№ 3. — С. 465–480.
3. Goncharov V.Yu. Existence criteria in some extremum problems involving eigenvalues of elliptic operators // J. Sib. Fed. Univ. Math. Phys. — 2016. — Т. 9. № 1. — С. 3

ПОСТРОЕНИЕ СЦЕНАРИЕВ ХРОМОСОМНЫХ ПЕРЕСТРОЕК

Горбунов К.Ю.

Даётся краткий обзор новых результатов по давно и активно изучаемой проблеме биоинформатики – анализу хромосомных структур и их перестроек. В нём рассматриваются вопросы построения наиболее экономного преобразования одной хромосомной структуры в другую. Даются основные определения, формулировки результатов, описываются идеи, лежащие в основе разработки эффективных алгоритмов.

СЛУЧАЙ РАВНОГО ГЕННОГО СОСТАВА И РАВНЫХ ЦЕН

Равный генный состава двух хромосомных структур означает, что каждая из них имеет одно и то же количество генов и между множествами генов данных структур задано взаимно однозначное соответствие, отражающее ортологичность. Гены располагаются по линейным или кольцевым хромосомам, при этом каждый ген имеет направление, отражающее цепь ДНК, на которой он расположен. Будем считать, что все гены на хромосоме можно упорядочить в линейном или циклическом порядке.

Таким образом, математически хромосомная структура представляется в виде ориентированного графа, который состоит из компонент – цепей и циклов, в том числе петель (кольцевых хромосом из одного гена). Изолированных вершин (*точек*) в графе нет. Дуги графа занумерованы натуральными числами. На рис. 1 приведены две хромосомные структуры – *a* и *b* с равным генным составом; гены с одинаковыми номерами считаются ортологичными.

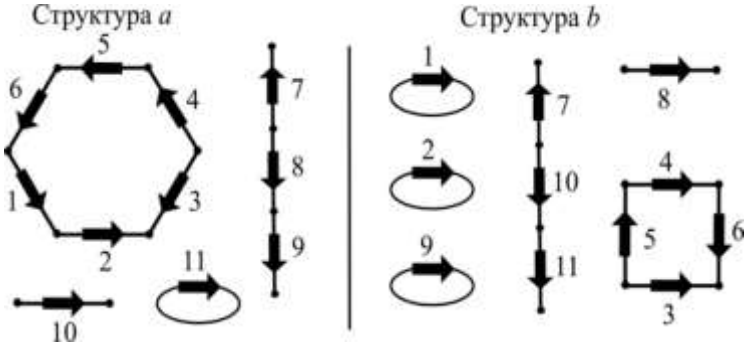


Рис. 1. Две хромосомные структуры, из 11 генов каждая

Фиксируются следующие операции над структурой (они известны также как DCJ-операции, см., напр., [1]).

1) *Двойная переклейка.* Разрезать две вершины, имеющиеся в графе, и по-новому отождествить (склеить) четыре образовавшихся края, рис. 2a.

2) *Полуторная переклейка.* Разрезать вершину и по-новому отождествить (склеить) один образовавшийся край с каким-то свободным краем в графе, рис. 2b.

3) и 4) *Одинарные переклейки*: *разрез* и *склейка*. Разрезать вершину или склеить два свободных края, рис. 2c. Эти операции взаимно обратны друг другу.

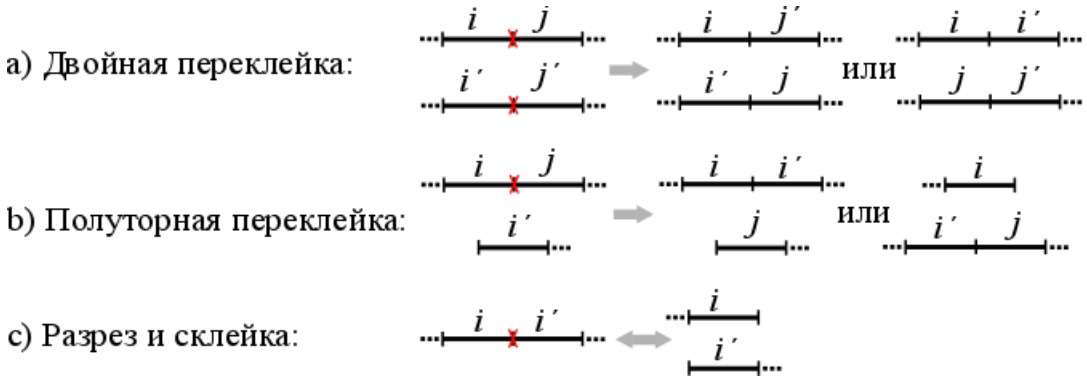


Рис. 2. Четыре операции над хромосомной структурой

Рассмотрим *задачу преобразования*: для двух данных структур *a* и *b* найти минимальную по числу операций цепочку операций, преобразующих структуру *a* в *b*. Заметим, что все операции обратимы: обратной операцией к двойной или полуторной переклейке является, соответственно, двойная или полуторная переклейка, операции разреза и склейки обратны друг другу. Поэтому данная задача эквивалентна такой: преобразовать отдельно структуру *a* и отдельно структуру *b* в некоторую одну структуру *c* двумя цепочками операций с минимальной суммой их длин.

Для её решения удобным оказывается понятие *общего графа a+b* двух структур *a* и *b*, рис.3.

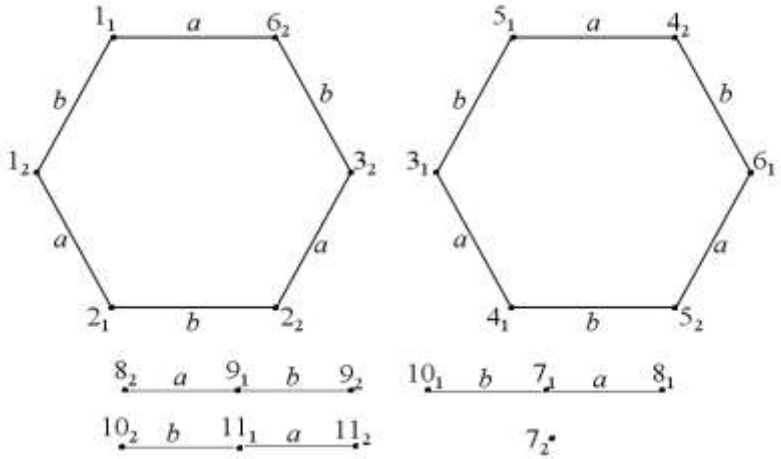


Рис. 3. Общий граф *a+b* двух структур, изображённых на рис. 1

Это неориентированный граф без петель, вершины которого – имена краёв всех генов; например, начало гена с номером 3 будет иметь имя 3_1, конец – имя 3_2. Ребро общего графа соединяет две вершины, если соответствующие им края отождествлены (склеены) в a или в b; оно помечается, соответственно, как a- или b-ребро. Например, общий граф структур, приведённых на рис. 1, показан на рис. 3. Таким образом, он несёт информацию о склейках одновременно в a и в b. Легко видеть, что граф $a+b$ всегда состоит из компонент – цепей (точки считаем цепями) и циклов, в которых a-рёбра и b-рёбра чередуются.

