МОЛЕКУЛЯРНАЯ БИОЛОГИЯ, 2003, том 37, № 5, с. 850–860

БИОИНФОРМАТИКА

УДК 577.1

ПОИСК КОНСЕРВАТИВНЫХ ВТОРИЧНЫХ СТРУКТУР РНК

© 2003 г. К. Ю. Горбунов[1]*, А. А. Миронов[2], В. А. Любецкий[1]

[1]*Институт проблем передачи информации Российской академии наук, Москва, 101447*

[2]*Федеральное государственное унитарное предприятие "ГосНИИгенетика", Москва, 113545*

Поступила в редакцию 03.04.2003 г.

Предложен новый алгоритм для поиска консервативных вторичных структур РНК в заданном наборе последовательностей. Алгоритм основан на выравнивании цепочек плеч потенциальных спиралей. Он позволяет учитывать консервацию нуклеотидных последовательностей в окрестностях спиральных участков. Алгоритм применим для поиска новых структурных РНК и регуляторных элементов. Его эффективность позволяет использовать алгоритм для полногеномного анализа. Приведены результаты разнообразного тестирования этого алгоритма.

Ключевые слова: вторичная структура РНК, алгоритм динамического программирования, структура тРНК, структура RFN, регуляция транскрипции, регуляция трансляции.

Роль вторичной структуры РНК хорошо известна. Она важна для: структурных РНК, таких как рибосомные или малые ядерные РНК; РНК ряда вирусных геномов; рибозимов; регуляции экспрессии генов. Большинство известных систем регуляции в бактериях на уровне матричных РНК работают по принципу аттенюации, когда реализуется одна из двух альтернативных структур. "Запрещающая" на уровне транскрипции содержит терминатор транскрипции, а на уровне трансляции блокирует сайт связывания рибосом. Альтернативная ("разрешающая") структура соответственно не позволяет образоваться терминатору транскрипции или блокировать сайт связывания рибосом. Часто "разрешающая" структура стабилизируется различными молекулами: белком в случае регуляции экспрессии рибосомных белков, низкомолекулярными лигандами в случае регуляции синтеза рибофлавина и тиамина [1, 2], другими РНК в случае Т-бокса [3]. В связи с биологической важностью вторичных структур РНК проблема ее предсказания по *одной* исходной последовательности или по *набору* исходных последовательностей является одной из классических задач биоинформатики.

Для анализа и предсказания вторичных структур РНК применяли различные подходы. Наиболее популярен подход, основанный на оптимизации какой-либо функции качества вторичной структуры. В 1966 г. для поиска наиболее длинных шпилек в одной последовательности Туманян предложил метод динамического программирования [5]. В 1980 г. Нуссинов и Якобсон предложили метод динамического программирования для поиска вторичной структуры с максимальным количеством спаренных оснований также в одной последовательности [6]. Позже Зукер предложил метод динамического программирования для поиска структур с минимальной свободной энергией (также в одной последовательности), который позволяет использовать дополнительные ограничения, получаемые из экспериментальных данных [7–9]. В настоящее время стандартным алгоритмом предсказания вторичных структур РНК является упомянутый алгоритм Зукера (http://www.bioinfo.rpi.edu/~zukerm/rna/). К сожалению, эти подходы не лишены недостатков. В частности, для транспортных РНК только примерно для 80% последовательностей удается предсказать структуру "клеверного листа". Другим существенным недостатком подхода, основанного на минимизации свободной энергии, является его принципиальная зависимость от исходных энергетических параметров. Кроме того, в ряде случаев, например при анализе аттенюаторов, наибольший интерес представляет как раз "разрешающая" структура, которая по механизму действия аттенюатора заведомо не является оптимальной. Поэтому авторам кажется, что вообще алгоритмы, основанные на оптимизации (по крайней мере, с доступными исследователям функционалами), не будут успешными.

Другим направлением в изучении и предсказании вторичных структур РНК является анализ динамики формирования вторичной структуры в процессе синтеза РНК. В работах [10–12] предложена кинетическая модель сворачивания РНК, которая реализована в форме монте-карловской процедуры. Такой подход улучшил понимание процессов формирования структуры РНК, одна-

* Эл. почта: gorbunov@ittp.ru

ко его предсказательная сила также не достаточно велика.

Наиболее успешен подход, основанный на поиске консервативных структур в последовательностях изофункциональных РНК. Таким образом, предсказаны структуры тРНК, рРНК, мяРНК. Однако при этом анализе, во-первых, использовали дополнительные сведения [13], получаемые из экспериментов, и значительную часть работы делали вручную, а во-вторых, эти последовательности заранее имели естественное выравнивание, что значительно упрощало задачу. Более поздние работы основывались на заранее построенных выравниваниях и использовали соображения о коррелированных заменах, сохраняющих потенциальные спирали (см., например, [14, 15]). Ряд работ основан на статистическом анализе взаимно комплементарных сегментов последовательностей [16, 17]. В последнее для поиска консервативных вторичных структур РНК время стали развиваться методы, основанные на генетических алгоритмах [18, 19].

