u_{s_2}$ для двух соседних рёбер s_1 и s_2 в общем графе. Соответственно, все переменные u_s принимают одно значение, причем ровно одна из них достигает максимального значения на каждом цикле, состоящем из обычных рёбер. Для циклов, содержащих блоки, эти переменные равны 0 и ни одна из них не достигает своего максимума. Таким образом, количество переменных u_s, достигающих своего максимума (и равное сумме переменных p_s) равно s_2.

2. Задача преобразования: произвольные структуры – сведением к квадратичному ЦЛП

Пусть a и b данные хромосомные структуры. Введём произвольную нумерацию генов с паралогами и без, полученные структуры обозначим a' и b'. Эту нумерацию назовём исходной. Далее везде будет подразумеваться наличие нумерации у структур.

Введём переменные z_{kij}, определенные в разделе 3.1. Для них верны следующие ограничения: $\sum_i z_{kij} \leqslant 1$ для любых фиксированных k и j и аналогично для суммы по индексу j. Также может быть введено нижнее ограничение на значение суммы $\sum_{i,j} z_{kij} \geqslant 1$ для некоторых генов k.

Будем называть ген *общим*, если он становится общим после того, как паралоги в b' перенумеруются в соответствии со значениями переменных z_{kij}. А именно, если $z_{kij} = 1$, ген $k.j$ в b' перенумеруется в ген $k.i$. После этого все гены, не участвующие в биекции, перенумеруются так, чтобы сохранилась полная нумерация структур. Ген будем называть *особым*, если он становится особым после перенумерации. Полученные после перенумерации структуры будем обозначать как $a'(z)$ и $b'(z)$. Напомним, кольцевые хромосомы, состоящие только из особых генов, называются *особыми*. Кольцевые хромосомы, состоящие из более одного гена, будем называть m-кольцевыми.

Для каждой кольцевой хромосомы d из a' определим $o(d,a) = \left(\sum_{k.i \in d, k.j \in b'} z_{kij} \right) / n_d$,

где n_d – число генов в d. Для линейной хромосомы d положим $o(d)=1$; $0 \leqslant o(d) \leqslant 1$. Можно показать, что d является особой тогда и только тогда, когда $o(d,a)=0$. Действительно, если каждый ген в d особый, значит, для любого паралога $k.i \in d$ не существует

соответствующего ему $k.j \in a'$, то есть $z_{kij} = 0$ для всех $k.i$ и $k.j$. Значит, вся сумма равна 0. В обратную сторону аналогично.

Значение $o(d,a)$ показывает, какая часть генов из d участвует в z-биекции с генами из b'. Аналогично определяется $o(d,b)$ для $d \in b'$.

Сделаем одинаковым генный состав в структурах $a'(z)$ и $b'(z)$ путем добавления в структуру $a'(z)$ особых генов из структуры $b'(z)$, не входящих в особые $b'(z)$-хромосомы, аналогично расширим $b'(z)$. Из добавленных генов составляются кольцевые хромосомы. Полученные хромосомы, а также их гены и склейки этих генов будем называть *новыми*. Новые склейки опишем переменной t, определение которой будет дано ниже. Таким образом построенные структуры будем обозначать $a^-(z,t)$ и $b^-(z,t)$, структуры без особых хромосом – $a''(z,t)$ и $b''(z,t)$. Обозначим общий граф $G'(z,t)=a''(z,t)+b''(z,t)$ и функцию $\Phi(z,t) = (C_0 + n + s_a + s_b) - C_1 - 0.5C_2$, где C_0 – общее количество особых хромосом в $a'(z,t)$ и $b'(z,t)$, C_1 – количество циклов в G', C_2 – количество чётных путей в G', n – количество общих генов в $a'(z,t)$ и $b'(z,t)$, s_a и s_b – количества новых генов в $a^-(z,t)$ и $b^-(z,t)$. Доказывается, что расстояние между $a^-(z,t)$ и $b^-(z,t)$ равно $\Phi(z,t)$ для любых z и t, $t_0 = t_0(z)$ – значение, на котором достигается минимум Φ.

Нетрудно проверить [13], что расстояние между $a^-(z,t_0)$ и $b^-(z,t_0)$ равно расстоянию между $a'(z)$ и $b'(z)$ для любого z. В [13] нет переменной z, так как не рассматриваются паралоги, также не используется переменная t. Поэтому решение задачи о расстоянии подразумевает нахождение $\min_z \min_t \Phi(z,t)$. По определению, новые склейки отвечают новым ребрам в $G'(z)$, остальные ребра называются *старыми*.

Определим переменную t, которая описывает новые склейки. Для каждой пары $s=(g,g')$ различных краёв генов из a' определим булеву переменную t_{bs}, показывающую, образуют ли края g и g' новую склейку в $b''(z,t)$. Для неё выполнены следующие ограничения: $t_{bs} \leqslant 1 - \sum_j z_{kij}$, $t_{bs} \leqslant n_g \cdot o(d_g)$, $\sum_{g'} t_{bgg'} \leqslant 1$, $\sum_{g'} \geqslant o(d_g) - \sum_j z_{kij}$, где $k.i$ – ген с краем g, d_g – хромосома, содержащая $k.i$, n_g – количество генов в d_g. Аналогично определяется t_{as} и соответствующие ограничения для b'.

Пункты 1-3 ниже описывают слагаемые функции Φ в терминах переменных ЦЛП. В итоге минимизируемая функция будет равна

$F = \left(\sum_d n_d + \sum_d (1-n_d) o_d - \sum_{k,i,j} z_{kij} \right) - \sum_s p_s - 0.5 \left(\sum_g r_g - \sum_g l_g \right)$, где d пробегает все хромосомы a' и b' и n_d – количество хромосом в d. Слагаемое $\sum_d n_d$ является константой и не влияет на результат минимизации. Переменные o_d, p_s, r_p, l_p и их линейные ограничения будут определены ниже. Задача минимизации принимает вид $\min_{z,t} \Phi(z,t) = \min F(o,z,p,r,l)$

1) Опишем количество C_1 циклов в графе G'. Для этого пронумеруем все склейки (g,g') из a' и b', начиная с единицы, обозначим как m_s номер склейки s. Для каждой склейки s введём целочисленную переменную u_s с ограничением $0 \leqslant u_s \leqslant m_s$. Потребуем равенства 0 всех u_s для s, принадлежащих особым хромосомам d в a'(z). Это можно выразить следующим неравенством $u_s \leqslant m_s \sum_{k.i \in d} \sum_j z_{kij}$ для любой кольцевой хромосомы d. Аналогичные ограничения вводятся для b'. Будем называть пару краёв генов краями одного типа, если они либо оба 5'-концы, либо 3' концы и принадлежат паралогам в разных структурах. Будем требовать $u_s = 0$ для всех таких склеек s в a', что один из входящих в неё краёв принадлежит общему гену и является крайним в цепи в G'. Пусть $g \in s$ является краем гена $k.i$, принадлежащего a'. Для всех генов $k.j$ из b' с краем такого же типа, как g, являющихся крайними в цепи из b', выполнены ограничения $u_s \leqslant m_s (1 - z_{kij})$. Аналогичные ограничения вводятся для b'.

Далее, будем требовать $u_s = 0$ для любой склейки $s \in a'$ такой, что один из входящих в неё краёв принадлежит особому a-гену и не является крайним в цепи, но является концом крайнего ребра в ней. Для каждого края $g_1 \in a'$, являющегося крайним в цепи из a', выполнено неравенство $u_s \leqslant m_s (1 - t_{g_1 g})$, где $g \in s$. Аналогичные ограничения для b'.

Потребуем, чтобы u_s были постоянны на всех рёбрах цикла или цепи в G'. А именно, для каждой пары склеек $s_1 = (g, g_1)$ и $s_2 = (g', g_2)$ в a' и b' соответственно, где g и g' одного типа, потребуем выполнение следующих ограничений: $u_{s_1} \leqslant u_{s_2} + m_{s_1} (1 - z_{kij})$, $u_{s_2} \leqslant u_{s_1} + m_{s_2} (1 - z_{kij})$, где $k.i$ и $k.j'$ – гены с краями g и g'. Эти ограничения гарантируют

равенство $u_{s_1} = u_{s_2}$ для двух соседних рёбер s_1 и s_2 в G', являющихся старыми рёбрами. Для каждой пары склеек $s_1 = (g_1, g_2)$ и $s_2 = (g_3, g_4)$ краёв, принадлежащих a' или b', потребуем $u_{s_1} \leqslant u_{s_2} + m_{s_1}(1 - t_{g_2 g_3})$ и $u_{s_2} \leqslant u_{s_1} + m_{s_2}(1 - t_{g_2 g_3})$. Данные ограничения гарантируют выполнение равенства $u_{s_1} = u_{s_2}$ для двух старых рёбер из G', соединённых ровно одним новым.

Для каждой склейки s определим булеву переменную p_s, показывающую, достигает ли u_s своего максимального значения m_s в точке минимума функции F. А именно, $p_s \cdot m_s \leqslant u_s$. Если $u_s \leqslant m_s$, то $p_s = 0$, иначе p_s может принимать любое значение. Но так как переменные p_s являются слагаемыми с отрицательными коэффициентами в F, $p_s = 1$.

Так как u_s принимает постоянное значение на всех рёбрах одного цикла и все максимальные значения различны, существует только одно ребро, на котором $u_s = m_s$ и только на одном ребре $p_s = 1$. На рёбрах цепей в силу наложенных ограничений $u_s = 0$ и ни на каком ребре не достигается максимальное значение. Учитывая, что каждый цикл содержит хотя бы одно старое ребро, получаем формулу числа циклов $C_1 = \sum_s p_s$.

2) Опишем число C_2 чётных путей в графе G'. Введём переменные r_{ag_1} и r_{bg_2} для двух краёв генов g_1 и g_2 в a' и b', принимающие значения в {–1,0,1}. Потребуем, чтобы сумма значений переменных r в точке минимума F по вершинам цепи или цикла в G' была равна 1, если это четная цепь и 0 иначе. Для каждой склейки (g_1, g_2) из a' или b' введём следующие ограничения: $r_{ag_1} + r_{ag_2} \leqslant 0$ и $r_{bg_1} + r_{bg_2} \leqslant 0$. Как следует из ограничений, эти переменные r не могут принимать значения 1 и 1, 1 и 0. Для каждой пары различных краёв генов g_1 и g_2, не являющейся склейкой, потребуем $r_{ag_1} + r_{ag_2} \leqslant 2(1 - t_{ag_1 g_2})$. Аналогичные ограничения наложим на b'. Для каждой пары (g,g') краёв одного типа из a' или b' потребуем $-2(1 - z_{kjj'}) \leqslant r_g - r_{g'} \leqslant 2(1 - z_{kjj'})$, где $k.j$ и $k.j'$ – гены с краями g и g'. Эти ограничения обеспечивают неравенство $r_g + r_{g'} \leqslant 0$ для (g,g'), являющихся рёбрами в G'. Также, если g и g' z-биективны, $r_{ag} = r_{bg'}$.

Учитывая, что переменные r_g входят в F с некоторыми отрицательными коэффициентами, они равны 1 в точке минимума в изолированных вершинах G'. Циклы имеют чётную длину, поэтому в вершинах циклов значения переменных либо нулевые, либо чередующиеся 1 и -1. Таким образом, сумма вдоль цикла равна 0. Значения переменных r_g на цепях чередуются, равны 1 на краях цепи чётной длины. Отсюда сумма вдоль такой цепи равна 1. Вдоль нечётных цепей чередование может прерываться нулевыми значениями, но сумма все равно будет равна 0. Отсюда получаем, что сумма $\sum r_g$ равна 1 только вдоль чётных цепей. Для особых хромосом d $\sum_{g \in d} r_g = 0$ в точке минимума, так как сумма очевидно не больше 0. Определим сумму, описанную в начале пункта 2. Для каждого края g гена из a', определим целочисленную переменную l_g, которая равна r_{ag} если g – край общего гена и 0 иначе. Это обеспечивается ограничениями $-\sum_j z_{kij} \leqslant l_g \leqslant \sum_j z_{kij}$, $l_g \leqslant r_{ag} + 2(1 - \sum_j z_{kij})$, $r_{ag} \leqslant l_g + 2(1 - \sum_j z_{kij})$, где $k.i$ – ген с краем g. Таким образом, вершине g в G', являющейся краем общего гена, соответствуют три переменных – r_{ag}, r_{bg} и l_g, принимающие одинаковые значения. Это позволяет сокращать r_{ag} и $-l_g$ при суммировании. Вершина g, являющаяся краем особого гена в $a'(z)$, отвечает двум переменным r_{ag} и l_{ag}, последняя равна 0. Таким образом, $C_2 = \sum_g r_g - \sum_g l_g$ в точке минимума F.

3) Опишем слагаемые $C_0 + n + s_a + s_b$. Для каждой хромосомы d в a' или b' определим булеву переменную o_d. Равенство $o_d = 1$ означает, что данная хромосома является особой m-кольцевой в точке минимума F. А именно, если d является m-кольцевой или линейной хромосомой, то $o_d \leqslant 1 - o(d)$; если же d – *1*-кольцевая или линейная, то $o_d = 0$. В самом деле, $o_d = 0$ следует из приведенных выше ограничений если d не особая или особая *1*-кольцевая. Для особой m-кольцевой хромосомы $o_d = 1$ в точке минимума F, так как o_d входят в F с отрицательными коэффициентами.

Покажем, что в точке минимума F мы имеем $C_0 + n + s_a + s_b = \sum_d n_d + \sum_d (1 - n_d) o_d - \sum_{k,i,j} z_{kij}$, где d пробегает по всем хромосомам в первой сумме, по всем m-кольцевым хромосомам во второй сумме и n_d – количество

генов в *d*. Число *n* эквивалентно сумме всех z_{kij}, s_a и s_b равны, соответственно, $n_b - n$ и $n_a - n$. Здесь n_a и n_b – количества генов в $a'(z)$ и $b'(z)$ не в особых хромосомах. Таким образом, $n + s_a + s_b = n_a + n_b - n$. Полагая, что $C_0 = \sum_d o_d + U$, $n = \sum_{k,i,j} z_{kij}$ и $n_a + n_b = \sum_d n_d(1 - o_d) - U$, где U – количество *1*-кольцевых хромосом, получаем нужное равенство.

Следующая теорема получена в [69]**.**

Теорема 5. Для данных *a* и *b*, минимальная нумерация паралогов и минимальное значение расстояния определяются точкой минимума *F*.

Замечание. Количество переменных и ограничений квадратично зависит от размера начальной задачи.

3. Задача реконструкции: произвольные структуры – сведением к кубическому ЦЛП

Сведём к ЦЛП задачу реконструкции хромосомных структур. Многие используемые в пункте 3.2 ограничения будут сохранены, рассмотрим отличия.

Пусть *T* – фиксированное укоренённое, возможно небинарное дерево. *Листовым* ребром будем называть ребро, одним из концов которого является лист, остальные ребра будем называть *внутренними*. Ребро будем обозначать как *T*-ребро (*G''*-ребро), если оно принадлежит дереву *T* (графу *G''*). Структуру в вершине *x* будем обозначать как *x*, то есть не будем различать вершину и соответствующую ей структуру. Начало и конец *T*-ребра будем обозначать как *a* и *b*, для самого ребра будем также использовать обозначение *e*=(*a*,*b*).

Зафиксируем начальную нумерацию во всех листовых структурах. Будем обозначать структуру листа *b* с фиксированной нумерацией как *b'*. Пусть *M* – множество всех полных имён *k.i*, где $1 \leqslant i \leqslant s(k)$.

Введём переменные z_{ukij} для каждого листа *u* и каждого гена *k.i* из *u'* и *k.j* из *M*; она равна 1, если *k.i* переименовывается в *k.j*, иначе $z_{ukij} = 0$. Существование и единственность *k.j* обеспечивается следующими ограничениями: для фиксированных *k* и *i* $\sum_j z_{ukij} = 1$, для фиксированных *k* и *j* $\sum_i z_{ukij} \leqslant 1$. Для краткости будем опускать индекс *u*.

Определим переменные $y_{\nu k.i}$ для каждой внутренней вершины ν и для каждого гена *k.i* из *M*; она равна 1, если *k.i* отсутствует в вершине ν, иначе 0. Для каждой внутренней вершины ν и каждой пары (*g*,*g′*) различных краёв из *M* определим переменную $x_{\nu gg'}$; равную 1, если *g* и *g′* присутствуют и отождествляются в вершине ν, иначе 0. Переменные $x_{\nu gg'}$ не определены в листьях, так как их значения в них зафиксированы. А именно, $\sum_{g' \neq g} x_{\nu gg'} \leqslant 1 - y_{\nu k.i}$ подразумевает, что край *g* любого гена $k.i \in M$, отсутствующий в ν, ни с чем не отождествляется, *g′* пробегает все значение краёв из *M*; также ограничения означают, что $\sum_{g'} x_{\nu gg'} \leqslant 1$ для любой фиксированной ν и *g*. Будем опускать индекс ν.

С целью исключить вырожденные сценарии с пустыми структурами во внутренних вершинах, наложим следующее ограничение: если ген отсутствует во внутренней вершине ν, он должен отсутствовать как минимум в половине своих прямых потомков. А именно, на каждое имя *k.j* из *M* налагаем следующее ограничение: $y_{\nu k.j} \leqslant 1.5 - \frac{1}{n_\nu}\left[\sum_{\nu'}(1 - y_{\nu' k.j}) + \sum_{\nu'}\sum_{i} z_{\nu' kij}\right]$, где n_ν – общее число прямых потомков ν′ из ν, ν' в первой и второй суммах пробегает по всем внутренним вершинам и листьям соответственно. Для бинарного дерева данное ограничение можно упростить до $y_{\nu k.j} \leqslant w(\nu') + w(\nu'')$, где $w(\nu^\alpha) = y_{\nu^\alpha k.j}$, если ν – внутренняя вершина, иначе $1 - \sum_i z_{\nu^\alpha kij}$.

Как в разделе 3.2 уравняем составы генов в *a′*(*z*) и *b′*(*z*), где *z* задаёт тождественные биекции паралогов для внутренних рёбер. Но теперь мы добавляем к *a′*(*z*) все особые *b′*(*z*) гены, а к *b′*(*z*) все особые *a′*(*z*) гены. Обозначаем полученные структуры как $a^+(z,t)$ и $b^+(z,t)$. Таким образом, особые хромосомы не удаляются. По этой причине общий граф *G″* структур $a^+(z,t)$ и $b^+(z,t)$ может отличаться от графа *G′*, определённого в разделе 3.2.

Для каждого ребра *e*=(*a*,*b*) и каждой пары *s*=(*g*,*g′*) различных краёв генов из *M* определим булеву переменную t_{ebs}, которая будет равна 1, если *g* и *g′* образуют новую склейку в $b^+(z,t)$. Аналогичная переменная t_{eas} вводится для *a*, однако, если *b* является листом, t_{eas} определяется только для краёв, присутствующих в *b′*. Индекс *e* можно опустить. Пусть *k.j* – ген с краем *g*. Для листового ребра *e* ограничения имеют

следующий вид: $t_{ebs} \leqslant 1 - y_{ak.j}$, $t_{ebs} \leqslant 1 - \sum_i z_{bkij}$, $\sum_{g_1 \in M} t_{ebgg_1} \leqslant 1$, $\sum_{g_1 \in b'} t_{eagg_1} \leqslant 1$, $\sum_{g_1 \in M} t_{ebgg_1} \geqslant 1 - y_{ak.j} - \sum_i z_{bkij}$; $t_{eas} \leqslant 1 + y_{ak.\alpha} - z_{bkj\alpha}$, $\sum_{g_1 \in b'} t_{eagg_1} \geqslant y_{ak.\alpha} + z_{bkj\alpha} - 1$. Последние два ограничения на самом деле являются системой неравенств для каждого $1 \leqslant \alpha \leqslant s(k)$.

Для внутреннего ребра e мы налагаем следующие ограничения: $t_{ebs} \leqslant 1 - y_{ak.j}$, $t_{ebs} \leqslant y_{bk.j}$, $\sum_{g_1 \in M} t_{bgg_1} \geqslant y_{bk.j} - y_{ak.j}$. Для t_{eas} ограничения аналогичные.

Пусть для любого листового ребра $e \in T$ $|M|$ - количество элементов в M, $c_e - |M|$ + количество генов в b. Целевая функция F' задачи минимизации тогда равна сумме двух выражений. Первое из них: $\left(c_e - \sum_{k.i \in M} y_{ak.i} - \sum_{k.j \in M} f_{k.j} \right) - \sum_s p_s - 0.5 \left(\sum_g r_g - \sum_g l_g \right)$, просуммированное по всем листовым T-рёбрам e. Второе выражение имеет вид: $\left(2 \cdot |M| - \sum_{k.i} y_{ak.i} - \sum_{k.i} y_{bk.i} - \sum_{k.j} f_{k.j} \right) - \sum_s p_s - 0.5 \left(\sum_g r_g - \sum_g l_g \right)$, просуммированное по всем внутренним T-рёбрам. Все переменные, кроме y определены в пунктах 1-3 ниже. Они соответствуют пунктам 1-3 раздела 3.2.

1) Пусть ребро $e=(a,b)$ является T-ребром и $G''(e)=a^+(z,t)+b^+(z,t)$. Определим переменные u_{es} и p_{es}, а также ограничения, обеспечивающие равенство числа C_1' циклов в графе $G''(e)$ в точке минимума F' сумме $\sum_s p_{es}$. А именно, для каждой пары $s=(g,g')$ разных краёв из M для внутреннего ребра $e=(a,b)$, определим целочисленные неотрицательные переменные u_{eas} и u_{ebs} и булевы переменные p_{eas} и p_{ebs}. Для листового ребра e и его конца b' мы определяем целочисленную неотрицательную u_{ebs} и булеву p_{ebs}, где s – любая склейка в b'. Для u_{eas} и u_{ebs} выполнено ограничение $u_s \leqslant m_s$. Здесь m_s – число упомянутых пар s, где s пробегает по всем парам, переменные u_{eas} и u_{ebs} определены для любого фиксированного $e \in T$. Для булевой переменной p_s потребуем $p_s \cdot m_s \leqslant u_s$.

Пусть $e=(a,b)$ является листовым ребром. Потребуем, чтобы $u_{as} \leqslant m_{as} \cdot x_{as}$, обеспечив равенство $u_{as} = 0$ для любой пары s несклеенных краёв из M. Для a, пусть s содержит ген $k.j$ из M, g – край этого гена. Для каждой переменной u_{as} и каждого края гена $k.j' \in b'$ того же типа, что и g, являющегося краем цепи в b', вводятся ограничения

$u_{as} \leqslant m_{as}(1 - z_{kj'j})$. Эти ограничения обеспечивают равенство $u_s = 0$ при условии, что g принадлежит общему гену $a'(z)$ и $b'(z)$, и в $G''(e)$ имеем следующее: g – край цепи и одновременно край G''-ребра, помеченного как a. Для b, пусть склейка $s \in b'$ и включает край $g \in k.j$. На каждую переменную u_{bs} и каждое i такое, что $1 \leqslant i \leqslant s(k)$, накладываются следующие ограничения: $u_{bs} \leqslant m_{bs}\left(1 - z_{kji} + \sum_{g_1 \in M} x_{ag'g_1}\right)$, где g' – край гена $k.i$ из M того же типа, что и g.

Эти ограничения обеспечивают равенство $u_s = 0$ при условии, что g принадлежит общему гену структур $a'(z)$ и $b'(z)$, и в G'' мы имеем: g – край цепи и в то же время край G''-ребра с пометкой b. Для каждого края g_1 из M потребуем, чтобы $u_{as} \leqslant m_{as}\left(1 - t_{bg_1g} + \sum_{g_2} x_{ag_1g_2}\right)$. Это неравенство обеспечивает $u_{bs} = 0$ если $g \in s$, g принадлежит особому гену из $a'(z)$ и g не является краем цепи в $G''(e)$, но является концом крайнего нового G''-ребра цепи. Для каждого края g_1 из b', который является краем цепи в b', выполняется ограничение $u_{bs} \leqslant m_{as}(1 - t_{ag_1g})$, что обеспечивает $u_{bs} = 0$ в случае, когда край $g \in b'$ принадлежит особому гену из $b'(z)$ и g не является краем цепи, но является концом крайнего нового G''-ребра цепи.

Напомним, что рассматривается случай листового ребра e=(a,b). Для каждой пары (s_1, s_2), где $s_1 = (g, g_1)$ – пара краёв из M и $s_2 = (g', g_2)$ – склейка из b', причём g и g' одного типа, выполнено $u_{as_1} \leqslant u_{bs_2} + m_{as_1}(1 - z_{kj'j})$, $u_{bs_2} \leqslant u_{as_1} + m_{bs_2}(1 - z_{kj'j} - x_{agg_1})$. Получаем $u_{s_1} = u_{s_2}$ для смежных старых G''-рёбер s_1 и s_2 графа $G''(e)$. Для каждой пары (s_1, s_2), где $s_1 = (g_1, g_2)$ и $s_2 = (g_3, g_4)$ – пары краёв из M потребуем $u_{as_1} \leqslant u_{as_2} + m_{as_1}(1 - t_{bg_2g_3} - x_{ag_3g_4})$, $u_{as_2} \leqslant u_{as_1} + m_{as_2}(2 - t_{bg_2g_3} - x_{ag_1g_2})$, все g_1, g_2, g_3, g_4 попарно различны. Эти ограничения обеспечивают $u_{s_1} = u_{s_2}$ для двух старых G''-рёбер s_1 и s_2 (помеченных как a) из $G''(e)$, разделённых ровно одним новым G''-ребром. Для каждой пары (s_1, s_2), где $s_1 = (g_1, g_2)$ и $s_2 = (g_3, g_4)$ – различные склейки из b', потребуем $u_{s_1} \leqslant u_{s_2} + m_{s_1}(1 - t_{ag_2g_3})$, $u_{s_2} \leqslant u_{s_1} + m_{s_2}(1 - t_{ag_2g_3})$. Эти ограничения обеспечивают

$u_{s_1} = u_{s_2}$ для двух старых *G''*-рёбер (помеченных как *b*) графа *G''*(*e*), разделённых ровно одним новым *G''*-ребром.

Для внутренних рёбер *e*=(*a*,*b*) полагаем, что $u_{as} \leqslant m_{as} x_{as}$, $u_{bs} \leqslant m_{bs} x_{bs}$, таким образом обеспечивая равенство $u_{as} = 0$ или $u_{bs} = 0$ для пары *s*=(*g*,*g'*), не являющейся склейкой. Для каждой переменной u_{as} требуем $u_{as} \leqslant m_{as} \left(y_{bk.j} + \sum_{g_1} x_{bgg_1} \right)$, где *s* включает $g \in k.j$. Подобные ограничения накладываются и на u_{bs}. Они обеспечивают $u_s = 0$ при условии, что *g* принадлежит общему гену и является краем цепи в *G''*(*e*). Равенство $u_s = 0$ (для u_{as} или u_{bs}) в случае, когда край *g* принадлежит особому гену (в *a'*(*z*) или *b'*(*z*)) и крайнему ребру цепи (в *G''*(*e*)), обеспечивается так же, как и для u_{as} на листовом ребре. На каждую пару (s_1, s_2), где $s_1 = (g, g_1)$ и $s_2 = (g, g_2)$ – различные пары краёв из *M*, налагаем следующие ограничения: $u_{as_1} \leqslant u_{bs_2} + m_{s_1}(1 - x_{bgg_2})$, $u_{bs_2} \leqslant u_{as_1} + m_{s_2}(1 - x_{agg_1})$. Данные ограничения обеспечивают равенство $u_{s_1} = u_{s_2}$ для старых смежных *G''*-рёбер s_1 и s_2 в *G''*(*e*). На каждую пару (s_1, s_2), где $s_1 = (g_1, g_2)$ и $s_2 = (g_3, g_4)$ – пары краёв из *M*, налагаем следующие ограничения:

$u_{as_1} \leqslant u_{as_2} + m_{as_1}(2 - t_{bg_2g_3} - x_{ag_3g_4})$, $u_{as_2} \leqslant u_{as_1} + m_{as_2}(2 - t_{bg_2g_3} - x_{ag_1g_2})$,

$u_{bs_1} \leqslant u_{bs_2} + m_{bs_1}(2 - t_{ag_2g_3} - x_{bg_3g_4})$, $u_{bs_2} \leqslant u_{bs_1} + m_{bs_2}(2 - t_{ag_2g_3} - x_{bg_1g_2})$, все g_1, g_2, g_3, g_4 попарно различны. Эти ограничения гарантируют равенство $u_{s_1} = u_{s_2}$ для двух старых *G''*-рёбер графа *G''*(*e*), разделённых ровно одним новым *G''*-ребром.

Утверждение о том, что $C_1' = \sum_s p_s$ в точке минимума для любого $e \in T$ доказывается так же, как в разделе 3.2.

2) Определим переменные и ограничения, обеспечивающие равенство количества C_2' чётных путей в *G''*(*e*) на ребре *e*=(*a*,*b*) в точке минимума *F'* величине $\sum_g r_g - \sum_g l_g$.

Для каждого края *g* из *M* определим целочисленную переменную r_{eag}, принимающую значения в {–1, 0, 1}. Аналогично для *b* в случае, если *b* – внутренняя,

иначе только для каждого края g из b'. Ограничения $-2(1-y_{ak.i}) \leqslant r_{eag} \leqslant 2(1-y_{ak.i})$ обеспечивают равенство $r_{eag}=0$ для любого края любого гена $k.i \in M$, отсутствующего в v. Аналогичное ограничение вводится для b, если она внутренняя.

Для каждой пары различных краёв g_1 и g_2 из M потребуем выполнение неравенства $r_{eag_1}+r_{eag_2} \leqslant 2(1-x_{ag_1g_2})$. Из него следует, что $r_{eag_1}+r_{eag_2} \leqslant 0$, если g_1 и g_2 склеены. Для внутреннего ребра e налагаются подобные ограничения с той разницей, что индекс a заменяется на b. Для листового ребра налагаются ограничения только для склеек (g_1, g_2) из b', в правой части стоит 0. Также потребуем $r_{eag_1}+r_{eag_2} \leqslant 2(1-t_{ebg_1g_2})$, таким образом обеспечив $r_{eag_1}+r_{eag_2} \leqslant 0$ при условии, что g_1 и g_2 образуют новую склейку. Для внутреннего ребра e вводятся аналогичные ограничения, в которых индекс a заменяется на b и наоборот; иначе ограничения вводятся только для пар (g_1, g_2) из b', которые не образуют склейку.

Для листового ребра e каждая пара (g,g'), где края g (из $k.j$) и g' (из $k.j'$) одного типа, g из M и g' из b', вводим следующее ограничение: $-2(1-z_{bkj'j}+y_{ak.j}) \leqslant r_{eag}-r_{ebg'} \leqslant 2(1-z_{1-z_{bkj'j}+y_{ak.j}})$. Из него следует, что $r_{ebg}=r_{eag'}$ для z-биективных краёв g и g' одного типа, если g присутствует в a. Для внутреннего ребра e и каждого края $g \in M$ гена $k.i$ потребуем $r_{eag} \leqslant r_{ebg}+2(y_{ak.i}+y_{bk.i})$, $r_{ebg} \leqslant r_{eag}+2(y_{ak.i}+y_{bk.i})$. Отсюда в случае, если g является общим для $a'(z)$ и $b'(z)$, получаем равенство $r_{eag}=r_{ebg}$.

Для каждого ребра e и гена $k.j$ из M определим булеву переменную $f_{ek.j}$, показывающую, является ли ген $k.j$ общим для структур $a'(z)$ и $b'(z)$. А именно, для внутреннего ребра e потребуем выполнение неравенств $f_{ek.j} \geqslant 1-y_{ak.j}-y_{bk.j}$, $f_{ek.j} \leqslant 1-y_{ak.j}$, $f_{ek.j} \leqslant 1-y_{bk.j}$. Для листового ребра переменная $y_{bk.j}$ заменяется на выражение $1-\sum_i z_{bkij}$, в результате чего получаются неравенства $f_{ek.j} \geqslant \sum_i z_{bkij}-y_{ak.j}$ и $f_{ek.j} \leqslant \sum_i z_{bkij}$.

Для каждого края g гена $k.i$ из M определим целочисленную переменную l_{eg}, которая равна r_{eag} при условии, что g – край общего гена в $a'(z)$ и $b'(z)$, или нулю в противном случае. Соответствующие неравенства имеют следующий вид:

$-f_{ek.i} \leqslant l_{eg} \leqslant f_{ek.i}$, $l_{eg} \leqslant r_{eag} + 2(1 - f_{ek.i})$, $r_{eag} \leqslant l_{eg} + 2(1 - f_{ek.i})$. Теперь утверждение, что $C_2' = \sum_g r_g - \sum_g l_g$ для любого $e \in T$ доказывается так же, как и в задаче нахождения расстояния.

