Любецкий В.А.[1], Горбунов К.Ю.[2]

[1] Институт проблем передачи информации им. А.А. Харкевича Российской академии наук (ИППИ РАН), зав. лаб. математических методов и моделей в биоинформатике, lyubetsk@iitp.ru

[2] Институт проблем передачи информации им. А.А. Харкевича Российской академии наук (ИППИ РАН), и.о. вед. н.с. лаб. математических методов и моделей в биоинформатике, gorbunov@iitp.ru

Задачи и алгоритмы, связанные с хромосомными перестройками

КЛЮЧЕВЫЕ СЛОВА:

Геном, хромосомная структура, хромосомная перестройка, эволюционный сценарий, эволюционное дерево, предковая структура, паралог, цена события.

АННОТАЦИЯ:

В настоящее время накоплено значительное количество данных о геномах различных организмов, в частности, об их хромосомных структурах. В связи с этим приобретает особую актуальность разработка эффективных алгоритмов обработки этих данных, построения возможных сценариев хромосомных перестроек, в частности, сценариев эволюции хромосомных структур вдоль эволюционного дерева. Мы рассматриваем основные задачи в этой области и алгоритмы их решения.

Задача поиска кратчайшей последовательности хромосомных перестроек. Пусть фиксировано натуральное число n. Хромосомной структурой без паралогов назовем множество линейных или циклических хромосом, составленных из n генов. Точнее, пусть каждый ген с номером i имеет два конца: i_1 и i_2. Если два гена в хромосоме идут друг за другом, то произвольный конец одного из них может следовать за произвольным концом другого. Эти два конца естественно представлять склеенными друг с другом. Каждый конец гена может быть склеен не более, чем с одним концом другого (или того же) гена. Таким образом, хромосомную структуру естественно представлять в виде паросочетания на множестве концов генов. Граф с вершинами – концами генов и ребрами – склейками, – будем называть графом хромосомной структуры. Очевидно, вершин в этом графе в два раза больше, чем генов в структуре.

Пусть даны две хромосомные структуры на одном и том же множестве из n генов и обе представлены своими графами. Рассмотрим задачу: найти кратчайшую последовательность операций, переводящую одну структуру в другую. Разрешены следующие операции [1]: разрыв двух склеек и переклейка четырех концов по другому (двойная переклейка), разрыв одной склейки и склейка одного ее конца с изолированной

вершиной (полуторная переклейка), просто разрыв одной склейки или обратная к нему операция склейки двух изолированных вершин (одинарные переклейки). Заметим, что в силу симметричности операций задача эквивалентна задаче поиска наименьшего числа таких операций, что каждая из них применяется к любой из двух структур и в конце структуры становятся совпадающими. Именно такую задачу решает приводимый ниже алгоритм.

Общим графом двух структур назовем граф, в котором вершинами являются $2n$ концов всех генов, а ребрами – ребра обеих паросочетаний, причем ребра одного из них помечены буквой a, а другого – буквой b. Легко видеть, что этот граф является объединением чередующихся (a,b)-цепей и циклов. Длиной цепи или цикла назовем число ребер в ней (в нем), изолированные вершины считаем цепями длины 0.

Мы предлагаем линейный по времени алгоритм, строящий кратчайшую последовательность хромосомных перестроек для двух данных структур. Он описывается в терминах общего графа, и не требует построения более сложных структур, таких, как, например, в [1–2] (обзор и дальнейшие ссылки по истории вопроса можно найти в сборнике [3]). Алгоритм легко обобщается на случай присутствия паралогов в структурах.

Алгоритм. Будем преобразовывать общий граф следующим образом.

1) Пока есть цепи длины больше 2 выполняем разбиение каждой такой цепи на цикл и более короткую цепь двойной переклейкой (очевидно, такая переклейка всегда существует и задается произвольной парой одинаково помеченных ребер цепи).

2) Пока есть циклы длины больше двух (очевидно, длина цикла всегда четная), выполняем разбиение каждого такого цикла на два цикла двойной переклейкой (очевидно, такая переклейка всегда существует и задается произвольной парой одинаково помеченных ребер цикла).

3) Для каждой цепи длины 2 разбиваем ее на цикл длины 2 и изолированную вершину полуторной переклейкой.

4) Для каждой цепи длины 1 разбиваем ее на 2 изолированных вершины одинарной переклейкой.

Обоснование алгоритма. Определим качество общего графа двух структур, как число циклов в нем плюс половина числа четных цепей (т.е. цепей четной длины, в том числе длины ноль). Заметим, что общий граф двух совпадающих структур всегда имеет качество, равное n, а граф двух несовпадающих структур – строго меньшее n. Оптимальность числа операций в алгоритме следует из того, что любая операция увеличивает качество общего графа не более, чем на единицу, а каждая операция алгоритма – ровно на единицу. Линейность времени алгоритма очевидна.

Случай наличия паралогов. Пусть каждый ген представлен в одной или двух копиях (эти копии называем паралогами). Тогда для каждого i в общем графе могут быть либо одна пара вершин (i_1,i_2), либо две: (i'_1,i'_2) и (i''_1,i''_2). Генным автоморфизмом назовем подстановку на множестве генов,

где каждый ген отображается либо в себя, либо в свой паралог. Две хромосомные структуры изоморфны, если существует автоморфизм, переводящий их друг в друга (т.е. если они неотличимы с точностью до паралогов). Теперь задача: найти кратчайшую последовательность операций, переводящую данную структуру A в структуру, изоморфную данной структуре B. Поскольку генных автоморфизмов не более 2^n, то при не очень больших n их можно перебрать. Для каждого из них решается задача, описанная в предыдущем пункте. На входе у нее структура A и структура B', получающаяся применением рассматриваемого автоморфизма к структуре B.

