Программа Distance-Common-GGL

для сведения к ЦЛП задачи вычисления кратчайших расстояния и преобразования в случае кольцевых и линейных хромосом

Руководство пользователя

(**http://lab6.iitp.ru/ru/chromoggl**)

(авторы: В.А. Любецкий, К.Ю. Горбунов, Р.А. Гершгорин)

2019 год

Оглавление

Общие сведения

Программа **Distance-Common-GGL** предназначена для автоматической конвертации задачи нахождения кратчайшего расстояния между произвольными геномными (синоним: хромосомными) структурами с паралогами в стандартный формат задачи целочисленного линейного программирования (ЦЛП). Рассматривается общее определение структуры как произвольного множества путей и циклов, представляющих линейные и кольцевые хромосомы, вместе с операциями, которые преобразуют одну структуру в другую. Структуры включают паралоги генов, последовательность операций допускает переменный генный состав. Задача состоит в минимизации числа операций в последовательности, которая преобразует одну структуру в другую. Последовательность, на которой достигается минимум, называется *кратчайшей*. Число операций этой последовательности называется *кратчайшей длиной*. Задача вычисления кратчайшей длины является NP-трудной, поэтому мы предлагаем её сведение к задаче ЦЛП с квадратичным числом переменных и ограничений.

Описание модели

Модель хромосомной структуры описывается как конечное множество ориентированных цепей и циклов, включая петли. Такое множество можно рассматривать как ориентированный граф, который будем называть *хромосомной структурой*. Ребро графа будем называть *геном*; отдельную цепь или отдельный цикл графа – *хромосомой* или *компонентой*. Каждому гену приписано имя, обычно *номер i* этого гена, который может повторяться (в случае паралогов) и тогда номер принимает вид *i.j*. Такая модель, как обычно, означает, что не учитываются длины генов и межгенных участков, как и состав генов и межгенных участков; направление ребра показывает, на какой цепи лежит ген. Вершина графа показывает «место» соединения соседних генов, независимо от их цепи, т.е. в вершине *отождествляются* (мы говорим, *склеиваются*) два края соседних генов. Обычно в структуре много цепей и циклов, что приводит к их своеобразному взаимодействию, поэтому ситуация многих хромосом в структуре решительно отличается от ситуации одной хромосомы.

Модель включает следующие *стандартные* операции над хромосомной структурой. *Двойная переклейка* – расклейка двух склеек краёв генов и новая переклейка четырёх краёв; *полуторная переклейка* – расклейка двух склеенных краёв и склеивание одного края с каким-то несклеенным краем, второй край остаётся *свободным*; *разрез* или *склейка* – соответственно расклейка двух склеенных краёв с образованием двух свободных краёв или склейка двух свободных краёв. Пусть даны хромосомные структуры *a* и *b*, *общим* (*особым*)

называется ген, который принадлежит обеим структурам (только одной из них); ген из структуры *a* называется *a-геном*, соответственно определяется *b*-ген. Модель включает две *дополнительные* операции (подразумевается преобразование *a* в *b*): *удаление* (связного максимального) участка особых *a*-генов и *вставка* участка особых *b*-генов. При удалении, если участок находился строго внутри цепи или цикла, два образовавшихся свободных конца общих генов склеиваются между собой; если он находился с краю цепи, край общего гена становится свободным; наконец, если он являлся отдельной хромосомой, она удаляется целиком. Если участок вставляется строго внутри цепи или цикла, место вставки предварительно расклеивается; вставка может выполняться с краю цепи или как новая хромосома. Нетрудно доказать, что использование немаксимальных удалений особых генов не приводит к расширению возможностей, как и разрезание участка особых *a*-генов в первых трёх операциях или операции вставки. Поэтому эти возможности не рассматриваются.

Напомним, задача состоит в поиске *кратчайшей* последовательности из этих операций, которая переводит структуру *a* в структуру *b.* Здесь «кратчайшая» означает последовательность, у которой число составляющих её операций минимально. В последовательности каждая операция рассматривается вместе с хромосомной структурой, к которой она применяется.