Сейчас доступны последовательности многих геномов. Их сравнительный анализ позволяет, в частности, искать новые консервативные (и альтернативные) структуры РНК. Для проведения такого рода массовых исследований необходимы методы, алгоритмы и компьютерные программы автоматического и, главное, быстрого поиска вторичных структур.

Эти методы должны допускать наличие в исходной выборке некоторого количества "мусорных" последовательностей, т.е. таких, которые заведомо не содержат искомой консервативной структуры. Другая их важная особенность должна состоять в том, чтобы заранее не требовалось выравнивания исходных последовательностей. И наконец, такие методы должны быть столь эффективными (быстрыми), чтобы за разумное время проводить полногеномный анализ.

ОБЩАЯ ПОСТАНОВКА ЗАДАЧИ

Будем называть потенциальной спиралью в данной последовательности пару взаимно комплементарных плеч (состоящих из отрезков и промежутков между ними – выпячиваний и внутренних петель). Пары потенциальных спиралей могут иметь различное взаимное расположение. Основные виды взаимного расположения спиралей показаны на рис. 1. Более полная классификация их взаимного расположения приведена в [12]. Частично перекрывающиеся спирали будем относить к одному из указанных здесь случаев. Рассмотрим множество всех возможных потенциальных спиралей (или их разумную, например, по значению энергии часть) в одной из данных последовательностей. Для этого множества спира-

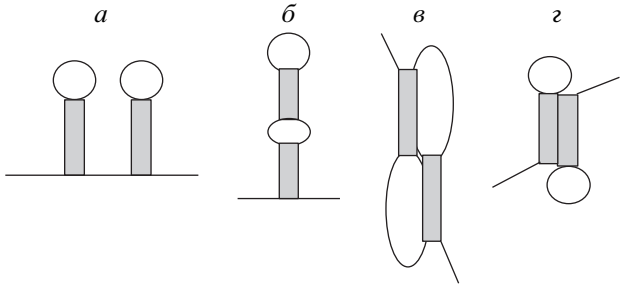

Рис. 1. Возможные взаимные расположения двух спиралей: *а* – две шпильки, *б* – вложенные спирали с внутренней петлей, *в* – псевдоузлы, *г* – взаимоисключающие спирали.

лей строится граф, вершины которого приписаны потенциальным спиралям. Две вершины графа (две спирали) соединены ребром в том случае, если эти спирали "совместимы в одной вторичной структуре", где понятие совместимости меняется в зависимости от биологической задачи. Обычно ситуации *в* и *г* на рис. 1 считают несовместимыми. Каждое ребро можно считать "покрашенным в цвет", который указывает на взаимное расположение спиралей, приписанных его концам. Такой граф будем называть *графом вторичных структур*.

Отношение "спираль E целиком содержит в своей петле спираль F", которое *обозначается* E > F, играет здесь существенную роль. Другое важное отношение состоит в том, что спираль E расположена целиком левее спирали F, в этом случае будем записывать E/F. Понятно, что взаиморасположение четырех спиралей клеверного листа и некоторые другие вторичные структуры целиком описываются этими двумя отношениями, рис. 2.

В самом общем виде задачу поиска вторичных структур *в одной* данной последовательности можно сформулировать как *задачу поиска максимальных клик или достаточно плотных подграфов в графе вторичных структур* [20].

Теперь, переходя к постановке, в которой *дан набор исходных последовательностей*, рассмотрим соответствующий ему набор таким образом полученных графов вторичных структур (по одному графу вторичных структур для каждой из данных нуклеотидных последовательностей). *Консервативной вторичной структурой* называется набор попарно изоморфных (с учетом цвета и всех других приписанных ребрам и вершинам характеристик) подграфов этих графов. В этом случае *задача состоит в поиске таких наборов*. Конечно, можно построить множество очень разных таких наборов (например, если взять по одной спирали в каждой последовательности).

Для поиска биологически содержательной вторичной структуры нужно сформулировать критерий качества консервативной вторичной