3) На каждом ребре $e \in T$, где e=(a,b), каждая из первых двух скобок в определении F' равна числу общих генов в $a'(z)$ и $b'(z)$, посчитанных по одному разу, плюс общее число особых генов в тех же структурах. Будем эту сумму обозначать как X. В самом деле, значения $c_e - \sum_{k.i} y_{ak.i}$ и $2|M| - \sum_{k,i} y_{ak.i} - \sum_{k,i} y_{bk.i}$ равны общему числу генов в a и b. Эти значения минус $\sum_{k,j} f_{k.j}$, число общих генов, посчитанных один раз, дают X.

Пусть $\Psi(e, x, y, z)$ равно $C_0 + n + s_a + s_b$ в Ф из раздела 3.2. Ψ здесь рассматривается на ребре e=(a,b), слагаемые определены в 3.2. Получаем $X - \Psi = C_3 - C_0$, где C_3 – общее число генов в особых хромосомах в a'(z) и b'(z), C_0 – общее число особых генов в $a'(z)$ и $b'(z)$. Определим $E = \sum_{e \in T} (C_3 - C_0)(e)$. Для любой расстановки A и начальной нумерации, $E(A)$ определяется аналогично.

Теорема 6 [69] гласит, что наш алгоритм сведения является *точным* при следующем условии (*): существует точка минимума целевой функции F', в которой в любой кольцевой хромосоме, входящей в структуру, соответствующую одному из концов какого-то ребра, существует ген, присутствующий в структуре, отвечающей второму концу этого ребра. Данное условие требует отсутствия особых кольцевых хромосом в структурах, отвечающих вершинам дерева. Введём определения. Скажем, что аргументы (x,y,z,t,f,u,p,r,l) функции F' являются расширением аргументов (x,y,z) функции F*, если переменная t для каждого e определяет новые склейки в $a^+(z,t)$ и $b^+(z,t)$ такие, что расстояние между структурами минимально, а другие переменные определяются через $a'(z)$, $b'(z)$ и G'' так, что вышеописанные ограничения вместе с равенствами $C_1' = \sum_s p_s$ и $C_2' = \sum_g r_g - \sum_g l_g$ выполнены для каждого ребра e. Очевидно, существует расширение для каждой расстановки A=(x,y,z); будем обозначать его как A_+. Напомним, что A определяет структуры a и b на концах ребра e=(a,b).

Теорема 6. При условии (*) минимальные значения функций $F^*(A)$ и $F'(x,y,z,t,f,u,p,r,l)$ равны. Иначе разница между минимальными значениями не больше общего числа особых хромосом в минимальной точке F'.

Константы $2\cdot|M|$ и c_e могут быть опущены при минимизации.

Заметим также, что условие (*) накладывает ограничения на особые (можно сказать, на кольцевые) хромосомы в структурах, то есть, ограничивает взаимосвязь между родительскими структурами и их прямыми потомками.

4. Согласование множеств контигов – сведением к ЦЛП с линейным числом переменных и ограничений

Для каждой пары $s=(g_1,g_2)$ концов различных контигов в a' определим булеву переменную t_{as}. Она равна 1, если g_1 и g_2 образуют склейку при соединении контигов в цикл; иначе t_{as} равна 0. Аналогично для b'. Стандартные ограничения обеспечивают соединение каждого контига ровно с одним другим.

Для каждой упорядоченной пары $d=(c_1,c_2)$ различных контигов из данных наборов a' и b' определим булеву переменную ν_d, равную 1, если контиг c_2 присоединяется к концу контига c_1; иначе 0. Набор переменных $\nu_d=1$ определяет порядок обхода контигов на искомом цикле по часовой стрелке. Ограничения $\nu_d \leqslant t_{s_1}+t_{s_2}+t_{s_3}+t_{s_4}$ для пар s_1,s_2,s_3,s_4 концов контигов c_1 и c_2 дают взаимоотношения между порядком и склейками концов контигов.

Определим переменные w_{ac} и w_{bc}, где c пробегает все контиги в a' и b' и $1\leqslant w_c\leqslant N$ (где N – число контигов в соответствующем наборе). Переменные w_{ac} задают строго возрастающую нумерацию контигов согласно их позиции на цикле. Этот порядок нарушается только для последнего контига. Аналогично для w_{bc}. Для каждой упорядоченной пары $d=(c_1,c_2)$ различных контигов из наборов a' и b' определим булеву переменную r_d, равную 1, если порядок для контига нарушается, то есть $\nu_d=1$ и $w_{c_2}\leqslant w_{c_1}$; иначе 0. Ограничения имеют вид $r_d\leqslant\nu_d$, $Nr_d\leqslant N-(w_{c_1}-w_{c_2}+1)$, $Nr_d\geqslant w_{c_1}-w_{c_2}+1$. Ограничение $\sum_d r_d=1$ гарантирует объединение всех контигов в одну циклическую хромосому, в которой они занумерованы переменными w в строго возрастающем порядке.

Введём переменные z_{kij}, каждая из которых равна 1, если ген *k.i* в *a'* отвечает гену *k.j* в *b'*; иначе $z_{kij}=0$. Стандартные ограничения обеспечивают наличие частичной биекции *k*-паралогов. Если $z_{kij}=1$, ген *k.j* в *b'* переименовывается в *k.i* в *a'* и становится синонимом *k.i* в *a'*, после чего гены в *z*-биекции случайным образом нумеруются, чтобы структуры оставались перенумерованы. Обозначим перенумерованные структуры *a'*(*z*) и *b'*(*z*). Склейки, определяющие объединения контигов в цикл определяются переменными *t*, аналогично разделу 3.2. Итоговые циклы будем обозначать как *a'*(*z*,*t*) и *b'*(*z*,*t*). Заметим, что эти структуры имеют неравный генный состав. Определим общий граф *G*=*a'*(*z*,*t*)+*b'*(*z*,*t*). Он состоит только из циклов.

Количество блоков *B* в *G* выражается с помощью переменных x_{as} для каждого *s*, где *s* – внутренняя склейка или пара внешних краёв контигов в *a'*. Она равна 1, если *s* – граница блока в *a'*(*z*,*t*) и 0 иначе. Аналогично для *b'*(*z*,*t*). На каждую *s* из *a'* накладывается ограничение $z_{as}\geqslant\sum_j z_{ki_1j}-\sum_j z_{li_2j}+(t_s-1)$, где $k.i_1$ и $l.i_2$ гены в *a'* с этими краями. Аналогично для *s* из *b'*. Для внутренних склеек *s* слагаемое t_s-1 опускается. Рассмотрим функцию $H=0.5\cdot\sum_s x_s+\sum_s y_s-\sum_s p_s$. Таким образом, $B=0.5\cdot\sum_s x_s$ в минимальной точке *H*.

Сумма s_1 целых частей половин длин максимальных связных участков обычных рёбер в *G* выражается с помощью переменных y_{as} и y_{bs} для всех *s* как в части 1 данного раздела. Они равны 0 если *s* находится на границе или внутри блока в *a'*(*z*,*t*) или *b'*(*z*,*t*); для склеек обычных генов значения y_{as} и y_{bs} на ребрах *G* чередуются и равны 0 на концах нечётных участков. Для каждой пары s_1 из *a'* и s_2 из *b'*, где ген *k.i* является смежным гену k_1i_1, а ген *k.j* является смежным гену k_2i_2 из s_2, вводим следующее ограничение: $y_{as_1}+y_{bs_2}\geqslant z_{kij}+\sum_j z_{k_1i_1j}+\sum_j z_{k_2ji_2}-2+(t_{as}-1)+(t_{bs}-1)$, где слагаемые $t_{as}-1$ и $t_{bs}-1$ пропускаются для внутренних смежных пар s_1 и s_2 соответственно. Это ограничение обеспечивает невозможность равенства y_s нулю на обоих смежных рёбрах. Соответственно, минимальная сумма на участке достигается для чередующихся

значений на рёбрах, причем последовательность начинается с 0. Таким образом, сумма $S_1 = \sum_s y_s$.

Количество S_2 циклов в *G*, состоящих из обычных рёбер выражается перменными u_S. и p_s для *s*, как в разделе 1. Для каждой *s* введём ограничения $u_s \leqslant m_s \sum_j z_{kij}$ (для *s* из *a'*) или $u_s \leqslant m_s \sum_j z_{kji}$ (для *s* из *b'*), где *k.i* – ген с краем из *s*. Для каждой пары *s* краёв введём ограничение $u_s \leqslant m_s t_s$. Также для каждой пары s_1 и s_2, которые содержат края гена *k.i* из *a'* и *k.j* из *b'* вводим ограничения $u_{s_1} \leqslant u_{s_2} + m_{s_1}(1 - z_{kij}) + m_{s_1}(1 - t_{s_2})$, $u_{s_2} \leqslant u_{s_1} + m_{s_2}(1 - z_{kij}) + m_{s_2}(1 - t_{s_1})$, где слагаемые $m_{s_2}(1 - t_{s_1})$ и $m_{s_1}(1 - t_{s_2})$ не входят для внутренних склеек s_1 и s_2 соответственно. Тогда $S_2 = \sum_s p_s$ в минимальной точке. Доказательство аналогично доказательству равенства числа циклов C_1 в *G'* сумме $\sum_s p_s$.

Таким образом, минимальное значение функции *H* равно $B + S_1 - S_2$, что равно расстоянию между искомыми циклами.

5. Тестирование на искусственных примерах.

Пример 1.

Рассмотрим задачу реконструкции для циклических структур с паралогами. Дано дерево ((*c*,*d*),(*e*,*f*)), в листьях которого заданы следующие структуры:

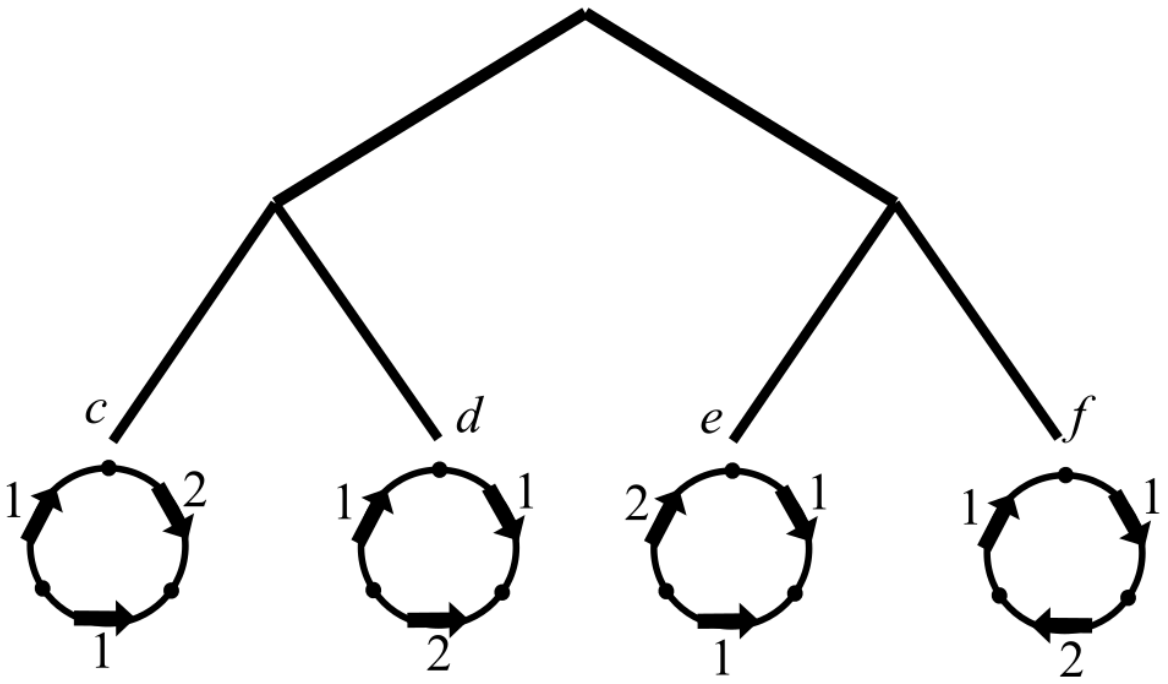


Рисунок 65. Исходное дерево с заданными листовыми структурами, присутствуют паралоги гена 1.

Пусть число паралогов гена 1 равно 3, гена 2 – 1. Начальная нумерация паралогов определена в таблице 3.1:

Таблица 3.1. Листовые структуры дерева

Звездочка перед именем гена указывает на его расположение гена на комплементарной цепи. Номер единственного паралога гена 2 опустим

Имя листа	Структура
c	1.1 2 *1.2 (C)
d	1.1 1.2 *2 (C)
e	2 1.1 *1.2 (C)
f	1.1 1.2 2 (C)

Задача ЦЛП для данной задачи выдала решение, показанное на рисунке 66. Суммарное число операций получилось равным 6, по одной операции на каждом ребре.

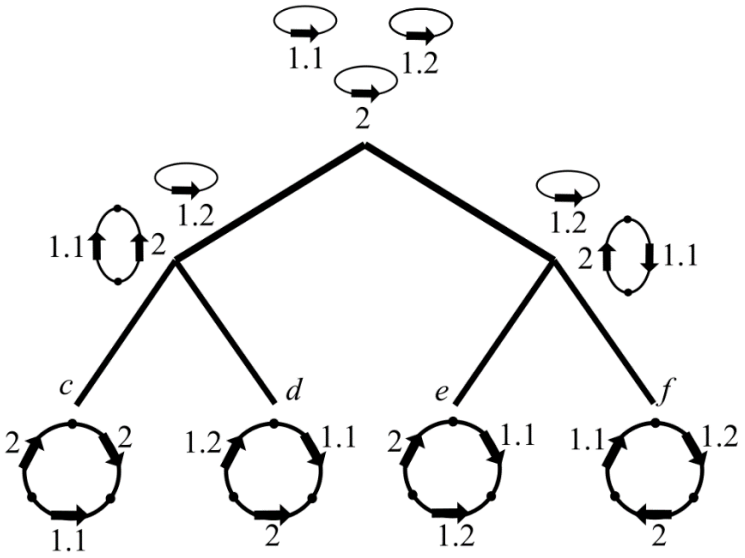


Рисунок 66. Решение задачи реконструкции, показанной на рисунке 65

В оптимальном решении паралоги гена 1 поменялись местами в структурах *c* и *d*.

Пример 2.

Рассмотрим задачу реконструкции для линейных структур с паралогами. Дано дерево ((*c*,*d*),(*e*,*f*)):

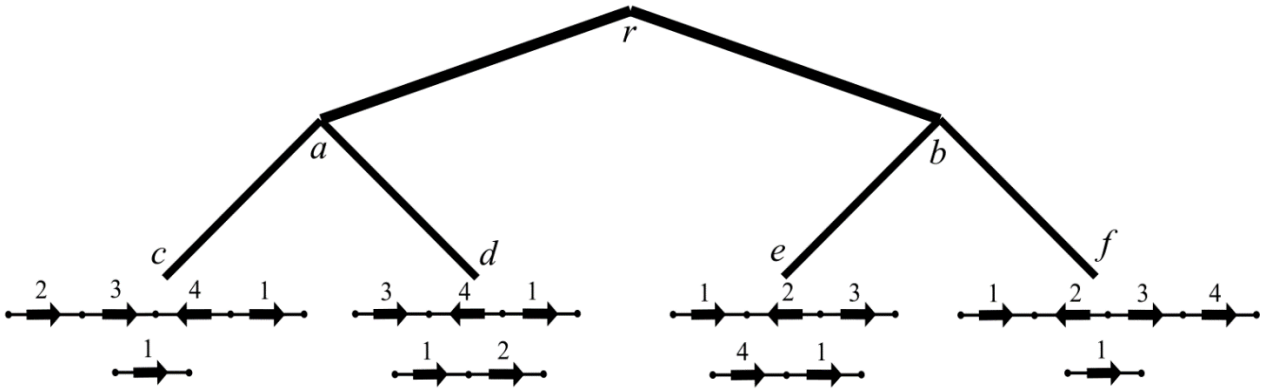


Рисунок 67. Исходное дерево с заданными листовыми структурами

Таблица 3.2. Листовые структуры дерева

Звездочка перед именем гена указывает на его расположение гена на комплементарной цепи

Имя листа	Структура
c	2 3 *4 1.1 (L); 1.2 (L)
d	3 *4 1.1 (L); 1.2 2 (L)
e	1.1 *2 3 (L); 4 1.2 (L)
f	1.1 *2 3 4 (L); 1.2 (L)

Задача ЦЛП для данной задачи выдала решение, показанное на рисунке 68. Суммарное число операций получилось равным 8.

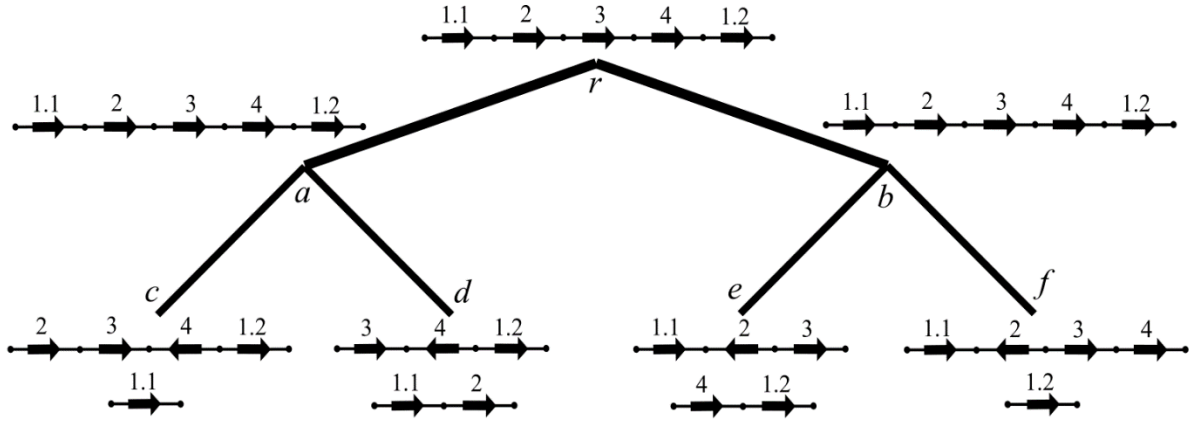


Рисунок 68. Решение задачи реконструкции, показанной на рисунке 67

ГЛАВА 4. ФИЛОГЕНЕТИЧЕСКИЕ ДЕРЕВЬЯ и РЕКОНСТРУКЦИЯ ХРОМОСОМНЫХ СТРУКТУР МИТОХОНДРИЙ ИНФУЗОРИЙ и СПОРОВИКОВ, ПЛАСТИД РОДОФИТНОЙ ВЕТВИ и БАКТЕРИЙ РОДА *Rhizobium*

1. Митохондрии инфузорий

В митохондриях синтезируются АТР и другие соединения [21]. В основном, многие эукариоты, живущие в анаэробных условиях, не имеют митохондрий [22] или содержат аналоги митохондрий, такие как гидрогеносомы или митосомы. Например, *Nyctotherus ovalis*, анаэробная инфузория, обитающая в кишечнике таракана *Periplaneta americana* и *Blaberus sp.* [23], содержит гидрогеносомы, которые производят молекулярный водород [24]. Роль митохондрий неодинакова в различных организмах, и это отражается в размере митохондриальных геномов. Митохондриальные геномы паразитирующих споровиков очень маленькие, они кодируют всего три белка и короткие фрагменты rRNA [25]. Инфузории (Ciliophora) включают паразита *Ichthyophthirius multifiliis*, который является причиной смерти многих пресноводных рыб, обитающих в аквариумах и на рыбных фермах [26]. Митохондрии споровиков могут служить мишенями для терапевтического воздействия на возбудителей протозойных инфекций. Поэтому анализ структуры и эволюции их геномов может иметь практическое значение, особенно в ветеринарной медицине. В диссертации рассмотрены митохондриальные геномы инфузорий: типа споровики (Apicomplexa) и типа инфузории (Ciliophora), которые принадлежат надтипу Альвеоляты (Alveolata); а именно, виды, принадлежащие трём классам: Armophorea (*Nyctotherus*), Oligohymenophorea (*Ichthyophthirius, Paramecium*, и *Tetrahymena*), и Spirotrichea (*Moneuplotes, Oxytricha*). Классы Oligohymenophorea и Spirotrichea существенно различаются [27]. Armophorea включает анаэробов, хотя является более близкой к Spirotrichea, чем к Oligohymenophorea [24]. Многие инфузории – свободно живущие организмы. Например, клетки *Moneuplotes minuta* можно обнаружить в Средиземном море около Корсики [27]. *Moneuplotes minuta* и *Oxytricha trifallax* могут быть выращены в растворе неорганических соединений, в качестве пищи можно использовать *Chlamydomonas reinhardtii* или *Klebsiella spp.* Инфузория *Ichthyophthirius multifiliis*, являющаяся патогеном пресноводных рыб, и встречающаяся как в умеренном, так и в тропическом регионах по всему миру [28], менее терпима к соли, чем рыба. *Tetrahymena* и *Paramecia* – свободно живущие инфузории. *Tetrahymena*, в основном, встречается в

пресноводных водоемах. *Paramecia* широко распространены как в пресных, так и в солёных средах. Особенно богатые популяции можно обнаружить в застоявшихся прудах.

Эндосимбионты *Paramecium aurelia* – грамм-отрицательные бактерии. Большинство эндосимбионтов выделяют токсины, которые убивают чувствительные штаммы *Paramecia* [29].

Из GenBank извлечены полные митохондриальные геномы следующих организмов: *Ichthyophthirius multifiliis* (NC_015981), *Paramecium aurelia* (NC_001324), *Paramecium caudatum* (NC_014262), *Tetrahymena malaccensis* (NC_008337), *Tetrahymena paravorax* (NC_008338), *Tetrahymena pigmentosa* (NC_008339), *Tetrahymena Pyriformis* (NC_000862), и *Tetrahymena thermophila* (NC_003029). Из той же базы были загружены четыре неполных генома *Moneuplotes minuta* (GQ903130), *Moneuplotes crassus* (GQ903131), *Nyctotherus ovalis* (GU057832), и *Oxytricha trifallax* (*Sterkiella histriomuscorum*; JN383843).

В рассматриваемых митохондриях кодируются десятки белков [24, 27-35]; функции некоторых из них не установлены, а соответствующие последовательности быстро накапливают мутации [36].

В *Ichthyophthirius* митохондриальная хромосома циклическая, в остальных рассматриваемых видах хромосомы линейные [35, 37]. В митохондриях *Tetrahymena*, *Moneuplotes*, и *Oxytricha* большинство генов транскрибируются в противоположных направлениях относительно середины линейной хромосомы. Большинство генов в митохондриях *Paramecium* и *Nyctotherus*, наоборот, считываются в одном направлении. Рассматриваемые геномы очень компактны, а гены формируют длинные опероны с короткими некодирующими участками; в некоторых случаях кодирующие участки могут перекрываться. Порядок генов в рассматриваемых классах различается, поэтому исследование эволюции хромосомных структур является нетривиальной задачей.

Филогенетическое дерево хромосомных структур митохондрий инфузорий построено на основе матрицы попарных расстояний, которая вычислена с использованием модели и алгоритма из Главы 1, программная реализация алгоритма доступна по ссылке [40], которая применялась эвристически: нумерация немногочисленных паралогов определялась ограниченным перебором вариантов. Для вычисления матрицы был использован линейный вариант цен: двойная переклейка – 1.2, полуторная переклейка – 1.1, вставка ребра – 1, удаление ребра – 0.9, удаление особых *a*-вершин – 0.8; удаление особых *b*-вершин – 1.5. В матрицу заносилось среднее

арифетическое расстояний от одной структуры к другой и наоборот. Полученное дерево показано на рисунке 69. Само дерево строится с помощью алгоритма присоединения соседей [20]. В следующем пункте получена реконструкция хромосомных структур по этому дереву.

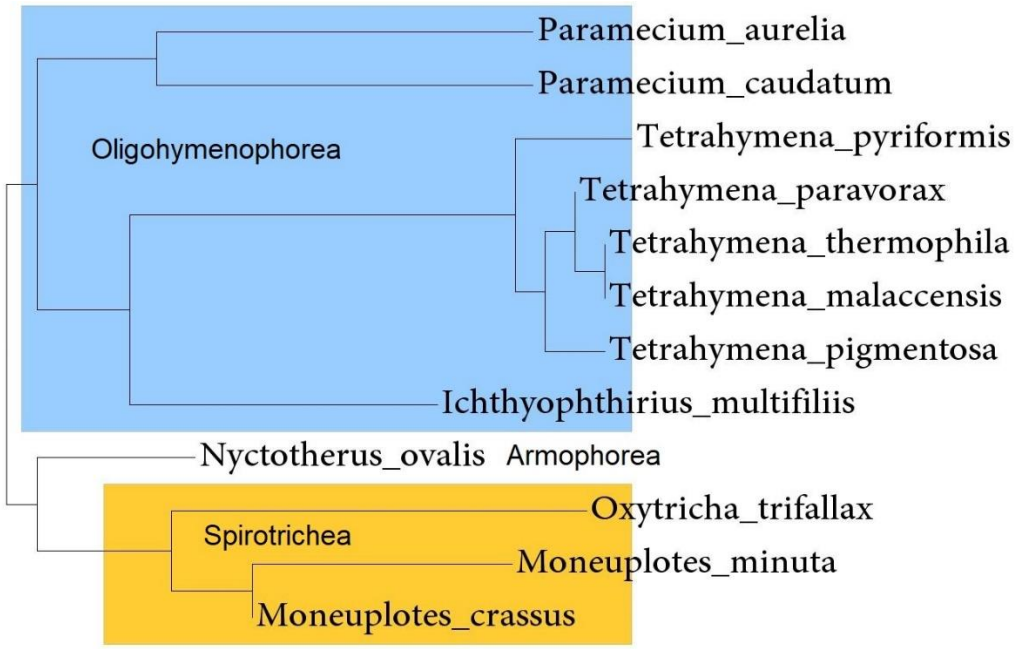


Рисунок 69. Филогенетическое дерево хромосомных структур митохондрий инфузорий
Дерево построено с помощью алгоритма присоединения соседей (neighbor-joining algorithm) на основе матрицы попарных расстояний, вычисленных с помощью алгоритма из Главы 1.

В полученном дереве каждый род формирует отдельную кладу. Классы Armophorea, Oligohymenophorea, и Spirotrichea также формируют клады. Близкое положение классов Armophorea и Spirotrichea на дереве согласуется с ранее опубликованными данными [24]. Таким образом, полученное дерево хорошо согласуется с деревьями, построеными по белковым семействам и с общепринятой таксономией. Небольшие различия между деревьями, показанными на рисунках 70, 71 [70], могут быть отнесены к небольшому размеру митохондриальных геномов.

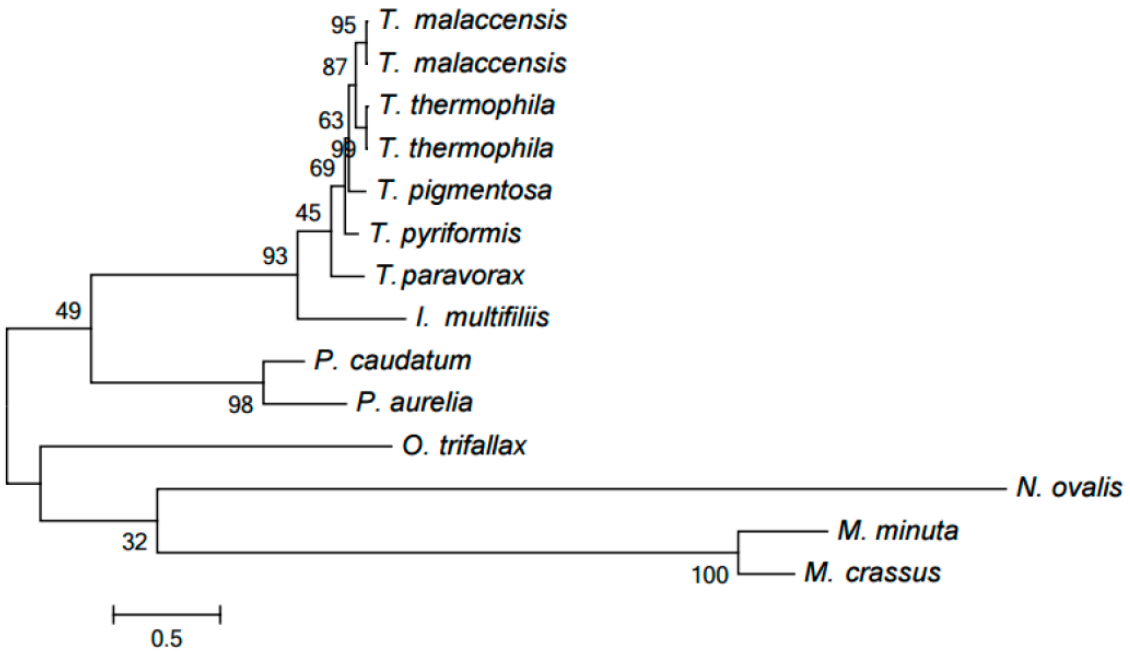


Рисунок 70. Дерево семейства субъединиц 9 NADH-дегидрогеназ (Nad9)
Построено по кластеризации [47].

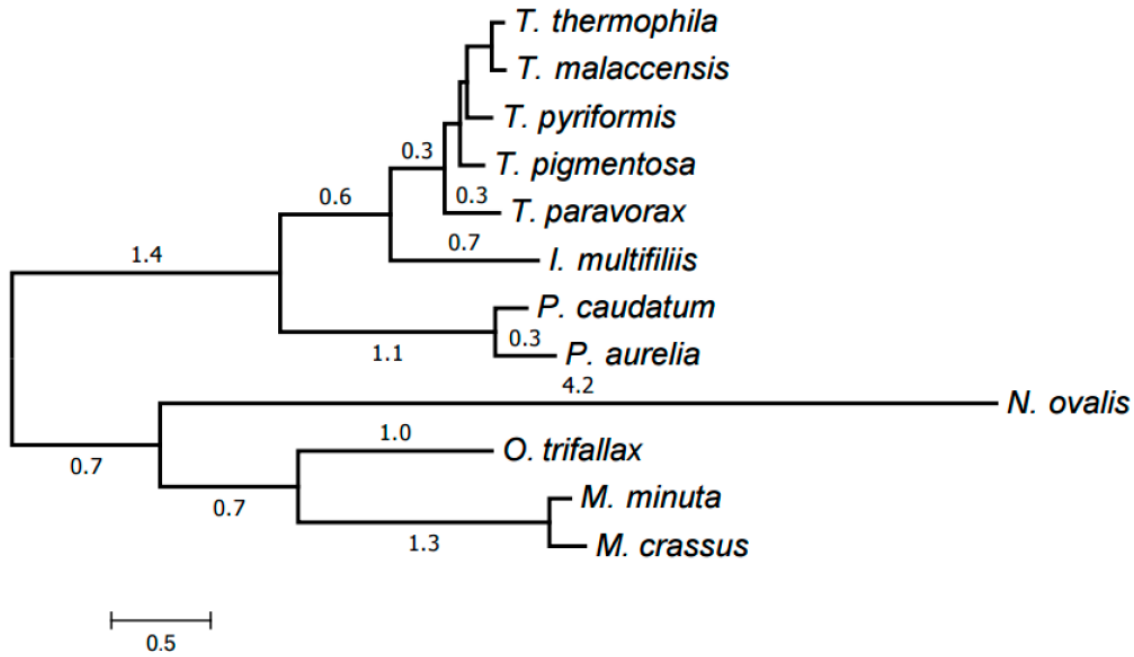


Рисунок 71. Филогенетическое дерево митохондрий.
Полученное с помощью PhyloBayes, использованы данные о белковых семействах из [70].

Реконструкция хромосомных структур митохондрий инфузорий выполнялась алгоритмом, описанным в разделе 3.3. Результаты реконструкции во внутренних вершинах дерева, показанного на рисунке 69, представлены в таблице 4.1. В корне

дерева и в двух его потомках предсказаны структуры, которые содержат по две хромосомы, циклическую и линейную. Остальным вершинам соответствует по одной линейной хромосоме.

Таблица 4.1. Реконструкция хромосомных структур митохондрий инфузорий
В левом столбце выписаны листовые вершины, помеченные знаком (*l*), или пары листовых вершин, ограничивающих кладу и, таким образом, определяющих внутреннюю вершину дерева. В среднем столбце показана хромомосомная структура, полученная в этой вершине; буквы *L* и *C* отмечают, что структура линейная или циклическая. Звездочка перед именем гена указывает на его расположение гена на комплементарной цепи. Номера паралогов указаны после знака подчёркивания. Вторая структура (хромосома, если их две) начинается с новой строки. Структуры (хромосомы) в листьях служат исходными данными для нашего алгоритма. В правом столбце приведены эволюционные события на ребре, входящем в указанную вершину, которые определяются построенной кратчайшей последовательностью.