Если $a=b=c$, граф $a+b$ состоит из точек и циклов длины 2 (такой цикл состоит из двух параллельных рёбер: a-ребро и b-ребро). Этот вид графа будем называть *финальным.* Легко видеть, что наша задача эквивалентна задаче приведения графа $a+b$ к финальному виду следующими операциями – аналогами операций над структурой, рис 4.

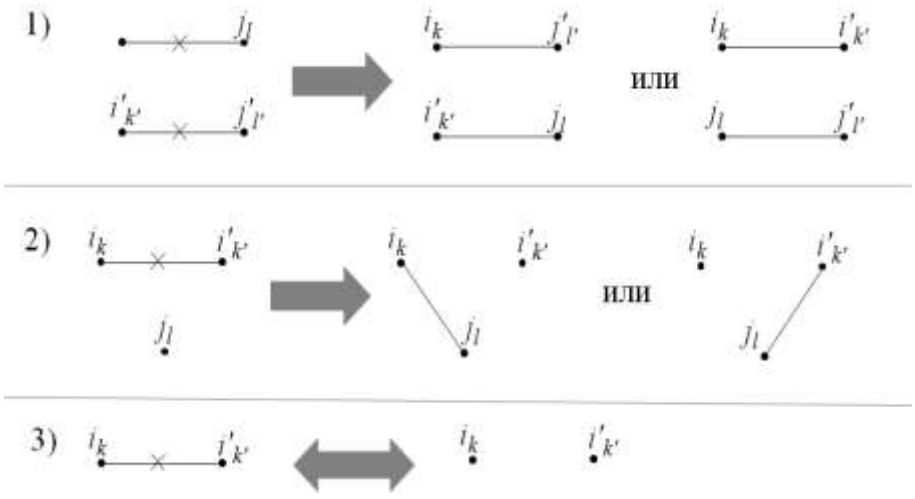


Рис. 4. Операции, разрешённые над общим графом

Двойная переклейка (п. 1 на рис. 4): удаление двух одинаково помеченных рёбер общего графа и соединение четырёх образовавшихся концов двумя новыми неинцидентными рёбрами с той же пометкой.

Полуторная переклейка (п. 2 на рис. 4): удаление ребра общего графа и соединение ребром с той же пометкой одного из его концов с вершиной, не инцидентной ребру с этой пометкой.

Разрез (п. 3 на рис. 4, слева направо): удаление любого ребра.

Склейка (п. 3 на рис. 4, справа налево): добавление ребра (скажем, с пометкой a) между вершинами, каждая из которых не инцидентна ребру с пометкой a.

Качеством $H(a+b)$ общего графа $a+b$ назовём число циклов, сложенное с половиной числа чётных цепей в нем. Чётной называется цепь с чётным числом рёбер, а также точка (цепь длины ноль). Пусть структуры a и b имеют по n генов. Для построения искомой последовательности операций

решающее значение имеет возрастание качества от $H(a+b)$ до n на 1 при выполнении каждой операции; таким образом, требуемое число операций можно указать сразу: оно равно $n–H(a+b)$. Следующая теорема это обосновывает. Назовём операцию *верной*, если она увеличивает качество общего графа на 1.

Теорема 1 [2]. *Каждая операция изменяет качество общего графа не более, чем на* 1. *Для любого нефинального графа существует верная операция, которую можно найти за линейное время*. *Качество финального графа равно* n.

Очевидно, из теоремы 1 следует существование линейного по времени алгоритма, строящего кратчайшее преобразование a в b. Отметим: линейный алгоритм для этой задачи был известен [1], однако у нас используется более простой вспомогательный граф (вышеописанный общий граф), в результате чего описание алгоритма значительно упростилось.

Если в обеих структурах a и b все хромосомы кольцевые, в общем графе, очевидно, все компоненты циклические. В этом случае достаточно применять лишь двойные переклейки. Тогда и все промежуточные структуры при переходе от a к b будут иметь лишь кольцевые хромосомы, что биологически естественно.

Если в обеих структурах a и b все хромосомы линейные, не всегда удаётся преобразовать a в b верными операциями так, чтобы и все промежуточные структуры имели лишь линейные хромосомы (хотя в большинстве случаев удаётся). Однако можно добиться того, что если возникает кольцевая хромосома, то следующая операция (двойная или полуторная переклейка) вставляет её в линейную хромосому. Это также биологически естественно с учётом действий мобильных элементов, которые часто зацикливают кусок хромосомы и вставляют его в другое место.

Вообще говоря, вопрос о том, когда можно обеспечить, чтобы топология промежуточных структур при переходе от a к b не отличалась от топологии самих a и b, довольно сложен. Пусть a и b состоят из одной кольцевой хромосомы каждая. В каких случаях можно преобразовать a в b минимальной цепочкой операций так, чтобы и все промежуточные структуры состояли только из одной кольцевой хромосомы? Рассмотрим более слабое требование: в промежуточной структуре может появиться вторая кольцевая хромосома, но тогда на следующем шаге снова остаётся одна хромосома. Теорема 2 [3] даёт частичный результат в этом направлении.

Теорема 2. *Пусть в a и b равный генный состав, обе структуры состоят из одной кольцевой хромосомы и разрешена лишь операция двойной переклейки. Существует минимальная цепочка операций, преобразующая a в b и такая, что любая промежуточная структура в ней состоит не более чем из двух кольцевых хромосом; если при некоторой операции в последовательности структур возникает вторая хромосома, то следующая операция оставляет одну хромосому.*

Из доказательства теоремы 2 нетрудно получить квадратичный по времени алгоритм [3], строящий описанную минимальную цепочку (автор не знает, возможен ли линейный алгоритм). Он имеет дополнительное свойство: если существует двойная переклейка (скажем, над a), увеличивающая качество графа $a+b$ и не создающая второй хромосомы в a, то алгоритм выбирает такую операцию.

Пусть теперь обе структуры a и b содержат лишь линейные хромосомы. Не всегда удаётся преобразовать a в b минимальной цепочкой так, чтобы и все промежуточные структуры имели лишь линейные хромосомы (хотя в большинстве случаев удаётся). Однако можно добиться того, что если возникает кольцевая хромосома, то следующая операция (двойная, полуторная переклейка или разрез) вставляет её в линейную хромосому или разрезает. Рассмотрим сначала случай, когда исходные структуры a и b состоят из одной хромосомы каждая.