Аналогичная постановка задачи и алгоритм имеют место и в случае произвольного числа паралогов у каждого гена, хотя число генных автоморфизмов (и, соответственно, время алгоритма) быстро растет с увеличением числа паралогов.

З**адача поиска кратчайшей последовательности хромосомных перестроек для случая наличия операций вставок и удалений.** Еще одно обобщение задачи состоит во введении операций вставок и удалений отрезков генома. Отрезок из одного или нескольких сцепленных генов может удалиться из хромосомной структуры в следующих случаях:

а) если он находится внутри линейной или циклической хромосомы (при этом два конца генов, между которыми он находился, склеиваются друг с другом);

б) если он находился с одного из краев линейной хромосомы (при этом соседний с ним конец гена остается не склеенным);

в) если он являлся отдельной линейной хромосомой.

В дополнение к пункту в) допускается удаление не только линейной, но и циклической хромосомы.

Вставка отрезка сцепленных генов – операция, обратная к удалению.

Рассматривается задача: даны две хромосомные структуры и требуется кратчайшим числом операций привести их к общей структуре (в дальнейшем это число будем называть расстоянием между структурами). При этом налагается условие – удаляться из структуры могут лишь гены, не принадлежащие другой структуре. Соответственно, вставляться могут лишь гены, не принадлежащие ни одной из двух текущих структур, но принадлежащие хоть одной из двух начальных структур. При таком условии вышеописанная задача без вставок и удалений является частным случаем данной задачи для случая, когда все гены принадлежат обеим структурам. Алгоритм решения этой задачи находится в стадии разработки.

Задача восстановления структуры вдоль дерева видов и алгоритм ее решения. В этой задаче дано эволюционное дерево видов и в каждом его листе задана своя хромосомная структура. Обозначим V – множество генов в объединении всех данных структур. Требуется восстановить хромосомные структуры на внутренних вершинах дерева так, чтобы минимизировать функционал F – сумму по всем ребрам дерева

расстояний между структурами, стоящими на концах ребра. Отметим, что разумно иметь на входе дерево, разбитое на временные слои (см. [4–6]), тогда длина каждого ребре (в смысле эволюционного времени) будет примерно одинаковым и не потребуется вводить поправочные коэффициенты в функционал *F*.

Чтобы задача могла быть решена эффективным алгоритмом, заменим функционал *F* на более простую функцию *f*. Для каждой внутренней вершины дерева введем свою переменную x_{ij} для каждой пары различных концов генов из *V* и переменную y_i для каждого гена из *V*. Смысл переменных: x_{ij}=1, если соответствующие концы склеены и x_{ij}=0 в противном случае; y_i=1 если соответствующий ген присутствует в структуре и y_i=0 в противном случае. Две переменные типа *x* назовем паралогичными, если они соответствуют одноименным концам генов с одинаковыми номерами (т.е. пары концов неотличимы с точностью до паралогов). Две переменные типа *y* назовем паралогичными, если они соответствуют генам с одинаковыми номерами (т.е. паралогам). Очевидно, все переменные типа *x* (соответственно, типа *y*) разбиваются на классы паралогичных переменных. Функция *f* равна сумме квадратов следующих выражений:

1) Для каждого ребра дерева и каждого класса паралогичных переменных типа *x*: сумма переменных данного класса на одном конце ребра минус их сумма на другом конце.

2) Для каждого ребра дерева и каждого класса паралогичных переменных типа *y*: сумма переменных данного класса на одном конце ребра минус их сумма на другом конце.

Условия, при которых минимизируется функция *f*, следующие:

1) Все переменные принимают значение 0 или 1.

2) Если переменная y_i представляет ген, один из концов которого соответствует переменной x_{ij}, то выполняется неравенство $y_i \geq x_{ij}$ (т.е. если x_{ij}=1, то и y_i=1).

Таким образом, получили задачу булева квадратичного программирования. В случае, если ее решение вызывает трудности, задачу можно ослабить до задачи обычного квадратичного программирования. При этом устовие 1) заменяется на условие 1'): все переменные больше или равны 0 и меньше или равны 1. Для такой задачи известны эффективные алгоритмы, но полученное решение может содержать дробные значения переменных и не соответствовать набору структур. В этом случае к алгоритму добавляется следующий эвристический этап.

Упорядочим все полученные значения переменных по группам равнозначных переменных в порядке убывания их значений. Перебираем варианты установки разделителя, т.е. разбиения переменных на две части, где значения в левой части строго больше значений в правой части. Для каждого варианта все переменные в левой части положим равными 1, а в правой – равными 0, вычислим функцию *f* (а предпочтительнее –

функционал *F*). Затем выбираем лучший вариант и объявляем его решением задачи.

Литература:

1. Bergeron A., Mixtacki J., Stoye J. A unifying view of genome rearrangements. *WABI 2006. LNCS* (*LNBI*), vol. 4175, pp. 163–173, 2006.
2. Fertin G., Labarre A., Rusu I., Tannier E., Vialette S. *Combinatorics of Genome Rearrangements*. Cambridge: MIT Press, 2009.
3. *Models and Algorithms for Genome Evolution*. By Cedric Chauve, Nadia El-Mabrouk, Eric Tannier (editors), Computational Biology series. Springer-Verlag, London, 2013.
4. Горбунов К.Ю., Любецкий В.А. Реконструкция эволюции генов вдоль дерева видов, *Молекулярная биология*, том 43, №5, с. 946–958, 2009.
5. Lyubetsky V.A., Rubanov L.I., Rusin L.Y. & Gorbunov K.Yu. Cubic time algorithms of amalgamating gene trees and building evolutionary scenarios. *Biology Direct*, vol. 7, no. 1, pp. 1–20, 2012.