Вершина	**Хромосомные структуры**	**События**
***T. thermophila* (*l*)**	*trnY *rnl_a_1 *trnL_1 *rnl_b_1 *ymf57 *ymf66 *ymf76 *rps13 *rps3 *rps19 *rpl2 *ymf74 *nad10 *rps12 *nad2 nad7 rps14 ymf60 *ymf64 *ymf75 *trnF *nad1_b *atp9 *ymf63 *ymf65 *ymf69 *ymf59 *rpl16 *yejR *ymf61 *trnH *nad3 *ymf72 *nad4L *nad9_2 *nad9_1 *ymf77 cob nad5 cox2 rns_a rns_b ymf56 ymf67 trnW ymf68 ymf71 cox1 nad1_a ymf62 rpl14 trnE ymf70 nad4 ymf73 rnl_b_2 trnL_2 rnl_a_2 trnM (L)	Нет событий
***T. malaccensis* (*l*)**	*trnY *rnl_a_1 *trnL_1 *rnl_b_1 *ymf57 *ymf66 *ymf76 *rps13 *rps3 *rps19 *rpl2 *ymf74 *nad10 *rps12 *nad2 nad7 rps14 ymf60 *ymf64 *ymf75 *trnF *nad1_b *atp9 *ymf63 *ymf65 *ymf69 *ymf59 *rpl16 *yejR *ymf61 *trnH *nad3 *ymf72 *nad4L *nad9_2 *nad9_1 *ymf77 cob nad5 cox2 rns_a rns_b ymf56 ymf67 trnW ymf68 ymf71 cox1 nad1_a ymf62 rpl14 trnE ymf70 nad4 ymf73 rnl_b_2 trnL_2 rnl_a_2 trnM (L)	Нет событий
T. thermophila* – *T. malaccensis	*trnY *rnl_a_1 *trnL_1 *rnl_b_1 *ymf57 *ymf66 *ymf76 *rps13 *rps3 *rps19 *rpl2 *ymf74 *nad10 *rps12 *nad2 nad7 rps14 ymf60 *ymf64 *ymf75 *trnF *nad1_b *atp9 *ymf63 *ymf65 *ymf69 *ymf59 *rpl16 *yejR *ymf61 *trnH *nad3 *ymf72 *nad4L *nad9_2 *nad9_1 *ymf77 cob nad5 cox2 rns_a rns_b ymf56 ymf67 trnW ymf68 ymf71 cox1 nad1_a ymf62 rpl14 trnE ymf70 nad4 ymf73 rnl_b_2 trnL_2 rnl_a_2 trnM (L)	1 дупликация

***T. paravorax* (*l*)**	*trnY *rnl_a_1 *trnL_1 *rnl_b_1 *ymf57 *ymf66 *ymf76 *rps13 *rps3 *rps19 *rpl2 *ymf74 *nad10 *rps12 *nad2 nad7 rps14 ymf60 *ymf64 *ymf75 *trnF *nad1_b *atp9 *ymf63 *ymf65 *ymf69 *ymf59 *rpl16 *yejR *ymf61 *trnH *nad3 *ymf72 *nad4L *nad9_1 *ymf77 cob nad5 cox2 rns_a rns_b ymf56 ymf67 trnW ymf68 ymf71 cox1 nad1_a ymf62 rpl14 trnE ymf70 nad4 ymf73 rnl_b_2 trnL_2 rnl_a_2 trnM (L)	Нет событий
T. paravorax – T. malaccensis	*trnY *rnl_a_1 *trnL_1 *rnl_b_1 *ymf57 *ymf66 *ymf76 *rps13 *rps3 *rps19 *rpl2 *ymf74 *nad10 *rps12 *nad2 nad7 rps14 ymf60 *ymf64 *ymf75 *trnF *nad1_b *atp9 *ymf63 *ymf65 *ymf69 *ymf59 *rpl16 *yejR *ymf61 *trnH *nad3 *ymf72 *nad4L *nad9_1 *ymf77 cob nad5 cox2 rns_a rns_b ymf56 ymf67 trnW ymf68 ymf71 cox1 nad1_a ymf62 rpl14 trnE ymf70 nad4 ymf73 rnl_b_2 trnL_2 rnl_a_2 trnM (L)	Нет событий
***T. pigmentosa* (*l*)**	*trnY *rnl_a_1 *trnL_1 *rnl_b_1 *ymf57 *ymf66 *ymf76 *rps13 *rps3 *rps19 *rpl2 *ymf74 *nad10 *rps12 *nad2 nad7 rps14 ymf60 *ymf64 *ymf75 trnF *nad1_b *atp9 *ymf63 *ymf65 *ymf69 *ymf59 *rpl16 *yejR *ymf61 trnH *nad3 *ymf72 *nad4L *nad9_1 *ymf77 cob nad5 cox2 rns_a rns_b ymf56 ymf67 trnW ymf68 ymf71 cox1 nad1_a ymf62 rpl14 trnE ymf70 nad4 ymf73 rnl_b_2 trnL_2 rnl_a_2 trnM (L)	2 инверсии
T. paravorax – T. pigmentosa	*trnY *rnl_a_1 *trnL_1 *rnl_b_1 *ymf57 *ymf66 *ymf76 *rps13 *rps3 *rps19 *rpl2 *ymf74 *nad10 *rps12 *nad2 nad7 rps14 ymf60 *ymf64 *ymf75 *trnF *nad1_b *atp9 *ymf63 *ymf65 *ymf69 *ymf59 *rpl16 *yejR *ymf61 *trnH *nad3 *ymf72 *nad4L *nad9_1 *ymf77 cob nad5 cox2 rns_a rns_b ymf56 ymf67 trnW ymf68 ymf71 cox1 nad1_a ymf62 rpl14 trnE ymf70 nad4 ymf73 rnl_b_2 trnL_2 rnl_a_2 trnM (L)	Нет событий
***T. pyriformis* (*l*)**	*trnY *rnl_a_1 *trnL_1 *rnl_b_1 *ymf57 *ymf66 *ymf76 *rps13 *rps3 *rps19 *rpl2 *ymf74 *nad10 *rps12 *nad2 nad7 rps14 rp16 *ymf64 *ymf75 *trnF *nad1_b *atp9 *ymf63 *ymf65 *ymf69 *ymf59 *rpl16 *yejR *ymf61 *trnH *nad3 *ymf72 *nad4L *nad9_1 *ymf77 cob nad5 cox2 rns_a rns_b ymf56 ymf67 trnW ymf68 ymf71 cox1 nad1_a ymf62 rpl14 trnE ymf70 nad4 ymf73 rnl_b_2 trnL_2 rnl_a_2 trnM (L)	1 замена гена

T. paravorax – T. pyriformis	*trnY *rnl_a_1 *trnL_1 *rnl_b_1 *ymf57 *ymf66 *ymf76 *rps13 *rps3 *rps19 *rpl2 *ymf74 *nad10 *rps12 *nad2 nad7 rps14 ymf60 *ymf64 *ymf75 *trnF *nad1_b *atp9 *ymf63 *ymf65 *ymf69 *ymf59 *rpl16 *yejR *ymf61 *trnH *nad3 *ymf72 *nad4L *nad9_1 *ymf77 cob nad5 cox2 rns_a rns_b ymf56 ymf67 trnW ymf68 ymf71 cox1 nad1_a ymf62 rpl14 trnE ymf70 nad4 ymf73 rnl_b_2 trnL_2 rnl_a_2 trnM (L)	6 вставок генов
***I. multifiliis* (*l*)**	*trnY *rnl_a_1 *rnl_b_1 *ymf66 *ymf57 *ymf76 *rps13 *rps3 *rps19 *rpl2 *nad10 *rps12 *nad2 nad7 rps14 ymf60 *ymf64 *ymf75 *trnF *atp9 *ymf63 *ymf65 *ymf59 *rpl16 *yejR *ymf61 *nad3 *nad4L *nad9_1 *ymf77 *nad1_b cob nad5 cox2 rns_a rns_b ymf67 trnW ymf68 cox1 nad1_a ymf62 rpl14 ymf70 nad4 ymf73 trnE rnl_b_2 rnl_a_2 trnY_2 (C)	2 потери генов, 1 замена гена, 1 перестановка, 1 зацикливание хромосомы
I. multifiliis – T. Pyriformis	*trnY *rnl_a_1 *rnl_b_1 *ymf57 *ymf66 *ymf76 *rps13 *rps3 *rps19 *rpl2 *nad10 *rps12 *nad2 nad7 rps14 ymf60 *ymf64 *ymf75 *trnF *nad1_b *atp9 *ymf63 *ymf65 *ymf59 *rpl16 *yejR *ymf61 *trnH *nad3 *nad4L *nad9_1 *ymf77 cob nad5 cox2 rns_a rns_b ymf56 ymf67 trnW ymf68 cox1 nad1_a ymf62 rpl14 trnE ymf70 nad4 ymf73 rnl_b_2 rnl_a_2 trnM (L)	1 вставка участка, 4 вставки генов, 1 замена гена, 1 трансверсия, 1 перестановка, 1 вставка циклической хромосомы в линейную
***P. aurelia* (*l*)**	*trnY *rnl_b *trnM *rnl_a *ymf66_1 *ymf66 *ymf57 *nad4 *ymf80 *rpl14 *ymf62 *nad1_a *cox1 *ymf68 *trnW *ymf67_b *ymf67_a *ymf56 *rns_b *rns_a *cox2 *nad5 *cob *ymf81 *ymf85 *rps13 *rps3 *rpl2 *ymf84 *nad10 *rps12 *nad2_a nad7 rps14 ymf79 ymf60 *ymf64 *ymf86 *ymf83 *atp9 *ymf63 *nad1_b *trnF *ymf65 *ymf65_1 *ymf78 *ymf59 *rpl16 *ymf82 *yejR *ymf61 *nad3 *nad4L *nad9_1 (L)	2 вставки генов 2 дупликации, 2 потери генов, 1 вставка участка
***P. caudatum* (*l*)**	*trnY *rnl_a_1 *rnl_b_1 *ymf66 *ymf57 *nad4 *ymf80 *rpl14 *nad6 *nad1_a *cox1 *ymf68 *trnW *ymf67_a *ymf56 *rns_b *rns_a *cox2 *nad5 *cob *ymf76 *rps13 *rps3 *rps19 *rpl2 *ymf84 *nad10 *rps12 *nad2_a nad7 rps14 ymf79 rpl6 *ymf64 *ymf83 *atp9 *ymf63 *ymf87 *nad1_b *trnF *ymf65 *ymf78 *ymf59 *rpl16 *yejR *ymf61 *nad3 *nad4L *nad9_1 (L)	2 замены генов, 1 потеря гена, 1 вставка гена

P. aurelia* –** ***P. caudatum	*trnY *rnl_a_1 *rnl_b_1 *trnM *ymf66 *ymf57 *nad4 *ymf80 *rpl14 *ymf62 *nad1_a *cox1 *ymf68 *trnW *ymf67_a *ymf56 *rns_b *rns_a *cox2 *nad5 *cob *ymf76 *rps13 *rps3 *rps19 *rpl2 *ymf84 *nad10 *rps12 *nad2_a nad7 rps14 ymf79 ymf60 *ymf64 *ymf83 *atp9 *ymf63 *nad1_b *trnF *ymf65 *ymf78 *ymf59 *rpl16 *yejR *ymf61 *nad3 *nad4L *nad9_1 (L)	1 потеря гена, 5 вставок генов, 1 замена гена, 1 инверсия, 1 перестановка, 1 трансверсия, 1 вставка циклической хромосомы в линейную
P. aurelia* –** ***T. pyriformis	*trnM *ymf66 *ymf76 *rps13 *rps3 *rps19 *rpl2 *nad10 *rps12 *nad2_a nad7 rps14 ymf60 *ymf64 *trnF *ymf65 *ymf59 *rpl16 *yejR *ymf61 *trnH *nad3 *nad4L *nad9_1 cob nad5 cox2 rns_a rns_b ymf56 trnW ymf68 cox1 nad1_a ymf62 rpl14 trnE nad4 ymf57 rnl_b_1 rnl_a_1 trnY(L) nad1_b ymf63 atp9 (C)	1 вставка гена, 1 вставка участка, 1 инверсия, 4 перестановки, 2 трансверсии
***M. minuta* (*l*)**	*nad5 *ccmF *cytb *trnM *rnl *cox2 *rpl14 *cox1 *nad4L *rps3_b *trnW *rns *trnY trnF nad9_1 nad2_a rpl16 nad4 rps12 nad10 rpl2 rps4 nad7 nad1_b trnG atp9 nad3 trnH nad1_a (L)	Нет событий
***M. crassus* (*l*)**	*nad5 *ccmF *cytb *trnM *rnl *cox2 *rpl14 *cox1 *nad4L *rps3_b *rns *trnY trnF nad9_1 nad2_a rpl16 nad4 rps12 nad10 rpl2 rps4 nad7 (L)	1 удаление участка, 1 потеря гена
M. minuta* – *M. crassus	*nad5 *ccmF *cytb *trnM *rnl *cox2 *rpl14 *cox1 *nad4L *rps3_b *trnW *rns *trnY trnF nad9_1 nad2_a rpl16 nad4 rps12 nad10 rpl2 rps4 nad7 nad1_b trnG atp9 nad3 trnH nad1_a (L)	3 потери гена, 1 удаление участка, 3 вставки генов, 1 инверсия

***O. trifallax* (*l*)**	*rps2_sin *nad5 *nad5_iii_sin *nad5_ii_sin *nad5_i_sin *ccmF *nad1_a *trnH *cob *nad3 *rpl6_ii_sin *rpl6_i_sin *rps7 *rps3_a_sin *atp9 *rps8_sin2 *trnG *nad1_b *rps14 *nad7 *rps4 *rps13 *rps19 *rpl2 *nad10 *rps12 *nad4_i_sin *rpl16 *nad2_a_sin *nad2_b *trnL *rps10 *trnE *nad9 *nad9_i_sin *trnF trnY trnW rps3_b nad4L cox1 nad6_sin rpl14 cox2 trnM (L)	4 вставки генов, 3 вставки участков, 3 замены генов, 2 дупликации
M. minuta – O. trifallax	*nad5 *ccmF *nad1_a *trnH *cob *nad3 *atp9 *trnG *nad1_b *rps14 *nad7 *rps4 *rps13 *rps19 *rpl2 *nad10 *rps12 *nad4 *rpl16 *nad2_a *trnE *nad9_1 *trnF trnY trnW rps3_b nad4L cox1 rpl14 cox2 trnM (L)	4 вставки генов, 1 инверсия, 1 перестановка, 1 трансверсия, 1 вставка циклической хромосомы в линейную
***N. ovalis* (*l*)**	nad1_b nad1_a nad3 nad9_1 nad4 nad10 nad4L rpl14 nad2_a nad5 rpl2a_sin rpl2 nad7 rps14_sin rps8_sin1 rps4_sin rpl6 rps12 (L)	5 потерь генов, 1 вставка участка, 1 дупликация, 1 замена гена, 4 перестановки, 1 вставка циклической хромосомы в линейную
M. minuta – N. ovalis	nad1_b atp9 rpl6 rps12 nad10 rpl2 rps19 rps13 nad7 rps14 *trnW *trnY trnF nad9_1 trnE nad2_a nad5 nad4L cox1 rpl14 cox2 trnM nad4 (L) trnH nad1_a nad3 cob(C)	4 удаления участков, 3 потери генов, 1 инверсия, 3 перестановки, 2 трансверсии, 1 вставка циклической хромосомы в линейную, 1 вырезание циклической хромосомы из линейной
The tree root	*cob *nad1_a *yejR *ymf61 *trnH *nad3 *nad4L *rpl14 *ymf62 *cox1 *ymf68 *trnW *ymf56 *rns_b *rns_a *ymf66 *rps13 *rps3 *rps19 *rpl2 *nad10 *rps12 nad9_1 trnE nad2_a nad5 cox2 trnM rpl16 ymf59 ymf65 nad7 rps14 nad4 ymf57 ymf60 *ymf64 *trnF trnY (L) nad1_b ymf63 atp9 (C)	–

2. Митохондрии споровиков видов класса Aconoidasida

Получены филогенетическое дерево и реконструкция вдоль него хромосомных структур митохондрий споровиков класса Aconoidasida. А именно, рассмотрены 15 видов из отряда Haemosporida и 3 вида из отряда Piroplasmida (таблица 4.2). Все данные получены из GenBank. Аннотация геномов уточнялась с помощью BLAST и Rfam. Как видно из таблицы 2, даже близкие из этих видов имеют кольцевые и линейные митохондриальные хромосомы; каждая структура содержит лишь одну хромосому. Однако в них отсутствуют паралоги.

Таблица 4.2. Хромосомные структуры митохондрий видов класса Aconoidasida
Циклические и линейные хромосомы помечены *C* и *L* соответственно. Звёздочка помечают гены, расположенные на комплементарной цепи. В правом столбце показаны последовательности генов, входящих в структуру, с использованием стандартных имён.

Отряд	Виды	Локус в GenBank	Тип	Структура
Haemosporida	*Leucocytozoon fringillinarum*	FJ168564.1	*C*	ss3 ls3 ls9 ss2 ls4 *cox3 ls8 ss5 ss1 cox1 cytb ls1 ss6 ls7 ss4
	Leucocytozoon majoris	FJ168563.1	*C*	ss3 ls3 ls9 ss2 ls4 *cox3 ls8 ss5 ss1 cox1 cytb ls1 ss6 ls7
	Leucocytozoon sabrazesi	NC_009336.1	*L*	ls1 ss4 ss6 ls7 ls6 ss3 ls3 ls9 ss2 ls4 ls5 *cox3 ls8 ss5 ss1 cox1 cytb ls2
	Plasmodium berghei	NC_015303.1	*L*	ls1 ss4 ss6 ls7 ls6 ss3 ls3 ls9 ss2 ls4 ls5 *cox3 ls8 ss5 ss1 cox1 cytb ls2
	Plasmodium falciparum	NC_002375.1	*L*	ss3 ls3 ls9 ss2 *cox3 ls8 ss5 ss1 cox1 cytb ls1 ss4 ss6 ls7
	Plasmodium floridense	NC_009961.2	*L*	ss3 ls3 ls9 ss2 ls4 *cox3 ls8 ss5 ss1 cox1 cytb ls1 ss4 ss6 ls7
	Plasmodium fragile	AY722799.1	*C*	ls1 ss6 ls7 ss3 ls3 ls9 ss2 ls4 *cox3 ls8 ss5 ss1 cox1 cytb
	Plasmodium gallinaceum	NC_008288.1	*L*	ls1 ss4 ss6 ls7 ls6 ss3 ls3 ls9 ss2 ls4 ls5 *cox3 ls8 ss5 ss1 cox1 cytb ls2
	Plasmodium juxtanucleare	NC_008279.1	*L*	ls1 ss4 ss6 ls7 ls6 ss3 ls3 ls9 ss2 ls4 ls5 *cox3 ls8 ss5 ss1 cox1 cytb ls2
	Plasmodium knowlesi	NC_007232.1	*C*	ls1 ss6 ls7 ss3 ls3 ls9 ss2 ls4 *cox3 ls8 ss5 ss1 cox1 cytb
	Plasmodium mexicanum	NC_009960.2	*L*	ss3 ls3 ls9 ss2 *cox3 ls8 ss5 ss1 cox1 cytb ls1 ss4 ss6 ls7
	Plasmodium reichenowi	NC_002235.1	*L*	ss3 ls3 ls9 ss2 *cox3 ls8 ss5 ss1 cox1 cytb ls1 ss4 ss6 ls7
	Plasmodium relictum	NC_012426.1	*C*	ss3 ls3 ls9 ss2 ls4 *cox3 ls8 ss5 ss1 cox1 cytb ls1 ss4 ss6 ls7
	Plasmodium simium	NC_007233.1	*C*	ls1 ss6 ls7 ss3 ls3 ls9 ss2 ls4 *cox3 cox1 cytb ls8 ss5 ss1

	Plasmodium vivax	NC_007243.1	*C*	ls1 ss6 ls7 ss3 ls3 ls9 ss2 ls4 *cox3 ls8 ss5 ss1 cox1 cytb
Piroplasmida	*Babesia bovis*	NC_009902.1	*L*	cox1 *cox3 ls1 *ls2 *ls3 *cytb *ls4 ls5
	Theileria parva	NC_011005.1	*L*	cox1 *cox3 ls1 *ls3 *cytb *ls5 ls4
	Theileria annulata	CR940346.1	*L*	cox1 *cox3 ls1 *ls3 *ls2 *cytb *ls5 ls4

Эволюция хромосомных структур. Для 18 видов из таблицы 4.2 алгоритмом из раздела 1.2 вычислена матрица попарных расстояний, по которой с помощью алгоритма UPGMA построено дерево эволюции (рисунок 72). Построенное дерево состоит из двух клад, включающих митохондрии пироплазмид (роды *Babesia* и *Theileria*) и гемоспоридий (роды *Plasmodium* и *Leucocytozoon*). Роды *Plasmodium* и *Leucocytozoon* не разделены на дереве, в частности, из-за наличия линейных и кольцевых хромосом в митохондриях.

Для уменьшения размера дерева листья с одинаковыми структурами в них объединены в одном листе, в каждой такой группе одинаковых структур выбран вид-представитель, и на дереве указано его имя. А именно, листу, соответствующему группе, приписано подчеркнутое. Это – три группы: *Plasmodium vivax*, *Leucocytozoon majoris*, *Plasmodium fragile*, *Plasmodium knowlesi*; *Plasmodium falciparum*, *Plasmodium reichenowi*, *Plasmodium mexicanum*; *Leucocytozoon sabrazesi*, *Plasmodium juxtanucleare*, *Plasmodium gallinaceum*, *Plasmodium berghei*. Корень дерева выбран так, чтобы получилось дерево, наиболее сбалансированное по расстоянию от корня до листьев.

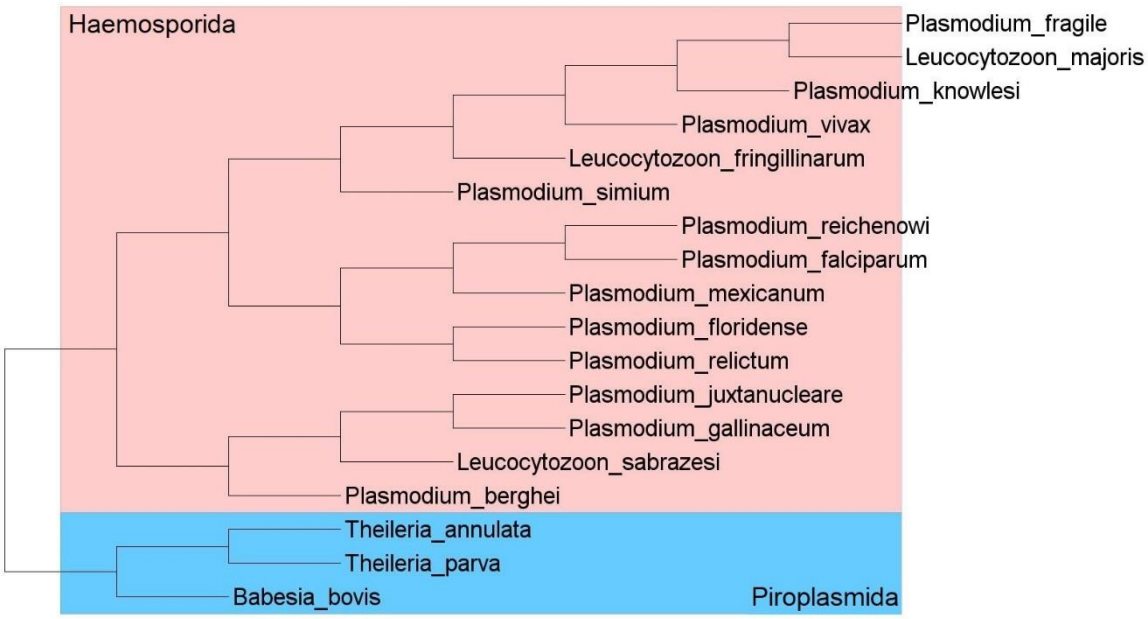


Рисунок 72. Дерево хромосомных структур митохондрий споровиков из класса Aconoidasida, построенное по матрице расстояний алгоритмом UPGMA.

Реконструкция хромосомных структур. Для полученного дерева (рисунок 72) проведена реконструкция хромосомных структур во внутренних вершинах. Для контроля реконструкция выполнялась двумя алгоритмами: эвристическим алгоритмом спуска [молбио] и сведение к ЦЛП, которое описано в разделе 3.3.

Первый алгоритм не был описан выше, поэтому сделаем это сейчас. С помощью алгоритма из раздела 2.2 решалась задача реконструкции для специального расстояния, результат которой рассматривался как начальная расстановка структур. Затем перебирались все внутренние вершины дерева и все операции над структурами с целью найти пару <вершина, операция>, которая максимально понижает суммарное минимальное расстояние для текущей расстановки. Найденная операция применялась к структуре в текущей вершине, если в результате операции получалась расстановка с меньшим суммарным расстоянием. Такая процедура применялась до тех пор, пока расстояние для текщей расстановки уменьшалось. Результаты реконструкции с помощью алгоритма спуска приведены в таблице 4.3.

Таблица 4.3. Филогенетическая реконструкция алгоритмом спуска хромосомных структур митохондрий споровиков класса Aconoidasida
Обозначения такие же, как в таблице 4.1. Порядок строк в таблице соответствует порядку обхода дерева на рисунке 72 от корня до листа и снизу вверх

Plasmodium fragile – Babesia bovis	ss1 cox1 *cox3 ls1 *ls3 *ss3 *ls6 ls8 ss5 (C) *ls7 *ss6 *ss4 ls9 ss2 ls4 ls5 cytb ls2 (L)
Theileria annulata – Babesia bovis	*ls4 ls5 cytb ls2 ls3 *ls1 cox3 *cox1 (L)
Theileria annulata – Theileria parva	*ls4 ls5 cytb ls2 ls3 *ls1 cox3 *cox1 (L)
Theileria annulata (*l*)	cox1 *cox3 ls1 *ls3 *cytb *ls5 ls4 (L)
Theileria parva (*l*)	cox1 *cox3 ls1 *ls3 *ls2 *cytb *ls5 ls4 (L)
Babesia bovis (*l*)	cox1 *cox3 ls1 *ls2 *ls3 *cytb *ls4 ls5 (L)
Plasmodium fragile – Plasmodium berghei	*ls7 *ss6 *ss4 *ls1 ls6 ss3 ls3 ls9 ss2 ls4 ls5 *cox3 ls8 ss5 ss1 cox1 cytb ls2 (L)
Plasmodium juxtanucleare – Plasmodium berghei	ls1 ss4 ss6 ls7 ls6 ss3 ls3 ls9 ss2 ls4 ls5 *cox3 ls8 ss5 ss1 cox1 cytb ls2 (L)
Plasmodium juxtanucleare – Leucocytozoon sabrazesi	ls1 ss4 ss6 ls7 ls6 ss3 ls3 ls9 ss2 ls4 ls5 *cox3 ls8 ss5 ss1 cox1 cytb ls2 (L)
Plasmodium juxtanucleare – Plasmodium gallinaceum	ls1 ss4 ss6 ls7 ls6 ss3 ls3 ls9 ss2 ls4 ls5 *cox3 ls8 ss5 ss1 cox1 cytb ls2 (L)
Plasmodium juxtanucleare (*l*)	ls1 ss4 ss6 ls7 ls6 ss3 ls3 ls9 ss2 ls4 ls5 *cox3 ls8 ss5 ss1 cox1 cytb ls2 (L)
Plasmodium gallinaceum (*l*)	ls1 ss4 ss6 ls7 ls6 ss3 ls3 ls9 ss2 ls4 ls5 *cox3 ls8 ss5 ss1 cox1 cytb ls2 (L)
Leucocytozoon sabrazesi (*l*)	ls1 ss4 ss6 ls7 ls6 ss3 ls3 ls9 ss2 ls4 ls5 *cox3 ls8 ss5 ss1 cox1 cytb ls2 (L)
Plasmodium berghei (*l*)	ls1 ss4 ss6 ls7 ls6 ss3 ls3 ls9 ss2 ls4 ls5 *cox3 ls8 ss5 ss1 cox1 cytb ls2 (L)
Plasmodium fragile – Plasmodium relictum	ss3 ls3 ls9 ss2 ls4 *cox3 ls8 ss5 ss1 cox1 cytb ls1 ss4 ss6 ls7 (L)
Plasmodium reichenowi – Plasmodium relictum	ss3 ls3 ls9 ss2 ls4 *cox3 ls8 ss5 ss1 cox1 cytb ls1 ss4 ss6 ls7 (L)
Plasmodium floridense – Plasmodium relictum	ss3 ls3 ls9 ss2 ls4 *cox3 ls8 ss5 ss1 cox1 cytb ls1 ss4 ss6 ls7 (L)
Plasmodium floridense (*l*)	ss3 ls3 ls9 ss2 ls4 *cox3 ls8 ss5 ss1 cox1 cytb ls1 ss4 ss6 ls7 (L)
Plasmodium relictum (*l*)	ss3 ls3 ls9 ss2 ls4 *cox3 ls8 ss5 ss1 cox1 cytb ls1 ss4 ss6 ls7 (C)
Plasmodium reichenowi – Plasmodium mexicanum	ss3 ls3 ls9 ss2 *cox3 ls8 ss5 ss1 cox1 cytb ls1 ss4 ss6 ls7 (L)
Plasmodium reichenowi – Plasmodium falciparum	ss3 ls3 ls9 ss2 *cox3 ls8 ss5 ss1 cox1 cytb ls1 ss4 ss6 ls7 (L)
Plasmodium reichenowi (*l*)	ss3 ls3 ls9 ss2 *cox3 ls8 ss5 ss1 cox1 cytb ls1 ss4 ss6 ls7 (L)
Plasmodium falciparum (*l*)	ss3 ls3 ls9 ss2 *cox3 ls8 ss5 ss1 cox1 cytb ls1 ss4 ss6 ls7 (L)
Plasmodium mexicanum (*l*)	ss3 ls3 ls9 ss2 *cox3 ls8 ss5 ss1 cox1 cytb ls1 ss4 ss6 ls7 (L)
Plasmodium fragile – Plasmodium simium	ss1 cox1 cytb ls1 ss4 ss6 ls7 ss3 ls3 ls9 ss2 ls4 *cox3 ls8 ss5 (C)
Plasmodium fragile – Leucocytozoon fringillinarum	ss1 cox1 cytb ls1 ss4 ss6 ls7 ss3 ls3 ls9 ss2 ls4 *cox3 ls8 ss5 (C)
Plasmodium fragile – Plasmodium vivax	ss1 cox1 cytb ls1 ss6 ls7 ss3 ls3 ls9 ss2 ls4 *cox3 ls8 ss5 (C)
Plasmodium fragile – Plasmodium knowlesi	ss1 cox1 cytb ls1 ss6 ls7 ss3 ls3 ls9 ss2 ls4 *cox3 ls8 ss5 (C)
Plasmodium fragile – Leucocytozoon majoris	ss1 cox1 cytb ls1 ss6 ls7 ss3 ls3 ls9 ss2 ls4 *cox3 ls8 ss5 (C)
Plasmodium fragile (*l*)	ls1 ss6 ls7 ss3 ls3 ls9 ss2 ls4 *cox3 ls8 ss5 ss1 cox1 cytb (C)
Leucocytozoon majoris (*l*)	ss3 ls3 ls9 ss2 ls4 *cox3 ls8 ss5 ss1 cox1 cytb ls1 ss6 ls7 (C)
Plasmodium knowlesi (*l*)	ls1 ss6 ls7 ss3 ls3 ls9 ss2 ls4 *cox3 ls8 ss5 ss1 cox1 cytb (C)
Plasmodium vivax (*l*)	ls1 ss6 ls7 ss3 ls3 ls9 ss2 ls4 *cox3 ls8 ss5 ss1 cox1 cytb (C)
Leucocytozoon fringillinarum (*l*)	ss3 ls3 ls9 ss2 ls4 *cox3 ls8 ss5 ss1 cox1 cytb ls1 ss6 ls7 ss4 (C)
Plasmodium simium (*l*)	ls1 ss6 ls7 ss3 ls3 ls9 ss2 ls4 *cox3 cox1 cytb ls8 ss5 ss1 (C)

В таблице 4.4 представлена реконструкция хромосомных структур для того же дерева на рисунке 72, полученнная сведением задачи к ЦЛП.

Таблица 4.4. Реконструкция для тех же данных, что в таблице 4.3, полученная сведением к ЦЛП.

