ВЫРАВНИВАНИЕ ПОСЛЕДОВАТЕЛЬНОСТЕЙ НА ОСНОВЕ ДЕРЕВА[1]

В.А. Любецкий[2], Л.И. Рубанов, А.В. Селиверстов

Институт проблем передачи информации им. А.А. Харкевича РАН
127994 Москва, Большой Каретный пер., 19
[2] lyubetsk@iitp.ru

Предложены алгоритм и компьютерная программа для построения выравнивания данного набора последовательностей. В приводимых примерах эти последовательности интерпретируются как участки, взятые из геномов различных организмов, перед одним и тем же геном. Выравнивание строится с помощью оригинального быстрого алгоритма, использующего бинарное дерево, которое указывает на степень родства любых двух данных последовательностей (фактически – на степень родства соответствующих им организмов). Если бинарное дерево неизвестно, то алгоритм получает его путем разрешения всех небинарных вершин в данном небинарном дереве, которое обычно уже известно в отличие от бинарного. Таким образом, алгоритм включает быстрый способ порождения без повторений всех бинарных деревьев, топологически совместимых с данным небинарным деревом, и, перебирая их все, он выдает выравнивание наилучшего качества. Алгоритм тестировался на искусственных и биологических данных.

Введение. Постановка задачи

Одна из центральных задач биоинформатики состоит в построении выравнивания данного набора последовательностей, каждая из которых представляет собой, например, участок перед одним и тем же геном, но из разных геномов (организмов). Обычно такое выравнивание называют «множественным», если последовательностей больше двух, и «парным», если их две. Нами предложен оригинальный быстрый алгоритм, который строит выравнивание с использованием бинарного дерева родства соответствующих организмов («дерева видов»). Такое дерево получается путем разрешения политомии (небинарности многих вершин) в некотором часто заранее известном небинарном дереве, для чего алгоритм включает оригинальную процедуру порождения без повторений всех бинарных деревьев, топологически совместных с данным небинарным деревом. Алгоритм строит выравнивание по текущему бинарному дереву, перебирая их все, и выдает то выравнивание, которое имеет наилучшее качество. Перейдем к более формальной постановке задачи.

Пусть дан набор (конечных) последовательностей произвольной длины в фиксированном алфавите, например, в 4-х буквенном: A, C, T, G. Задача состоит в том, чтобы расположить эти последовательности друг под другом, вставляя в них произвольное число раз знак пробела «=» так, чтобы все полученные в результате последовательности имели уже одинаковую длину и при этом в них образовалось как можно больше «консервативных» (т.е. состоящих из как можно меньшего числа букв) столбцов (в итоговой матрице). Эта матрица и называется выравниванием. «Больше консервативных столбцов» означает максимизацию некоторого фиксированного функционала от выравнивания, который должен быть указан. Очевидно, прямой перебор всех возможных вариантов слишком велик, даже если ограничить общее число пробелов (иными словами, число столбцов в этой матрице).

Метод решения

Пусть кроме набора последовательностей дано бинарное дерево, листьям которого приписаны номера последовательностей.

Родственные («близкие») по дереву последовательности должны расположиться в выравнивании особенно консервативно между собой, а менее родственные («далекие») последовательности могут расположиться и относительно менее консервативно. Конечно, здесь снова речь идет о функционале, зависящем уже от выравнивания и от дерева, который должен быть заранее указан или подразумеваться. «Расстояние по дереву» в простейшем случае – просто число ребер между двумя листьями. Если пользователь имеет такое бинарное дерево, то он задает его нашей программе. Однако в ней предусмотрено и получение этого дерева следующим образом. Обычно близость в наборе последовательностей в некоторой мере известна и может быть задана в форме небинарного дерева (что отражает недостаток знания): например, последовательности с номерами от 1 до 5 могут быть (по доступным исследователю данным) равно близки между собой, т.е. иметь общего отца. Но по смыслу алгоритма у каждого отца должно быть ровно два сына. Тогда все такие небинарные вершины (политомии) разрешаются: точнее, путем добавления в исходное небинарное дерево промежуточных вершин строятся все варианты бинарных деревьев, согласованных с исходно данным небинарным деревом относительно предиката «*x* – потомок *y*». При этом цепочка бинарных деревьев должна порождаться без повторений, так как число вариантов и без того велико: каждая политомическая вершина с *n* сыновьями допускает

$$R(n) = \frac{(2n-3)!}{2^{n-2}\cdot(n-2)!} \qquad (1)$$

в топологически различных вариантов разрешения. При нескольких политомиях разрешение каждой выполняется независимо; таким образом, общее число деревьев есть произведение чисел вариантов (1) для всех небинарных вершин.