Теорема 3. *Пусть в a и b равный генный состав и обе структуры состоят из одной линейной хромосомы. Существует минимальная цепочка, преобразующая a в b и такая, что любая промежуточная структура в ней содержит не более одной линейной хромосомы; если при некоторой операции в последовательности структур возникает кольцевая хромосома, то следующая операция оставляет одну линейную.*

Доказательство. Пусть обе текущие структуры состоят из одной линейной хромосомы. Если в графе $a+b$ нет нечётных b-цепей (т.е. цепей, в которых оба крайних ребра – b-рёбра), применяем любую a-операцию (возможны все операции кроме разреза) увеличивающую его качество; в результате число линейных хромосом в каждой структуре не больше 1, а в $a+b$ по-прежнему нет нечётных b-цепей (заметим: при наших условиях если в $a+b$ нет нечётных b-цепей, то нет и нечётных a-цепей). Иначе, склейкой замыкаем нечётную b-цепь в цикл. Поскольку нечётная b-цепь могла быть лишь одна, теперь таких цепей в графе $a+b$ нет (хотя может иметься нечётная a-цепь). Далее, доказываем теорему 2 по той же схеме, что и теорему 2 [3]. Пусть в результате последней операции в структуре a (последняя операция является a-операцией) появилась кольцевая хромосома H и в $a+b$ нет нечётных b-цепей. Рассмотрим четыре случая.

1) В структуре a нет линейной хромосомы. Тогда в $a+b$ есть нечётная a-цепь. Применим к ней операцию разреза, отрезав крайнюю вершину. Тогда в a хромосома H превратится в линейную, а качество графа $a+b$ увеличится, что и требуется.

2) Существует компонента в графе $a+b$, содержащая a-ребро e_1 из линейной хромосомы и a-ребро e_2 из H. Применим к паре (e_1,e_2) двойную переклейку, увеличивающую число циклов в $a+b$. Тогда в a хромосома H объединится с линейной, а качество графа $a+b$ увеличится, что и требуется.

3) Существует чётная цепь в графе $a+b$, содержащая a-ребро e из H. Тогда внешний край v её крайнего b-ребра является краем линейной хромосомы в a. Применим к паре (e,v) полуторную переклейку, увеличивающую число циклов в $a+b$. Тогда в a хромосома H объединится с линейной, а качество графа $a+b$ увеличится, что и требуется.

4) Каждая компонента в $a+b$ содержит только края генов из линейной хромосомы в a или только края генов из H. Тогда не существует b-ребра, соединяющего край гена из первой хромосомы с краем из H, что противоречит тому, что b состоит из одной хромосомы. Теорема 3 доказана.

Из доказательства теоремы 3 получается квадратичный по времени алгоритм, строящий описанную минимальную цепочку. Рассуждения аналогичны приведённым после доказательства теоремы 1 [3]. Аналогичным образом обеспечивается и дополнительное свойство алгоритма: если существует операция (скажем, над a), увеличивающая качество $a+b$ и не создающая второй хромосомы в a, то алгоритм выбирает такую операцию.

Теперь рассмотрим случай, когда исходные структуры a и b имеют только линейные хромосомы (возможно, по несколько).

Теорема 4. *Пусть в a и b равный генный состав, обе структуры содержат лишь линейные хромосомы. Существует минимальная цепочка, преобразующая a в b и такая, что любая промежуточная структура в ней содержит не более одной кольцевой хромосомы; если при некоторой операции в последовательности структур возникает кольцевая хромосома, то следующая операция оставляет только линейные.*

Доказательство. Пусть обе текущие структуры содержат только линейные хромосомы. Если в графе $a+b$ нет нечётных цепей, применяем любую a-операцию, увеличивающую его качество; в результате в $a+b$ по-прежнему нет нечётных цепей. Иначе, операцией разреза отрезаем от каждой нечётной цепи крайнюю вершину, устраняя, таким образом, нечётные цепи. Далее, доказываем теорему 4 по той же схеме, что и теорему 3. Пусть в результате последней операции в структуре a (последняя операция является a-операцией) появилась кольцевая хромосома H и в $a+b$ нет нечётных цепей. Рассмотрим четыре случая.

1) В a нет линейной хромосомы. Тогда в $a+b$ есть нечётная a-цепь, что противоречит нашему предположению.

2) То же, что и в доказательстве теоремы 3.

3) То же, что и в доказательстве теоремы 3.

4) Каждый цикл в $a+b$ содержит только края генов из линейных хромосом в a или только края генов из H; каждая цепь в $a+b$ содержит только края генов из линейных хромосом в a. Тогда для любого края из H существует b-ребро, соединяющее его с краем из H. Это противоречит тому, что b содержит лишь линейные хромосомы. Теорема 4 доказана.

Квадратичный по времени алгоритм, строящий описанную минимальную цепочку, обосновывается теми же рассуждениями, что и в теоремах 2

и 3. Аналогичным образом обеспечивается и дополнительное свойство алгоритма: если существует операция (скажем, над a), увеличивающая качество $a+b$ и не создающая кольцевой хромосомы в a, то алгоритм выбирает такую операцию.

СЛУЧАЙ РАВНОГО ГЕННОГО СОСТАВА И НЕРАВНЫХ ЦЕН

Если каждой операции приписана цена, то задача преобразования, соответственно, уточняется. Требуется найти цепочку преобразующих a в b операций, чтобы их суммарная цена была минимальной. Такую цепочку будем называть *кратчайшей* (если цены всех операций равны, она совпадает с минимальной).

Задача поиска кратчайшей цепочки при произвольных ценах представляется сложной. Рассмотрим упрощённую задачу, когда самая дешёвая последовательность ищется не среди всех возможных последовательностей, а лишь среди кратчайших. Эту задачу назовём *условной*. Ограничимся случаем, когда цены операций удовлетворяют одному из двух условий: $c_2 \le c_1 \le c'_1 \le c_{1.5}$ (*циклический* вариант) и $c_1 \le c'_1 \le c_{1.5} \le c_2$ (*линейный* вариант). Здесь указаны соотношения между ценами разреза c_1, склейки c'_1, полуторной переклейки $c_{1.5}$, двойной переклейки c_2. Названия вариантов соответствуют случаям, когда в структурах преобладают кольцевые хромосомы (тогда двойные переклейки происходят чаще полуторных и их цена меньше) или линейные (тогда наоборот). Остаётся в силе и эквивалентная формулировка условной задачи в терминах общего графа с оговоркой, что разрез b-ребра имеет цену склейки, а склейка b-ребром имеет цену разреза.

Теорема 5 [2]. *Для каждого из двух указанных вариантов цен существует линейный алгоритм, решающий условную задачу преобразования a в b.*

Случай неравного генного состава. Теперь рассмотрим случай *неравного генного состава*, т.е. множества номеров генов в структурах a и b могут не совпадать. Ген, который присутствует (одним и тем же номером) и в a и в b назовём *общим*; ген, присутствующий лишь в одной из структур – *особым* (соответственно, есть a-особые и b-особые гены). Кроме четырёх прежних операций над структурой (которые назовём *стандартными*) фиксируются две следующие *дополнительные* операции: *удаление и вставка*, рис. 5.