Вершина	Структуры
Plasmodium fragile – Babesia bovis	*ls5 ls6 ls2 (L) ss4 ss6 ls7 ss3 ls3 ls9 ss2 ls4 *cox3 ls8 ss5 ss1 cox1 cytb ls1 (C)
Theileria annulata – Babesia bovis	cox1 *cox3 ls1 *ls3 *cytb *ls5 ls4 (L)
Theileria annulata – Theileria parva	cox1 *cox3 ls1 *ls3 *cytb *ls5 ls4 (L)
Theileria annulata (l)	cox1 *cox3 ls1 *ls3 *cytb *ls5 ls4 (L)
Theileria parva (l)	cox1 *cox3 ls1 *ls3 *ls2 *cytb *ls5 ls4 (L)
Babesia bovis (l)	cox1 *cox3 ls1 *ls2 *ls3 *cytb *ls4 ls5 (L)
Plasmodium fragile – Plasmodium berghei	ss4 ss6 ls7 ss3 ls3 ls9 ss2 ls4 *cox3 ls8 ss5 ss1 cox1 cytb ls1 (C) ls2 (L)
Plasmodium juxtanucleare – Plasmodium berghei	ls1 ss4 ss6 ls7 ls6 ss3 ls3 ls9 ss2 ls4 ls5 *cox3 ls8 ss5 ss1 cox1 cytb ls2 (L)
Plasmodium juxtanucleare – Leucocytozoon sabrazesi	ls1 ss4 ss6 ls7 ls6 ss3 ls3 ls9 ss2 ls4 ls5 *cox3 ls8 ss5 ss1 cox1 cytb ls2 (L)
Plasmodium juxtanucleare – Plasmodium gallinaceum	ls1 ss4 ss6 ls7 ls6 ss3 ls3 ls9 ss2 ls4 ls5 *cox3 ls8 ss5 ss1 cox1 cytb ls2 (L)
Plasmodium juxtanucleare (l)	ls1 ss4 ss6 ls7 ls6 ss3 ls3 ls9 ss2 ls4 ls5 *cox3 ls8 ss5 ss1 cox1 cytb ls2 (L)
Plasmodium gallinaceum (l)	ls1 ss4 ss6 ls7 ls6 ss3 ls3 ls9 ss2 ls4 ls5 *cox3 ls8 ss5 ss1 cox1 cytb ls2 (L)
Leucocytozoon sabrazesi (l)	ls1 ss4 ss6 ls7 ls6 ss3 ls3 ls9 ss2 ls4 ls5 *cox3 ls8 ss5 ss1 cox1 cytb ls2 (L)
Plasmodium berghei (l)	ls1 ss4 ss6 ls7 ls6 ss3 ls3 ls9 ss2 ls4 ls5 *cox3 ls8 ss5 ss1 cox1 cytb ls2 (L)
Plasmodium fragile – Plasmodium relictum	ss4 ss6 ls7 ss3 ls3 ls9 ss2 ls4 *cox3 ls8 ss5 ss1 cox1 cytb ls1 (C) ls2 (L)
Plasmodium reichenowi – Plasmodium relictum	ss4 ss6 ls7 ss3 ls3 ls9 ss2 ls4 *cox3 ls8 ss5 ss1 cox1 cytb ls1 (C) ls2 (L)
Plasmodium floridense – Plasmodium relictum	ls7 ss3 ls3 ls9 ss2 ls4 *cox3 ls8 ss5 ss1 cox1 cytb ls1 ss6 (C)
Plasmodium floridense (l)	ss3 ls3 ls9 ss2 ls4 *cox3 ls8 ss5 ss1 cox1 cytb ls1 ss4 ss6 ls7 (L)
Plasmodium relictum (l)	ss3 ls3 ls9 ss2 ls4 *cox3 ls8 ss5 ss1 cox1 cytb ls1 ss4 ss6 ls7 (C)
Plasmodium reichenowi – Plasmodium mexicanum	ss3 ls3 ls9 ss2 *cox3 ls8 ss5 ss1 cox1 cytb ls1 ss4 ss6 ls7 (L)
Plasmodium reichenowi – Plasmodium falciparum	ss3 ls3 ls9 ss2 *cox3 ls8 ss5 ss1 cox1 cytb ls1 ss4 ss6 ls7 (L)
Plasmodium reichenowi (l)	ss3 ls3 ls9 ss2 *cox3 ls8 ss5 ss1 cox1 cytb ls1 ss4 ss6 ls7 (L)
Plasmodium falciparum (l)	ss3 ls3 ls9 ss2 *cox3 ls8 ss5 ss1 cox1 cytb ls1 ss4 ss6 ls7 (L)
Plasmodium mexicanum (l)	ss3 ls3 ls9 ss2 *cox3 ls8 ss5 ss1 cox1 cytb ls1 ss4 ss6 ls7 (L)
Plasmodium fragile – Plasmodium simium	ss4 ss6 ls7 ss3 ls3 ls9 ss2 ls4 *cox3 ls8 ss5 ss1 cox1 cytb ls1 (C)
Plasmodium fragile – Leucocytozoon fringillinarum	ss3 ls3 ls9 ss2 ls4 *cox3 ls8 ss5 ss1 cox1 cytb ls1 ss4 ss6 ls7 (L)
Plasmodium fragile – Plasmodium vivax	ls1 ss6 ls7 ss3 ls3 ls9 ss2 ls4 *cox3 ls8 ss5 ss1 cox1 cytb (C)
Plasmodium fragile – Plasmodium knowlesi	ls1 ss6 ls7 ss3 ls3 ls9 ss2 ls4 *cox3 ls8 ss5 ss1 cox1 cytb (C)
Plasmodium fragile – Leucocytozoon majoris	ls1 ss6 ls7 ss3 ls3 ls9 ss2 ls4 *cox3 ls8 ss5 ss1 cox1 cytb (C)

Plasmodium fragile (l)	ls1 ss6 ls7 ss3 ls3 ls9 ss2 ls4 *cox3 ls8 ss5 ss1 cox1 cytb (C)
Leucocytozoon majoris (l)	ss3 ls3 ls9 ss2 ls4 *cox3 ls8 ss5 ss1 cox1 cytb ls1 ss6 ls7 (C)
Plasmodium knowlesi (l)	ls1 ss6 ls7 ss3 ls3 ls9 ss2 ls4 *cox3 ls8 ss5 ss1 cox1 cytb (C)
Plasmodium vivax (l)	ls1 ss6 ls7 ss3 ls3 ls9 ss2 ls4 *cox3 ls8 ss5 ss1 cox1 cytb (C)
Leucocytozoon fringillinarum (l)	ss3 ls3 ls9 ss2 ls4 *cox3 ls8 ss5 ss1 cox1 cytb ls1 ss6 ls7 ss4 (C)
Plasmodium simium (l)	ls1 ss6 ls7 ss3 ls3 ls9 ss2 ls4 *cox3 cox1 cytb ls8 ss5 ss1 (C)

Два использованных алгоритма привели к незначительно отличающимся деревьям. Например, ген *ls2*, кодирующий фрагмент большой субъединицы рРНК становится отдельной линейной хромосомой во внутренних вершинах дерева. Хотя гены рРНК редко разделяются, такой феномен может говорить о высокой мобильности конкретного фрагмента.

3. Пластиды родофитной ветви

Из GenBank выбраны пластидные геномы родофитной ветви 66 видов с довольно сложной структурой. Белки, кодируемые в них, кластеризованы алгоритмом, описанным в [45, 46, 75] с параметрами E=0.001, L=0, H=0.6. По так полученным кластерам уточнена ортологичность генов. База данных этих кластеров доступна в [74]. Виды и сведения об их кластеризации представлены в таблице 4.5.

Таблица 4.5. Шестьдесят шесть видов родофитной ветви, кодируемые в их пластдах белки, и соответствующие семейства (кластеры) белков. Здесь #Prot – число белков, кодируемых в пластидах указанного вида; #clust – число кластеров, в которых представлен вид; #sing – число белков вида, кодируемых в пластидах и не включенных ни в один кластер (синглетонов).

Locus in GenBank	Species	#prot	#clust	#sing
NC_024079.1	*Asterionella formosa*	134	129	0
NC_024080.1	*Asterionellopsis glacialis*	145	138	1
NC_012898.1	*Aureococcus anophagefferens*	105	105	0
NC_012903.1	*Aureoumbra lagunensis*	110	110	0
NC_011395.1	*Babesia bovis T2Bo*	32	22	7
NC_021075.1	*Calliarthron tuberculosum*	201	200	1
NC_025313.1	*Cerataulina daemon*	132	130	0
NC_025310.1	*Chaetoceros simplex*	131	128	0
NC_020795.1	*Chondrus crispus*	204	204	0
NC_026522.1	*Choreocolax polysiphoniae*	71	71	0
NC_014340.2	*Chromera velia*	78	51	24
NC_014345.1	*Chromerida sp. RM11*	81	69	5
NC_024081.1	*Coscinodiscus radiatus*	139	130	0

Locus in GenBank	Species	#prot	#clust	#sing
NC_013703.1	*Cryptomonas paramecium*	82	79	3
NC_004799.1	*Cyanidioschyzon merolae strain 10D*	207	189	18
NC_001840.1	*Cyanidium caldarium*	197	186	11
NC_024082.1	*Cylindrotheca closterium*	161	141	13
NC_024083.1	*Didymosphenia geminata*	130	128	0
NC_014287.1	*Durinskia baltica*	129	127	0
NC_013498.1	*Ectocarpus siliculosus*	148	143	1
NC_004823.1	*Eimeria tenella strain Penn State*	28	21	7
NC_007288.1	*Emiliania huxleyi*	119	112	7
NC_024928.1	*Eunotia naegelii*	160	136	2
NC_015403.1	*Fistulifera solaris*	135	130	1
NC_016735.1	*Fucus vesiculosus*	139	139	0
NC_024665.1	*Galdieria sulphuraria*	182	181	1
NC_023785.1	*Gracilaria salicornia*	202	200	2
NC_006137.1	*Gracilaria tenuistipitata var. liui*	203	201	2
NC_021618.1	*Grateloupia taiwanensis*	233	201	32
NC_000926.1	*Guillardia theta*	147	142	5
NC_010772.1	*Heterosigma akashiwo*	156	139	3
NC_014267.1	*Kryptoperidinium foliaceum*	139	132	6
NC_027093.1	*Lepidodinium chlorophorum*	62	52	7
NC_024084.1	*Leptocylindrus danicus*	132	130	0
NC_022667.1	*Leucocytozoon caulleryi*	30	30	0
NC_024085.1	*Lithodesmium undulatum*	138	129	0
NC_020014.1	*Nannochloropsis gaditana*	119	116	3
NC_022259.1	*Nannochloropsis granulata*	125	123	0
NC_022262.1	*Nannochloropsis limnetica*	124	123	0
NC_022263.1	*Nannochloropsis oceanica*	126	123	1
NC_022260.1	*Nannochloropsis oculata*	126	123	0
NC_022261.1	*Nannochloropsis salina*	123	123	0
NC_001713.1	*Odontella sinensis*	140	128	9
NC_020371.1	*Pavlova lutheri*	111	102	9
NC_016703.2	*Phaeocystis antarctica*	108	108	0
NC_021637.1	*Phaeocystis globosa*	108	108	0
NC_008588.1	*Phaeodactylum tricornutum*	132	130	0
NC_023293.1	*Plasmodium chabaudi chabaudi*	31	31	0
NC_000925.1	*Porphyra purpurea*	209	209	0
NC_023133.1	*Porphyridium purpureum*	224	183	40
NC_021189.1	*Pyropia haitanensis*	211	210	1

Locus in GenBank	Species	#prot	#clust	#sing
NC_024050.1	*Pyropia perforata*	209	207	2
NC_007932.1	*Pyropia yezoensis*	209	206	3
NC_025311.1	*Rhizosolenia imbricata*	135	123	1
NC_009573.1	*Rhodomonas salina*	146	143	3
NC_025312.1	*Roundia cardiophora*	140	126	0
NC_018523.1	*Saccharina japonica*	139	139	0
NC_014808.1	*Thalassiosira oceanica CCMP1005*	142	126	1
NC_008589.1	*Thalassiosira pseudonana*	141	127	0
NC_025314.1	*Thalassiosira weissflogii*	141	127	0
NC_007758.1	*Theileria parva strain Muguga*	44	27	12
NC_001799.1	*Toxoplasma gondii RH*	26	21	5
NC_026851.1	*Trachydiscus minutus*	137	124	8
NC_016731.1	*Ulnaria acus*	130	128	0
NC_011600.1	*Vaucheria litorea*	139	138	1
NC_026523.1	*Vertebrata lanosa*	192	191	1

В хромосомные структуры включались только пластидные гены, присутствующие во многих видах и кодирующие следующие белки. Это – шапероны *clpC*; субъединицы фотосистемы I *psaA, psaB, psaC, psaD, psaE, psaF, psaI, psaJ, psaK, psaL,* и *psaM*; субъединицы фотосистемы II *psb28, psb30, psbA, psbB, psbC, psbD, psbE, psbF, psbH, psbI, psbJ, psbK, psbL, psbN, psbT, psbV, psbX, psbY*, и *psbZ*; большая субъединица rubisco *rbcL*; субъединицы РНК полимеразы *rpoA, rpoB, rpoC1, rpoC2,* и *rpoZ*; рибосомные белки *rpl1, rpl2, rpl3,rpl4, rpl5, rpl6, rpl9, rpl11, rpl12, rpl13, rpl14, rpl16, rpl18, rpl19, rpl20, rpl21, rpl22, rpl23, rpl24, rpl27, rpl28, rpl29, rpl31, rpl32, rpl33, rpl34, rpl35, rpl36, rps1, rps2, rps20, rps3, rps4, rps5, rps6, rps7, rps8, rps9, rps10, rps11, rps12, rps13, rps14, rps16, rps17, rps18,* и *rps19*; фактор элонгации *tufA*. Паралоги гена *psbY* можно найти в *Odontella sinensis, Phaeodactylum tricornutum, Thalassiosira pseudonana, Thalassiosira oceanica, Ulnaria acus, Asterionella formosa, Asterionellopsis glacialis, Didymosphenia geminata, Lithodesmium undulatum, Eunotia naegelii, Chaetoceros simplex, Roundia cardiophora, Cerataulina daemon, Thalassiosira weissflogii*. Паралоги гена *clpC* можно найти в *Theileria parva, Babesia bovis, Chromera velia, Thalassiosira oceanica, Nannochloropsis gaditana, Nannochloropsis granulata, Nannochloropsis oculata, Nannochloropsis salina, Nannochloropsis limnetica, Nannochloropsis oceanica, Rhizosolenia imbricata*. Последовательные и сонаправленные паралоги гена *rpoC2* присутствуют в *Theileria parva, Leucocytozoon caulleryi, Plasmodium chabaudi*. В *Rhizosolenia imbricata*

присутствует большой повтор, включающий гены *psbA, psaC, rps6, clpC, rps10, rps7, rps12*. В ходе подготовки данных были найдены несколько ошибок в аннотациях: *rpo* вместо *proC1* в *Nannochloropsis gaditana*, *rpoC* вместо *proC1* в *Cyanidioschyzon merolae*, *rpoC2-n-terminal* вместо *proC2* в *Babesia bovis*. Здесь все хромосомы кольцевые.

Филогенетическое дерево хромосомных структур пластид родофитной ветви. Дерево, показанное на рисунке 73, построено по матрице попарных расстояний, которая вычислена алгоритмом из раздела 1.2.

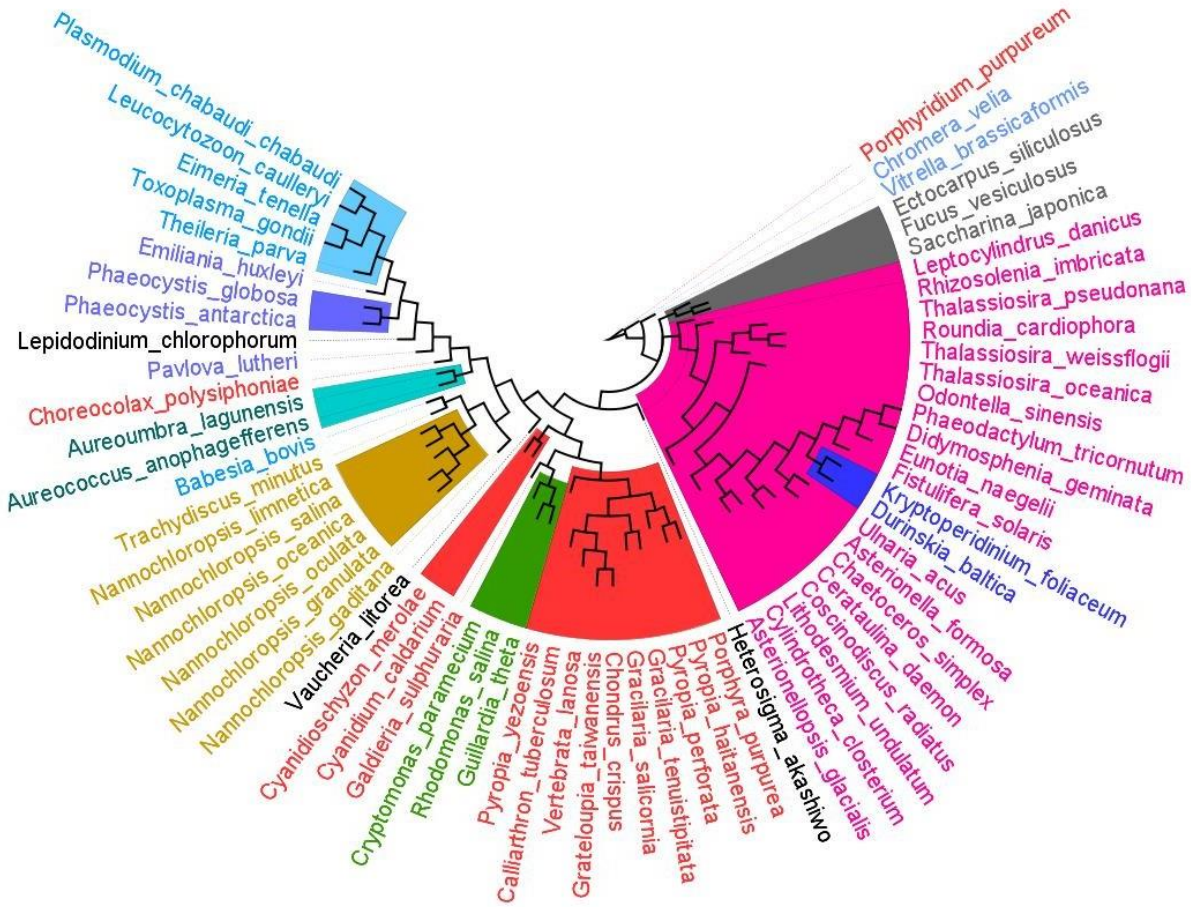


Рисунок 73. Дерево хромосомных структур пластид родофитной ветви
Хромосомные структуры, которые использованы для вычисления матрицы расстояний, приведены в таблицах 6*a* и 6*b* в строках, помеченных знаком (*l*)

Полученное дерево хорошо согласуется с известными результатами о филогении соответствующих видов. Наиболее значительные различия – позиции фотосинтезирующей альвеоляты *Chromera velia* и родофитной водоросли *Porphyridium purpureum*, у которых порядки генов в пластидных хромосомах значительно отличаются от порядков пластидных генов в родственных видах. Это отмечалось при изучении

регуляции гена *moeB* [47]. Отдельная клада сформирована пластидами рода *Nannochloropsis,* которые образуют изолированную группу в большой группе Stramenopiles [58].

Все диатомовые водоросли образуют большую кладу, включающую также некоторые страменопилы и два вида альвеолят *Durinskia baltica* и *Kryptoperidinium foliaceum,* у которых пластиды произошли от пластид диатомовых водорослей [59]

Другая большая клада сформирована пластидами родофитных водорослей, исключая *Porphyridium purpureum*, криптофитовыми водорослями, некоторыми альвеолятами, гаптофитовыми водорослями и страменопилами *Aureococcus anophagefferens* и *Aureoumbra lagunensis* [60], а также рафидофитовыми водорослями *Heterosigma akashiwo* [61] и желто-зелеными водорослями (Xanthophyta) *Vaucheria litorea*.

Все бурые водоросли *Ectocarpus siliculosus, Fucus vesiculosus*, и *Saccharina japonica* [62, 63] образуют еще одну кладу.

Виды альвеолят, у которых пластиды похожи на пластиды родофитных водорослей, включают всех рассмотренных споровиков, а также фотосинтезирующую алвеолату *Chromerida* sp. RM11. Общий предок этих пластид подтвержден с помощью выравнимания белков [64, 65]. Более того, в [66] предсказана однотипная экспрессия *ycf24* (*sufB*) в пластидах споровиков и некоторых родофитных водорослей, что также подтверждает их близкое расположение на построенном дереве. Значительное разнообразие в пластидах Страменопил и Гаптофитовых водорослей было отмечено в [67]. Тем не менее, независимое происхождение криптофитовых пластид не подтверждено. В целом, можно утверждать, что пластиды родофитной ветви являются монофилетической группой и происходят от пластид родофитных водорослей, однако для криптофитовых водорослей и споровиков обоснованность этого утверждения неясна.

В [68] дерево хромосомных структур пластид споровиков построено с помощью другого подхода. То дерево весьма похоже на соответствующее поддерево в нашем дереве пластид. А именно, *Chromerida* в обоих деревьях расположена на рано отделившейся ветви. Согласно длинам путей, *Plasmodium* расположен рядом с *Toxoplasma*, и оба они расположены рядом с *Theileria*. В то же время, деревья различаются, причиной, вероятно, является различное число рассматриваемых генов и видов; в нашем случае присутствует много далеких видов.

Реконструкция хромосомных структур пластид родофитной ветви вдоль дерева их эволюции. Рассмотрим реконструкции на двух поддеревьях полученного дерева (рисунке 73). Первое дерево – от общего предка *Leptocylindrus danicus* и *Odontella sinensis* («малое дерево»), а второе – от общего предка *Porphyra purpurea* и *Vaucheria litorea* («большое дерево»). Получившиеся структуры показаны в таблицах 4.6*a*, 4.6*b* (для малого и большого дерева соответственно). Обе реконструкции были получены с использованием Целочисленного линейного программирования с двумя миллионами переменных и четырьмя миллионами линейных ограничений. Паралоги генов *psbY*, *rpoC2*, *clpC* и некоторых других генов различаются индексами. Повторим, что все хромосомы кольцевые. Эволюционные события (составляющие сценарии) для этих реконструкций показаны на рисунках 74*a* и 74*b*.

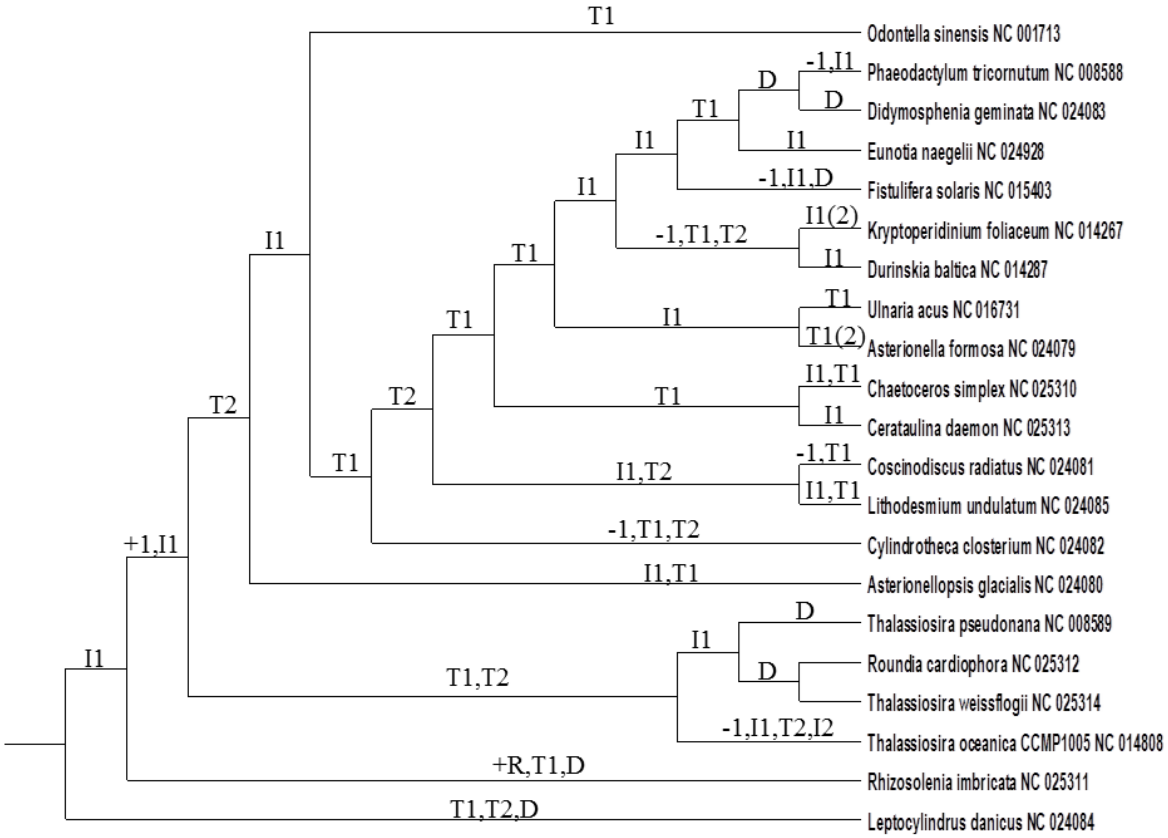


Рисунок 74*a*. Эволюционный сценарий хромосомных структур вдоль малого дерева.
Обозначения событий: −1 – потеря одного из двух паралогов гена *psbY*, +1 – возникновение паралога гена *psbY*, +R – возниконовение инвертированного повтора участка хромосомы; I1 – инверсия участка хромосомы, T1 – трансверсия участка хромосомы; T2 – перестановка участка хромосомы; I2 – вставка участка хромосомы,

D – удаление участка хромосомы. Число событий указано в скобках, если их больше одного.

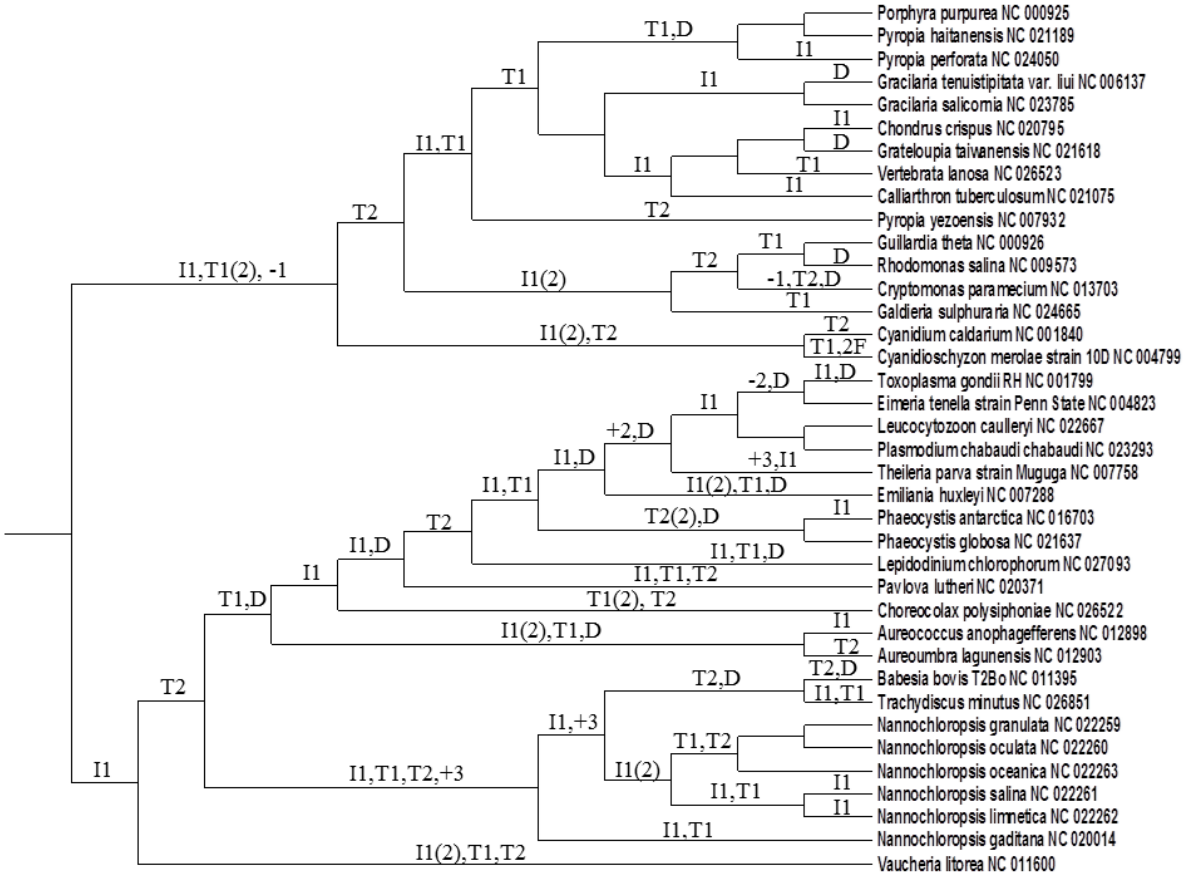


Рисунок 74*b*. Эволюционный сценарий хромосомных структур вдоль большого дерева

Обозначения событий: −1 – потеря гена *psbY*; −2 – потеря одного из двух паралогов гена *rpoC2*; +2 – возникновение паралога гена *rpoC2*; +3 – возникновение паралога гена *clpC*; I1 – инверсия участка хромосомы; T1 – трансверсия участка хромосомы; T2 – перестановка участка хромосомы; 2F – слияние двух паралогов гена *rpoC2* в один большой ген, и D – удаление участка хромосомы.

Таблица 4.6*a*. Реконструкция хромосомных структур пластид родофитной ветви вдоль малого дерева. Строки, помеченные (*l*) соответствуют листьям дерева и исходным хромосомным структурам. Левый столбец указывает и на внутренние вершины дерева путём указания первого и последнего листа на дереве с рисунка 74*a*. В правом столбце указанна структура, приписанная соответствующей вершине. Номера паралогов указаны после подчёркивания. Вертикальная черта разделяет хромосомы. Все хромосомы кольцевые.