Для текущего бинарного дерева наш алгоритм строит соответствующее ему множественное выравнивание следующим образом. В вершинах дерева рассматриваются последовательности распределений частот букв в поддереве с корнем в данной вершине: в листьях последовательности распределений задаются исходными данными тривиально, а именно, в каждой позиции берется распределение с одной единицей и остальными нулями. При переходе от двух сыновей к их отцу две такие последовательности распределений выравниваются с помощью процедуры динамического программирования, в которой величина приза a_j за совпадение частот x_i и y_i соответствующей буквы не фиксирована заранее, а вычисляется своя по каждой позиции j с учетом расстояния по дереву между двумя сыновьями. Алгоритм использует скалярный квадрат разности ненулевых распределений

$$a_j = 1 - \sum_{i=1}^{4} w_i (x_i - y_i)^2 , \qquad (2)$$

где w_i – веса разных букв, в сумме составляющие 1. (В программе предусмотрены и другие варианты вместо (2), в частности, метрики L_1 и L_2 в пространствах распределений.) Штраф a'_j за вставляемые пробелы (т.е. за выравнивание нулевого распределения на ненулевое) задается убывающей функцией от длины участка пробелов: если участок длиннее, то штраф за него меньше. В этом парном выравнивании нулевому распределению запрещено выравниться на себя.

Теперь последовательность распределений для отца в каждой ее позиции принимается равной полусумме полученных после выравнивания распределений в сыновьях для этой позиции выравнивания:

$$\mathbf{Z} = \frac{1}{2}(\mathbf{X} + \mathbf{Y}) \qquad (3)$$

Когда для корня дерева таким образом построена последовательность, то выполняется обратный ход алгоритма, при котором пробелы, вставленные в нее, опускаются вниз дерева вплоть до его листьев. Последовательности с пробелами, полученные в листьях таким образом, – искомое множественное выравнивание и результат работы алгоритма при данном бинарном дереве.

Функционал, о котором говорилось выше, равен сумме $\sum_j a_j + a'_j$ призов или штрафов по всем позициям j парного выравнивания последовательностей распределений в двух прикорневых вершинах.

Такая процедура характеризуется сложностью, линейной от числа листьев. В частности, для случая сбалансированного дерева с $m = 2^k$ листьями она состоит из m-1 попарных выравниваний последовательностей, последовательно от листьев к корню. Само парное выравнивание выполняется очень быстро.

Для сравнения множественных выравниваний, полученных при различных вариантах бинарного разрешения политомического дерева, используется «индекс консервативности», задаваемый функционалом

$$I = \left(N_a + N_s\right)b + \sum_{i=1}^{N_s}\left(b+s\right)\left(l_i - 1\right) + N_b c\,, \quad (4)$$

где N_a – число одиночных абсолютно консервативных (т.е. с ровно одной буквой в столбце) позиций в выравнивании, N_s – число непрерывных абсолютно консервативных участков длиной 2 и более нуклеотидов (l_i обозначает длину i-го такого участка), N_b – число «почти» абсолютно консервативных (с только одной отличающейся буквой в столбце) позиций выравнивания; b, c и s – величины призов, являющиеся параметрами программы.

По окончании перебора всех вариантов разрешения программа выдает выравнивание, на котором указанный функционал достигает наибольшего значения, а при равенстве значений – выравнивание с наибольшим значением индекса консервативности (4).

Результаты

Построен эффективный алгоритм и соответствующая программа (общедоступная – сейчас по запросу от авторов, к началу конференции она будет выложена на сайте http://lab6.iitp.ru), которая решают поставленную задачу. Например, с ее помощью построение множественного выравнивания 16 последовательностей с длинами 120-223 позиций занимает менее секунды на ПК Pentium-4, 3 ГГц.