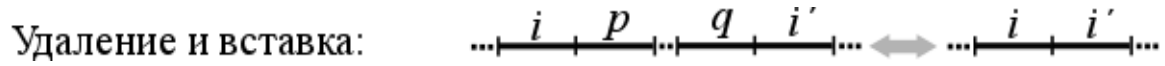


Рис. 5. Две дополнительные операции над хромосомной структурой

Удаление: *удалить* из структуры связный отрезок из a-особых генов (a-*отрезок*) и вставка: *вставить* в структуру связный отрезок из b-особых

генов (*b-отрезок*); если у удаляемого отрезка были соседние гены, то их края склеиваются.

Понятие *общего графа* $a+b$ нуждается в некоторой модификации. Теперь он имеет следующие вершины: *обычные* – номера краёв общих генов; и *особые* – максимальные по включению a-отрезки и b-отрезки, называемые *блоками*. Блок помечается как a- или b-вершина в $a+b$ и ей приписывается последовательность генов, составляющих блок. *Особое* ребро в $a+b$ соединяет обычную вершину с особой, если в a или b край, соответствующий обычной вершине, склеен с краем блока, соответствующего особой вершине (a- или b-ребро). *Петля* в $a+b$ соответствует циклу, который является блоком. *Висячим* называется особое ребро, инцидентное особой вершине степени 1.

На рис. 6 приведён типичный вид общего графа. Жирными кружочками показаны особые вершины

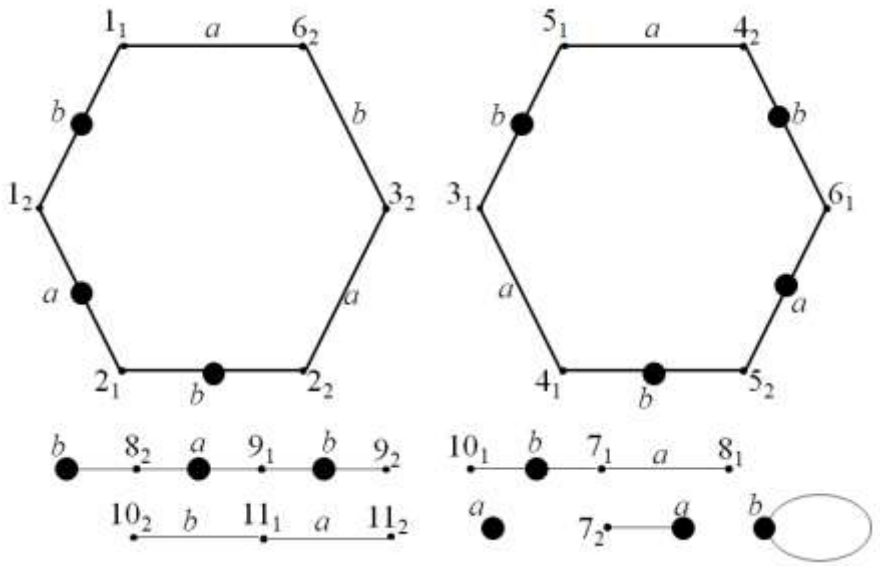


Рис. 6. Типичный вид общего графа в случае неравного генного состава

Как и прежде, общий граф неориентированный и состоит из связных компонент – цепей и циклов, а также петель, которые мы считаем отдельным типом компонент. Невисячие особые рёбра присутствуют в нём парами – это рёбра, инцидентные одной особой вершине; такую пару удобно считать за одно двойное ребро (с этой оговоркой сохраняется чередование a- и b-рёбер). *Размером компоненты* общего графа назовём сумму в ней числа обычных рёбер с половиной числа особых невисячих рёбер. Для обычных точек и петель размер считаем равным 0, для особых точек – равным "минус 1". Определение финального вида общего графа остаётся прежним.

На рис. 7 изображены две структуры (a) и их общий граф (b).

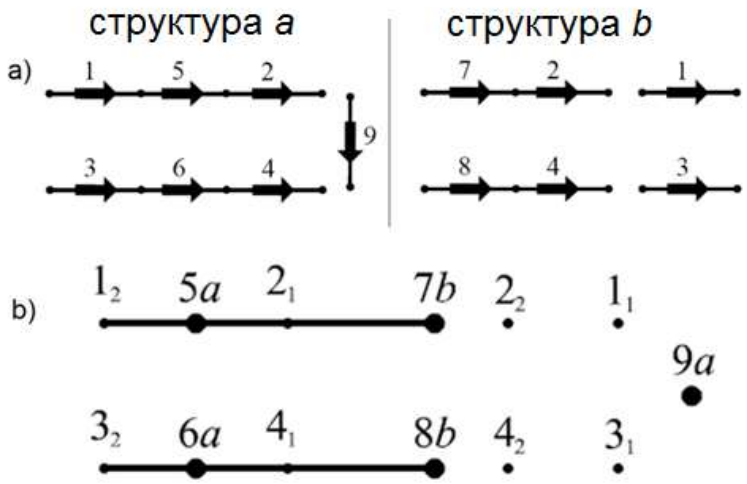


Рис. 7. Две хромосомные структуры (a) и их общий граф (b)

При определении операций над общим графом – аналогов операций над структурой, – возникает две трудности. Основная – не удаётся определить операцию вставки, менее значимая – нежелательно разбивать особые вершины на части. Поэтому, чтобы свести задачу преобразования структур к задаче приведения общего графа к финальному виду, требуется провести дополнительные рассуждения (см. теорему 6 ниже). Финальный вид, как и прежде, допускает лишь циклы длины 2 и обычные точки. Сведение доказано нами [2] для случая, когда цены всех стандартных операций одинаковы, а цены операций вставки и удаления любые (такое соотношение цен назовём *допустимым*). Как и раньше, ищем структуру c, к которой независимо приводятся a и b, суммарно кратчайшими цепочками (тогда a приводится к b *через* c).

Скажем, что операция вставки *разбивает* a-блок, если он линейный, и отрезок вставляется между двумя его генами. Операции двойной или полуторной переклейки, или разреза *разбивают* a-блок, если они включают одну или две расклейки внутри линейного a-блока, или две расклейки внутри циклического a-блока. Операция удаления *разбивает* a-блок, если удаляется лишь его часть.

Теорема 6 [2]. *При любом допустимом соотношении цен среди кратчайших* (от a к b) *цепочек существует такая, у которой никакая операция не разбивает* a*-блок, и все удаления происходят раньше всех вставок*.

Благодаря теореме 6 можно рассматривать только пять операций над общим графом, рис. 8.

Двойная переклейка (рис. 8.1): удаление двух одинаково помеченных рёбер общего графа и соединение четырёх образовавшихся концов двумя новыми неинцидентными рёбрами с той же пометкой. Если при этом образуется ребро с особыми концами (оба относятся к a или оба к b), то оно заменяется одной особой вершиной. *Полуторная переклейка* (рис. 8.2): удаление ребра графа и соединение ребром с той же пометкой одного из его

концов с обычной вершиной, не инцидентной ребру с этой пометкой, или с особой вершиной степени не больше 1 с той же пометкой (с возможным последующим отождествлением двух особых вершин). *Склейка* (рис. 8.3 справа налево и 8.3'): добавление ребра (скажем, с пометкой a) между вершинами, каждая из которых является либо обычной, не инцидентной ребру с пометкой a, либо особой, степени не больше 1 с той же пометкой (с возможным последующим отождествлением двух особых вершин). *Разрез* (рис. 8.3 слева направо): удаление любого ребра.