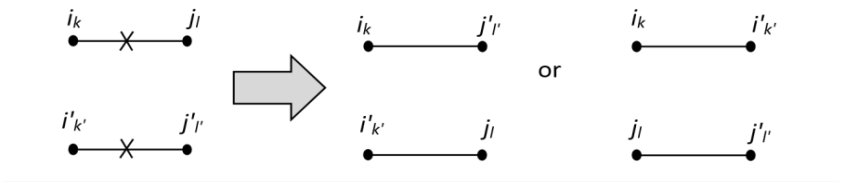
Определение общего графа и его финального вида. В общем графе *a*+*b* двух структур *a* и *b* вершины – края общих генов и, кроме того, все максимальные участки особых генов. Каждый край берётся один раз; более формально: вместо края пишется имя гена с индексом 1 или 2, указывающим на его начало или конец. Вершины первого типа называются *обычными*, а второго – *особыми*. Ребро соединяет две обычные вершины, если в одной из структур края склеены, т.е. примыкают друг к другу на хромосоме. Ребро соединяет обычную вершину с особой, если край общего гена склеен с крайним геном участка особых генов. Рёбра из первого случая называются *обычными*, а из второго – *особыми*. В цепи крайнее ребро с особым краем назовём *висячим*. Ребро помечается *a* или *b* в зависимости от того, в какой из них имеется склейка; вершины могут соединяться двумя рёбрами. Особые вершины делятся на *a*- и *b*-вершины. В графе могут быть изолированные вершины – участки из особых генов: если такой участок – цикл, то проводим в нём петлю, которую назовём *особой*. Получается неориентированный граф.

К общему графу *a*+*b* применяются аналоги операций над структурами, которые описываются следующим образом. (1) Удалить два одинаково помеченных ребра и четыре образовавшихся конца соединить двумя новыми неинцидентными рёбрами с той же пометкой. (2) Удалить ребро (скажем, с пометкой *a*) и соединить *a*-ребром один из его

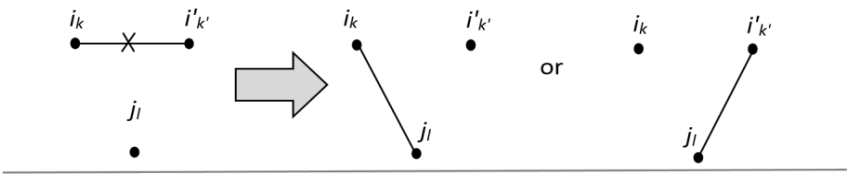
концов с обычной вершиной, не инцидентной *a*-ребру или с особой *a*-вершиной, имеющей не более одного инцидентного *a*-ребра. (3) Удалить любое ребро. (4) Добавить ребро (скажем, с пометкой *a*) между вершинами, не инцидентными *a*-ребру. Если в результате операции получаются две инцидентные особые вершины, они сливаются в одну вершину, что входит в определение операции; получаемой вершине приписывается объединение имён исходных вершин. (5) Удаления особой вершины или особой петли; если эта вершина имела две инцидентные ей обычные вершины, они соединяются ребром. Легко определить аналог операция вставки, но оказывается, что без неё можно обойтись без потери общности, что является нетривиальным утверждением.

Стандартные операции

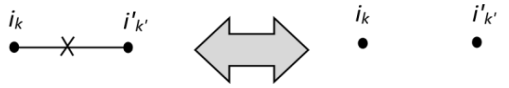
1) Double-cut-and-paste (DP). Исходные рёбра одноимённые.



2) Sesqui-cut-and-paste (SP)



3) Удаление *a*-ребра или добавление *b*-ребра (C) и удаление *b*-ребра или добавление *a*-ребра (J)



Финальный вид общего графа *a+b* определяется как общий граф, состоящий из изолированных обычных вершин и *финальных 2-циклов*. Последнее определяется как граф из двух обычных вершин, соединённых обычными рёбрами, одно из *a* и одно из *b.* Легко показать, что *исходная задача эквивалентна* приведению графа *a+b* к финальному виду.

Сведение к ЦЛП с квадратичным числом переменных и ограничений

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.

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*. 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 [1], что расстояние между $a^-(z,t_0)$ и $b^-(z,t_0)$ равно расстоянию между $a'(z)$ и $b'(z)$ для любого z. В [1] нет переменной z, так как не рассматриваются паралоги, также не используется переменная t. Поэтому решение задачи о расстоянии подразумевает нахождение $\min\limits_z \min\limits_t \Phi(z,t)$. 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 $\min\limits_{z,t} \Phi(z,t) = \min F(o,z,p,r,l)$

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 $r_{ag_1} + r_{ag_2} \leqslant 2(1 - t_{ag_1g_2})$. 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 [2]**.**

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.

