Информационные процессы, Том 2, № 1, 2002, стр. 100-105.

ТЕОРИЯ И МЕТОДЫ ОБРАБОТКИ ИНФОРМАЦИИ

Модифицированный алгоритм поиска альтернативных вторичных структур РНК и результаты счета

Л.А.Леонтьев, Е.В.Любецкая, В.А.Любецкий

Институт проблем передачи информации РАН
101447, Москва, Большой Каретный переулок, 19, Россия
E-mail Lin@iitp.ru Тел. (7095) 2998354, 4916780. Факс (7095) 2090579

Поступила в редколлегию 21.05.2002

Аннотация—Представлен алгоритм и программа, решающий новую задачу — поиск альтернативных вторичных структур РНК и их особенностей. Программа показала хорошие результаты при нахождении экспериментально найденных альтернативных вторичных структур на 60 фрагментах мРНК.

1. ВВЕДЕНИЕ

Задача поиска альтернативных вторичных структур РНК является одной из важнейших в проблематике компьютерной геномики. В последние годы выяснилась большая роль вторичных структур на уровне мРНК в регуляции процессов биосинтеза в клетке (процессов аттенуации). В моделях такой регуляции существенная роль отводится как раз неспаренным основаниям, в частности, нуклеотидам петель шпилек. Для поиска координат начала и конца петли шпильки нами модифицирован фундаментальный алгоритм (также предложенный и развитый в ИППИ РАН) из [1] (в частности, в [1] приведена биологическая аргументация в пользу этого алгоритма).

Модификация алгоритма состояла в следующем.

А) Использовались различные мощности пар: различным комплиментарным парам нуклеотидов сопоставляются различные мощности (паре $\langle$C,G$\rangle$ — мощность 2, паре $\langle$A,T$\rangle$ — 1 и паре $\langle$G,T$\rangle$ — 0). Это приблизительно отображает биологическую идею свободной энергии шпильки.

Б) В результате работы алгоритма строился массив шпилек и упорядочивался по убыванию параметра B — начала петли. Далее выделялись такие пары нуклеотидов (т.е. параметров (B,C) найденных шпилек), которые являлись координатами петель более чем p шпилек (для последовательностей длины 350-600 эмпирически подобрано p=9).

Алгоритм был компьютерно реализован программой на языке Object Pascal в среде Delphi 5. Модифицированный таким образом алгоритм показал высокую эффективность на природных последовательностях мРНК. На данный момент не существует четкого доказательства правильности работы алгоритма, но есть предположения, обосновывающие его правильность. Например, в пользу этого алгоритма можно привести такое соображение: алгоритм при построении шпилек стремится к наиболее мощному отрезку прибавить другие наиболее мощные отрезки, т.е. максимизирует шпильку по отрезкам, начиная от петли. Во многих случаях биологически значимые шпильки также состоят из локально максимальных по мощности отрезков и также начинают образовываться от петли.

2. НАХОЖДЕНИЕ КООРДИНАТ ПЕТЕЛЬ АЛЬТЕРНАТИВНЫХ ВТОРИЧНЫХ СТРУКТУР

Особенности подготовки исходных данных и результаты нахождения координат петель шпилек таковы.

1. Некоторые типы шпилек (например, шпилька-терминатор) в большинстве организмов находится с высокой точностью. Координаты петель некоторых шпилек не находятся. Подробная статистика нахождения координат петель шпилек приведена ниже, в таблице 1.

2. В результате просчетов 60 последовательностей выяснилось, что количество координат петель шпилек, повторяющихся более 9 раз, мало относительно общего количества координат петель шпилек, вычисляемых алгоритмом. Общее количество шпилек для разных последовательностей было 300-500; количество таких координат было от 10 до 25 для всех рассмотренных нами последовательностей.