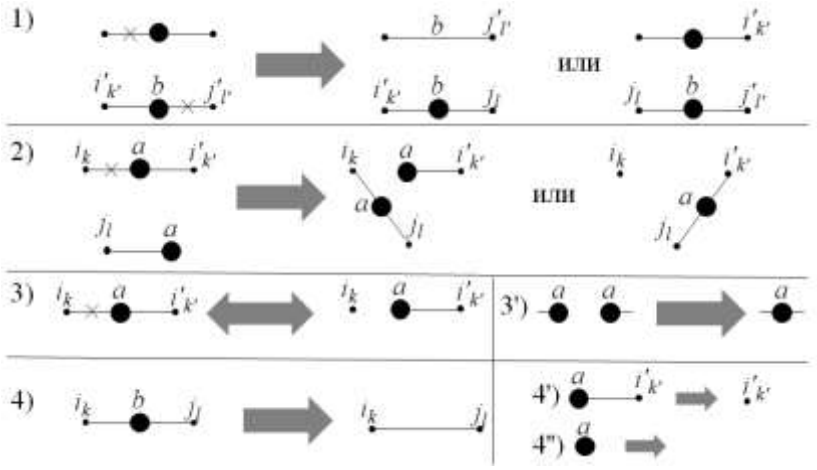


Рис. 8. Операции над общим графом в случае неравного состава

Также вводится ещё одна *дополнительная* операция *удаления* особой вершины (рис. 8.4, 8.4', 8.4"). Если её степень равна 2, то она удаляется и инцидентные ей ребра сливаются в одно ребро с той же пометкой; если эта вершина висячая, то она удаляется вместе с инцидентным ей ребром; если это точка или вершина с петлей, то вершина и петля удаляются. Удаление a-вершины является аналогом операции удаления в структуре, удаление b-вершины – аналогом операции вставки.

Ценой приведения общего графа к финальному виду называем сумму цен операций с оговоркой, что удаление b-вершины имеет цену вставки (и что разрез b-ребра имеет цену склейки, а склейка b-ребром имеет цену разреза, что при допустимом соотношении цен несущественно). Взяв в качестве c структуру, полученную после всех удалений, но до всех вставок, получаем такое следствие теоремы 6.

Следствие теоремы 6. *При допустимом соотношении цен задача поиска кратчайшей цепочки от a к b сводится к задаче приведения общего графа $a+b$ к финальному виду исходными операциями и только одной дополнительной операцией удаления особой вершины.*

Итак, задача преобразования a к b кратчайшей цепочкой эквивалентна (для допустимого соотношения цен) задаче приведения графа $a+b$ к финальному виду цепочкой с минимальной суммарной ценой операций, кото-

рую также будем называть *кратчайшей*. В [2] эта задача решена в частном случае *равных цен всех операций*, что эквивалентно поиску цепочки, минимальной по числу операций. Изложим общий план алгоритма. На этапе 1 из компонент (кроме циклов размера 2) вырезаются обычные рёбра и замыкаются в циклы длины 2. Затем на этапе 2 обрабатываются цепи. Это связано с тем, что совместная обработка цепей (их *взаимодействия*) позволяет экономить число операций, по сравнению с тем их числом, которое получилось бы при обработке каждой цепи в отдельности. Например, одно из взаимодействий (полуторная переклейка) двух цепей показано на рис. 9.



Рис. 9. Взаимодействие, экономящее одну операцию

Взаимодействия делятся на 19 типов. В [2] указан не зависящий от графа $a+b$ порядок типов взаимодействий, который приводит к максимально возможной экономии числа операций, т.е. к минимизации числа операций. После этапа 2 между оставшимися компонентами уже невозможны взаимодействия, которые экономили бы число операций.

На последнем этапе оставшиеся компоненты по отдельности приводятся к финальному виду: петли удаляются, цепи размера больше 0 замыкаются в циклы, циклы разбиваются на циклы размера 2, удаляются особые вершины. Время алгоритма линейное.

На рис. 10 показана цепочка операций, построенная алгоритмом для общего графа, изображённого на рис. 7.



Рис. 10. Приведение общего графа с рис. 7 к финальному виду

Здесь этап 1 пустой (нет обычных рёбер), затем следует этап 2 – взаимодействие трёх цепей, состоящее из двух операций (пп. 2, 3 на рис. 10) и в конце удаление двух особых вершин.

Доказательство точности алгоритма проходит по следующей схеме [2]. Число операций алгоритма выражается формулой через характеристики исходного общего графа, обозначим это число как $T(a+b)$. После этого проверяется, что в любом общем графе G никакая операция не может уменьшить $T(G)$ больше, чем на 1. Отметим: в [4] описан другой способ построения линейного по времени алгоритма решения той же задачи для равных цен операций. В отличие от него, наш способ обобщается на случай их неравных цен, рассматриваемый ниже. Частные случаи таких цен рассмотрены в [3, 5–6].

Отдельно рассмотрим *случай неравных цен*. Пусть цены четырёх стандартных операций над структурой одинаковы (считаем их равными 1), а цены операций вставки и удаления равны друг другу, но могут быть отличны от 1. Это соотношение цен, в котором цены вставки и удаления одинаковы (обозначим эту цену как w), можно назвать *стационарным*, подразумевая, что при нём не меняется количество генов в геноме. Если $w \leq 0.5$, линейный по времени алгоритм [6] решения задачи преобразования (и доказательство его точности) строится по той же вышеописанной схеме, что и для случая $w=1$ (только число типов взаимодействий цепей на этапе 3 становится существенно меньше). Если $0.5<w<1$, та же схема даёт линейный по времени алгоритм, отличающийся от описанного порядком взаимодействий. Отличие связано с тем, что в этом случае не удаётся указать фиксированный порядок типов взаимодействий цепей, приводящий к минимизации числа операций, но удаётся свести задачу поиска взаимодействий к задаче целочисленного линейного программирования (ЦЛП) с константой переменных и ограничений [7]. Известно, что такая задача решается за линейное время [8].

Еще одно соотношение цен – все цены всех стандартных операций равны 1, а цены w_a и w_b операций, соответственно, удаления и вставки удовлетворяют соотношению $1 \leq w_a \leq w_b$. Для этого случая получен линейный по времени алгоритм, аддитивная ошибка которого не превышает минимума из двух значений: 2 и w_a+w_b-2 [7]. Это соотношение цен можно назвать *нестационарным*, предполагая, что при нём идёт уменьшение числа генов в геноме. Алгоритм отличается от алгоритма для стационарного случая наличием дополнительных этапов после этапа 2. На этапе 3 в определённом порядке производятся взаимодействия между компонентами общего графа. Эти взаимодействия не меняют суммарное число операций, но позволяют заменить "дорогую" операцию удаления вершины на более дешёвую стандартную операцию. Например, если удалить две b-петли по отдельности, будет произведено две операции удаления b-вершины, если же сначала двойной переклейкой объединить эти две петли в одну, то одно удаление заменится на двойную переклейку. Ещё один пример такого взаимодействия изображён на рис. 11.