Входные данные программы Distance-Common-GGL

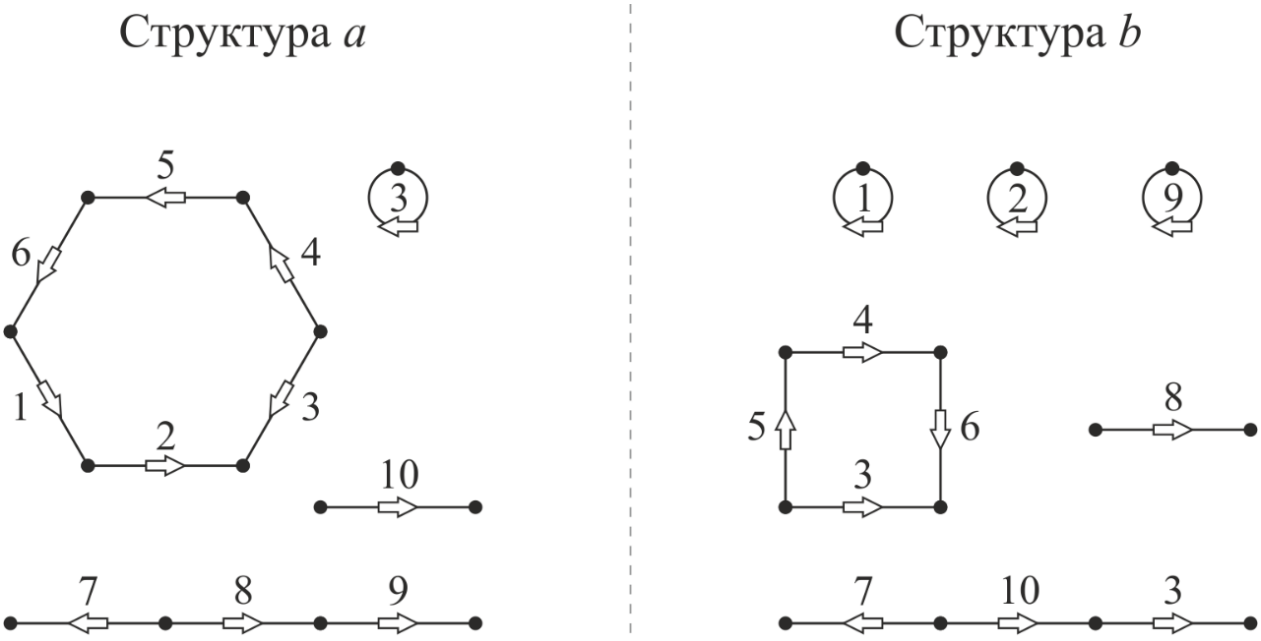


Рис. 1. Пример двух входных структур.

На вход программы подается **файл структур**, содержащий следующие данные.

I. Количество N различных имён генов, присутствующих в структурах, и далее N строк, каждая из которых содержит по два числа, первое – номер гена, второе – максимальное число паралогов для гена с данным номером. Например, для рис. 1:

```
10
1 1
2 1
3 2
4 1
5 1
6 1
7 1
8 1
9 1
10 1
```

II. Количество структур, которое в задаче преобразования равно 2, далее две строки, каждая из которых содержит описание структуры в следующем формате: имя вида/штамма; индекс соответствующего этой структуре листа в дереве; число хромосом, включённых в структуру из этого вида, т.е. число путей и циклов в структуре; пометка (L), если хромосома линейная; или (C), если она кольцевая; число генов в хромосоме; последовательность генов в хромосоме, записанная как имена генов со знаками «+» или «–», которые указывают на транскрибируемую цепь. Для рис. 1 описание имеет следующий вид:

```
2
StructA; 0; 4; C6: +1.1+2.1-3.1+4.1+5.1+6.1; C1: +3.2; L3: -7.1+8.1+9.1; L1: +10.1;
StructB; 1; 6; C4: +5.1+4.1+6.1-3.1; C1: +1.1; C1: +2.1; C1: +9.1; L1: +8.1; L3: -7.1+10.1+3.2;
```

Пример входного файла содержится в составе дистрибутива программы (**data/input_common.txt**).

Входные параметры программы и её работа

Программа запускается с двумя параметрами командной строки:

-i [имя файла с входными данными] -o [имя файла, содержащего задачу ЦЛП]

Если расширение имени выходного файла не равно .lp, программа автоматически добавит его. Данное расширение используется утилитой cplex.

В процессе работы программы на консоль выводится текущий этап, например, считывание дерева, структур, или вычисление ограничений для переменных.

Выходные данные программы Distance-Common-GGL

В результате работы программа записывает в выходной файл соответствующую задачу целочисленного линейного программирования в формате LP [3]:

Minimize

- 2 p_0_1.1h2.1t - 2 p_0_1.1t6.1h - 2 p_0_2.1h3.1h - 2 p_0_3.1t4.1t - 2 p_0_3.2h3.2t - 2 p_0_4.1h5.1t - 2 p_0_5.1h6.1t - 2 p_0_7.1t8.1t - 2 p_0_8.1h9.1t - 2 p_1_1.1h1.1t - 2 p_1_10.1h3.2t - 2 p_1_10.1t7.1t - 2 p_1_2.1h2.1t - 2 p_1_3.1h6.1h - 2 p_1_3.1t5.1t - 2 p_1_4.1h6.1t - 2 p_1_4.1t5.1h - 2 p_1_9.1h9.1t - 2 z_1.1.1 - 2 z_10.1.1 - 2 z_2.1.1 - 2 z_3.1.1 - 2 z_3.1.2 - 2 z_3.2.1 - 2 z_3.2.2 - 2 z_4.1.1 –