3. Выяснилось, что координаты петель шпилек, повторяющихся более 9 раз, почти не зависят от изменения последовательности. Для проверки этого положения было сделано следующее:

a. Исходные 37 последовательностей многократно "нарезались" по-другому, т.е. из мРНК исследуемых организмов выбирались несколько другие фрагменты. А именно, увеличивалась и уменьшалась длина кусков мРНК, поступающих на вход алгоритму, а также из мРНК брались смежные фрагменты, включающие те же биологические ответы. "Нарезание" осуществлялось таким образом, чтобы эти биологические ответы располагались то в начале, то в середине, то в конце фрагмента; т.е. фрагменты "сдвигались" на +100-300 нуклеотидов. Таким образом, были получены дополнительные 47 последовательностей с теми же биологическими ответами. При этом в 75% случаев были найдены те же самые 10-20 координат петель шпилек; в 5 последовательностях были найдены улучшенные ответы; в 6 — ухудшенные; в 1 последовательности произошли равнозначные изменения. Во всех последовательностях, где произошли изменения, изменялись координаты петель 1-2 шпилек из их общего числа 3-5.

b. Последовательности, содержащие структуры типа T-box (исходное количество 23 шт.) также были "нарезаны" аналогичными способами, получились 23 новых последовательности. Результат не изменился в 82% случаев (т.е. в 19 последовательностях).

4. Чтобы не порождалось избыточное количество не биологически значимых шпилек алгоритм содержит ограничения, которые в некоторых случаях делают теоретически невозможным нахождение биологически значимых шпилек. Перечислим эти ограничения.

a. Минимальная длина отрезка в алгоритме — 3 нуклеотида; в биологически значимых шпильках встречаются отрезки длины 2.

b. Некоторые биологически значимые шпильки содержат боковые подшпильки, алгоритм же не умеет искать такие структуры.

c. Биологически значимые шпильки иногда содержат комплиментарную пару нуклеотидов (G,T) (или (G,U) в разных видах РНК) на концах и началах отрезков. Алгоритм же не считает пару (G,T) комплиментарной, если она находится в конце или в начале отрезка.

5. При точном нахождении координат петли биологически значимой шпильки в подавляющем большинстве случаев правильно находится первый ее отрезок; в среднем правильно находится два отрезка шпильки; а в некоторых случаях находится вся шпилька (все ее 1-5 отрезков). Эти данные были получены с помощью *метода ко́ров(cores)*; суть метода заключается в следующем: берется группа шпилек с одинаковыми координатами петель (координаты петель удовлетворяют п. 1Б) и, начиная от петли, выделяются пары нуклеотидов, повторяющиеся в более чем половине шпилек группы. Эти нуклеотиды и составляют кор; если же на некотором шаге такой пары не нашлось, то кор заканчивается. Метод является продолжением идеи п. 1Б для группы шпилек.

3. НАХОЖДЕНИЕ ШПИЛЕК-ТЕРМИНАТОРОВ АЛЬТЕРНАТИВНЫХ ВТОРИЧНЫХ СТРУКТУР

Алгоритм применялся для поиска шпилек, обладающих ярко выраженными свойствами, например, шпилек-терминаторов.

Терминатором считается шпилька небольшой длины (15-23 нуклеотидов), состоящая из одного непрерывного отрезка (из 6-9 комплиментарных пар нуклеотидов) или имеющая одно небольшое выпячивание (1-2 нуклеотидов). Также известно, что либо с середины правого основания, либо после окончания правого основания терминатора находится “поле t” (т.е. более 5 нуклеотидов t подряд). Мы предположили, что мощный отрезок, составляющий терминатор, должен повторяться во многих локально-оптимальных шпильках, находимых алгоритмом. И поэтому перебирали первые отрезки коров при всех координатах петель шпилек, повторяющихся более 9 раз, проверяя наличие поля t у каждого кора. Все полученные терминаторы ранжировались по мощности (т.е. лучшим терминатором являлся самый мощный первый отрезок кора с полем t). Результаты работы этого алгоритма для трехшпилечных аттенуаторных структур иллюстрирует табл. 2, т.к. для них использовалась эта программа поиска терминатора и, видно, что точность нахождения терминатора в большинстве случаев оценивается “2” (см. комментарий к табл. 2). Для остальных структур применялся более старый алгоритм, в котором терминатором назывался самый мощный первый отрезок кора (без учета поля t).

Результаты работы первой версии алгоритма поиска шпильки-терминатора таковы: найдено 14 терминаторов из 43 (т.е. 33%); при этом в 5 последовательностях в биологических ответах шпилька-терминатор отсутствует.

Более подробно результаты нахождения терминаторов приведены в таблице 1.

Таблица 1.