Рис. 11. Взаимодействие, заменяющее удаление *b*-вершины на полуторную переклейку

На этапе 4 производим замыкание цепей положительного размера в циклы. Цепи нулевого размера невозможно замкнуть в циклы, также затруднено их взаимодействие с циклами. Поэтому такие «короткие» цепи являются источником вышеупомянутой аддитивной ошибки алгоритма.

На этапе 5 производим объединение циклов в один цикл (рис. 12) и разбиение полученного цикла на циклы размера 2 (рис. 13).

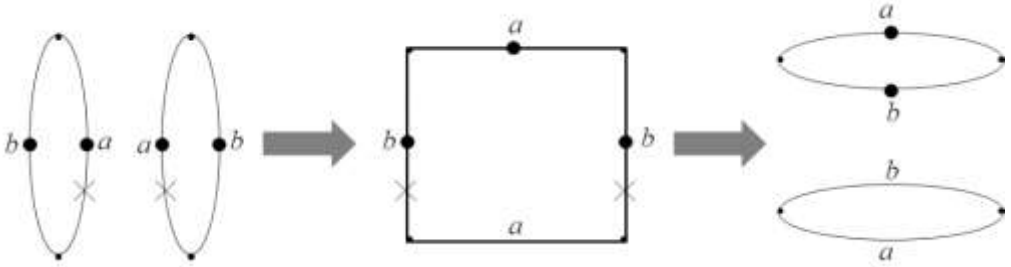


Рис. 12. Объединение двух циклов в один

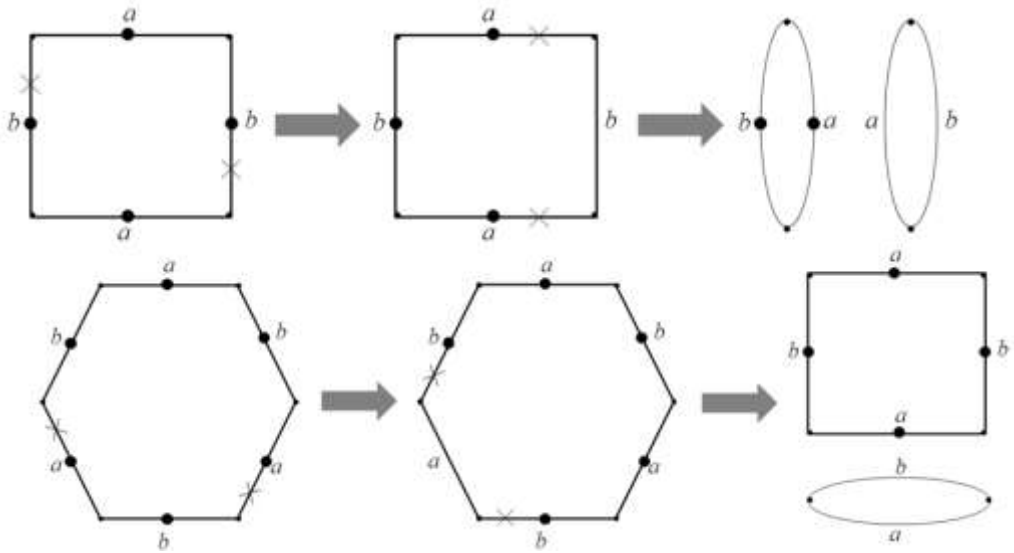


Рис. 13. Отщепление финального цикла от (*a*,*b*)-цикла размера 4 (сверху) и размера 6 (снизу)

Этап 6 применяется, если цена удаления превышает удвоенную цену переклейки. На нём производятся *взаимодействия*, каждое из которых увеличивает на 1 число операций алгоритма, но заменяет операцию удаления на две переклейки. Например, взаимодействие, изображённое на рис. 14, заменяет удаление *a*-вершины на две двойных переклейки.

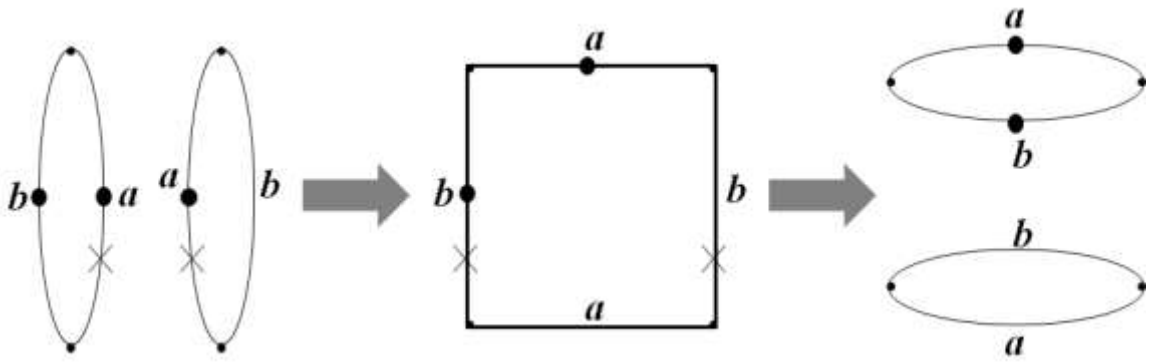


Рис. 14. Взаимодействие, заменяющее удаление на две переклейки

На финальном этапе из оставшихся цепей и циклов (все они имеют размер не больше 2) удаляем особые вершины и получаем граф финального вида.

В частном случае, когда $1=w_a \le w_b \le 2$ аддитивная ошибка алгоритма не превышает w_b-1 [5, 9]. В другом частном случае, когда все хромосомы в исходных структурах кольцевые, алгоритм становится точным [3]. Это связано с тем, что ошибка может возникнуть из-за наличия в графе $a+b$ цепи, содержащей особые вершины лишь одного типа (a- или b-вершины). Если таких цепей нет, алгоритм точен.

РЕШЕНИЕ ЗАДАЧИ ПРЕОБРАЗОВАНИЯ ПРИ НАЛИЧИИ ПАРАЛОГОВ В СТРУКТУРАХ

Мы рассмотрели задачу построения наиболее экономного преобразования одной хромосомной структуры в другую для случая, когда каждому гену присвоено уникальное имя (или номер). Однако, важное в прикладных вопросах обобщение рассмотренной задачи состоит в том, что в структурах разрешается повторение имён. Это обобщение назовём *задачей с паралогами*. Наличие паралогов делает рассматриваемую задачу NP-трудной. Однако при равных ценах всех операций удалось [10–11] свести её к задаче целочисленного линейного программирования (ЦЛП) с квадратичным числом переменных и ограничений, что позволяет использовать эффективные алгоритмы, разработанные для задач ЦЛП.