…

Subject To

z_1.1.1 <= 1

z_10.1.1 <= 1

z_2.1.1 <= 1

z_3.1.1 + z_3.1.2 <= 1

z_3.2.1 + z_3.2.2 <= 1

…

Bounds

u_1_1.1h1.1t >= 0

u_1_1.1h1.1t <= 14

u_0_1.1h2.1t >= 0

u_0_1.1h2.1t <= 1

u_0_1.1t6.1h >= 0

u_0_1.1t6.1h <= 6

…

Binary

p_0_1.1h2.1t

p_0_1.1t6.1h

p_0_2.1h3.1h

p_0_3.1t4.1t

…

General

l_1.1h

l_1.1t

l_10.1h

l_10.1t

l_2.1h

…

End

Полностью файл приведен в контрольном примере в составе дистрибутива программы (**results/output_common.lp**). Подробное описание всех переменных, участвующих в записи задачи ЦЛП, можно найти в [4].

Решение задачи ЦЛП утилитой cplex

Для решения задачи ЦЛП применялись как облачные вычисления, так и вычисления с помощью утилиты IBM на локальном сервере. Для вычисления в облаке IBM необходимо воспользоваться ссылкой [5]. Для вычисления на локальной машине необходимо скачать IBM ILOG CPLEX Optimization Studio [6] и воспользоваться утилитой **cplex.exe**. Утилита запускается в командной строке и далее поддерживает набор команд. Для загрузки файла (например, output_common.lp) с задачей ЦЛП формата LP необходимо ввести команду

```
> read output_common.lp
```

Далее необходимо запустить оптимизатор командой

```
> optimize
```

И, наконец, после окончания вычислений, необходимо записать найденное решение в файл командой

```
> write ilp_solution_common.sol
```

В результате запишется XML-файл ilp_solution_common.sol, в котором будет представлено найденное решение (оптимальное значение функционала, соответствующие значения всех переменных).

```
<?xml version = "1.0" encoding="UTF-8" standalone="yes"?>
<CPLEXSolution version="1.2">
 <header
  problemName="output_common.lp"
  ...
  objectiveValue="-27"
  ...
 <quality
  epInt="1.0000000000000001e-05"
  epRHS="9.9999999999999995e-07"
  maxIntInfeas="0"
  maxPrimalInfeas="0"
  maxX="3"
  maxSlack="49"/>
 ...
 <variables>
  <variable name="p_0_1.1h2.1t" index="0" value="1"/>
  ...
  <variable name="z_1.1.1" index="18" value="1"/>
  ...
```

```
 <variable name="l_1.1h" index="31" value="-0"/>
 ...
 <variable name="r_0_1.1h" index="53" value="0"/>
 ...
 <variable name="t_1_1.1h1.1t" index="99" value="0"/>
 ...
 <variable name="u_0_1.1h2.1t" index="561" value="1"/>
 ...
</variables>
</CPLEXSolution>
```

Полностью файл решения для рассматриваемого примера приведен в составе дистрибутива программы distance-to-structs-ggl (**data/ilp_solution_common.sol**).

Утилита cplex.exe доступна в бесплатном варианте, однако она имеет внутренние ограничения на размер задачи. Для получения бесплатной версии без ограничений необходимо запросить академическую лицензию по следующей ссылке.

Преобразование решения в структуры

Для того, чтобы восстановить из решения ЦЛП структуры с оптимальным расстоянием, используется утилита **distance-to-structs-ggl.exe**. Утилита имеет три параметра запуска: -i [имя файла с решением задачи ЦЛП, выданным CPLEX], -s [имя файла с исходными структурами, формат такой же, как в разделе **Входные параметры программы**], -o [имя файла, в который будут записаны структуры с оптимальной расстановкой паралогов, формат в точности соответствует формату входного файла для ChromoGGL]. Для рассматриваемого примера файл структур приведён в составе дистрибутива программы distance-to-structs-ggl (**results/resolved_structures_common.txt**).

Литература

[1] 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.

[2] 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.

[3] https://www.ibm.com/support/knowledgecenter/SSSA5P_12.7.0/ilog.odms.cplex.help/CPLEX/FileFormats/topics/LP.html - описание формата LP задач линейного программирования, поддерживаемого IBM CPLEX Optimizer

[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
[5] https://www.ibm.com/us-en/marketplace/decision-optimization-cloud - облачный оптимизатор IBM для решения задач ЦЛП.
[6] https://www.ibm.com/analytics/cplex-optimizer - IBM ILOG CPLEX Optimization Studio, локальная утилита, позволяющая оптимизировать задачи ЦЛП.