Вершина	Структуры
Leptocylindrus danicus – Odontella sinensis	rbcl *psbz rpl35_1 rpl20_1 psam psad *rpl12 *rpl1 *rpl11 *rps14 psae_1 psbx psbv rpl19 *psab *psaa *psaj *psaf *psb1 *ycf39 *psal psbe psbf psbl psbj *psbc *psbd *rps2 *rpoc2_1 *rpoc1 *rpob *rps20 rpl33 rps18 ycf3 ycf33 *psa1 psbk *psbh psbn *psbt *psbb *ycf31 *psbh psbn *psbt *psbb *rpoc1 | rpl32_1 rpl21_1 rpl27_1 rpl34_1 psba_1 psby_1 psac_1 rps6_1 *clpc_1 *rps10_1 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl24 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4

	*rpl3 psb28_2 rps4_2 rps16_2 ycf35_2 *psba_2 *rpl34_2 *rpl27_2 *rpl21_2 *rpl32_2 *psby_2 *rpl20_2 *rpl35_2 *psae_2 *ycf35_1 *rps16_1 *rps4_1 *psb28_1 psac_2 psbw
Leptocylindrus danicus (*l*)	rbcl *psbz rpl35_1 rpl20_1 psam psad *rpl12 *rpl1 *rpl11 *rps14 psae_1 psbx psbv rpl19 *ycf3 *rps18 *rpl33 rps20 rpob rpoc1 rpoc2_1 rps2 *psbk psa1 psbd psbc psaa psab psbb psbt *psbn psbh ycf33 *psbj *psbl *psbf *psbe psal ycf39 psb1 psaf psaj psac rps6_1 *rpl34_1 *rpl27_1 *rpl21 rpl3 rpl4 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rpl29 rps17 rpl14 rpl24 rpl5 rps8 rpl6 rpl18 rps5 rpl36 rps13 rps11 rpoa rpl13 rps9 rpl31 rps12 rps7 tufa rps10_1 clpc_1 *rps16_1 psb28_1 rps4_1 psby_1 rpl32_1 ycf35_1 psba_1 *psac_1
Rhizosolenia imbricata – *Odontella sinensis*	*psab *psaa *psaj *psaf *psb1 *ycf39 *psal psbe psbf psbl psbj *psad *psbc *psbd *psbz *rpl12 *rpl1 *rpl11 psbx psbv rpl19 rps14 psam *rps2 *rpoc2_1 *rpoc1 *rpob *rps20 rpl33 rps18 ycf3 psae_1 ycf33 *psa1 psbk *psbh psbn *psbt *psbb *ycf31 *psbh psbn *psbt *psbb rbcl *rpoc1 | rpl32_1 rpl21_1 rpl27_1 rpl34_1 psba_1 rpl35_1 rpl20_1 psby_1 psac_1 rps6_1 *clpc_1 *rps10_1 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl24 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3 psb28_2 rps4_2 rps16_2 ycf35_2 *psba_2 *rpl34_2 *rpl27_2 *rpl21_2 *rpl32_2 *psby_2 *rpl20_2 *rpl35_2 *psae_2 *ycf35_2 *rps16_2 *rps4_2 *psb28_2 psac_2 psbw
Rhizosolenia imbricata (*l*)	*psab *psaa *psaj *psaf *psb1 *ycf39 *psal psbe psbf psbl psbj *psad *psbc *psbd *psbz *rpl12 *rpl1 *rpl11 psbx psbv rpl19 *ycf3 *rps18 *rpl33 rps20 rpob rpoc1 rpoc2_1 rps2 *psbk *rbcl psbb psbt *psbn psbh ycf33 *rps14 rpl35 rpl20 *psba_1 psac_1 rps6_1 *clpc_1 *rps10_1 *rps7_1 *rps12_1 *rpl34_1 *rpl27_1 *rpl21_1 *rpl32_1 *psby_1 *ycf33 *rps16 *rps4 *psb28_1 rpl3 rpl4 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rpl29 rps17 rpl14 rpl24 rpl5 rps8 rpl6 rpl18 rps5 rpl36 rps13 rps11 rpoa rpl13 rps9 rpl31 rps12_2 rps7_2 rps10_2 clpc_2 *rps6_2 *psac_2 psba_2
Thalassiosira oceanica – *Odontella sinensis*	*psaj *psaf rps14 psam rpl11 rpl1 rpl12 *rps2 *rpoc2_1 *rpoc1 *rpob *rps20 rpl33 rps18 ycf3 *psb1 *ycf39 *psal psbe psbf psbl psaa psab psae_1 ycf33 *psbc *psbd *psa1 psbk psbz *psbj *psbh psbn *psbt *psbb *ycf31 *psbh psbn *psbt *psbb rbcl *rpoc1 *rpl19 *psbv *psbx | rpl32_1 rpl21_1 rpl27_1 rpl34_1 psba_1 rpl35_1 rpl20_1 psby_1 psac_1 rps6_1 *clpc_1 *rps10_1 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl24 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3 psb28_2 rps4_2 rps16_2 ycf35_2 *psba_2 *rpl34_2 *rpl27_2 *rpl21_2 *rpl32_2 *psby_2 *rpl20_2 *rpl35_2 *psae_2 *psad *ycf35_1 *rps16_1 *rps4_1 *psb28_1 psac_2 psbw
Thalassiosira oceanica – *Thalassiosira pseudonana*	*psaj *psaf rps14 psam rpl11 rpl1 rpl12 *rps2 *rpoc2_1 *rpoc1 *rpob *rps20 rpl33 rps18 ycf3 *psb1 *ycf39 *psal psbe psbf psbl psaa psab *rpl19 *psbv *psbx *rbcl psbb psbt *psbn psbh psae_1 rpl35_1 rpl20 *ycf33 *psbk psa1 psbd psbc psbz psad *psbj psby_2 rpl32_2 rpl21_2 rpl27_2 rpl34_2 psba_2 ycf35_2 *psac_2 *rps16 *rps4 *psbw *psb28_2 rpl3 rpl4 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rpl29 rps17 rpl14 rpl24 rpl5 rps8 rpl6 rpl18 rps5 rpl36 rps13 rps11 rpoa rpl13 rps9 rpl31 rps12 rps7 tufa rps10_1 clpc_1 *rps6_1 psac_1 *ycf35_1 *psba_1 *rpl34_1 *rpl27_1 *rpl21_1 *rpl32_1 *psby_1
Thalassiosira oceanica (*l*)	*psaj *psaf rps14 psam rpl11 rpl1 rpl12 *rps2 *rpoc2_1 *rpoc1 *rpob *rps20 rpl33 rps18 ycf3 *psb1 *ycf39 *psal psbe psbf psbl psbj *psad psbd psbc psbz *psa1 psbk ycf33 psbx psbv psbb psbt *psbn psbh psae rpl35 rpl20 *rbcl rpl19 *psab *psaa *psby_2 rpl32_2 rpl21_2 rpl27_2 rpl34_2 psac *psba_1 ycf35_1 clpc_1 psb28_1 rps4_1 rps16_1 rps6_1 rpl3 rpl4 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rpl29 rps17 rpl14 rpl24 rpl5 rps8 rpl6 rpl18 rps5 rpl36 rps13 rps11 rpoa rpl13 rps9 rpl31 rps12 rps7 tufa rps10_1 *clpc' *ycf35' psba' *psac' *rpl34' *rpl27' *rpl21' *rpl32' psby'
Thalassiosira weissflogii – *Thalassiosira pseudonana*	psaa psab *rpl19 *psbv *psbx *rbcl psbb psbt *psbn psbh psae_1 rpl35_1 rpl20 rpl11 rpl1 rpl12 *rps2 *rpoc2_1 *rpoc1 *rpob *rps20 rpl33 rps18 ycf3 *psam *rps14 *ycf33 *psbk psa1 psbd psbc psbz psad *psbj *psbl *psbf *psbe psal ycf39 psb1 psaf psaj psby_2 rpl32_2 rpl21_2 rpl27_2 rpl34_2 psba_2 ycf35 *psac *rps16 *rps4 *psbw *psb28_2 rpl3 rpl4 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rpl29 rps17 rpl14 rpl24 rpl5 rps8 rpl6 rpl18 rps5 rpl36 rps13 rps11 rpoa rpl13 rps9 rpl31 rps12 rps7 tufa rps10_1 clpc_1 *rps6_1 psac_1 *ycf35_1 *psba_1 *rpl34_1 *rpl27_1 *rpl21_1 *rpl32_1 *psby_1
Thalassiosira weissflogii – *Roundia cardiophora*	psaa psab *rpl19 *psbv *psbx *rbcl psbb psbt *psbn psbh psae_1 rpl35_1 rpl20 rpl11 rpl1 rpl12 *rps2 *rpoc2_1 *rpoc1 *rpob *rps20 rpl33 rps18 ycf3 *psam *rps14 *ycf33 *psbk psa1 psbd psbc psbz psad *psbj *psbl *psbf *psbe psal ycf39 psb1 psaf psaj psby_2 rpl32_2 rpl21_2 rpl27_2 rpl34_2 psba_2 ycf35 *psac *rps16 *rps4 *psb28 rpl3 rpl4 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rpl29 rps17 rpl14 rpl24 rpl5 rps8 rpl6 rpl18 rps5 rpl36 rps13 rps11 rpoa rpl13 rps9 rpl31 rps12 rps7 tufa rps10_1 clpc_1 *rps6_1 psac_1 *ycf35_1 *psba_1 *rpl34_1 *rpl27_1 *rpl21_1 *rpl32_1 *psby_1
Thalassiosira weissflogii (*l*)	psaa psab *rpl19 *psbv *psbx *rbcl psbb psbt *psbn psbh psae_1 rpl35_1 rpl20 rpl11 rpl1 rpl12 *rps2 *rpoc2_1 *rpoc1 *rpob *rps20 rpl33 rps18 ycf3 *psam *rps14 *ycf33 *psbk psa1 psbd psbc psbz psad *psbj *psbl *psbf *psbe psal ycf39 psb1 psaf psaj psby_2 rpl32_2 rpl21_2 rpl27_2 rpl34_2 psba_2 ycf35_2 *psac_2 *rps16_2 *rps4_2 *psb28_2 rpl3 rpl4 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rpl29 rps17 rpl14 rpl24 rpl5 rps8 rpl6 rpl18 rps5 rpl36 rps13 rps11 rpoa rpl13 rps9 rpl31 rps12 rps7 tufa rps10_1 clpc_1 *rps6_1 psac_1 *ycf35_1 *psba_1 *rpl34_1 *rpl27_1 *rpl21_1 *rpl32_1 *psby_1
Roundia cardiophora (*l*)	psaa psab *rpl19 *psbv *psbx *rbcl psbb psbt *psbn psbh psae_1 rpl35_1 rpl20 rpl11 rpl1 rpl12 *rps2 *rpoc2_1 *rpoc1 *rpob *rps20 rpl33 rps18 ycf3 *psam *rps14 *ycf33 *psbk psa1 psbd psbc psbz psad *psbj *psbl *psbf *psbe psal ycf39 psb1 psaf psaj psby_2 rpl32_2 rpl21_2 rpl27_2 rpl34_2 psba_2 ycf35_2 *psac_2 *rps16_2 *rps4_2 *psb28_2 rpl3 rpl4 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rpl29 rps17 rpl14 rpl24 rpl5 rps8 rpl6 rpl18 rps5 rpl36 rps13 rps11 rpoa rpl13 rps9 rpl31 rps12 rps7 tufa rps10_1 clpc_1 *rps6_1 psac_1 *ycf35_1 *psba_1 *rpl34_1 *rpl27_1 *rpl21_1 *rpl32_1 *psby_1
Thalassiosira pseudonana (*l*)	psaa psab *rpl19 *psbv *psbx *rbcl psbb psbt *psbn psbh psae_1 rpl35_1 rpl20 rpl11 rpl1 rpl12 *rps2 *rpoc2_1 *rpoc1 *rpob *rps20 rpl33 rps18 ycf3 *psam *rps14 *ycf33 *psbk psa1 psbd psbc psbz psad *psbj *psbl *psbf *psbe psal ycf39 psb1 psaf psaj psby_2 rpl32_2 rpl21_2 rpl27_2 rpl34_2 psba_2 ycf35_2 *psac_2 *rps16_2 *rps4_2 *psbw rpl3 rpl4 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rpl29 rps17 rpl14 rpl24 rpl5 rps8 rpl6 rpl18 rps5 rpl36 rps13 rps11 rpoa rpl13 rps9 rpl31 rps12 rps7 tufa rps10_1 clpc_1 *rps6_1 psac_1 *ycf35_1 *psba_1 *rpl34_1 *rpl27_1 *rpl21_1 *rpl32_1 *psby_1
Asterionellopsis glacialis – *Odontella sinensis*	psaa psab psaf psaj psae_1 *ycf33 *psbc *psbd *psa1 psbk psbz *psbj *psbl *psbf *psbe psal ycf39 psb1 *psbh psbn *psbt *psbb *ycf31 *psbh psbn *psbt *psbb rbcl *rps2 *rpoc2_1 *rpoc1 *rpob *rps20 rpl33 rps18 ycf3

	*rpl19 *psbv *psbx rpl11 rpl1 rpl12 | rpl32_1 rpl21_1 rpl27_1 rpl34_1 psba_1 rpl35_1 rpl20_1 psby_1 psac_1 rps6_1 *clpc_1 *rps10_1 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl24 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3 psb28_2 rps4_2 rps16_2 ycf35_2 *psba_2 *rpl34_2 *rpl27_2 *rpl21_2 *rpl32_2 *psby_2 *rpl20_2 *rpl35_2 *psae_2 rps14 psam *psad *ycf35_1 *rps16_1 *rps4_1 *psb28_1 psac_2 psbw
Asterionellopsis glacialis (*l*)	psaa psab psaf psaj psae_1 ycf33 *psbc *psbd *psa1 psbk psbz *psbj *psbl *psbf *psbe psal ycf39 psb1 *psbh psbn *psbt *psbb rps14 psam psad *rpl12 *rpl1 *rpl11 psbx psbv rpl19 *rps2 *rpoc2_1 *rpoc1 *rpob *rps20 rpl33 rps18 ycf3 *rbcl rpl35_2 rpl20_2 psby_2 rpl32_2 *psac_2 rpl3 rpl4 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rpl29 rps17 rpl14 rpl24 rpl5 rps8 rpl6 rpl18 rps5 rpl36 rps13 rps11 rpoa rpl13 rps9 rpl31 rps12 rps7 tufa rps10_1 rpl21_1 rpl27_1 rpl34_1 psba_1 psb28_1 rps4_1 rps16_1 ycf35_1 clpc_1 *rps6_1 *rpl32_1 *psby_1
Cylindrotheca closterium – *Odontella sinensis*	psbz *psbj *psbl *psbf *psbe psal ycf39 psb1 psaf psaj *ycf31 *psbh psbn *psbt *psbb rbcl ycf33 *rps2 *rpoc2_1 *rpoc1 *rpob *rps20 rpl33 rps18 ycf3 *rpl19 *psbv *psbx rpl11 rpl1 rpl12 *psbk psa1 psbd psbc | rpl32_1 rpl21_1 rpl27_1 rpl34_1 psba_1 psae_1 rpl35_1 rpl20_1 psby_1 psac_1 rps6_1 *clpc_1 *rps10_1 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl24 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3 psb28_2 rps4_2 rps16_2 ycf35_2 *psba_2 *rpl34_2 *rpl27_2 *rpl21_2 *rpl32_2 *psby_2 *rpl20_2 *rpl35_2 *psae_2 rps14 psam *psad *psab *psaa *ycf35_1 *rps16_1 *rps4_1 *psb28_1 psac_2 psbw
Cylindrotheca closterium – *Phaeodactylum tricornutum*	*psaa ycf33 *rpoc2_1 *rpoc1 *rpob *rps20 rpl33 rps18 ycf3 *rpl19 *psbv *psbx rpl11 rpl1 rpl12 psad *psam *rps14 psbb psbt *psbn psbh psab *psaj *psaf *psb1 *ycf39 *psal psbe psbf psbl psbj *psbz *psbc *psbd *psa1 psbk rpl32_1 rpl21_1 rpl27_1 rpl34_1 psba_1 *ycf35_1 *rps16_1 *rps4_1 *psb28_1 | psb28_2 rps4_2 rps16_2 ycf35_2 *psba_2 *rpl34_2 *rpl27_2 *rpl21_2 *rpl32_2 *psby_2 *rpl20_2 *rpl35_2 *psae_2 rps2 *rbcl psae_1 rpl35_1 rpl20_1 psby_1 psac_1 rps6_1 *clpc_1 *rps10_1 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl24 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3
Cylindrotheca closterium (*l*)	*psaa ycf33 *rpoc2_1 *rpoc1 *rpob *rps20 rpl33 rps18 ycf3 *rpl19 *psbv *psbx rpl11 rpl1 rpl12 psad *psam *rps14 psbb psbt *psbn psbh *psaj *psaf *psb1 *ycf39 *psal psbe psbf psbl psbj *psbz *psbk psa1 psbd psbc *rbcl *psae_1 rps2 rpl35_1 rpl20_1 *clpc_1 *rps6_1 rpl3 rpl4 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rpl29 rps17 rpl14 rpl24 rpl5 rps8 rpl6 rpl18 rps5 rpl36 rps13 rps11 rpoa rpl13 rps9 rpl31 rps12 rps7 tufa rps10_1 *ycf35_1 *rps16_1 *rps4_1 *psb28_1 *psba_1 *rpl34_1 *rpl27_1 *rpl21_1 *rpl32_1 *psac_1 *psby_1 *psab
Lithodesmium undulatum – *Phaeodactylum tricornutum*	psaa psab *psaj *psaf *psb1 *ycf39 *psal psbe psbf psbl psbj *psbz *psbc *psbd *psa1 psbk rps14 psam psad *psbv *psbx rpl32_1 rpl21_1 rpl27_1 rpl34_1 psba_1 *ycf35_1 *rps16_1 *rps4_1 *psb28_1 | psb28_2 rps4_2 rps16_2 ycf35_2 *psba_2 *rpl34_2 *rpl27_2 *rpl21_2 *rpl32_2 *psby_2 *rpl20_2 *rpl35_2 *psae_2 *ycf33 *psbh psbn *psbt *psbb *rpl12 *rpl1 *rpl11 rpl19 *ycf3 *rps18 *rpl33 rps20 rpob rpoc1 rpoc2_1 rps2 *rbcl psae_1 rpl35_1 rpl20_1 psby_1 psac_1 rps6_1 *clpc_1 *rps10_1 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl24 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3
Lithodesmium undulatum – *Coscinodiscus radiates*	psaa psab *psaj *psaf *psb1 *ycf39 *psal psbe psbf psbl psbj *psbz *psbc *psbd *psa1 psbk rps14 psam psad *psbv *psbx rpl11 rpl1 rpl12 rpl19 *ycf3 *rps18 *rpl33 rps20 rpob rpoc1 rpoc2_1 rps2 *ycf33 *psbh psbn *psbt *psbb rbcl psae_2 psby_2 rpl35_2 rpl20_2 *psba_2 *rpl34_2 *rpl27_2 *rpl21_2 *rpl32_2 *psby_2 ycf35_2 *rps16_2 *rps4_2 *psb28_2 rpl3 rpl4 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rpl29 rps17 rpl14 rpl24 rpl5 rps8 rpl6 rpl18 rps5 rpl36 rps13 rps11 rpoa rpl13 rps9 rpl31 rps12 rps7 tufa rps10_1 clpc_1 *rps6_1 *psac_1 psba_1 *rpl20_1 *rpl35_1 *psae_1
Lithodesmium undulatum (*l*)	psaa psab *psaj *psaf *psb1 *ycf39 *psal psbe psbf psbl psbj *psbz *psbc *psbd *psa1 psbk rps14 psam psad *psbv *psbx rpl11 rpl1 rpl12 rpl19 *ycf3 *rps18 *rpl33 rps20 rpob rpoc1 rpoc2_1 rps2 *ycf33 *psbh psbn *psbt *psbb rbcl *rpl20_2 *rpl35_2 *psae_2 psby_2 *rps16_2 *rps4_2 *psb28_2 rpl3 rpl4 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rpl29 rps17 rpl14 rpl24 rpl5 rps8 rpl6 rpl18 rps5 rpl36 rps13 rps11 rpoa rpl13 rps9 rpl31 rps12 rps7 tufa rps10_1 clpc_1 *rps6_1 *psac_1 ycf35_1 rpl32_1 rpl21_1 rpl27_1 rpl34_1 psba_1 *psby_1 psae_1 rpl35_1 rpl20_1
Coscinodiscus radiates (*l*)	psaa psab *psaj *psaf *psb1 *ycf39 *psal psbe psbf psbl psbj *psbz *psbc *psbd psam psad *rpl12 *rpl1 *rpl11 psbx psbv rpl19 *ycf3 *rps18 *rpl33 rps20 rpob rpoc1 rpoc2_1 rps2 *ycf33 *psbh psbn *psbt *psbb rbcl *psa1 psbk *rps14 psae_2 rpl35_2 rpl20_2 *psba_2 *rpl34_2 *rpl27_2 *rpl21_2 *rpl32_2 *psby_2 ycf35_2 *rps16_2 *rps4_2 *psb28_2 rpl3 rpl4 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rpl29 rps17 rpl14 rpl24 rpl5 rps8 rpl6 rpl18 rps5 rpl36 rps13 rps11 rpoa rpl13 rps9 rpl31 rps12 rps7 tufa rps10_1 clpc_1 *rps6_1 *psac_1 psba_1 *rpl20_1 *rpl35_1 *psae_1
Cerataulina daemon – *Phaeodactylum tricornutum*	rpl32_1 rpl21_1 rpl27_1 rpl34_1 psba_1 *ycf35_1 *rps16_1 *rps4_1 *psb28_1 | psb28_2 rps4_2 rps16_2 ycf35_2 *psba_2 *rpl34_2 *rpl27_2 *rpl21_2 *rpl32_2 *psby_2 *rpl20_2 *rpl35_2 *psae_2 *psab *psaa *psaj *psaf *psb1 *ycf39 *psal psbe psbf psbl psbj *psbz *psbc *psbd *psa1 psbk *ycf33 *psbh psbn *psbt *psbb rps14 psam *psad *rpl12 *rpl1 *rpl11 psbx psbv rpl19 *ycf3 *rps18 *rpl33 rps20 rpob rpoc1 rpoc2_1 rps2 *rbcl psae_1 rpl35_1 rpl20_1 psby_1 psac_1 rps6_1 *clpc_1 *rps10_1 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl24 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3
Cerataulina daemon – *Chaetoceros simplex*	ycf35_1 psb28_1 rps4_1 rps16_1 | psb28_2 rps4_2 rps16_2 ycf35_2 *psba_2 *rpl34_2 *rpl27_2 *rpl21_2 *rpl32_2 *psby_2 *rpl20_2 *rpl35_2 *psae_2 psaa psab ycf33 psad *psam *rps14 psae_1 rpl35_1 rpl20_1 psby_1 psac_1 rps6_1 *clpc_1 *rps10_1 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl24 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3 | rbcl *rps2 *rpoc2_1 *rpoc1 *rpob *rps20 rpl33 rps18 ycf3 *rpl19 *psbv *psbx rpl11 rpl1 rpl12 *psbk psa1 psbd psbc psbz *psbj *psbl *psbf *psbe psal ycf39 psb1 psaf psaj *psbh psbn *psbt *psbb C
Cerataulina daemon (*l*)	psaa psab ycf33 *rps2 *rpoc2_1 *rpoc1 *rpob *rps20 rpl33 rps18 ycf3 *rpl19 *psbv *psbx rpl11 rpl1 rpl12 *psbk psa1 psbd psbc psbz *psbj *psbl *psbf *psbe psal ycf39 psb1 psaf psaj *psbh psbn *psbt *psbb rbcl *psad *psam *rps14 psae_2 rpl35_2 rpl20_2 psby_2 rpl32_2 rpl21_2 rpl27_2 rpl34_2 psba_2 *ycf35_2 *rps16_2 *rps4_2

	*psb28_2 rpl3 rpl4 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rpl29 rps17 rpl14 rpl24 rpl5 rps8 rpl6 rpl18 rps5 rpl36 rps13 rps11 rpoa rpl13 rps9 rpl31 rps12 rps7 tufa rps10_1 clpc_1 *rps6_1 *psac_1 *psby_1
Chaetoceros simplex (*l*)	*psad *rps2 *rpoc2_1 *rpoc1 *rpob *rps20 rpl33 rps18 psaa psab ycf3 *rpl19 *psbv *psbx rpl11 rpl1 rpl12 *psbk psa1 psbd psbc psbz *psb1 *ycf39 *psal psbe psbf psbl psbj psaf psaj *psbh psbn *psbt *psbb rbcl *ycf33 *psam *rps14 psae_1 rpl35_1 rpl20_1 psby_1 *ycf35_1 *rps16_1 *rps4_1 *psb28_1 rpl3 rpl4 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rpl29 rps17 rpl14 rpl24 rpl5 rps8 rpl6 rpl18 rps5 rpl36 rps13 rps11 rpoa rpl13 rps9 rpl31 rps12 rps7 tufa rps10_2 clpc_2 *rps6_2 *psac_2 *psba_2 *rpl34_2 *rpl27_2 *rpl21_2 *rpl32_2 *psby_2
Asterionella Formosa – *Phaeodactylum tricornutum*	rpl32_1 rpl21_1 rpl27_1 rpl34_1 psba_1 *ycf35_1 *rps16_1 *rps4_1 *psb28_1 | psb28_2 rps4_2 rps16_2 ycf35_2 *psba_2 *rpl34_2 *rpl27_2 *rpl21_2 *rpl32_2 *psby_2 *rpl20_2 *rpl35_2 *psae_2 *psab *psaa *psaj *psaf *psb1 *ycf39 *psal psbe psbf psbl psbj *psbz *psbc *psbd *psa1 psbk *ycf33 *psbh psbn *psbt *psbb rps14 psam *psad *rpl12 *rpl1 *rpl11 psbx psbv rpl19 *ycf3 *rps18 *rpl33 rps20 rpob rpoc1 rpoc2_1 rps2 *rbcl psae_1 rpl35_1 rpl20_1 psby_1 psac_1 rps6_1 *clpc_1 *rps10_1 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl24 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3
Asterionella Formosa – *Ulnaria acus*	psb28_2 rps4_2 rps16_2 ycf35_2 *psba_2 *rpl34_2 *rpl27_2 *rpl21_2 *rpl32_2 *psby_2 *rpl20_2 *rpl35_2 *psae_2 *psab *psaa *psaj *psaf *psb1 *ycf39 *psal psbe psbf psbl psbj *psbz *psbc *psbd *psa1 psbk *ycf33 *psbh psbn *psbt *psbb rps14 psam psad *rpl12 *rpl1 *rpl11 psbx psbv rpl19 *ycf3 *rps18 *rpl33 rps20 rpob rpoc1 rpoc2_1 rps2 rbcl psae_1 rpl35_1 rpl20_1 psby_1 psac_1 rps6_1 *clpc_1 *rps10_1 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl24 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3
Asterionella Formosa (*l*)	psaa psab *rbcl *rps2 *rpoc2_1 *rpoc1 *rpob *rps20 rpl33 rps18 ycf3 *rpl19 *psbv *psbx rpl11 rpl1 rpl12 *psbh psbn *psbt *psbb rps14 psam psad ycf33 *psbk psa1 psbd psbc psbz *psbj *psbl *psbf *psbe psal ycf39 psb1 psaf psaj psae_2 rpl35_2 rpl20_2 psby_2 psac_2 rpl32_2 rpl21_2 rpl27_2 rpl34_2 psba_2 *ycf35_2 *rps16_2 *rps4_2 *psb28_2 rpl3 rpl4 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rpl29 rps17 rpl14 rpl24 rpl5 rps8 rpl6 rpl18 rps5 rpl36 rps13 rps11 rpoa rpl13 rps9 rpl31 rps12 rps7 tufa rps10_1 rps6_1 *clpc_1 *psac_1 *psby_1
Ulnaria acus (*l*)	psaa psab *psaj *psaf *psb1 *ycf39 *psal psbe psbf psbl psbj *psbz *ycf33 psbd psbc *psbk psa1 *psbh psbn *psbt *psbb rps14 psam psad *rpl12 *rpl1 *rpl11 psbx psbv rpl19 *ycf3 *rps18 *rpl33 rps20 rpob rpoc1 rpoc2_1 rps2 rbcl psae_1 rpl35_1 rpl20_1 psby_1 psac_1 rps6_1 *clpc_1 *rps10_1 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl24 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3 psb28_2 rps4_2 rps16_2 ycf35_2 *psba_2 *rpl34_2 *rpl27_2 *rpl21_2 *rpl32_2 *psby_2
Durinskia baltica – *Phaeodactylum tricornutum*	rpl32_1 rpl21_1 rpl27_1 rpl34_1 psba_1 *ycf35_1 *rps16_1 *rps4_1 *psb28_1 | psb28_2 rps4_2 rps16_2 ycf35_2 *psba_2 *rpl34_2 *rpl27_2 *rpl21_2 *rpl32_2 *psby_2 *rpl20_2 *rpl35_2 *psae_2 *psab *psaa *psaj *psaf *psb1 *ycf39 *psal psbe psbf psbl psbj *psbz *psbc *psbd *psa1 psbk *ycf33 *psbh psbn *psbt *psbb rps14 psam *psad *rpl12 *rpl1 *rpl11 psbx psbv rpl19 *ycf3 *rps18 *rpl33 rps20 rpob rpoc1 rpoc2_1 rps2 *rbcl psae_1 rpl35_1 rpl20_1 psby_1 psac_1 rps6_1 *clpc_1 *rps10_1 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl24 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3
Durinskia baltica – *Kryptoperidinium foliaceum*	rpl32_1 rpl21_1 rpl27_1 rpl34_1 psba_1 *ycf35_1 *rps16_1 *rps4_1 *psb28_1 rps6_1 *clpc_1 *rps10_1 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl24 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3 *rpl20_1 *rpl35_1 *psae_1 rbcl *rps2 *rpoc2_1 *rpoc1 *rpob *rps20 rpl33 rps18 ycf3 *rpl19 *psbv *psbx rpl11 rpl1 rpl12 psad *psam *rps14 psbb psbt *psbn psbh ycf33 *psbk psa1 psbd psbc psbz *psbj *psbl *psbf *psbe psal ycf39 psb1 psaf psaj psaa psab psby_1 psac_1
Durinskia baltica (*l*)	*psab *psaa *psaj *psaf *psb1 *ycf39 *psal psbe psbf psbl psbj *psbz *psbc *psbd *psa1 psbk *ycf33 *psbh psbn *psbt *psbb rps14 psam *psad *rpl12 *rpl1 *rpl11 psbx psbv rpl19 *ycf3 *rps18 *rpl33 rps20 rpob rpoc1 rpoc2_1 rps2 *rbcl psae_1 *rpl20_1 *rpl35_1 rps6_1 psby_1 psac_1 rpl32_1 rpl21_1 rpl27_1 rpl34_1 psba_1 *ycf35_1 *rps16_1 *rps4_1 *psb28_1 rpl3 rpl4 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rpl29 rps17 rpl14 rpl24 rpl5 rps8 rpl6 rpl18 rps5 rpl36 rps13 rps11 rpoa rpl13 rps9 rpl31 rps12 rps7 tufa rps10_1 clpc_1
Kryptoperidinium foliaceum (*l*)	*psbj *psbl *psbf *psbe psal ycf39 psb1 psaf psaj psaa psab *psbz *psbc *psbd *psa1 psbk *ycf33 *psbh psbn *psbt *psbb rps14 psam *psad *rpl12 *rpl1 *rpl11 psbx psbv rpl19 *ycf3 *rps18 *rpl33 rps20 rpob rpoc1 rpoc2_1 rps2 *rbcl psae_1 rpl35_1 rpl20_1 rpl3 rpl4 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rpl29 rps17 rpl14 rpl24 rpl5 rps8 rpl6 rpl18 rps5 rpl36 rps13 rps11 rpoa rpl13 rps9 rpl31 rps12 rps7 tufa rps10_1 clpc_1 *rps6_1 psb28_1 rps4_1 rps16_1 ycf35_1 *psba_1 *rpl34_1 *rpl27_1 *rpl21_1 *rpl32_1 *psac_1 *psby_1
Fistulifera solaris – *Phaeodactylum tricornutum*	rpl32_1 rpl21_1 rpl27_1 rpl34_1 psba_1 *ycf35_1 *rps16_1 *rps4_1 *psb28_1 psac_1 *rps6_1 *clpc_1 *rps10_1 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl24 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3 psb28_2 rps4_2 rps16_2 ycf35_2 *psba_2 *rpl34_2 *rpl27_2 *rpl21_2 *rpl32_2 *psby_2 *rpl20_2 *rpl35_2 *psae_2 *psab *psaa *psaj *psaf *psb1 *ycf39 *psal psbe psbf psbl psbj *psbz *psbc *psbd *psa1 psbk *ycf33 *psbh psbn *psbt *psbb rps14 psam *psad *rpl12 *rpl1 *rpl11 psbx psbv rpl19 *ycf3 *rps18 *rpl33 rps20 rpob rpoc1 rpoc2_1 rps2 *rbcl psae_1 rpl35_1 rpl20_1 psby_1
Fistulifera solaris (*l*)	*rpob *rps20 rpl33 rps18 ycf3 *rpl19 *psbv *psbx rpl11 rpl1 rpl12 psad *psam *rps14 psae_1 rpl35_1 rpl20_1 psby_1 psac_1 *rps6_1 *clpc_1 *rps10_1 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl24 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3 psb28_2 rps4_2 rps16_2 ycf35_2 *psba_2 *rpl34_2 *rpl27_2 *rpl21_2 *rpl32_2 *psab *psaa *psaj *psaf *ycf39 *psal psbe psbf psbl psbj *psbz *psbc *psbd *psa1 psbk *ycf33 *psbh psbn *psbt *psbb *rbcl *rps2 *rpoc2_1 *rpoc1
Eunotia naegelii – *Phaeodactylum tricornutum*	rpl32_1 rpl21_1 rpl27_1 rpl34_1 psba_1 *ycf35_1 *rps16_1 *rps4_1 *psb28_1 psac_1 *rps6_1 *clpc_1 *rps10_1 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl24 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3 psb28_2 rps4_2 rps16_2 ycf35_2 *psba_2 *rpl34_2 *rpl27_2 *rpl21_2 *rpl32_2 *psby_2 *rpl20_2 *rpl35_2 *psae_2 rbcl *rps2 *rpoc2_1 *rpoc1 *rpob *rps20 rpl33 rps18 ycf3 *rpl19 *psbv *psbx rpl11 rpl1 rpl12 psad *psam *rps14 psbb psbt *psbn psbh ycf33

	*psbk psa1 psbd psbc psbz *psbj *psbl *psbf *psbe psal ycf39 psb1 psaf psaj psaa psab psae_1 rpl35_1 rpl20_1 psby_1
Eunotia naegelii (*l*)	psaa psab *psaj *psaf *psb1 *ycf39 *psal psbe psbf psbl psbj *psbz *psbc *psbd *psa1 psbk *ycf33 *psbh psbn *psbt *psbb rps14 psam *psad *rpl12 *rpl1 *rpl11 psbx psbv rpl19 *ycf3 *rps18 *rpl33 rps20 rpob rpoc1 rpoc2_1 rps2 *rbcl psae_2 rpl35_2 rpl20_2 psby_2 rpl32_2 rpl21_2 rpl27_2 rpl34_2 psba_2 *ycf35_2 *rps16_2 *rps4_2 *psb28_2 rpl3 rpl4 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rpl29 rps17 rpl14 rpl24 rpl5 rps8 rpl6 rpl18 rps5 rpl36 rps13 rps11 rpoa rpl13 rps9 rpl31 rps12 rps7 tufa rps10_1 clpc_1 *rps6_1 *psac_1 psb28_1 rps4_1 rps16_1 ycf35_1 *psba_1 *rpl34_1 *rpl27_1 *rpl21_1 *rpl32_1 *psby_1
Didymosphenia geminate – *Phaeodactylum tricornutum*	psb28_2 rps4_2 rps16_2 ycf35_2 *psba_2 *rpl34_2 *rpl27_2 *rpl21_2 *rpl32_2 *psby_2 *rpl20_2 *rpl35_2 *psae_2 rbcl *rps2 *rpoc2_1 *rpoc1 *rpob *rps20 rpl33 rps18 ycf3 *rpl19 *psbv *psbx rpl11 rpl1 rpl12 psad *psam *rps14 psbb psbt *psbn psbh ycf33 *psbk psa1 psbd psbc psbz *psbj *psbl *psbf *psbe psal ycf39 psb1 psaf psaj psaa psab psae_1 rpl35_1 rpl20_1 psby_1 psac_1 *rps6_1 *clpc_1 *rps10_1 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl24 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3
Didymosphenia geminate (*l*)	*psab *psaa *psaj *psaf *psb1 *ycf39 *psal psbe psbf psbl psbj *psbz *psbc *psbd *psa1 psbk *ycf33 *psbh psbn *psbt *psbb rps14 psam *psad *rpl12 *rpl1 *rpl11 psbx psbv rpl19 *ycf3 *rps18 *rpl33 rps20 rpob rpoc1 rpoc2_1 rps2 *rbcl psae_1 rpl35_1 rpl20_1 psby_1 psac_1 *rps6_1 *clpc_1 *rps10_1 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl24 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3 psb28_2 rps4_2 rps16_2 ycf35_2 *psba_2 *rpl34_2 *rpl27_2 *rpl21_2 *rpl32_2 *psby_2
Phaeodactylum tricornutum (*l*)	*psab *psaa *psaj *psaf *psb1 *ycf39 *psal psbe psbf psbl psbj *psbz *psbc *psbd *psa1 psbk *ycf33 *psbh psbn *psbt *psbb rps14 psam *psad *rpl12 *rpl1 *rpl11 psbx psbv rpl19 *ycf3 *rps18 *rpl33 rps20 rpob rpoc1 rpoc2_1 rps2 *rbcl psae_2 rpl35_2 rpl20_2 psby_2 rpl32_2 rpl21_2 rpl27_2 rpl34_2 psba_2 *ycf35_2 *rps16_2 *rps4_2 *psbw rpl3 rpl4 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rpl29 rps17 rpl14 rpl24 rpl5 rps8 rpl6 rpl18 rps5 rpl36 rps13 rps11 rpoa rpl13 rps9 rpl31 rps12 rps7 tufa rps10_1 clpc_1 rps6_1 *psac_1 *psby_1
Odontella sinensis (*l*)	psby_1 rpl32_1 psac_1 rps6_1 *clpc_1 *rps10_1 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl24 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3 psbw rps4_2 rps16_2 ycf35_2 *psba_2 *rpl34_2 *rpl27_2 *rpl21_2 *rpl32_2 *psby_2 *rpl20_2 *rpl35_2 *psae_2 rps14 psam psad ycf33 *rbcl psbb psbt *psbn psbh ycf31 *psaj *psaf *psb1 *ycf39 *psal psbe psbf psbl psbj *psbc *psbd *psa1 psbk *rpl12 *rpl1 *rpl11 psbx psbv rpl19 *ycf3 *rps18 *rpl33 rps20 rpob rpoc1 rpoc2_1 rps2 *psab *psaa

Результат реконструкции для большого дерева представлен в таблице 4.6*b*. Из таблицы видно, что большинство предковых структур состоят из одной хромосомы в минимальном поддереве, содержащем *Porphyra purpurea* и *Galdieria sulphuraria*. Остальные структуры содержат по несколько хромосом, что может указывать на активные перестройки хромосом в предковых структурах соответствующей части дерева.