Итак, *паралоги* – гены с одинаковым номером k, для их различения вводим двойную нумерацию вида $k.i$. Например, пусть в a имеется три гена с номером k, а в b – два гена с тем же номером. Требуется занумеровать эти пять генов *вторыми* номерами i (т.е. полные номера будут иметь вид $k.i$) так, чтобы получившаяся нумерация (уже без повторений номеров) давала наиболее дешёвый переход от a к b. Нумерация вторыми номерами определяет частичное соответствие между генами в a и b, в частности, какие гены будут общими, а какие особыми. Например, если первые три гена

занумеровать как *k*.1, *k*.2, *k*.3, а вторые два как *k*.2 и *k*.4, то ребро с номером *k*.2 будет общим для *a* и *b*, а рёбра с остальными номерами – особыми.

Пусть *a* и *b* – две данные хромосомные структуры (уже с паралогами). Зададим на них произвольные полные нумерации генов (получившиеся структуры по-прежнему обозначаем *a* и *b*). Введём булевы переменные z_{kij}, где z_{kij}=1, если ген *k.i* в структуре *a* соответствует гену *k.j* в структуре *b*, иначе z_{kij}=0. Основная идея сведения к ЦЛП: с помощью этих и других переменных выразить линейными выражениями величины, входящие в формулу вычисления минимального числа операций, требуемых для преобразования *a* в *b*, и получить из этой формулы целевую функцию для задачи ЦЛП.

Продемонстрируем эту идею на простом примере с равным (точнее, с сбалансированным) генным составом. Пусть в каждой из структур по одной «треугольной» хромосоме, рис. 15, и надо определить, какой из генов с номером 1 в структуре *a* соответствует какому гену с тем же номером в структуре *b* (выбрать один из двух вариантов).

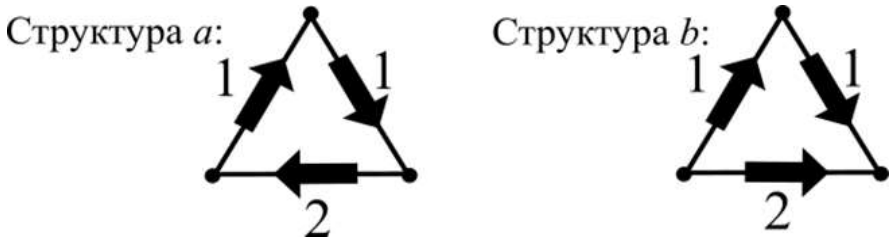


Рис. 15. Две структуры с паралогами

Как мы знаем, для этого следует максимизировать качество общего графа *a*+*b*, т.е. число циклов в нём. Начнём с произвольной нумерации паралогов, чтобы на них можно было указывать, рис. 16.

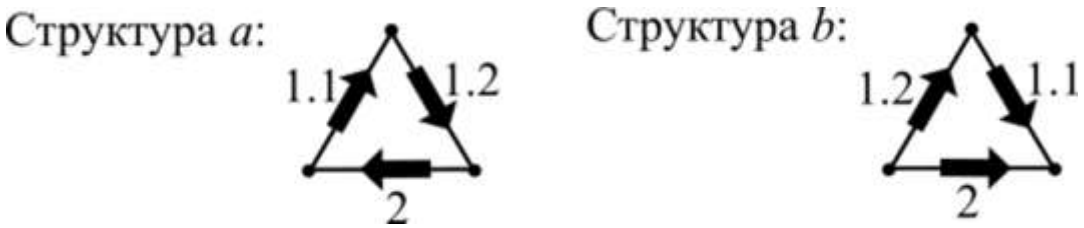


Рис. 16. Произвольная начальная нумерация паралогов

Как ранее было описано, соответствие между паралогами определяется переменными z_{11}, z_{12}, z_{21}, z_{22}. Ограничения $z_{11}+z_{12}=1$, $z_{21}+z_{22}=1$, $z_{11}+z_{21}=1$, $z_{12}+z_{22}=1$, обеспечивают его биективность. Чтобы выразить число *C* циклов в *a*+*b* естественно сопоставить каждой склейке *s* краёв генов в исходных структурах (а следовательно, и каждому ребру общего графа) бу-

леву переменную p_s так, чтобы на каждом цикле в $a+b$ ровно одна из этих переменных была равна 1, а остальные равны 0; тогда C будет равно сумме всех шести переменных p_s. Чтобы этого добиться, сопоставим каждой склейке s целочисленную неотрицательную переменную u_s (рис. 17) так, чтобы на рёбрах каждого цикла графа $a+b$ эти переменные принимали одно и то же значение (в разных циклах оно может быть разным).

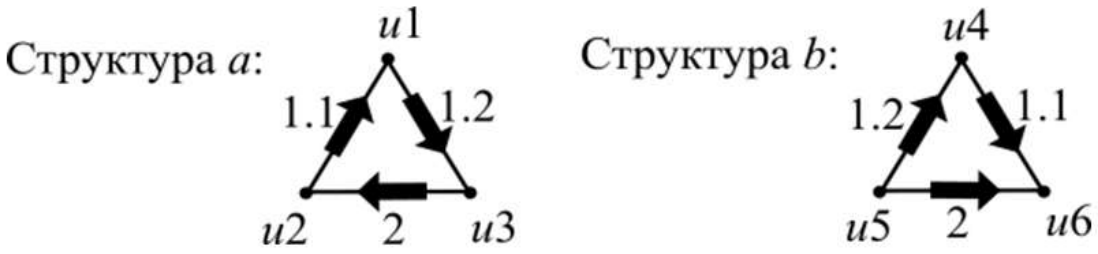


Рис. 17. Сопоставление переменных u_s склейкам краёв генов

Достаточно обеспечить равенство каждых двух переменных u_s на соседних рёбрах в $a+b$. Произвольно занумеруем склейки s (и, соответственно, переменные u_s), начиная с 1, и наложим ограничения $u_{s1}\le 1$, $u_{s2}\le 2,\ldots,u_{s6}\le 6$, т.е. каждая переменная u_s не превышает своего номера n_s. Легко видеть, что требуемое свойство будет выполнено, если для каждой из четырёх пар генов $<1.i$ из a, $1.j$ из $b>$ и склеек s_{1i1}, s_{1j1} их начал (и, аналогично, склеек s_{1i2}, s_{1j2} их концов) наложить, пару ограничений $u_{s1i1}\le u_{s1j1}+n_{s1i1}(1-z_{1ij})$, $u_{s1j1}\le u_{s1i1}+n_{s1j1}(1-z_{1ij})$ (а для гена 2 потребовать $u_{s21a}=u_{s21b}$, $u_{s22a}=u_{s22b}$).

Осталось показать, как обеспечить единственность единичного значения переменных p_s на цикле в $a+b$. Свяжем каждую пару переменных u_s и p_s ограниченем $p_s n_s\le u_s$, т.е. $p_s\le u_s/n_s$. Если переменная u_s меньше n_s, то соответствующая переменная p_s равна 0. Иначе, из ограничения ничего не следует, но в точке максимума целевой функции (напомним, она равен сумме всех p_s) эта p_s равна 1, иначе это не был бы максимум.