	Название гена	Количество биологических шпилек	Найденные координаты петли с учетом точности Δ	Терминатор
1	Bs_pyrB	2	T --- точно; A --- точно;	Не найден
2	Bs_pyrP	2	T --- 8,10*; A --- 4,1;	Не найден
3	Bs_pyrR	2	T --- точно; A --- точно	Не найден
4	Be_serS	5	T --- точно, Sh,2,3,A --- точно	Найден точно
5	Be_tyrS	5	T --- точно; Sh --- 2,5; 2 --- не найдена; 3,A --- точно	Найден точно
6	Bq_serS	5	T --- точно; Sh --- не найден; 2 --- 2,3; 3 --- точно; A --- 0,2	Найден точно
7	Bq_tyrS1	5	T --- точно; Sh --- точно; 2,3 --- не найден; A --- точно	Не найден
8	Bq_tyrS2	5	T --- точно; Sh --- точно; 2 --- 0,3; 3,A --- точно	Найден без 3 пар
9	Bs_serS	5	T --- 1,1; Sh --- 3,5; 2,3,A --- точно	Без одной пары
10	Bs_tyrS	6	T --- точно; Sh --- не найден; 2 --- 2,2; 3 --- 3,3; 4 --- точно; A --- 0,1	Найден точно
11	Bs_tyrZ	6	T не найден; Sh --- точно; 2 --- не найден; 3 --- 1,0; 4 --- точно; A --- не найден	Не найден
12	Ca_tyrZ	3	T не найден; Sh --- 1,4; A --- 0,1	Не найден
13	Ca_yurG	5	T --- 0,1; Sh --- 4,1; 2,3 --- не найден A --- 2,0	Не найден
14	DF_serS	5	T не найден; Sh,2,3 --- точно; A --- 4,1	Не найден
15	DF_tyrZ	3	T --- 1,0; Sh --- 4,1; A --- точно	Без одной пары
16	DHA_tyrZ	3	T --- 0,1; Sh --- 0, 2; A --- 0,1	Не найден
17	EF_serS	3	T --- 3,2; Sh --- не найден; A --- 1,4	Не найден

18	EF_tyrS	5	T точно; Sh --- не найден; 2 --- точно; 3 --- 6,2; A --- точно	Не найден
19	HD_serS	5	T точно; Sh --- точно; 2 --- 0,6; 3 --- 5,3; A --- 0,1	Не найден
20	HD_tyrZ	6	T точно; Sh --- 5,3; 2 --- не найден; 3 --- 4,1; 4 --- 1,1; A --- точно	Не найден
21	LLX_serS	3	T --- 1,2; Sh --- 1,0; A --- 0,1	Не найден
22	LO_serS	3	T --- 0,1; Sh --- 3,1; A --- не найден	Найдена часть
23	LO_tyrS	5	T точно; Sh --- 4,1; 2,3 --- точно; A --- 0,1	Найден точно
24	PN_serS	3	T --- 1,2; Sh --- 2,6; A --- точно	Не найден
25	Sa_serS	5	T точно; Sh --- 5,3; 2 --- 0,1; 3,A --- не найден	Найден точно
26	SEQ_serS	3	T --- 1,1; Sh --- не найден; A --- точно	Не найден
27	Bs_thrS	5	T --- 1,0; Sh --- не найден; 2,3,A --- точно	Не найден
28	LL_his	5	T --- 5,4; Sh,2,3 --- точно; A не найден	Не найден
29	LL_trp	5	T точно; Sh --- точно; 2 --- не найден; 3 --- 0,2; A --- точно	Найден точно
30	BS_purE	3	T не найден; Sh --- точно; A --- не найден	Не найден
31	BS_purM	1	1 --- 2,1	Без 2 пар
32	Bs_tyrS	5	T точно; Sh,2 --- не найден; 3,A - точно	Найден точно
33	EC_pyrB	2	T --- 0,1; A --- не найден	Не найден
34	EC_ilvG	3	T точно; Sh,A --- точно	Не найден
35	EC_rpsJ	6	1 --- не найден; 2 --- 1,1; 3,4 не найден; 5 --- точно; 6 --- 3,2	Нет термин-а
36	Hi_rpsJ	5	1 --- 3,1; 2 --- 1,1; 3,4 --- точно; 5 --- 6,2	Нет термин-а
37	BS_ilv_leu	5	T точно; Sh --- точно; 2 --- 2,6; 3 --- не найден; A --- точно	Не найден
38	BS_yczA	5	T точно; Sh --- не найден; 2 --- 1,1; 3 --- точно; A --- не найден	Найден точно
39	BS_trpE	2	1 --- точно; 2 --- 0,1	Нет термин-а
40	Sa_ileS	4	T точно; Sh,2,A --- не найден	Не найден
41	Ec_rplK	2	1,2 --- точно	Нет термин-а
42	Hi_rplK	2	1,2 --- точно	Нет термин-а
43	Bs_valS	3	T --- 0,1; Sh --- 4,3; A --- не найден	Не найден

Обозначения и примечание к таблице 1:

1. Sh — specifier hairpin, A — антитерминатор, T — терминатор.

2. Выражение “точно” (в 4-ом столбце) означает, что алгоритм нашел координаты петли шпильки в точном соответствии с биологическим ответом. “Не найден” означает, что те же координаты различаются на более, чем 10 позиций по B или по C.