Таблица 4.6*b*. Реконструкция хромосомных структур пластид родофитной ветви вдоль большого дерева. Обозначения те же, что в Таблице 4.6*a*.

Вершина	Структуры
Porphyra purpyrea – *Vaucheria litorea*	psac *psak *psba psby_2 rpl32 rpl21 rpl27 rps6 psbd psbc rps16 psbw rps1 *rpl12 *rpl1 *rpl11 *rpoz ycf33 *rpl19 clpc_1 *rpl9 | rps4 *rpl28 rbcl *psbv *psbx *psaj *psaf *ycf37 *rpl34 psam psby_1 rbcl29 | rpl33 rps18 ycf3 *ycf39 *rps2 *rpoc2_1 *rpoc1 *rpob *rps20 *psb28 ycf36 psad psb1 psal psbk *rpl20 *rpl35 *ycf35 *ycf31 *rps10 *tufa *rps12 *rpoa *rps19 *rpl2 *rpl3 *rps14 rpl23 rpl22 rps3 rpl16 rpl29 rps17 rpl14 rpl24 rpl5 rps8 rpl6 rpl18 rps5 ycf38 psbb psbt *psbn psbh *psbz psbm *psb30 psae psbe psbf psbl psbj *rpl4 *rps7 *rpl31 *rps9 *rpl13 *rps11 *rps13 *rpl36 *psa1 psaa psab
Porphyra purpyrea – *Cyanidioschyzon merolae*	psac *psak *psba psby_2 rpl32 rpl21 rpl27 rps6 psbd psbc rps16 psbw rps1 *rpl12 *rpl1 *rpl11 *rpoz ycf33 *rpl19 clpc_1 *rpl9 | rps4 *rpl28 rbcl *psbv *psbx *psaj *psaf *ycf37 *rpl34 psam | psal *psb1 *ycf39 *rps2 *rpoc2_1 *rpoc1 *rpob *rps20 rpl33 rps18 ycf3 *rpl20 *rpl35 ycf31 ycf35 psb28 ycf36 psad *psbz psbk *rps14 *psab *psaa rpl3 rpl4 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rpl29 rps17 rpl14 rpl24 rpl5 rps8 rpl6 rpl18 rps5 rpl36 rps13 rps11 rpoa rpl13 rps9 rpl31 rps12 rps7 tufa rps10 ycf38 psbb psbt *psbn psbh *psae psb30 psa1 *psbj *psbl *psbf *psbe
Cyanidium caldarium – *Cyanidioschyzon merolae*	psac *psak *psba *rpl32 *psby_2 *rpl27 *rpl21 ycf39 psb1 *psal psbe psbf psbl psbj *psa1 *rps2 *rps1 *rpoc2_1 *rpoc1 *rpob *rps20 rpl33 rps18 *psbv *psbx *psaj *psaf *ycf37 psam rps4 *rpl28 rbcl ycf3 ycf33 *rpl19 clpc_1 rpl11 rpl1 rpl12 *psbw *rps16 *psbc *psbd *rps6 *rpl34 *rpoz | rpl20 rpl3 rpl4 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rpl29 rps17 rpl14 rpl24 rpl5 rps8 rpl6 rpl18 rps5 rpl36 rps13 rps11 rpoa rpl13 rps9 rpl31 rps12 rps7 tufa rps10 ycf38 psbb psbt *psbn psbh *psae psaa psab rps14 *psbk psbz *psad rpl35

Cyanidium caldarium (*l*)	psbd psbc ycf3 *rps18 *rpl33 rpob rpoc1 rpoc2_1 rps2 psa1 *psbj *psbl *psbf *psbe psal *psb1 *ycf39 rpl21 rpl27 psby_2 rpl32 psba psak *psac rpl34 rps6 psae *psbh psbn *psbt *psbb *rpl19 clpc_1 rpl11 rpl1 rpl12 *psbw *rps16 rps4 *rpl28 rbcl *psam ycf37 psaf psaj psbv *rps10 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl24 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3 *rpl20 *rpl35 psad psbk *rps14 *psab *psaa
Cyanidioschyzon merolae (*l*)	rps4 *rpl28 rbcl psam rpl21 rpl27 psby_2 rpl32 psba *psak *psac rpoz rpl34 rps6 psbd psbc rps16 psbw rps1 rpl11 rpl1 rpl12 *clpc_1 rpl19 *ycf33 psbb psbt *psbn psbh *psae psaa psab rps14 *psbk psbz *psad rpl35 rpl20 rpl3 rpl4 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rpl29 rps17 rpl14 rpl24 rpl5 rps8 rpl6 rpl18 rps5 rpl36 rps13 rps11 rpoa rpl13 rps9 rpl31 rps12 rps7 tufa rps10 ycf38 *ycf3 psaf psaj psbx psbv *rps18 *rpl33 rps20 rpob rpoc1 rpoc2_1 rps2 psa1 *psbj *psbl *psbf *psbe psal *psb1 *ycf39
Porphyra purpyrea – Galdieria sulphuraria	psac *psak *psba ycf35 psby_2 rpl32 rpl21 rpl27 rps6 psbd psbc rps16 psbw rps1 *rpl12 *rpl1 *rpl11 *rpoz ycf33 *rpl19 clpc_1 *rpl9 | rps4 *rpl28 rbcl *psbv *psbx *psaj *psaf *ycf37 *rpl34 psam | psal *psb1 *ycf39 *rps2 *rpoc2_1 *rpoc1 *rpob *rps20 rpl33 rps18 ycf3 *rpl20 *rpl35 ycf31 psb28 ycf36 psad *psbz psbk *rps14 *psab *psaa rpl3 rpl4 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rpl29 rps17 rpl14 rpl24 rpl5 rps8 rpl6 rpl18 rps5 rpl36 rps13 rps11 rpoa rpl13 rps9 rpl31 rps12 rps7 tufa rps10 ycf38 psbb psbt *psbn psbh *psae psb30 psa1 *psbj *psbl *psbf *psbe
Guillardia theta – Galdieria sulphuraria	psac *psak *psba rps6 *rpl34 psbd psbc rps16 psbw *rpl12 *rpl1 *rpl11 *rpoz | rps4 *rpl28 rbcl *psbv *psbx *psaj *psaf *ycf37 psam | psal *psb1 *ycf39 *rps2 *rpoc2_1 *rpoc1 *rpob *rps20 rpl33 rps18 ycf3 *rpl20 *rpl35 ycf31 ycf36 psad *psbz psbk *rps14 *psab *psaa rpl3 rpl4 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rpl29 rps17 rpl14 rpl24 rpl5 rps8 rpl6 rpl18 rps5 rpl36 rps13 rps11 rpoa rpl13 rps9 rpl31 rps12 rps7 tufa rps10 ycf38 psbb psbt *psbn psbh *psae ycf33 *rpl19 clpc_1 *rpl9 psb30 ycf35 rpl21 rpl27 psby_2 rpl32 psa1 *psbj *psbl *psbf *psbe
Galdieria sulphuraria (*l*)	psbd psbc rps16 psbw *rpl12 *rpl1 *rpl11 rpl9 *clpc_1 rpl19 psae *psbh psbn *psbt *psbb *ycf38 *rps10 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3 psaa psab rps14 *psbk psbz *psad *ycf36 rpl35 rpl20 *ycf3 *rps18 *rpl33 rps20 rpob rpoc1 rpoc2_1 rps2 ycf39 psb1 *psal psbe psbf psbl psbj *psa1 *psb30 ycf37 psaf psaj psbx psbv *rbcl rpl28 *rps4 *psam rpl21 rpl27 psby_2 rpl32 *rpoz psac *psak *psba rps6
Guillardia theta – Cryptomonas paramecium	rps4 rbcl *psbv *psbx *psaj *psaf *ycf37 psac *psak *psba rps6 *rpl34 psam | psal *psb1 *ycf39 *rps2 *rpoc2_1 *rpoc1 *rpob *rps20 rpl33 rps18 ycf3 *rpl20 *rpl35 ycf31 ycf36 psad *psbz psbk *rps14 *psab *psaa rpl3 rpl4 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rpl29 rps17 rpl14 rpl24 rpl5 rps8 rpl6 rpl18 rps5 rpl36 rps13 rps11 rpoa rpl13 rps9 rpl31 rps12 rps7 tufa rps10 ycf38 psbb psbt *psbn psbh *psae ycf33 *rpl19 clpc_1 rpl11 rpl1 rpl12 *psbw *rps16 *psbc *psbd rpl21 rpl27 psby_2 rpl32 *ycf35 psa1 *psbj *psbl *psbf *psbe
Cryptomonas paramecium (*l*)	rps4 rbcl *rps2 *rpoc2_1 *rpoc1 *rpob *rps20 rpl33 rps18 *rpl20 *rpl35 *rps14 rpl3 rpl4 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rpl29 rps17 rpl14 rpl24 rpl5 rps8 rpl6 rpl18 rps5 rpl36 rps13 rps11 rpoa rpl13 rps9 rpl31 rps12 rps7 tufa rps10 *rpl19 clpc_1 rpl11 rpl1 rpl12 rpl34 rps16 *rpl27 *rpl21
Guillardia theta – Rhodomonas salina	psac *psak *psba rps6 psam rps4 rbcl *psbv *psbx *psaj *psaf *ycf37 psa1 *psbj *psbl *psbf *psbe psal *psb1 *ycf39 *rps2 *rpoc2_1 *rpoc1 *rpob *rps20 rpl33 rps18 ycf3 *rpl20 *rpl35 ycf31 ycf36 psad *psbz psbk *rps14 *psab *psaa rpl3 rpl4 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rpl29 rps17 rpl14 rpl24 rpl5 rps8 rpl6 rpl18 rps5 rpl36 rps13 rps11 rpoa rpl13 rps9 rpl31 rps12 rps7 tufa rps10 psbb psbt *psbn psbh *psae ycf33 *rpl19 clpc_1 rpl11 rpl1 rpl12 *psbw *rps16 *psbc *psbd rpl21 rpl27 psby_2 rpl32 *ycf35 *rpl34
Guillardia theta (*l*)	*rpl19 clpc_1 rpl11 rpl1 rpl12 *psbw *rps16 *psbc *psbd rpl21 rpl27 psby_2 rpl32 *ycf35 *rpl34 psac *psak *psba rps6 psam rps4 rbcl *psbv *psbx *psaj *psaf *ycf37 psa1 *psbj *psbl *psbf *psbe psal *psb1 *ycf39 *rps2 *rpoc2_1 *rpoc1 *rpob *rps20 rpl33 rps18 ycf3 *rpl20 *rpl35 ycf31 ycf36 psad *psbz psbk *rps14 *psab *psaa rpl3 rpl4 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rpl29 rps17 rpl14 rpl24 rpl5 rps8 rpl6 rpl18 rps5 rpl36 rps13 rps11 rpoa rpl13 rps9 rpl31 rps12 rps7 tufa rps10 psbb psbt *psbn psbh *psae ycf33
Rhodomonas salina (*l*)	*rpl19 clpc_1 rpl11 rpl1 rpl12 *psbw *rps16 *psbc *psbd rpl21 rpl27 psby_2 rpl32 *ycf35 *rpl34 psac *psak *psba rps6 *psam rps4 rbcl *psbv *psbx *psaj *psaf *ycf37 psa1 *psbj *psbl *psbf *psbe psal *psb1 *ycf39 *rps2 *rpoc2_1 *rpoc1 *rpob *rps20 rpl33 rps18 ycf3 *rpl20 *rpl35 ycf36 psad *psbz psbk *rps14 *psab *psaa rpl3 rpl4 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rpl29 rps17 rpl14 rpl24 rpl5 rps8 rpl6 rpl18 rps5 rpl36 rps13 rps11 rpoa rpl13 rps9 rpl31 rps12 rps7 tufa rps10 psbb psbt *psbn psbh *psae ycf33
Porphyra purpyrea – Pyropia yezoensis	psac *psak *rps4 psal *rpl28 rbcl *psb1 *ycf39 *rps2 *rpoc2_1 *rpoc1 *rpob *rps20 rpl33 rps18 ycf3 *rpl20 *rpl35 ycf36 psad psaa psab rps14 *psbk psbz rpl3 rpl4 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rpl29 rps17 rpl14 rpl24 rpl5 rps8 rpl6 rpl18 rps5 rpl36 rps13 rps11 rpoa rpl13 rps9 rpl31 rps12 rps7 tufa rps10 ycf38 psbb psbt *psbn psbh *psae psb28 psb30 psa1 *psbj *psbl *psbf *psbe *psam *rpl34 ycf37 psaf psaj psbx psbv psby_2 rpl32 rpl21 rpl27 *psba ycf35 rps6 psbd psbc rps16 psbw rps1 *rpl12 *rpl1 *rpl11 rpl9 *clpc_1 rpl19 *ycf33 *rpoz
Pyropia yezoensis (*l*)	*rpl34 ycf37 psaf psaj psbx psbv rpl32 rpl21 rpl27 *psba ycf35 *rps4 psal *rpl28 rbcl *psb1 *ycf39 *rps2 *rpoc2_1 *rpoc1 *rpob *rps20 rpl33 rps18 ycf3 *rpl20 *rpl35 psad psaa psab *rps10 *tufa *rps7 *rps12 rpl31 *rps9 *rpl13 *rpoa rps11 rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl24 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3 *psbz psbk *rps14 psbb psbt *psbn psbh *psae *rpl19 clpc_1 *rpl9 rpl11 rpl1 rpl12 *rps1 *psbw *rps16 *psbc *psbd *rps6 psak *psac psa1 *psbj *psbl *psbf *psbe *psam
Porphyra purpyrea – Calliarthron tuberculosum	rpoz psb28 psb30 psa1 *psbj *psbl *psbf *psbe *psam *rpl34 ycf37 psaf psaj psbx psbv psby_2 rpl32 rpl21 rpl27 *psba ycf35 rps6 psbd psbc rps16 psbw rps1 *rpl12 *rpl1 *rpl11 rpl9 *clpc_1 rpl19 *ycf33 psae *psbh psbn *psbt *psbb *ycf38 *rps10 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl24 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3 *psbz psbk *rps14 *psab *psaa *psad ycf36 rpl35 rpl20 *ycf3 *rps18 *rpl33 rps20 rpob rpoc1 rpoc2_1 rps2 ycf39 psb1 psal *rpl28 rbcl ycf34 rps4 psak *psac
Porphyra purpyrea – Pyropia perforata	psam psbe psbf psbl psbj *psa1 psac *psak rps6 psbd psbc rps16 psbw rps1 *rpl12 *rpl1 *rpl11 rpl9 *clpc_1 rpl19 *ycf33 psae *psbh psbn *psbt *psbb *ycf38 rps14 *psbk psbz rpl3 rpl4 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rpl29 rps17 rpl14 rpl24 rpl5 rps8 rpl6 rpl18 rps5 rpl36 rps13 rps11 rpoa rpl13 rps9 rpl31 rps12 rps7 tufa rps10

	*psab *psaa *psad ycf31 ycf36 rpl35 rpl20 *ycf3 *rps18 *rpl33 rps20 rpob rpoc1 rpoc2_1 rps2 ycf39 psb1 *ycf34 *rbcl rpl28 *psal rps4 *ycf35 psba *rpl27 *rpl21 *rpl32 *psby_2 *psbv *psbx *psaj *psaf *ycf37 rpl34
Porphyra purpyrea – Pyropia haitanensis	*rpl34 ycf37 psaf psaj psbx psbv psby_2 rpl32 rpl21 rpl27 *psba ycf35 *rps4 psal *rpl28 rbcl ycf34 *psb1 *ycf39 *rps2 *rpoc2_1 *rpoc1 *rpob *rps20 rpl33 rps18 ycf3 *rpl20 *rpl35 *ycf36 *ycf31 psad psaa psab *rps10 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl24 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3 *psbz psbk *rps14 ycf38 psbb psbt *psbn psbh *psae ycf33 *rpl19 clpc_1 *rpl9 rpl11 rpl1 rpl12 *rps1 *psbw *rps16 *psbc *psbd *rps6 psak *psac psa1 *psbj *psbl *psbf *psbe *psam
Porphyra purpyrea (*l*)	*rpl34 ycf37 psaf psaj psbx psbv psby_2 rpl32 rpl21 rpl27 *psba ycf35 *rps4 psal *rpl28 rbcl ycf34 *psb1 *ycf39 *rps2 *rpoc2_1 *rpoc1 *rpob *rps20 rpl33 rps18 ycf3 *rpl20 *rpl35 *ycf36 *ycf31 psad psaa psab *rps10 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl24 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3 *psbz psbk *rps14 ycf38 psbb psbt *psbn psbh *psae ycf33 *rpl19 clpc_1 *rpl9 rpl11 rpl1 rpl12 *rps1 *psbw *rps16 *psbc *psbd *rps6 psak *psac psa1 *psbj *psbl *psbf *psbe *psam
Pyropia haitanensis (*l*)	*rpl34 ycf37 psaf psaj psbx psbv psby_2 rpl32 rpl21 rpl27 *psba ycf35 *rps4 psal *rpl28 rbcl ycf34 *psb1 *ycf39 *rps2 *rpoc2_1 *rpoc1 *rpob *rps20 rpl33 rps18 ycf3 *rpl20 *rpl35 *ycf36 *ycf31 psad psaa psab *rps10 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl24 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3 *psbz psbk *rps14 ycf38 psbb psbt *psbn psbh *psae ycf33 *rpl19 clpc_1 *rpl9 rpl11 rpl1 rpl12 *rps1 *psbw *rps16 *psbc *psbd *rps6 psak *psac psa1 *psbj *psbl *psbf *psbe *psam
Pyropia perforata (*l*)	*rpl34 ycf37 psaf psaj psbx psbv psby_2 rpl32 rpl21 rpl27 *psba ycf35 *rps4 psal *rpl28 rbcl ycf34 *psb1 *ycf39 *rps2 *rpoc2_1 *rpoc1 *rpob *rps20 rpl33 rps18 ycf3 *rpl20 *rpl35 *ycf36 psad psaa psab *rps10 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl24 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl3 *rpl4 *psbz psbk *rps14 ycf38 psbt *psbn psbh *psae *ycf33 *rpl19 clpc_1 *rpl9 rpl11 rpl1 rpl12 *rps1 *psbw *rps16 *psbc *psbd *rps6 psak *psac psa1 *psbj *psbl *psbf *psbe *psam
Gracilaria tenuistipilata – Calliarthron tuberculosum	rpoz psb28 psb30 psa1 *psbj *psbl *psbf *psbe *psam *rpl34 ycf37 psaf psaj psbx psbv psby_2 rpl32 rpl21 rpl27 *psba ycf35 rps6 psbd psbc rps16 psbw rps1 *rpl12 *rpl1 *rpl11 rpl9 *clpc_1 rpl19 *ycf33 psae *psbh psbn *psbt *psbb *ycf38 *rps10 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl24 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3 *psbz psbk *rps14 *psab *psaa *psad ycf36 rpl35 rpl20 *ycf3 *rps18 *rpl33 rps20 rpob rpoc1 rpoc2_1 rps2 ycf39 psb1 psal *rpl28 rbcl ycf34 rps4 psak *psac
Gracilaria tenuistipilata – Gracilaria Salicornia	psb30 psa1 *psbj *psbl *psbf *psbe *psam *rpl34 ycf37 psaf psaj psbx psbv psby_2 rpl32 rpl21 rpl27 *psba ycf35 rps6 psbd psbc rps16 *rpl9 rpl11 rpl1 rpl12 *rps1 *psbw clpc_1 rpl19 *ycf33 psae *psbh psbn *psbt *psbb *ycf38 *rps10 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl24 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3 *psbz psbk *rps14 *psab *psaa *psad ycf36 rpl35 rpl20 *ycf3 *rps18 *rpl33 rps20 rpob rpoc1 rpoc2_1 rps2 ycf39 psb1 psal *rpl28 rbcl ycf34 rps4 psak *psac
Gracilaria tenuistipilata (*l*)	*rpl34 ycf37 psaf psaj psbx psbv psby_2 rpl32 rpl21 rpl27 *psba ycf35 rps6 psbd psbc rps16 *rpl9 rpl11 rpl1 rpl12 *rps1 *psbw clpc_1 rpl19 *ycf33 psae *psbh psbn *psbt *psbb *ycf38 *rps10 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl24 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3 psbk *rps14 *psab *psaa *psad ycf36 rpl35 rpl20 *ycf3 *rps18 *rpl33 rps20 rpob rpoc1 rpoc2_1 rps2 ycf39 psb1 psal *rpl28 rbcl ycf34 rps4 psak *psac psa1 *psbj *psbl *psbf *psbe *psam
Gracilaria salicornia (*l*)	*rpl34 ycf37 psaf psaj psbx psbv psby_2 rpl32 rpl21 rpl27 *psba ycf35 rps6 psbd psbc rps16 *rpl9 rpl11 rpl1 rpl12 *rps1 *psbw clpc_1 rpl19 *ycf33 psae *psbh psbn *psbt *psbb *ycf38 *rps10 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl24 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3 *psbz psbk *rps14 *psab *psaa *psad ycf36 rpl35 rpl20 *ycf3 *rps18 *rpl33 rps20 rpob rpoc1 rpoc2_1 rps2 ycf39 psb1 psal *rpl28 rbcl ycf34 rps4 psak *psac psb30 psa1 *psbj *psbl *psbf *psbe *psam
Chondrus crispus – Calliarthron tuberculosum	rpoz psb30 psa1 *psbj *psbl *psbf *psbe *psam *rps6 *ycf35 psba *rpl27 *rpl21 *rpl32 *psby_2 *psbv *psbx *psaj *psaf *ycf37 rpl34 psbd psbc rps16 psb28 rps1 *rpl12 *rpl1 *rpl11 rpl9 *clpc_1 rpl19 *ycf33 psae *psbh psbn *psbt *psbb *ycf38 *rps10 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl24 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3 *psbz psbk *rps14 *psab *psaa *psad ycf36 rpl35 rpl20 *ycf3 *rps18 *rpl33 rps20 rpob rpoc1 rpoc2_1 rps2 ycf39 psb1 psal *rpl28 rbcl ycf34 rps4 psak *psac
Calliarthron tuberculosum (*l*)	*rps6 psba *rpl27 *rpl21 *rpl32 *psby_2 *psbv *psbx *psaj *psaf *ycf37 rpl34 psbd psbc rps16 psb28 rps1 *rpl12 *rpl1 *rpl11 rpl9 *clpc_1 rpl19 *ycf33 psae *psbh psbn *psbt *psbb *ycf38 *rps10 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl24 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3 *psbz psbk *rps14 *psab *psaa *psad ycf36 rpl35 rpl20 *ycf3 *rps18 *rpl33 rps20 rpob rpoc1 rpoc2_1 rps2 ycf39 psb1 psal *rpl28 rbcl rps4 psak *psac rpoz psb30 psa1 *psbj *psbl *psbf *psbe *psam
Chondrus crispus – Vertebrata lanosa	rpoz psb30 psa1 *psbj *psbl *psbf *psbe *psam *rps6 *ycf35 psba *rpl27 *rpl21 *rpl32 *psby_2 *psbv *psbx *psaj *psaf *ycf37 rpl34 psbd psbc rps16 psb28 rps1 *rpl12 *rpl1 *rpl11 rpl9 *clpc_1 rpl19 *ycf33 psae *psbh psbn *psbt *psbb *ycf38 *rps10 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl24 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3 *psbz psbk *rps14 *psab *psaa *psad ycf36 rpl35 rpl20 *ycf3 *rps18 *rpl33 rps20 rpob rpoc1 rpoc2_1 rps2 ycf39 psb1 psal *rpl28 rbcl ycf34 rps4 psak *psac
Vertebrata lanosa (*l*)	rpl9 *clpc_1 rpl19 *ycf33 psae *psbh psbn *psbt *psbb *ycf38 *rps10 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 rpl24 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3 *psbz psbk *rps14 *psab *psaa *psad ycf36 rpl35 rpl20 *ycf3 *rps18 *rpl33