Действительно, на цикле в $a+b$ все переменные u_s принимают одно значение. В точке максимума ровно одна из них равна своему номеру n_s, т.е. принимает своё максимальное значение. Больше одной быть не может, поскольку все эти значения различны, а если бы никакая не достигала своего максимума, их все можно было бы одинаково увеличить, чтобы одна достигла его и соответствующая p_s стала равна 1, увеличив целевую функцию. В решении полученной задачи ЦЛП имеем $z_{112}=1$ и $z_{121}=1$.

Таким образом, в структуре b следует поменять местами имена 1.1 и 1.2, рис. 18.

Максимум целевой функции равен 2 – легко проверить, что в итоговом общем графе два цикла (а в начальном – один цикл).

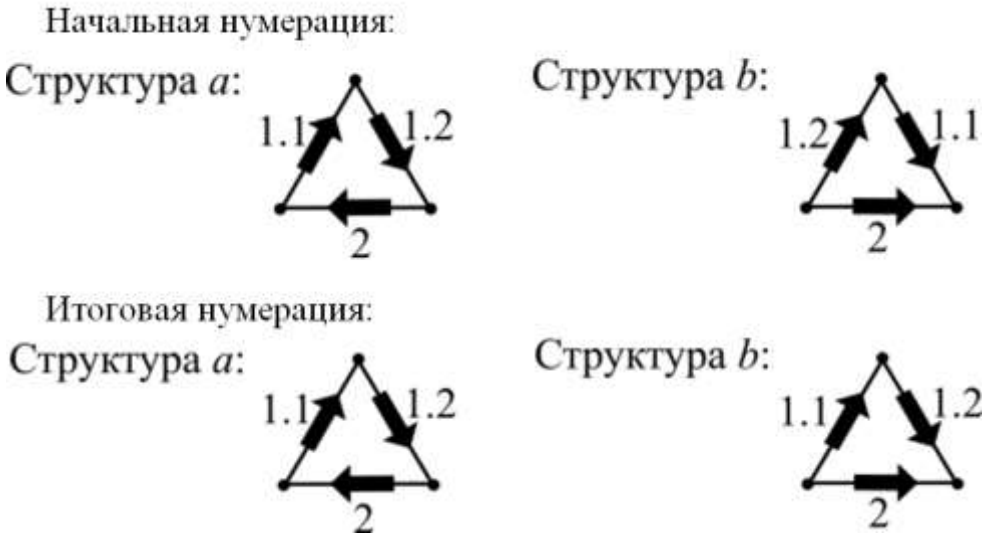


Рис. 18. Перенумерация паралогов в соответствии с результатом решения задачи ЦЛП

Благодарности. Работа поддержана РФФИ (грант № 18-29-13037).

Л И Т Е Р А Т У Р А

1. Bergeron A., Mixtacki J., Stoye J. A unifying view of genome rearrangements. Algorithms in Bioinformatics, LNCS, 2006; 4175:163–173.

2. Горбунов К.Ю., Любецкий В.А. Линейный алгоритм минимальной перестройки структур. Проблемы передачи информации, 2017, т. 53, вып. 1, стр. 60–78.

3. Lyubetsky V.A., Lyubetskaya E.V., Gorbunov K.Yu. Linear algorithm for a cyclic graph transformation. Lobachevskii Journal of Mathematics, 2018, Vol. 39, No. 9, P. 1217–1227.

4. Compeau P.E.C. DCJ-indel sorting revisited. Algorithms for Molecular Biology, 2013, vol. 8, pp. 6.1–6.9.

5. Горбунов К.Ю., Любецкий В.А. Линейный алгоритм кратчайшей перестройки графов при разных ценах операций. Информационные процессы, 2016, т. 16, № 2, стр. 223–236.

6. Горбунов К.Ю., Любецкий В.А. Линейный алгоритм перестройки графа. Автоматика и телемеханика, 2018, том 79, № 12, стр. 124–141.

7. Gorbunov K.Yu., Lyubetsky V.A. A Fast Algorithm of Chromosome Structure Transformation. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2019, на рецензировании.

8. Lenstra JR H.W. Integer Programming with a Fixed Number of Variables. Mathematics of Operations Research. 1983. Vol. 8. No. 4. P. 538–548.

9. Горбунов К.Ю., Любецкий В.А. Модифицированный алгоритм преобразования хромосомных структур: условия абсолютной точности. Современные информационные технологии и ИТ-образование, 2016, т. 12, № 1, стр. 162–172.

10. Lyubetsky V.A., Gershgorin R.A., Seliverstov A.V., Gorbunov K.Yu. Algorithms for Reconstruction of Chromosomal Structures. BMC Bioinformatics, 2016, vol. 17, no. 40, pp. 1–23.

11. 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, no. 537, pp. 1–18.

МОДЕЛЬ ГЕННОЙ РЕГУЛЯТОРНОЙ СЕТИ

Горбунов К.Ю.

Дискретная булевская *генетическая сеть* [1] – это конечное множество $g_1,g_2,\ldots,g_n$ из n генов, где каждый ген может находиться в одном из двух состояний (0 – пассивное и 1 – активное). Каждой упорядоченной паре генов (g,g') (возможно, $g=g'$) сопоставлено рациональное число – интенсивность воздействия гена g на ген g', когда g находится в активном состоянии. Оно может быть положительным (активация) или отрицательным (репрессия). Также, каждому гену сопоставлена некоторая интенсивность стационарного воздействия, независимая от состояний других генов, и неотрицательное рациональное число – порог активности. Функционирование сети происходит по шагам: в каждый следующий момент $t+1$ времени состояние $g(t+1)$ каждого гена g определяется суммой интенсивностей воздействий на него в момент t. Если эта сумма превышает порог его активности, ген g переходит в активное состояние, иначе – в пассивное (независимо от значения $g(t)$). *Стационарным состоянием* (или *аттрактором*) сети называется такой набор состояний всех генов, который не меняется со временем, т.е. для каждого гена g выполняется $g(t+1)=g(t)$. Последовательность состояний сети называется её *эволюционированием*. Первый повтор состояний в ней является либо стационарным состоянием (если повторилось предыдущее состояние), либо предельным циклом (если повторилось более раннее состояние). Задача поиска аттракторов (и предельных циклов заданной длины) в описанной сети является известной NP-трудной задачей, один из возможных подходов к ней описан в [1] и основан на использовании временной логики GRN. Мы предлагаем два других подхода: первый основан на сведении задачи к целочисленному линейному программированию (ЦЛП), второй – на обобщении задачи и сведении её к линейному или квадратичному программированию. Число переменных и ограничений в этих задачах линейно зависит от числа генов в сети.

ОПИСАНИЕ МОДЕЛИ. СВЕДЕНИЕ ЗАДАЧИ К ЦЛП

Каждому гену g_i сопоставим одноименную булеву переменную, выражающую его состояние. Пусть $g_1,g_2,\ldots,g_m$ – гены, активирующие ген g_i, а