Из обширного тестирования, которое мы провели для нашей программы, приведем выравнивания, которые позволяют предложить оригинальные гипотезы о регуляции генов. Последние здесь не обсуждаются, они представлены в журнал «Молекулярная биология» и также были доложены авторами на международной конференции BGRS'2008.

На рис. 1 показано полученное программой множественное выравнивание участков: перед геном *ycf24* и псевдогеном у *Th. annulata*; на рис. 2 – перед генами у *T. gondii* и на рис. 3 – перед геном *rpoB*. Здесь и на рисунках курсивом приводятся имена организмов (видов). В выравниваниях 1-3 последовательности расположены непосредственно перед началом гена, и гипотеза состоит в том, что они задают сайты связывания регуляторного фактора с РНК. На рис. 4 показано полученное программой множественное выравнивание участков перед геном *rps20*. Здесь гипотеза состоит в том, что имеется регуляция, основанная на конкуренции при считывании генов с разных цепей ДНК.

Рис. 1. Выравнивание перед геном *ycf24*. Абсолютно консервативные позиции подчеркнуты, консервативные – выделены полужирным.

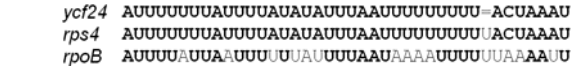

Рис. 2. Выравнивание перед генами у *T. gondii*. Обозначения те же, что на рис. 1.

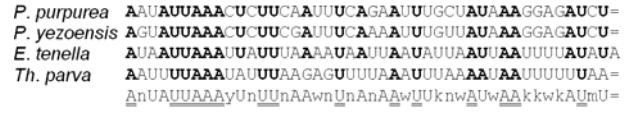

Рис. 3. Выравнивание перед геном *rpoB*. Обозначения те же, что на рис. 1.

Cyanidioschyzon merolae	ACTC**TTGCTT**TTTGCCATCTGCT=ATTT**TATCTT**TATGTAGACT	-33
Cyanidium caldarium	AAAT**TTGTTT**ATTTTACTTTAAT=A**TG**A**TACAGT**AATTTATAAC	-32
Porphyra purpurea	GCTA**TTGCCT**ATTCTTTTTTTTAA**TG**T**TATAAT**ACGGCGCATA	-78
Porphyra yezoensis	ACTA**TTGCCT**ATTGTTTTTCTTTAA**TG**T**TATAAT**ACGCCGCATA	-78
Gracilaria tenuistipitata	GTTC**TTGTCT**ATTTTAATGTATTAA**TG**A**TATAAT**CCAATTAGAT	-63
Guillardia theta	TTAA**TTTATT**CCATTATTTCTTATA**TG**T**TATAAT**CTTTTATTAC	-59
Rhodomonas salina	TCTT**CTTATT**C=ATAATTTGTTCTA**TG**T**TATAAT**CACTAATCGT	-55

Рис. 4. Выравнивание перед геном *rps20*. Полужирным показаны консервативные участки. Двойным подчеркиванием и серым тоном показаны места связывания факторов.

Заключение

Предложенный и реализованный алгоритм множественного выравнивания с использованием известного или динамически строящегося бинарного дерева имеет следующие достоинства:

- высокую скорость работы, обусловленную линейной зависимостью вычислительной сложности от числа данных последовательностей, что позволяет обрабатывать большие наборы длинных последовательностей; возможность распараллеливания алгоритма, что еще более увеличивает скорость вычислений;
- возможность проверки всех вариантов разрешения политомии в случае, если бинарное дерево неизвестно пользователю;
- естественный способ работы с частично известными последовательностями, когда приходится рассматривать последовательности в расширенном алфавите (например, в популярной номенклатуре IUPAC-IUB);
- возможность учета различных весов нуклеотидов и их оперативного варьирования;
- возможность задавать программе любой разумной сложности функционал, подлежащий минимизации.

Авторы благодарят Л. Русина за ценное обсуждение этой работы и большую помощь.