	rps20 rpob rpoc1 rpoc2_1 rps2 ycf39 psb1 psal *rpl28 rbcl ycf34 rps4 psak *psac rpoz psb30 psa1 *psbj *psbl *psbf *psbe *psam *rps6 *ycf35 psba *rpl27 *rpl21 *rpl32 *psby_2 *psbv *psbx *psaj *psaf rpl34 psbd psbc rps16 psb28 rps1 *rpl12 *rpl1 *rpl11
Chondrus crispus – Grateloupia taiwanensis	rpoz psb30 psa1 *psbj *psbl *psbf *psbe *psam *rps6 *ycf35 psba *rpl27 *rpl21 *rpl32 *psby_2 *psbv *psbx *psaj *psaf *ycf37 rpl34 psbd psbc rps16 psb28 rps1 *rpl12 *rpl1 *rpl11 rpl9 *clpc_1 rpl19 *ycf33 psae *psbh psbn *psbt *psbb *ycf38 *rps10 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl24 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3 *psbz psbk *rps14 *psab *psaa *psad ycf36 rpl35 rpl20 *ycf3 *rps18 *rpl33 rps20 rpob rpoc1 rpoc2_1 rps2 ycf39 psb1 psal *rpl28 rbcl ycf34 rps4 psak *psac
Chondrus crispus (*l*)	*rpl34 ycf37 psaf psaj psbx psbv psby_2 rpl32 rpl21 rpl27 *psba ycf35 rps6 psbd psbc rps16 psb28 rps1 *rpl12 *rpl1 *rpl11 rpl9 *clpc_1 rpl19 *ycf33 psae *psbh psbn *psbt *psbb *ycf38 *rps10 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl24 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3 *psbz psbk *rps14 *psab *psaa *psad ycf36 rpl35 rpl20 *ycf3 *rps18 *rpl33 rps20 rpob rpoc1 rpoc2_1 rps2 ycf39 psb1 psal *rpl28 rbcl ycf34 rps4 psak *psac rpoz psb30 psa1 *psbj *psbl *psbf *psbe *psam
Grateloupia taiwanensis (*l*)	rbcl ycf34 rps4 psak *psac rpoz psa1 *psbj *psbl *psbf *psbe *psam *rps6 *ycf35 psba *rpl27 *rpl21 *rpl32 *psby_2 *psbv *psbx *psaj *psaf ycf37 rpl34 psbd psbc rps16 psb28 rps1 *rpl12 *rpl1 *rpl11 rpl9 clpc_1 rpl19 *ycf33 psae *psbh psbn *psbt *psbb *ycf38 *rps10 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl24 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3 *psbz psbk *rps14 *psab *psaa *psad ycf36 rpl35 rpl20 *ycf3 *rps18 *rpl33 rps20 rpob rpoc1 rpoc2_1 rps2 ycf39 psb1 psal *rpl28
Toxoplasma gondii – Vaucheria litorea	psby_1 rpoc1 rbcl29 psbv_1 psac_2 psaj_2 | psam rpl33 rps18 ycf3 *rps2 *rpoc2_1 *rpoc1 *rpob *rps20 psaf psaj_1 psbd psbc rps16 rps4 ycf33 *rps1 *psb28 rpl9 rpl11 ycf36 psad psb1 psba *rbcl *psby_2 *rps6 *rpl34 rpl21 rpl27 *psac_1 rpl32 psal psbk *rpl20 *rpl35 ycf37 *ycf35 *tufa *rps12 *rpoa *rps19 *rpl2 *rpl3 *rps14 rpl23 rpl22 rps3 rpl16 rps17 rpl14 rpl5 rps8 rpl6 rps5 psbb psbt *psbn psbh *psbv_2 *clpc_1 *psbz psbm *psb30 psbe psbf psbl psbj *rpl4 *rps7 *rpl31 *rps9 *rps11 *rps13 *rpl36 *psa1 psaa psab | psbx rps10 *ycf39 rpoz *rpl12 *rpl1 rpl19
Vaucheria litorea (*l*)	psam rpl33 rps18 ycf3 *rps2 *rpoc2_1 *rpoc1 *rpob *rps20 psaf psaj_1 psbd psbc rps16 rps4 ycf33 *rps1 *psb28 rpl9 rpl11 rpl1 rpl12 *psae psbb psbt *psbn psbh *rps10 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl24 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3 *rpl20 *rpl35 *ycf39 *ycf37 *rbcl *psba psb1 *rpl19 clpc_1 *psbv_2 *psbx *psal psbe psbf psbl psbj *psa1 *psbk psbz psac_1 *psad *ycf36 *rpl34 *rpl32 *psby_2 *rpl27 *rpl21 psaa psab rps14
Toxoplasma gondii – Nannochloropsis gaditana	psby_1 rpoc1 rbcl29 psbv_1 psac_2 psaj_2 | ycf36 psad psb1 psba *rbcl *ycf3 *rps18 *rpl33 rps20 rpob rpoc1 rpoc2_1 rps2 *psby_2 *rps6 *rpl34 rpl21 rpl27 *psaj_1 *psac_1 rpl32 psaf psal psbk *rpl20 *rpl35 ycf37 *ycf35 rps16 *psbc *psbd *tufa *rps12 *rpoa *rps19 *rpl2 *rpl3 *rps14 rpl23 rpl22 rps3 rpl16 rps17 rpl14 rpl5 rps8 rpl6 rps5 psbb psbt *psbn psbh rps4 *psbv_2 *clpc_1 *psbz psam psbm *psb30 psbe psbf psbl psbj rpl11 *rpl4 *rps7 *rpl31 *rps9 *rps11 *rps13 *rpl36 *psa1 psaa psab | psbx rps10 *ycf39 rpoz *rpl12 *rpl1 rpl19
Babesia bovis – Nannochloropsis gaditana	psby_1 rpoc1 rbcl29 psbv_1 psac_2 psaj_2 | ycf36 psad psb1 psba *rbcl *ycf3 *rps18 *rpl33 rps20 rpob rpoc1 rpoc2_1 rps2 *psby_2 *clpc_1 *rps6 *rpl34 rpl21 rpl27 *psaj_1 *psac_1 rpl32 psaf psbx *psbc *psbd *ycf34 rpl3 rpl4 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rpl29 rps17 rpl14 rpl5 rps8 rpl6 rpl18 rps5 rpl36 rps13 rps11 rpoa rpl13 rps9 rpl31 rps12 rps7 tufa rps10 psbb psbt *psbn psbh *psae *rpl12 *rpl1 *rpl11 *psbz psbk *rps14 *psab *psaa *psbv_2 *clpc_3 *rpl19_2 *rpl20 *rpl35 *psal psbe psbf psbl psbj *psa1 psbw *rps4 rps16
Nannochloropsis gaditana (*l*)	clpc_3 psbv_2 psaa psab *psbk rpl11 rpl1 rpl12 psae *psbh psbn *psbt *rps10 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl14 *rps17 *rbcl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3 psbd *psbx *psaf *rpl32 *rpl19_2 *rpl20 *rpl35 *psal psbe psbf psbj *rps4 rps16 ycf36 psad psb1 psba *rbcl *ycf3 *rps18 *rpl33 rps20 rpob rpoc1 rpoc2_1 rps2 *psby_1 *clpc_1 *rps6 *rpl34 rpl21 rpl27 *psaj_1 *psac_1
Babesia bovis – Nannochloropsis limnetica	ycf36 psad psb1 psba *rbcl *ycf3 *rps18 *rpl33 rps20 rpob rpoc1 rpoc2_1 rps2 *psby_2 *clpc_1 *rps6 *rpl34 rpl21 rpl27 *psaj_1 *psac_1 clpc_2 psbv_1 psac_2 psaj_2 rpl32 psaf psbx *psbc *psbd *ycf34 rpl3 rpl4 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rpl29 rps17 rpl14 rpl5 rps8 rpl6 rpl18 rps5 rpl36 rps13 rps11 rpoa rpl13 rps9 rpl31 rps12 rps7 tufa rps10 psbb psbt *psbn psbh *psae *rpl12 *rpl1 *rpl11 *psbz psbk *rps14 *psab *psaa *psbv_2 *clpc_3 *rpl19_2 *rpl20 *rpl35 *psal psbe psbf psbl psbj *psa1 psbw *rps4 rps16
Babesia bovis – Trachydiscus minutus	psbh *psae *rpl12 *rpl1 *rpl11 psbv_1 *psbz psbk *rps14 *psab *psaa *psbv_2 rpl3 rpl4 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rpl29 rps17 rpl14 rpl5 rps8 rpl6 rpl18 rps5 rpl36 rps13 rps11 rpoa rpl13 rps9 rpl31 rps12 rps7 tufa rps10 psbb psbt *psbn | clpc_3 ycf34 psbd psbc *psbx *psaf *rpl32 *clpc_2 psac_1 psaj_1 *rpl19_1 *rpl27 *rpl21 rpl34 rps6 clpc_1 psby_2 *rbcl *ycf3 *rps18 *rpl33 rps20 rpob rpoc1 rpoc2_1 rps2 *psba *psb1 *psad *rps16 rps4 *psbw psa1 *psbj *psbl *psbf *psbe psal rpl35 rpl20 rpl19_2 *psaj_2 *psac_2
Babesia bovis (*l*)	*rps2 *rpoc2_1 *rpoc1 *rpob *clpc_2 *clpc_1 *tufa *rps12 *rps11 *rps13 *rps8 *rpl14 *rpl16 *rps3 *rpl2
Trachydiscus minutus (*l*)	*psae *rpl12 *rpl1 *rpl11 ycf34 psbd psbc psam *psbx *psaf *rpl32 *psbz psbk *rps14 *psab *psaa *psbv_2 *clpc_2 psac_1 psaj_1 *rpl19_1 *rpl27 *rpl21 rpl34 rps6 clpc_1 psby_2 *rbcl *rps2 *rpoc2_1 *rpoc1 *rpob *rps20 rpl33 rps18 ycf3 *psba *psb1 *psad *rps16 rps4 *psbw psa1 *psbj *psbl *psbf *psbe psal rpl35 rpl20 rpl19_2 *psaj_2 *psac_2 clpc_3 psbv_1 rpl3 rpl4 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rpl29 rps17 rpl14 rpl5 rps8 rpl6 rpl18 rps5 rpl36 rps13 rps11 rpoa rpl13 rps9 rpl31 rps12 rps7 tufa rps10 psbb psbt *psbn psbh
Nannochloropsis granulata – Nannochloropsis limnetica	ycf36 psad psb1 psba *rbcl *ycf3 *rps18 *rpl33 rps20 rpob rpoc1 rpoc2_1 rps2 *psby_2 *clpc_1 *rps6 *rpl34 rpl21 rpl27 *psaj_1 *psac_1 clpc_2 psbv_1 psac_2 psaj_2 rpl32 psaf psbx *psbc *psbd *ycf34 rpl3 rpl4 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rpl29 rps17 rpl14 rpl5 rps8 rpl6 rpl18 rps5 rpl36 rps13 rps11 rpoa rpl13 rps9 rpl31 rps12 rps7 tufa rps10 psbb psbt *psbn psbh *psae *rpl12 *rpl1 *rpl11 *psbz psbk *rps14 *psab *psaa *psbv_2 *clpc_3 *rpl19_2 *rpl20 *rpl35 *psal psbe psbf psbl psbj *psa1 psbw *rps4 rps16
Nannochloropsis granulata – Nannochloropsis oceanica	psaa psab rps14 *psbk psbz rpl11 rpl1 rpl12 psae *psbh psbn *psbt *psbb *rps10 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl14 *rps17 *rpl29 *rpl16 *rps3

	*rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3 ycf34 psbd psbc *psbx *psaf *rpl32 *psbv_1 *clpc_2 psac_1 psaj_1 *rpl27 *rpl21 rpl34 rps6 clpc_1 psby_2 *rps2 *rpoc2_1 *rpoc1 *rpob *rps20 rpl33 rps18 ycf3 rbcl *psba *psb1 *psad *ycf36 *rps16 rps4 *psbw psa1 *psbj *psbl *psbf *psbe psal rpl35 rpl20 rpl19_2 clpc_3 psbv_2
Nannochloropsis granulata – Nannochloropsis oculata	psaa psab rps14 *psbk psbz rpl11 rpl1 rpl12 psae *psbh psbn *psbt *psbb *rps10 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3 ycf34 psbd psbc *psbx *psaf *rpl32 *psbv_1 *clpc_2 psac_1 psaj_1 *rpl27 *rpl21 rpl34 rps6 clpc_1 psby_2 *rps2 *rpoc2_1 *rpoc1 *rpob *rps20 rpl33 rps18 ycf3 rbcl *psba *psb1 *psad *ycf36 *rps16 rps4 *psbw psa1 *psbj *psbl *psbf *psbe psal rpl35 rpl20 rpl19_2 clpc_3 psbv_2
Nannochloropsis granulate (*l*)	psaa psab rps14 *psbk psbz rpl11 rpl1 rpl12 psae *psbh psbn *psbt *psbb *rps10 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3 ycf34 psbd psbc *psbx *psaf *rpl32 *psbv_1 *clpc_2 psac_1 psaj_1 *rpl27 *rpl21 rpl34 rps6 clpc_1 psby_2 *rps2 *rpoc2_1 *rpoc1 *rpob *rps20 rpl33 rps18 ycf3 rbcl *psba *psb1 *psad *ycf36 *rps16 rps4 *psbw psa1 *psbj *psbl *psbf *psbe psal rpl35 rpl20 rpl19_2 clpc_3 psbv_2
Nannochloropsis oculata (*l*)	psaa psab rps14 *psbk psbz rpl11 rpl1 rpl12 psae *psbh psbn *psbt *psbb *rps10 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3 ycf34 psbd psbc *psbx *psaf *rpl32 *psbv_1 *clpc_2 psac_1 psaj_1 *rpl27 *rpl21 rpl34 rps6 clpc_1 psby_2 *rps2 *rpoc2_1 *rpoc1 *rpob *rps20 rpl33 rps18 ycf3 rbcl *psba *psb1 *psad *ycf36 *rps16 rps4 *psbw psa1 *psbj *psbl *psbf *psbe psal rpl35 rpl20 rpl19_2 clpc_3 psbv_2
Nannochloropsis oceanica (*l*)	psaa psab rps14 *psbk psbz rpl11 rpl1 rpl12 psae *psbh psbn *psbt *psbb *rps10 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3 ycf34 psbd psbc *psbx *psaf *rpl32 *psbv_1 *clpc_2 psac_1 psaj_1 *rpl27 *rpl21 rpl34 rps6 clpc_1 psby_2 *rps2 *rpoc2_1 *rpoc1 *rpob *rps20 rpl33 rps18 ycf3 rbcl *psba *psb1 *psad *ycf36 *rps16 rps4 *psbw psa1 *psbj *psbl *psbf *psbe psal rpl35 rpl20 rpl19_2 clpc_3 psbv_2
Nannochloropsis salina – Nannochloropsis limnetica	ycf36 psad psb1 psba *rbcl *ycf3 *rps18 *rpl33 rps20 rpob rpoc1 rpoc2_1 rps2 *psby_2 *clpc_1 *rps6 *rpl34 rpl21 rpl27 *psaj_1 *psac_1 clpc_2 rpl32 psaf psbx *psbc *psbd *ycf34 rpl3 rpl4 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rpl29 rps17 rpl14 rpl5 rps8 rpl6 rpl18 rps5 rpl36 rps13 rps11 rpoa rpl13 rps9 rpl31 rps12 rps7 tufa rps10 psbb psbt *psbn psbh *psae *rpl12 *rpl1 *rpl11 *psbz psbk *rps14 *psab *psaa *psbv_2 *clpc_3 *rpl19_2 *rpl20 *rpl35 *psal psbe psbf psbl psbj *psa1 psbw *rps4 rps16
Nannochloropsis salina (*l*)	psaa psab rps14 *psbk psbz rpl11 rpl1 rpl12 psae *psbh psbn *psbt *psbb *rps10 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3 ycf34 psbd psbc *psbx *psaf *rpl32 psac_1 psaj_1 *rpl27 *rpl21 rpl34 rps6 clpc_1 psby_2 *rps2 *rpoc2_1 *rpoc1 *rpob *rps20 rpl33 rps18 ycf3 rbcl *psba *psb1 *psad *ycf36 *rps16 rps4 *psbw psa1 *psbj *psbl *psbf *psbe psal rpl35 rpl20 rpl19_2 clpc_3 psbv_2
Nannochloropsis limnetica (*l*)	psaa psab rps14 *psbk psbz rpl11 rpl1 rpl12 psae *psbh psbn *psbt *psbb *rps10 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3 ycf34 psbd psbc *psbx *psaf *rpl32 *clpc_2 psac_1 psaj_1 *rpl27 *rpl21 rpl34 rps6 clpc_1 psby_2 *rps2 *rpoc2_1 *rpoc1 *rpob *rps20 rpl33 rps18 ycf3 rbcl *psba *psb1 *psad *ycf36 *rps16 rps4 *psbw psa1 *psbj *psbl *psbf *psbe psal rpl35 rpl20 rpl19_2 clpc_3 psbv_2
Toxoplasma gondii – Aureoumbra lagunensis	psal psbk *rpl20 *rpl35 ycf37 *ycf35 rpob rpoc1 rpoc2_1 rps2 rps16 *psad *psbc *psbd *tufa *rps12 *rpoa *rps19 *rpl2 *rpl3 *rps14 rpl27 *psac rpl23 rpl22 rps3 rpl16 rps17 rpl14 rpl5 rps8 rpl6 rps5 rbcl psbb psbt *psbn psbh rps4 *psbv *clpc_1 *psbz psam psbm ycf3 psaf psaj *psb30 psbe rps6 psbf psbl psbj psby_1 *rpl34 rpl11 *rpl4 *rps7 *rpl31 *rps9 *rps11 *rps13 *rpl36 *psa1 psba *rpl21 psaa psab | psbx rps10 *ycf39 psb1 rpoz *rpl12 *rpl1 rpl19 rpl33 rps18
Aureococcus anophagefferens – Aureoumbra lagunensis	psal psbk *rpl20 *rpl35 ycf37 *ycf35 rpob rpoc1 rpoc2_1 rps2 rps16 *psad *psbc *psbd psaj rpl3 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rps17 rpl14 rpl24 rpl5 rps8 rpl6 rps5 rpl36 rps13 rps11 rpoa rpl13 rps9 rpl31 rps12 rps7 tufa rps10 psbb psbt *psbn psbh psbx psbv ycf33 *psbj *psbl *psbf *psbe psaa psab rps14 rbcl ycf39 psb1 *psam psaf *rpl27 *rpl21 *ycf30 *rpl34 *rps6 *rpl1 *rpl11 *rps4 *ycf3 *rps18 *rpl33 psac *clpc_1 psba
Aureococcus anophagefferens (*l*)	psal psbk *rpl20 *rpl35 ycf37 *ycf35 rpob rpoc1 rpoc2_1 rps2 rps16 *psad *psbc *psbd psaj rpl3 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rps17 rpl14 rpl24 rpl5 rps8 rpl6 rps5 rpl36 rps13 rps11 rpoa rpl13 rps9 rpl31 rps12 rps7 tufa rps10 psbb psbt *psbn psbh psbx psbv ycf33 *psbj *psbl *psbf *psbe psaa psab rps14 rbcl ycf39 psb1 *psam rps4 rpl11 rpl1 rps6 rpl34 ycf30 rpl21 rpl27 *psac rpl33 rps18 ycf3 *clpc_1 psba
Aureoumbra lagunensis (*l*)	rps16 *psad *psbc *psbd psaf psaj rpl3 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rps17 rpl14 rpl24 rpl5 rps8 rpl6 rps5 rpl36 rps13 rps11 rpoa rpl13 rps9 rpl31 rps12 rps7 tufa rps10 psbb psbt *psbn psbh psbx psbv ycf33 *psbj *psbl *psbf *psbe psaa psab rps14 rbcl ycf39 psb1 *psam *rpl27 *rpl21 *ycf30 *rpl34 *rps6 *rpl1 *rpl11 *rps4 *ycf3 *rps18 *rpl33 rpob rpoc1 rpoc2_1 rps2 psac ycf37 *ycf35 *psbk *psal rpl35 rpl20 *clpc_1 psba
Toxoplasma gondii – Choreocolax polysiphoniae	*tufa *rps12 *rpoa *rps19 *rpl2 *rpl3 *rps14 rpl20 rpoc1 rpl27 *psac rpl23 rpl22 rps3 rpl16 rps17 rpl14 rpl5 rps8 rpl6 rps5 rbcl psbb psbt *psbn psbh rps4 *psbv *clpc_1 *psbz psbk psam psbm ycf3 psbd psbc *rps2 *rpoc2_1 *rpob psaf psaj psal *psb30 psbe rps6 psbf psbl psbj psby_1 *rps16 *rpl34 rpl11 *rpl4 *rps7 *rpl31 *rps9 *rps11 *rps13 *rpl36 *psad *psa1 psba *rpl21 psaa psab | psbx *ycf35 rps10 *ycf39 psb1 rpoz *rpl12 *rpl1 rpl19 rpl33 rps18
Choreocolax polysiphoniae (*l*)	*tufa *rps12 *rpoa *rps19 *rpl2 *rpl3 *rps14 rpl20 rpoc1 rpl27 *rpl12 *rpl1 *clpc_1 rpl19 *rps10 *rps7 *rpl31 *rps9 *rpl13 *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl14 *rpl29 *rpl16 *rps3 *rpl22 *rpl23 *rpl4 rpob rpoc2_1 rps2 rps4 *rps6 *rpl21 rps16 *rpl11
Toxoplasma gondii – Pavlova lutheri	*psac rpl3 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rps17 rpl14 rpl5 rps8 rpl6 rps5 rbcl psbb psbt *psbn psbh rps4 *psbv *clpc_1 *psbz psbk psam psbm ycf3 psbd psbc *rps2 *rpoc2_1 *rpoc1 *rpob psaf psaj psal *psb30 psbe rps6 psbf psbl psbj psby_1 *rps16 *rpl34 *rps14 rpl11 *rpl4 *rps7 *rps12 *rpl31 *rps9 *rpoa *rps11 *rps13 *rpl36 *psad *psa1 *rpl20 psba *rpl21 psaa psab *rpl27 *tufa | psbx *ycf35 rps10 *ycf39 psb1 rpoz *rpl19 rpl33 rps18
Pavlova lutheri (*l*)	*psac rpl3 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rps17 rpl14 rpl5 rps8 rpl6 rps5 rbcl psbb psbt *psbn psbh rps4 *psbv *clpc_1 *psbz psbk psam rpoz *rpl19 *rps10 *tufa *rps7 *rps12 *rpl31 *rps9 *rpoa *rps11 *rps13 *rpl36 *rps6

	psaf psaj *rps16 *psad *rpl27 *rpl21 psal *psbc *psbd *rps2 *rpoc2_1 *rpoc1 *rpob *psby_2 rpl20 ycf3 psa1 *psbj *psbl *psbf *psbe *psb1 *psbx rpl33 rps18 ycf39 rps14 psba psaa psab
Toxoplasma gondii – Lepidodinium chlorophorum	psbm ycf3 psbd psbc *rps2 *rpoc2_1 *rpoc1 *rpob psaf psaj psal *psb30 psbe rps6 psbf psbl psbj psby_1 *rps16 *rpl34 *rps14 rpl11 *rpl4 *rps4 *rps7 *rps12 *rpl31 *rps9 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl6 *rps8 *rpl5 *rpl14 *rps17 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl3 *psbv *psad psam *psa1 psac *psbh psbn *psbt *psbb *psbz psbk *rpl20 psba *rpl21 psaa psab rbcl *rpl27 *clpc_1 *tufa | psbx *ycf35 rps10 *ycf39 psb1 *rpl19 rpl33 rps18
Lepidodinium chlorophorum (*l*)	psaa psab rpl23 rpl2 rps19 rps3 rpl16 rpl14 rpl5 rpl36 rps11 rpoa rps9 psbd psbc rpob *psba *rps14 *psac *rpoc2_1 *psb1 *psbz *psbj *psbl *psbf *psbe *ycf3 *psbm tufa rpl19 rps2 rbcl psbb psbt psbn psbh psbk *rpl20 rps18 psaj rps12 rps7 rps4 psa1
Toxoplasma gondii – Phaeocystis globosa	psbm rpl11 *rpl4 *rps4 | psby_1 *rps16 *rpl34 *rps14 ycf3 psbd psbc *rps2 *rpoc2_1 *rpoc1 *rpob psaf psaj psal *psb30 psbe rps6 psbf psbl psbj | psbx *ycf35 rps10 *ycf39 psb1 rpl21 rpl20 *psba rpl27 *rbcl *psab *psaa *rpl19 rpl33 rps18 | rpl31 rps12 rps7 tufa clpc_1 *psbk psbz psbb psbt *psbn psbh *psac psa1 *psam psad psbv rpl3 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rps17 rpl14 rpl5 rps8 rpl6 rps5 rpl36 rps13 rps11 rpoa rps9
Phaeocystis antarctica – Phaeocystis globosa	psby_1 *rps16 *rpl34 *rps14 ycf3 psbd psbc *rps2 *rps4 *rpoc2_1 *rpoc1 *rpob *psaj *psaf rbcl *rpl27 *rpl20 *rpl21 ycf39 psb1 *rpl19 rpl33 rps18 psbx *ycf35 rps6 *psba *psam psad psbv rpl3 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rps17 rpl14 rpl5 rps8 rpl6 rps5 rpl36 rps13 rps11 rpoa rps9 rpl31 rps12 rps7 tufa rps10 psaa psab psal *psb30 clpc_1 *psbk psbz psbb psbt *psbn psbh psac psa1 psbe psbf psbl psbj
Phaeocystis antarctica (*l*)	ycf39 psb1 *rpl19 rpl33 rps18 psbx *ycf35 rps6 *psba *psbv *psad psam rpl3 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rps17 rpl14 rpl5 rps8 rpl6 rps5 rpl36 rps13 rps11 rpoa rps9 rpl31 rps12 rps7 tufa rps10 psaa psab psal *psb30 clpc_1 *psbk psbz psbb psbt *psbn psbh psac psa1 psbe psbf psbl psbj psby_1 *rps16 *rpl34 *rps14 ycf3 psbd psbc *rps2 *rps4 *rpoc2_1 *rpoc1 *rpob *psaj *psaf rbcl *rpl27 *rpl20 *rpl21
Phaeocystis globosa (*l*)	ycf39 psb1 *rpl19 rpl33 rps18 psbx *ycf35 rps6 *psba *psam psad psbv rpl3 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rps17 rpl14 rpl5 rps8 rpl6 rps5 rpl36 rps13 rps11 rpoa rps9 rpl31 rps12 rps7 tufa rps10 psaa psab psal *psb30 clpc_1 *psbk psbz psbb psbt *psbn psbh psac psa1 psbe psbf psbl psbj psby_1 *rps16 *rpl34 *rps14 ycf3 psbd psbc *rps2 *rps4 *rpoc2_1 *rpoc1 *rpob *psaj *psaf rbcl *rpl27 *rpl20 *rpl21
Toxoplasma gondii – Emiliania huxleyi	psbm rpl11 *rpl4 *rps4 | ycf39 *rps10 ycf35 *rps18 *rpl33 rpl19 psb1 rpl21 rpl20 *psba *rbcl *psab *psaa | rpl31 rps12 rps7 tufa clpc_1 *psbk psbz psbb psbt *psbn psbh *psac psaf psaj psal *rpl27 rpob rpoc1 rpoc2_1 rps2 *psbc *psbd *ycf3 rps14 rpl34 rps16 *psbj *psbl *psbf *psbe rps6 psa1 *psam psad psbv rpl3 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rps17 rpl14 rpl5 rps8 rpl6 rps5 rpl36 rps13 rps11 rpoa rps9
Emiliania huxleyi (*l*)	psaa psab rbcl psba *rpl20 *rpl21 *psb1 *rpl19 rpl33 rps18 *ycf35 clpc_1 *psbk psbz psbb psbt *psbn psbh *psac psaf psaj psal *rpl27 rpob rpoc1 rpoc2_1 rps4 rps2 *psbc *psbd *ycf3 rps14 rpl34 rps16 *psbj *psbl *psbf *psbe rps6 psa1 *psam psad psbv rpl3 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rps17 rpl14 rpl5 rps8 rpl6 rps5 rpl36 rps13 rps11 rpoa rps9 rpl31 rps12 rps7 tufa rps10 *ycf39
Toxoplasma gondii – Theileria parva	rpob rpoc1 rpoc2_1 rpoc2_2 rps2 rps4 rpl4 *clpc_1 *tufa *rps7 *rps12 *rps11 *rpl36 *rps5 *rpl6 *rps8 *rpl14 *rps17 *rpl16 *rps3 *rps19 *rpl2 *rpl23
Theileria parva (*l*)	rps4 rpl4 rpl2 rps19 rps3 rpl16 rpl14 rps8 rpl6 rps5 rpl36 rps11 rps12 rps7 tufa clpc_1 clpc_2 rpob rpoc1 rpoc2_1 rpoc2_2 rps2
Toxoplasma gondii – Plasmodium chabaudi	rpoc2_2 rps2 *clpc_1 *tufa *rps7 *rps12 *rps11 *rpl36 *rps5 *rpl6 *rps8 *rpl14 *rps17 *rpl16 *rps3 *rps19 *rpl2 *rpl23 *rpl4 *rps4 rpob rpoc1 rpoc2_1
Leucocytozoon caulleryi – Plasmodium chabaudi	rps4 rpl4 rpl23 rpl2 rps19 rps3 rpl16 rps17 rpl14 rps8 rpl6 rps5 rpl36 rps11 rps12 rps7 tufa clpc_1 *rps2 *rpoc2_2 *rpoc2_1 *rpoc1 *rpob
Leucocytozoon caulleryi (*l*)	rps4 rpl4 rpl23 rpl2 rps19 rps3 rpl16 rps17 rpl14 rps8 rpl6 rps5 rpl36 rps11 rps12 rps7 tufa clpc_1 *rps2 *rpoc2_2 *rpoc2_1 *rpoc1 *rpob
Plasmodium chabaudi (*l*)	rps4 rpl4 rpl23 rpl2 rps19 rps3 rpl16 rps17 rpl14 rps8 rpl6 rps5 rpl36 rps11 rps12 rps7 tufa clpc_1 *rps2 *rpoc2_2 *rpoc2_1 *rpoc1 *rpob
Toxoplasma gondii – Eimeria tenella	rps2 *clpc_1 *rpl11 *tufa *rps7 *rps12 *rps11 *rpl36 *rps5 *rpl6 *rps8 *rpl14 *rps17 *rpl16 *rps3 *rps19 *rpl2 *rpl4 *rps4 rpob rpoc1 rpoc2_1
Toxoplasma gondii (*l*)	rps4 rpl4 rpl2 rps19 rps3 rpl16 rps17 rpl14 rpl6 rps5 rpl36 rps11 rps12 rps7 tufa *rpl11 clpc_1 *rps2 *rpoc1 *rpob
Eimeria tenella (*l*)	rps4 rpl4 rpl2 rps19 rps3 rpl16 rps17 rpl14 rps8 rpl6 rps5 rpl36 rps11 rps12 rps7 tufa rpl11 clpc_1 *rps2 *rpoc2_1 *rpoc1 *rpob

Реконструкция филогенетического дерева структур хромосом пластид трёх видов бурых водорослей алгоритмом из раздела 3.3. Для реконструкции хромосомных структур выбрано дерево из трёх видов бурых водорослей, а именно *Ectocarpus siliculosus*, *Fucus vesiculosus*, *Saccharina japonica*. Дерево, построеное по данным хромосомным структурам, представлено на рисунке 74с и совпадает с частью дерева на рисунке 73. Все хромосомы кольцевые.

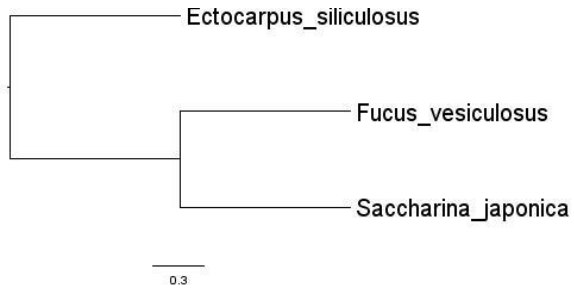


Рисунок 74c. Дерево хромосомных структур пластид трёх видов бурых водорослей.

Полученная реконструкция приведена в таблице 4.6*c*.

Таблица 4.6*c*. Реконструкция хромосомных структур пластид бурых водорослей. Обозначения те же, что в Таблице 4.6*a*.

Вершина	Структуры
Ectocarpus siliculosus (l)	rpl32_1 rpl21_1 *rps4 *rps16 *rps1 rpl9 rpl11 rpl1 rpl12 *rps10 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl24 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3 *rpl21_2 *rpl32_2 *rpl35 rpl20 *rpl19 rpl27 rpl34 rps20 rpob rpoc1 rpoc2 rps2 rps14 *rps18 *rpl33 clpc rbcl
Fucus vesiculosus (l)	*rpl19 rpl27 rpl34 rps20 rpob rpoc1 rpoc2 rps2 rpl35 rpl20 rbcl rps14 *clpc rpl33 rps18 *rpl32_2 rps16 rps4 rps1 rpl9 rpl11 rpl1 rpl12 *rps10 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl24 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3 *rpl21_2
Saccharina japonica (l)	*rps2 *rpoc2 *rpoc1 *rpob *rps20 *rpl34 *rpl27 rpl19 rpl35 rpl20 rbcl rps14 *rps18 *rpl33 clpc rpl32_1 rpl21_1 rpl3 rpl4 rpl23 rpl2 rps19 rpl22 rps3 rpl16 rpl29 rps17 rpl14 rpl24 rpl5 rps8 rpl6 rpl18 rps5 rpl36 rps13 rps11 rpoa rpl13 rps9 rpl31 rps12 rps7 tufa rps10 *rpl12 *rpl1 *rpl11 *rpl9 rps1 *rps4 *rps16
Fucus vesiculosus – Saccharina japonica	*rpl19 rpl27 rpl34 rps20 rpob rpoc1 rpoc2 rps2 rpl35 rpl20 rbcl rps14 rpl32_2 *rps18 *rpl33 clpc rpl32_1 rpl21_1 *rps4 *rps16 *rps1 rpl9 rpl11 rpl1 rpl12 *rps10 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl24 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3 *rpl21_2
Tree root	rpl32_1 rpl21_1 *rps4 *rps16 *rps1 rpl9 rpl11 rpl1 rpl12 *rps10 *tufa *rps7 *rps12 *rpl31 *rps9 *rpl13 *rpoa *rps11 *rps13 *rpl36 *rps5 *rpl18 *rpl6 *rps8 *rpl5 *rpl24 *rpl14 *rps17 *rpl29 *rpl16 *rps3 *rpl22 *rps19 *rpl2 *rpl23 *rpl4 *rpl3 *rpl21_2 *rpl19 rpl27 rpl34 rps20 rpob rpoc1 rpoc2 rps2 rps14 rpl32_2 *rps18 *rpl33 clpc rpl35 rpl20 rbcl

4. Хромосомные структруры бактерий рода *Rhizobium*

Данные получены из GenBank. Филогенетическое дерево хромосомных структур бактерий рода *Rhizobium* построено на основе матрицы попарных расстояний, которая вычислена с использованием алгоритмов из разделов 3.1 и 3.2. Оно показано на рисунке 7.

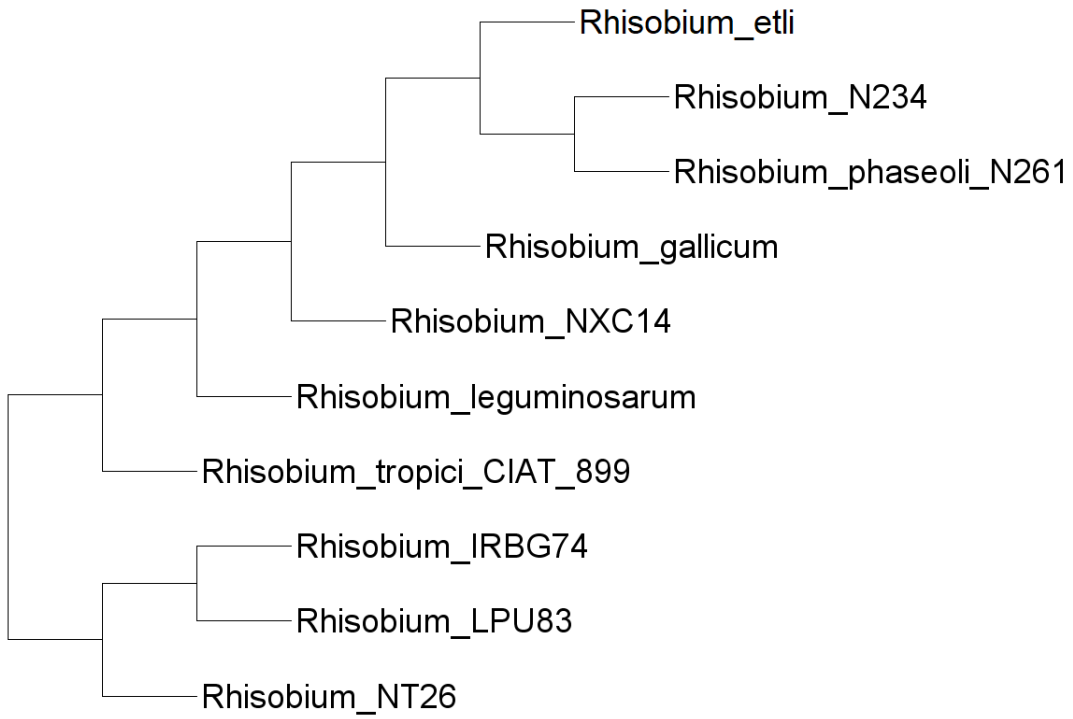


Рисунок 75. Дерево хромосомных структур бактерий рода Rhizobium spp. Получено с использованием хромосомных структур, указанных в таблице 4.7 в строках, помеченных (l).