***** Запись означает, что для данной шпильки расхождение между значениями (B,C) в биологическом ответе и ответе алгоритма составляет 8 нуклеотидов по B и 10 нуклеотидов по C. Выражение “без двух пар” (и т.п.) (в 5-ом столбце) означает, что на концах алгоритмически найденного терминатора не хватает двух пар нуклеотидов из биологически правильного ответа.

4. АЛГОРИТМ ПОИСКА ТРЕХШПИЛЕЧНЫХ АТТЕНУАТОРНЫХ СТРУКТУР

Алгоритм обосновывается двумя фактами (факты подтвердились для всех 17 последовательностей и имеют объяснение; строго доказаны они не были).

Факт 1. Для трехшпилечных структур с точностью близкой к 100% находится шпилька-терминатор (см. подробную статистику в п.3).

Факт 2. Упорядочим координаты петель, найденные в алгоритме из п.1 по убыванию параметра B. Пусть некоторая петля является петлей терминатора (найденная, например, с помощью алгоритма из п.3). Тогда следующая петля является петлей второй шпильки (с некоторой точностью), а петля после следующей является петлей первой шпильки.
Факт 3. По определению трехшпилечной структуры шпильки, входящие в нее (это 1-ая, 2-ая и терминатор) образуют два перекрытия — (2-ая,T) и (1-ая,2-ая).

Описание алгоритма: Шаг 1. Найти шпильку терминатор с помощью алгоритма из п.3. Тогда знаем координаты петель второй и первой шпилек.
Шаг 2. Возьмем координаты петли второй шпильки. Возьмем группу шпилек с данным (B,C) и будем перебирать эти шпильки, включая подшпильки, и проверять условие перекрытия ($|C'D' \cap AD| >= 5$; A>B′, C>D′). Все полученные шпильки — кандидаты на вторую шпильку. По некоторому критерию выберем одну из них. Сейчас это простейший критерий — самая мощная шпилька. В результате получаем более длинную шпильку, чем биологически значимая.
Шаг 3. Возьмем группу первых шпилек (т.е. группу шпилек с соответствующим (B,C)). Аналогично шагу 2 выберем среди них и их подшпилек шпильки, образующие перекрытие с выбранной второй шпилькой. Из этих полученных шпилек выберем одну первую шпильку по некоторому критерию (сейчас это критерий наибольшей мощности; в результате шпилька оказывается длиннее биологически значимой).
Шаг 4. Если для терминатора не оказалось второй и/или первой шпильки, выведем “возможную шпильку” — это кор (см. алгоритм из п.2) при соответствующем (B,C). Очевидно, кор окажется короче, чем биологически значимые шпильки.

Перспективы: Улучшить критерий выбора первой и второй шпилек из соответствующих групп. Возможны следующие критерии: “лучшей” является самая мощная комбинация первой и второй шпилек или комбинация с наилучшей энергией и т.д. Возможно, следует организовать выбор шпилек из группы возможных шпилек следующим образом: выбираются сочетания первой и второй шпильки с наилучшей энергией, с наибольшей суммарной мощностью, наибольшей областью перекрытия и другим различным критериям.

Таблица 2. Результаты поиска трехшпилечных структур.