Реконструкция хромосомных структур бактерий рода *Rhizobium*.

Реконструкция вдоль дерева, показанного на рисунке 75, выполнялась алгоритмом, описанным в разделе 3.3. Результаты представлены в таблице 4.7. Как видно из таблицы, всем внутренним вершинам приписано по одной кольцевой хромосоме, как и всем листьям, кроме одного, в котором кроме кольцевой имеется и короткая линейная хромосома.

Таблица 4.7. Реконструкция хромосомных структур бактерий рода *Rhizobium*. Обозначения такие же, как в предыдущих таблицах.

Вершина	Структуры
Rhizobium_N324_CP013630 (l)	*rpsA *rpsO rplT rpsT rpoN rpoE_1 rpsU_1 rpoZ *rplI *rpsR *rpsF *rpsI *rplM rplK rplA rplJ rplL rpoB rpoC rpsL rpsG rpsJ rplC rplD rplW rplB rpsS rplV rpsC rplP rpsQ rplN rplX rplE

	rpsN rpsH rplF rplR rpsE rplO rpsM rpsK rpoA rplQ rpsB *rpsD rpoE_2 rplY rpoH_1 rpsU_2 rpoE_3 rpsP rplS *rpoH_2 rplU (C)
Rhizobium_phaseoli_straiNN261_CP013580 (l)	*rpsA *rpsO rplT rpsT rpoN rpoE_1 rpsU_1 rpoZ *rplI *rpsR *rpsF *rpsI *rplM rplK rplA rplJ rplL rpoB rpoC rpsL rpsG rpsJ rplC rplD rplW rplB rpsS rplV rpsC rplP rpsQ rplN rplX rplE rpsN rpsH rplF rplR rpsE rplO rpsM rpsK rpoA rplQ rpsB *rpsD rpoE2 rplY rpoH_1 rpsU_2 rpoE_3 rpsP rplS rpoH_2 rplU (C)
Rhizobium_N324_CP013630 – Rhizobium_phaseoli_N261_CP013580	*rpsA *rpsO rplT rpsT rpoN rpoE_1 rpsU_1 rpoZ *rplI *rpsR *rpsF *rpsI *rplM rplK rplA rplJ rplL rpoB rpoC rpsL rpsG rpsJ rplC rplD rplW rplB rpsS rplV rpsC rplP rpsQ rplN rplX rplE rpsN rpsH rplF rplR rpsE rplO rpsM rpsK rpoA rplQ rpsB *rpsD rpoE2 rplY rpoH_1 rpsU_2 rpoE_3 rpsP rplS rpoH_2 rplU (C)
Rhizobium_etli_CP001074 (l)	*rpsA *rpsO rplT rpsT rpoN rpoE_1 rpsU_1 rpoZ *rplI *rpsR *rpsF *rpsI *rplM rplK rplA rplJ rplL rpoB rpoC rpsL rpsG rpsJ rplC rplD rplW rplB rpsS rplV rpsC rplP rpsQ rplN rplX rplE rpsN rpsH rplF rplR rpsE rplO rpsM rpsK rpoA rplQ rpsB *rpsD rpoE2 rplY rpoE_3 *rpoE_4 rpoH_1 rpsU_2 rpsP rplS rpoH_2 rplU (C)
Rhizobium_etli_CP001074 – Rhizobium_phaseoli_N261_CP013580	*rpsA *rpsO rplT rpsT rpoN rpoE_1 rpsU_1 rpoZ *rplI *rpsR *rpsF *rpsI *rplM rplK rplA rplJ rplL rpoB rpoC rpsL rpsG rpsJ rplC rplD rplW rplB rpsS rplV rpsC rplP rpsQ rplN rplX rplE rpsN rpsH rplF rplR rpsE rplO rpsM rpsK rpoA rplQ rpsB *rpsD rpoE_2 rplY rpoH_1 rpsU_2 rpoE_3 rpsP rplS rpoH_2 rplU (C)
Rhizobium_gallicum_CP006877 (l)	*rpsA rpsO rplT rpsT rpoN rpoE_1 rpoZ *rplI *rpsR *rpsF *rpsI *rplM rplK rplA rplJ rplL rpoB rpoC rpsL rpsG rpsJ rplC rplD rplW rplB rpsS rplV rpsC rplP rpsQ rplN rplX rplE rpsN rpsH rplF rplR rpsE rplO rpsM rpsK rpoA rplQ rpsB *rpsD rpoE_2 rplY rpoH_1 *rpsU_1 rpsU_2 rpsP rplS *rpoH_2 rplU (C)
Rhizobium_etli_CP001074 – Rhizobium_gallicum_CP006877	*rpsA *rpsO rplT rpsT rpoN rpoE_1 rpsU_1 rpoZ *rplI *rpsR *rpsF *rpsI *rplM rplK rplA rplJ rplL rpoB rpoC rpsL rpsG rpsJ rplC rplD rplW rplB rpsS rplV rpsC rplP rpsQ rplN rplX rplE rpsN rpsH rplF rplR rpsE rplO rpsM rpsK rpoA rplQ rpsB *rpsD rpoE2 rplY rpoH_1 rpsU_2 rpsP rplS rpoH_2 rplU (C)
Rhizobium_NXC14_CP021030 (l)	*rpsA *rpsO rplT rpsT rpoN rpoE_1 rpoE_3 rpsU_1 rpoZ *rplI *rpsR *rpsF *rpsI *rplM rplK rplA rplJ rplL rpoB rpoC rpsL rpsG rpsJ rplC rplD rplW rplB rpsS rplV rpsC rplP rpsQ rplN rplX rplE rpsN rpsH rplF rplR rpsE rplO rpsM rpsK rpoA rplQ rpsB *rpsD rpoE_2 rplY rpoH_1 rpsU_2 rpsP rplS rpoH_2 rplU (C)
Rhizobium_etli_CP001074 – Rhizobium_NXC14_CP021030	*rpsA *rpsO rplT rpsT rpoN rpoE_1 rpsU_1 rpoZ *rplI *rpsR *rpsF *rpsI *rplM rplK rplA rplJ rplL rpoB rpoC rpsL rpsG rpsJ rplC rplD rplW rplB rpsS rplV rpsC rplP rpsQ rplN rplX rplE rpsN rpsH rplF rplR rpsE rplO rpsM rpsK rpoA rplQ rpsB *rpsD rpoE_2 rplY rpoH_1 rpsU_2 rpsP rplS rpoH_2 rplU (C)
Rhizobium_leguminosarum_AM236080 (l)	*rpsA *rpsO rplT rpsT rpoN rpsU_1 *rplI *rpsR *rpsF *rpsI *rplM rplK rplA rplJ rplL rpoB rpoC rpsL rpsG rpsJ rplC rplD rplW rplB rpsS rplV rpsC rplP rpsQ rplN rplX rplE rpsN rpsH rplF rplR rpsE rplO rpsM rpsK rpoA rplQ rpsB *rpsD *rpoD *rpoZ rpoH_1 rpsU_2 rpoE_3 rplU (C)
Rhizobium_etli_CP001074 – Rhizobium_leguminosarum_AM236080	*rpsA *rpsO rplT rpsT rpoN rpoE_1 rpsU_1 rpoZ *rplI *rpsR *rpsF *rpsI *rplM rplK rplA rplJ rplL rpoB rpoC rpsL rpsG rpsJ

	rplC rplD rplW rplB rpsS rplV rpsC rplP rpsQ rplN rplX rplE rpsN rpsH rplF rplR rpsE rplO rpsM rpsK rpoA rplQ rpsB *rpsD rpoH_1 rpsU_2 rpoH_2 rplU (C)
Rhizobium_tropici_CIAT_899_CP004015 (l)	*rpsA *rpsO rplT rpsT *rpoN *rplI *rpsR *rpsI *rplM rplK rplA rplL rpoB rpoC rpsL rpsG rpsJ rplC rplD rplW rplB rplV rpsC rplP rpsQ rplN rplX rpsH rplF rplR rpsE rplO rpsM rpsK rpoA rplQ rpsB *rpsD rpoH_1 *rpoH_2 rplU (C)
Rhizobium_etli_CP001074 – Rhizobium_tropici_CIAT_899_CP004015	*rpsA *rpsO rplT rpsT rpoN rpsU1 rpoZ *rplI *rpsR *rpsF *rpsI *rplM rplK rplA rplJ rplL rpoB rpoC rpsL rpsG rpsJ rplC rplD rplW rplB rpsS rplV rpsC rplP rpsQ rplN rplX rplE rpsN rpsH rplF rplR rpsE rplO rpsM rpsK rpoA rplQ rpsB *rpsD rpoH_1 rpsU_2 rpoH_2 rplU (C)
Rhizobium_IRBG74_HG518322 (l)	*rpsO rplT rpsT *rpoN rpoZ *rpsR *rpsF *rpsI *rplM rpsB *rpsD *rplQ *rpoA *rpsK *rpsM *rplO *rpsE *rplR *rplF *rpsH *rpsN *rplE *rplX *rplN *rplP *rpsC *rplV *rpsS *rplB *rplW *rplD *rplC *rpsJ *rpsG *rpsL *rpoC *rpoB *rplL *rplJ *rplA *rplK *rpoD rplY rpoH_1 rpsP rplS *rplU (C) rpsU_1 *rpsU_2 rpsA (L)
Rhizobium_LPU83_HG916852 (l)	rpsO rpsA rplT rpsT rpoN rpoZ *rplI *rpsR *rpsF rplK rplA rplJ rplL rpoB rpoC rpsL rpsG rpsJ rplC rplD rplW rplB rpsS rplV rpsC rplP rpsQ rplN rplX rplE rpsN rpsH rplF rplR rpsE rplO rpsM rpsK rpoA rplQ *rpsI *rplM rpsD *rpsB *rpoD rplY rpoH_1 rpsU_1 *rpsU_2 rpsU_3 rpsP rplS rpoH2 *rplU (C)
Rhizobium_IRBG74_HG518322 – Rhizobium_LPU83_HG916852	rpsO rpsA rplT rpsT rpoN rpoZ *rplI *rpsR *rpsF *rpsI *rplM rplK rplA rplJ rplL rpoB rpoC rpsL rpsG rpsJ rplC rplD rplW rplB rpsS rplV rpsC rplP rpsQ rplN rplX rplE rpsN rpsH rplF rplR rpsE rplO rpsM rpsK rpoA rplQ rpsD *rpsB *rpoD rplY rpoH_1 rpsU_1 *rpsU_2 rpsU_3 rpsP rplS rpoH_2 *rplU (C)
Rhizobium_NT26_FO082820 (l)	rpsO rpsA rplT *rpoN rpoZ *rplI *rpsR *rpsF *rpsI *rplM rplK rplA rplJ rplL rpoB rpoC rpsL rpsG rpsJ rplC rplD rplW rplB rpsS rplV rpsC rplP rplN rplX rplE rpsN rpsH rplF rplR rpsE rplO rpsM rpsK rpoA rplQ rpsB rpsU_2 *rpsD *rpoD rplY rpoH_1 rpsU_1 rplU rpsP rplS *rpoH_2 (C)
Rhizobium_IRBG74_HG518322 – Rhizobium_NT26_FO082820	rpsO rpsA rplT rpsT rpoN rpoZ *rplI *rpsR *rpsF *rpsI *rplM rplK rplA rplJ rplL rpoB rpoC rpsL rpsG rpsJ rplC rplD rplW rplB rpsS rplV rpsC rplP rpsQ rplN rplX rplE rpsN rpsH rplF rplR rpsE rplO rpsM rpsK rpoA rplQ rpsB *rpsD *rpoD rplY rpoH_1 rpsU_1 rpsP rplS rpoH_2 *rplU (C)
Tree root	rpsO rpsA rplT rpsT rpoN rpsU_2 rpoZ *rplI *rpsR *rpsF *rpsI *rplM rplK rplA rplJ rplL rpoB rpoC rpsL rpsG rpsJ rplC rplD rplW rplB rpsS rplV rpsC rplP rpsQ rplN rplX rplE rpsN rpsH rplF rplR rpsE rplO rpsM rpsK rpoA rplQ rpsB *rpsD *rpoD rplY rpoH_1 rpsU_1 rpsP rplS rpoH_2 rplU (C)

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Watterson G.A., Ewens W.J., Hall T.E. The Chromosome Inversion Problem // *Journal of Theoretical Biology*. 1982. Vol. 99. P. 1–7. DOI: 10.1016/0022-5193(82)90384-8.

2 Blanchette M., Bourque G., Sankoff D. Breakpoint phylogenies. In: S. Miyano, T. Takagi Genome Informatics // *Univ. Academy Press*. 1997. P. 25–34.

3 Chin Lung Lu. An Efficient Algorithm for the Contig Ordering Problem under Algebraic Rearrangement Distance // *Journal of Computational Biology*. 2015. Vol. 22(11). P. 975–987. DOI: 10.1089/cmb.2015.0073.

4 Sankoff D., Leduc G., Antoine N., Paquin B., Lang B.F., Cedergren R. Gene order comparisons for phylogenetic inference: evolution of mitochondrial genome // *Proc. Natl. Acad. Sci.* 1992. Vol. 89. P. 6575-6579.

5 Hannenhalli S., Pevzner P. Transforming cabbage into turnip: polynomial algorithm for sorting signed permutations by reversals // *Journal of the ACM*. 1999. Vol. 46. P. 1–27. DOI: 10.1145/300515.300516.

6 Hannenhalli S., Pevzner P.A. Transforming man into mice (polynomial algorithm for genomic distance problem) // *Proceedings of the 36th Annual Symposium on Foundations of Computer Science*. 1995. P. 581. DOI: 10.1109/SFCS.1995.492588.

7 Yancopoulos S., Attie O., Friedberg R. Efficient sorting genomic permutations by translocation, inversion and block interchange // *Bioinformatics*. 2005. Vol. 21(16). P. 3340–3346. DOI: 10.1093/bioinformatics/bti535.

8 Bergeron A., Mixtacki J., Stoye J. A unifying view of genome rearrangements // *Algorithms in Bioinformatics, LNCS*. 2006. Vol. 4175. P. 163–173. DOI: 10.1007/11851561_16.

9 Shao M., Lin Y., Moret B. An Exact Algorithm to Compute the DCJ Distance for Genomes with Duplicate Genes. In: Sharan R. (eds) // *Research in Computational Molecular Biology, LNCS*. 2014. Vol. 8394. P. 280–292. DOI: 10.1007/978-3-319-05269-4_22.

10 Martinez F.V., Feijao P., Braga M.D., Stoye J. On the family-free DCJ distance and similarity // *Algorithms for molecular biology*. 2015, Vol. 10(1). DOI: 10.1186/s13015-015-0041-9.

11 Braga M.D.V., Willing E., Stoye J. Double cut and join with insertions and deletions // *Journal of computational biology*. 2011. Vol. 18. P. 1167–1184. DOI: 10.1089/cmb.2011.0118.

12 da Silva P.H., Machado R., Dantas S., Braga M.D.V. DCJ-indel and DCJ-substitution distances with distinct operation costs // *Algorithms for Molecular Biology*. 2013. V. 8. P. 21.1–21.15. DOI: 10.1186/1748-7188-8-21.

13 Compeau P.E.C. DCJ-Indel sorting revisited // *Algorithms for Molecular Biology*. 2013, Vol. 8, P. 6.1–6.9. DOI: 10.1186/1748-7188-8-6.

14 Compeau P.E.C. A Generalized Cost Model for DCJ-Indel Sorting // *Proceedings of 14-th International Workshop "Algorithms in Bioinformatics", LNCS*. 2014. Vol. 8701. P. 38–51. DOI: 10.1007/978-3-662-44753-6_4.

15 Alekseyev M.A., Pevzner P.A. Multi-Break Rearrangements and Chromosomal Evolution // *Theoretical Computer Science*. 2008. Vol. 395, № 2-3. P. 193–202. DOI: 10.1089/cmb.2008.0080.

16 Alekseyev M.A., Pevzner P.A. Breakpoint graphs and ancestral genome reconstructions // *Genome Research*. 2009. Vol. 19(5). P. 943-957. DOI: 10.1101/gr.082784.108.

17 Avdeyev P., Jiang S., Aganezov S., Hu F., Alekseyev M.A. Reconstruction of Ancestral Genomes in Presence of Gene Gain and Loss // *Journal of Computational Biology*. 2016. Vol. 23(3). P. 150-164. DOI: 10.1089/cmb.2015.0160.

18 Горбунов К.Ю., Любецкий В.А. Линейный алгоритм кратчайшей перестройки графов при разных ценах операций // *Информационные процессы*. 2016. Т. 16, № 2. С. 223–236.

19 Sokal R.R., Michener C.D. A statistical method for evaluating systematic relationships // *University of Kansas Science Bulletin.* 1958. V. 38. P. 1409-1438.

20 Saitou N, Nei M. The neighbor-joining method: a new method for reconstructing phylogenetic trees // Molecular Biology and Evolution.1987. V. 4, № 4. P. 406-425.

21 Garmash E.V. Mitochondrial respiration of the photosynthesizing cell // *Russian Journal of Plant Physiology*. 2016. Vol. 63. P. 13–25. DOI: 10.1134/S1021443715060072.

22 Karnkowska A., Vacek V., Zubáčová Z., Treitli S.C., Petrželková R., Eme L., Novák L., Žárský V., Barlow L.D., Herman E.K., et al. A eukaryote without a mitochondrial organelle // *Current Biology*. 2016. Vol. 26. P. 1274–1284. DOI: 10.1016/j.cub.2016.03.053.

23 Van Hoek A.H., van Alen T.A., Sprakel V.S., Hackstein J.H., Vogels G.D. Evolution of anaerobic ciliates from the gastrointestinal tract: Phylogenetic analysis of the ribosomal repeat from Nyctotherus ovalis and its relatives // *Molecular Biology Evolution*. 1998. Vol. 15(9). P. 1195–1206.

24 De Graaf R.M., Ricard G., van Alen T.A., Duarte I., Dutilh B.E., Burgtorf C., Kuiper J.W., van der Staay G.W., Tielens A.G., Huynen M.A., et al. The organellar genome and metabolic potential of the hydrogen-producing mitochondrion of Nyctotherus ovalis // *Molecular Biology Evolution.* 2011. Vol. 28(8). P. 2379–2391. DOI: 10.1093/molbev/msr059.

25 Kairo A., Fairlamb A.H., Gobright E., Nene V. A 7.1 kb linear DNA molecule of Theileria parva has scrambled rDNA sequences and open reading frames for mitochondrially encoded proteins // *EMBO J.* 1994. Vol. 13(4). P. 898–905.

26 Ванятинский В.Ф., Мирзоева Л.М., Поддубная А.В. Болезни рыб // *Пищевая промышленность*. 1979.

27 de Graaf R.M., van Alen T.A., Dutilh B.E., Kuiper J.W.P., van Zoggel H.J.A.A., Huynh M.B., Görtz H.-D., Huynen M.A., Hackstein J.H.P. The mitochondrial genomes of the ciliates Euplotes minuta and Euplotes crassus // *BMC Genomics*. 2009. Vol. 10. P. 514. DOI: 10.1186/1471-2164-10-514.

28 Matthews R.A. Ichthyophthirius multifiliis Fouquet and ichthyophthiriosis in freshwater teleosts // *Advances in Parasitology*. 2005. Vol. 59. P. 159–241. DOI: 10.1016/S0065-308X(05)59003-1.

29 Preer J.R., Jr., Preer L.B., Jurand A. Kappa and other endosymbionts in Paramecium aurelia // *Bacteriological Reviews*. 1974. Vol 38. P. 113–163.

30 Barth D., Berendonk T.U. The mitochondrial genome sequence of the ciliate Paramecium caudatum reveals a shift in nucleotide composition and codon usage within the genus Paramecium // *BMC Genom*ics. 2011. Vol. 12. P. 272. DOI: 10.1186/1471-2164-12-272.

31 Brunk C.F., Lee L.C., Tran A.B., Li J. Complete sequence of the mitochondrial genome of Tetrahymena thermophila and comparative methods for identifying highly divergent genes // *Nucleic Acids Research*. 2003. Vol. 31. P. 1673–1682.

32 Burger G., Zhu Y., Littlejohn T.G., Greenwood S.J., Schnare M.N., Lang B.F., Gray M.W. Complete sequence of the mitochondrial genome of Tetrahymena pyriformis and comparison with Paramecium Aurelia mitochondrial DNA // *Journal of Molecular Biology*. 2000. Vol. 297(2). P. 365–380. DOI: 10.1006/jmbi.2000.3529.

33 Cummings D.J. Mitochondrial genomes of the ciliates // *International Review of Cytology*. 1992. Vol. 141. P. 1–64. DOI: 10.1016/S0074-7696(08)62062-8.

34 Edqvist J., Burger G., Gray M.W. Expression of mitochondrial protein-coding genes in Tetrahymena pyriformis // *Journal of Molecular Biology*. 2000. Vol. 297. P. 381–393. DOI: 10.1006/jmbi.2000.3530.

35 Swart E.C., Nowacki M., Shum J., Stiles H., Higgins B.P., Doak T.G., Schotanus K., Magrini V.J., Minx P., Mardis E.R., et al. The Oxytricha trifallax mitochondrial genome // *Genome Biology and Evolution*, 2012. Vol. 4. P. 136–154. DOI: 10.1093/gbe/evr136.

36 Moradian M.M., Beglaryan D., Skozylas J.M., Kerikorian V. Complete mitochondrial genome sequence of three tetrahymena species reveals mutation hot spots and accelerated nonsynonymous substitutions in Ymf genes // *PLoS ONE*. 2007. Vol. 2(7). E650. DOI: 10.1371/journal.pone.0000650.

37 Pritchard A.E., Cummings D.J. Replication of linear mitochondrial DNA from Paramecium: Sequence and structure of the initiation-end crosslink // *Proc. Natl. Acad. Sci. USA*. 1981. Vol. 78(12). P. 7341–7345.

38 Rubanov L.I., Seliverstov A.V., Zverkov O.A., Lyubetsky V.A. A method for identification of highly conserved elements and evolutionary analysis of superphylum Alveolata // *BMC Bioinformatics*. 2016. Vol. 17. P. 385. DOI: 10.1186/s12859-016-1257-5.

39 Lyubetsky V.A., Gershgorin R.A., Seliverstov A.V., Gorbunov K.Y. Algorithms for reconstruction of chromosomal structures // *BMC Bioinformatics*. 2016. Vol. 17. P. 40. DOI: 10.1186/s12859-016-0878-z.

40 Электронный ресурс http://lab6.iitp.ru/en/chromoggl/ (The ChromoGGL Programs).

41 Seliverstov A.V. Monomials in quadratic forms // *Journal of Applied and Industrial Mathematics*. 2013. Vol. 7. P. 431–434. DOI: 10.1134/S1990478913030162.

42 Gorbunov K.Y., Gershgorin R.A., Lyubetsky V.A. Rearrangement and inference of chromosome structures // *Molecular Biology*. 2015. Vol. 49. P. 327–338. DOI: 10.1134/S0026893315030073.

43 Горбунов К.Ю., Любецкий В.А., Линейный алгоритм минимальной перестройки структур // Проблемы передачи информации. 2017. Т. 53, №. 1. С. 60–78

44 Saitou N., Nei M. The neighbor-joining method: A new method for reconstructing phylogenetic trees // *Molecular Biology and Evolution*. 1987. Vol. 4. P. 406–425.

45 Любецкий В.А., Селиверстов А.В., Зверков О.А. Построение разделяющих паралоги семейств гомологичных белков, кодируемых в пластидах цветковых растений // *Математическая биология и биоинформатика*. 2013. Т. 8, № 1. С. 225–233.

46 Zverkov O.A., Seliverstov A.V., Lyubetsky V.A. Plastid-encoded protein families specific for narrow taxonomic groups of algae and protozoa // *Molecular Biology*. 2012. Vol. 46. P. 717–726. DOI: 10.1134/S0026893312050123.

47 Zverkov O.A., Seliverstov A.V., Lyubetsky V.A. A Database of plastid protein families from red algae and Apicomplexa and expression regulation of the moeB gene // *BioMed Research International*. 2015. Vol. 2015. 510598. DOI: 10.1155/2015/510598.

48 Zverkov O.A., Seliverstov A.V., Lyubetsky V.A. Regulation of expression and evolution of genes in plastids of rhodophytic branch // *Life*. 2016. Vol. 6(1). P. 7. DOI: 10.3390/life6010007.

49 Edgar R.C. MUSCLE: Multiple sequence alignment with high accuracy and high throughput // *Nucleic Acids Research*. 2014. Vol. 32. P. 1792–1797. DOI: 10.1093/nar/gkh340.

50 Capella-Gutierrez S., Silla-Martinez J.M., Gabaldon T. trimAl: A tool for automated alignment trimming in large-scale phylogenetic analyses // *Bioinformatics*. 2009. Vol. 25(15). P. 1972–1973. DOI: 10.1093/bioinformatics/btp348.

51 Lartillot N., Philippe H. A Bayesian mixture model for across-site heterogeneities in the amino-acid replacement process // *Molecular Biology and Evolution*. 2004. Vol. 21. P. 1095–1109. DOI: 10.1093/molbev/msh112.

52 Lartillot N., Philippe H. Computing Bayes factors using thermodynamic integration // *Systematic Biology*. 2006. Vol. 55. P. 195–207. DOI:10.1080/10635150500433722.

53 Lartillot N., Brinkmann H., Philippe H. Suppression of long-branch attraction artefacts in the animal phylogeny using a site-heterogeneous model // *BMC Evolutionary Biology*. 2007. Vol. 7 (Suppl. 1). S4. DOI: 10.1186/1471-2148-7-S1-S4.

54 Rota-Stabelli O., Yang Z., Telford M.J. MtZoa: A general mitochondrial amino acid substitutions model for animal evolutionary studies // *Mol. Phylogenet. Evol.* 2009. Vol. 52. P. 268–272. DOI: 10.1016/j.ympev.2009.01.011.

55 Seliverstov A.V., Lysenko E.A., Lyubetsky V.A. Rapid evolution of promoters for the plastome gene *ndhF* in flowering plants // *Russian Journal of Plant Physiology*. 2009. Vol. 56. P. 838–845. DOI: 10.1134/S1021443709060144.

56 Lyubetsky V.A., Rubanov L.I., Seliverstov A.V. Lack of conservation of bacterial type promoters in plastids of Streptophyta // *Biology Direct*. 2010. Vol. 5. P. 34. DOI: 10.1186/1745-6150-5-34.

57 Nawrocki E.P., Burge S.W., Bateman A., Daub J., Eberhardt R.Y., Eddy S.R., Floden E.W., Gardner P.P., Jones T.A., Tate J., et al. Rfam 12.0: Updates to the RNA families database // *Nucleic Acids Research*. 2015. Vol. 43. D130–D137. DOI: 10.1093/nar/gku1063

58 Wei L., Xin Y., Wang D., Jing X., Zhou Q., Su X., et al. Nannochloropsis plastid and mitochondrial phylogenomes reveal organelle diversification mechanism and intragenus phylotyping strategy in microalgae // *BMC Genomics*. 2013. Vol. 14. P. 534. DOI: 10.1186/1471-2164-14-534.

59 Imanian B., Pombert J.F., Keeling P.J. The complete plastid genomes of the two 'dinotoms' *Durinskia baltica* and *Kryptoperidinium foliaceum* // *PLoS ONE*. 2010. Vol. 5(5). E10711. DOI: 10.1371/journal.pone.0010711.

60 Ong H.C., Wilhelm S.W., Gobler C.J., Bullerjahn G., Jacobs M.A., McKay J., et al. Analyses of the complete chloroplast genome sequences of two members of the Pelagophyceae: *Aureococcus anophagefferens* CCMP1984 and *Aureoumbra lagunensis* CCMP1507 // *Journal of Phycology*. 2010. Vol. 46(3). P. 602–615. DOI: 10.1111/j.1529-8817.2010.00841.x.

61 Cattolico R.A., Jacobs M.A., Zhou Y., Chang J., Duplessis M., Lybrand T., et al. Chloroplast genome sequencing analysis of *Heterosigma akashiwo* CCMP452 (West Atlantic) and NIES293 (West Pacific) strains // *BMC Genomics*. 2009. Vol. 9. P. 211. DOI: 10.1186/1471-2164-9-211.

62 Wang X., Shao Z., Fu W., Yao J., Hu Q., Duan D. Chloroplast genome of one brown seaweed, *Saccharina japonica* (Laminariales, Phaeophyta): its structural features and phylogenetic analyses with other photosynthetic plastids // *Marine Genomics*. 2013. Vol. 10. P. 1–9. DOI: 10.1016/j.margen.2012.12.002.

63 Le Corguille G., Pearson G., Valente M., Viegas C., Gschloessl B., Corre E., et al. Plastid genomes of two brown algae, *Ectocarpus siliculosus* and *Fucus vesiculosus*: further insights on the evolution of red-algal derived plastids // *BMC Evolutionary Biology*. 2009. Vol. 9. P. 253. DOI: 10.1186/1471-2148-9-253.

64 Janouškovec J., Horak A., Obornik M., Lukes J., Keeling P.J. A common red algal origin of the apicomplexan, dinoflagellate, and heterokont plastids // *Proc. Natl. Acad. Sci. USA*. 2010. Vol. 107(24). P. 10949–10954. DOI: 10.1073/pnas.1003335107.

65 Janouškovec J., Liu S.L., Martone P.T., Carre W., Leblanc C., Collen J., et al. Evolution of red algal plastid genomes: ancient architectures, introns, horizontal gene transfer, and taxonomic utility of plastid markers // *PLoS ONE*. 2013. Vol. 8(3). E59001. DOI: 10.1371/journal.pone.0059001.

66 Sadovskaya T.A., Seliverstov A.V. Analysis of the 5′-leader regions of several plastid genes in protozoa of the phylum apicomplexa and red algae // *Molecular Biology*. 2009. Vol. 43(4). P. 552–556. DOI: 10.1134/S0026893309040037.

67 Baurain D., Brinkmann H., Petersen J., Rodriguez-Ezpeleta N., Stechmann A., Demoulin V., et al. Phylogenomic evidence for separate acquisition of plastids in cryptophytes, haptophytes, and stramenopiles // *Molecular Biology and Evolution*. 2010. Vol. 27(7). P. 1698–1709. DOI: 10.1093/molbev/msq059.

68 Garg A., Stein A., Zhao W., Dwivedi A., Frutos R., Cornillot E., et al. Sequence and annotation of the apicoplast genome of the human pathogen babesia microti // *PLoS ONE*. 2014. Vol. 9(10). e107939. DOI: 10.1371/journal.pone.0107939.

69 Lyubetsky V.A., Gershgorin R.A., Gorbunov K.Yu. Chromosome structures: reduction of certain problems with unequal gene content and gene paralogs to Integer linear programming // *BMC Bioinformatics*. 2017, Vol. 18, № 537, 18 pages.

70 Gershgorin R.A., Gorbunov K.Yu., Zverkov O.A., Rubanov L.I., Seliverstov A.V., Lyubetsky V.A. Highly Conserved Elements and Chromosome Structure Evolution in Mitochondrial Genomes in Ciliates // *Life*. 2017, Vol. 7(1). DOI: 10.3390/life7010009.

71 Gershgorin R.A., Rubanov L.I., Seliverstov A.V. Easily Computable Invariants for Hypersurface Recognition // *Journal of Communications Technology and Electronics*. 2015, Vol. 60, № 12, P. 1429–1431. DOI: 10.1134/S1064226915120074.

72 Gershgorin R.A., Gorbunov K.Yu., Seliverstov A.V., Lyubetsky V.A. Evolution of Chromosome Structures // Proceedings of the 39th IITP RAS Interdisciplinary Conference & School “Information Technology and Systems 2015” (ITaS’15), Sochi, Russia, Sep 7–11 2015.

73 Lyubetsky V.A., Gershgorin R.A., Rubanov L.I., Seliverstov A.V., Zverkov O.A. Evolution and Systematics of Plastids of Rhodophytic Branch // *Proceedings of the International Moscow Conference on Computational Molecular Biology (MCCMB’17)*, Moscow, Russia, July 27–30, 2017, 4 стр.

74 Электронный ресурс http://lab6.iitp.ru/ppc/redline67/

75 Lyubetsky V.A., Seliverstov A.V., Zverkov O.A. Transcription regulation of plastid genes involved in sulfate transport in Viridiplantae // *BioMed Research International*. 2013, Vol. 2013, № 412450, 6 pages. DOI: 10.1155/2013/413450.

76 Gorbunov K.Yu., Lyubetsky V.A. A linear algorithm for the shortest transformation of graphs with different operation costs // *Journal of Communications Technology and Electronics*. 2017, Vol. 62, № 6, P. 653–662. DOI: 10.1134/S1064226917060092.

77 Gorbunov K.Yu., Lyubetsky V.A. Linear algorithm for minimal rearrangement of structures // *Problems of Information Transmission*. 2017, Vol. 53, № 1, P. 55–72.