Aa_aroma_pheA	**2 1 0**	0 0 0	-	-	-
Ec_aroma_pheA	0 0 0	0 0 0	**1 2 1**	-	-
Ec_aroma_pheS	**2 2 1**	0 0 0	-	-	-
Ec_aroma_trpE	**2 1 0**	0 0 0	-	-	-
Hi_aroma_pheA	**2 1 0**	-	-	-	-
Hi_aroma_pheST	0 0 0	0 0 0	0 0 0	**2 1 0**	0 0 0
Hi_aroma_trpBA	0 0 0	-	-	-	-
Hi_aroma_trpE	0 0 0	0 0 0	-	-	-
St_aroma_pheA	**2 1 0**	0 0 0	0 0 0	-	-
St_aroma_pheS	**2 1 2**	0 0 0	-	-	-
St_aroma_trpE	**2 1 0**	-	-	-	-
Vc_aroma_pheA	**2 1 1**	0 0 0	0 0 0	0 0 0	-
Vc_aroma_trpE	**2 1 0**	0 0 0	0 0 0	0 0 0	-
Yp_aroma_pheA1	**2 2 1**	-	-	-	-
Yp_aroma_pheA2	**2 1 0**	-	-	-	-
Yp_aroma_PheS	0 0 0	**2 1 1**	0 0 0	0 0 0	-
Yp_aroma_trpE	**2 1 0**	-	-	-	-

Примечания к таблице 2:

1. В первом столбце указаны названия организмов.

2. Курсивом выделены названия организмов, у которых шпилька-терминатор разрывна; напомним, что такие терминаторы алгоритм почти не находит.

3. Столбцы 2-6 указывают количество структур, которые алгоритм счел за биологически значимые; во втором столбце — лучшая структура, в пятом — худшая (с точки зрения алгоритма).

4. Прочерк "-" означает, что таких структур найдено не было, т.е., например, для организма Aa_aroma_pheA было найдено две структуры, организма Ec_aroma_pheA — три.

5. Каждой структуре выставляется "оценка". Оценкой является вектор, где первая координата — оценка шпильки-терминатора, вторая координата — оценка второй шпильки и третья координата — оценка первой шпильки.

6. Оценка шпильке выставляется по трехбальной шкале:
"2" — "шпилька найдена условно точно", т.е. delta(B,C)<=5, delta(A,D)<=7;
"1" — "шпилька найдена приблизительно", т.е. delta(B,C)<=5, delta(A,D)>=7, шпилька требует уточнения по (A,D);
"0" — "шпилька не найдена", т.е. delta(B,C)>=5, delta(A,D)>=7.

7. Очевидно, что если некоторая структура оценивается двойками и единицами, то остальные структуры будут оцениваться нулями, т.к. лежат в других частях последовательности.

5. БИОЛОГИЧЕСКОЕ ОБОСНОВАНИЕ РЕЗУЛЬТАТОВ

Этот алгоритм является одним из первых решающих задачу поиска альтернативных вторичных структур. Поэтому у авторов не было возможности сравнить результаты его работы с каким-либо другим алгоритмом, и алгоритм тестировался только на вторичных альтернативных структурах, уже найденных экспериментально, а также на случайных последовательностях. Алгоритм годен для нахождения альтернативных вторичных структур и в достаточно общей ситуации. Сейчас програмно реализована версия алгоритма, для поиска трехшпилечных альтернативных структур (здесь проведен обширный счет) и для поиска структур типа T-box.

Перечислим типы структур, на которых он тестировался:
1. Аттенюаторы транскрипции генов pheA, trp биосинтеза ароматических аминокислот гамма-протобактерий (Escherichia coli, Salmonella typhi, Yersinia pestis, Vibrio cholerae, Haemophilus influenzae, Actinobacillus actinomycetemcomitans); гена, кодирующего фенилаланин-тРНК синтетазу pheS.

Таких структур было 17, результаты нахождения вторичных альтернативных структур близки к 100%, более подробно см. таблицу 2.
2. Генов биосинтеза пиримидина (pyr) в Bacillus subtilis.

Таких структур было 3; для них находились координаты петель шпилек, осуществляющих регуляцию. Более подробно см. табл. 1, п.1-3.
3. T-box terminator-antiterminator structures, которые принимают участие в регуляции генов биосинтеза аминокислот, а также генов, кодирующих аминоацил-тРНК синтетазы Gram+ бактерий.

Для таких структур находились координаты петель шпилек, осуществляющих регуляцию. Более подробно см. табл. 1, п.4-43.

6. БЛАГОДАРНОСТИ

Авторы выражают глубокую благодарность М.С. Гельфанду и А.А. Миронову за помощь и объяснение биологического содержания задачи.

СПИСОК ЛИТЕРАТУРЫ

1. Верещагин Н.К., Любецкий В.А. Алгоритм определения вторичной структуры РНК. *Труды научно-исследовательского семинара логического центра ИФ РАН*, 2000, выпуск 14, Москва, Издательство РАН, сс. 99–109.