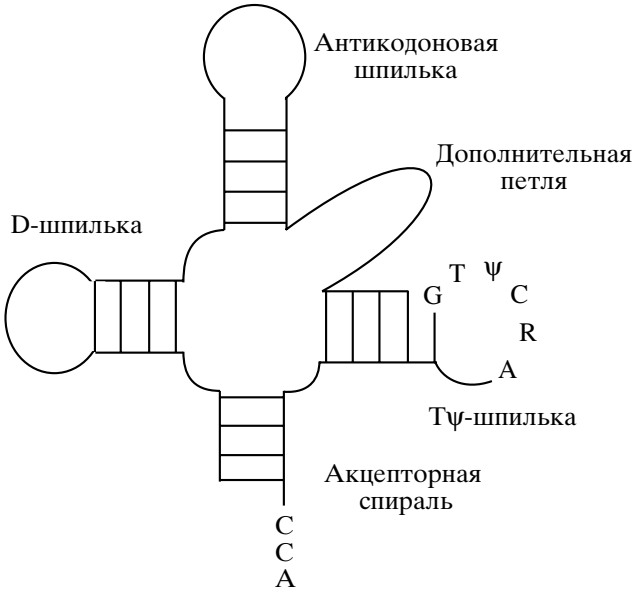

Рис. 2. Клеверная структура тРНК. Ac – акцепторная спираль, D – "левая боковая" D-спираль, An – антикодонная спираль, ψ – "правая боковая" спираль. Очевидно, что следующая система отношений описывает топологию тРНК: Ac > D, Ac > An, Ac > ψ, D/An, An/ψ.

структуры, который зависит от самой биологической задачи. Снабдим вершины и ребра дополнительными характеристиками, например расстоянием между плечами спиралей, и(или) последовательностями нуклеотидов в промежутках между спиральными участками (петлях) и в самих спиралях и(или) статистическими свойствами спиралей. В этом случае качество консервативной структуры определяется как функция этих характеристик.

К сожалению, попытка прямо перейти от этой постановки задачи к алгоритму ее решения не может привести к успеху, так как не вызывает сомнения, что таким образом получится переборный алгоритм.

Итак, пусть дан набор из n фрагментов РНК, например набор регуляторных областей ортологичных генов. Далее переменные i и j пробегают номера этих фрагментов от 1 до n. Задача состоит в том, чтобы найти в этих последовательностях подобные (гомологичные) вторичные структуры, причем некоторые из последовательностей могут не содержать этой структуры (так называемые "*мусорные*").

Часто интересно и достаточно было бы найти только ранжирование потенциальных спиралей в каждой из немусорных последовательностей, при котором спираль с меньшим номером (из i-ой последовательности) имела бы большие основания претендовать на вхождение в искомую вторичную структуру этой i-ой последовательности. В такой постановке вопрос о нахождении самой вторичной структуры в i-ой последовательности не ставится (он может возникать только как средство узнать, правильно ли были ранжированы спирали в наборе последовательностей путем сравнения так возникших в них "наилучших" вторичных структур между собой).

В работе описана упрощенная постановка задачи и алгоритм ее решения; приводятся результаты тестирования. Она основана на предварительных публикациях [21–23].

АЛГОРИТМ ПОИСКА КОНСЕРВАТИВНЫХ ВТОРИЧНЫХ СТРУКТУР

Упрощенная постановка задачи и схема алгоритма

В каждой нуклеотидной последовательности найдем все или подходящую часть потенциальных спиралей (критерий отбора ниже уточняется). Затем сравним каждую последовательность с каждой и отметим все аналогичные спирали (критерий аналогичности определяется ниже). Далее для каждой спирали определим ее рейтинг как количество последовательностей, в которых имеются спирали достаточно аналогичные к исходной спирали. В каждой из исходных последовательностей отметим спирали, у которых рейтинг выше некоторого порога (в частности, эти спирали линейно упорядочиваются). Они и объявляются алгоритмом как спирали, принадлежащие консервативной структуре в этой последовательности. Для эффективной работы алгоритма часто существенно учитывать контекст спирали, т.е. наличие в окрестности данной спирали других определенным образом расположенных относительно нее спиралей, наличие консервативных блоков (т.е. слов из нуклеотидов, расположенных определенным образом в этой окрестности) и т.п.

Эти блоки можно искать и независимо от описанного ниже алгоритма, например с помощью алгоритмов множественного выравнивания.

Однако поиск аналогичных спиралей в двух данных нуклеотидных последовательностях попрежнему сложная задача. Поэтому продолжим упрощения в постановке исходной задачи: каждая спираль состоит из двух плеч – левого и правого. И если рассматривать плечи независимо друг от друга (но помнить о каждом плече: от какой спирали оно образовалось и является ли в ней "левым" или "правым"), то задачу поиска аналогичных спиралей в двух нуклеотидных последовательностях можно свести к задаче выравнивания новых *специальных последовательностей* (*состоящих уже из плеч спиралей*), которая эффективно решается методом динамического программирования. Такие последовательности будем называть *последовательностями плеч* (точное определение приведено ниже).

Теперь можно уточнить определение рейтинга спирали, представленной ее двумя плечами в *i*-ой последовательности плеч (которая, в свою очередь, составлена по *i*-ой нуклеотидной последовательности) как сумму (по двум плечам и по всем *j*) весов, которые получит каждое из этих плеч при выравнивании *i*-ой последовательности плеч с каждой другой *j*-ой последовательностью плеч (которая составлена уже по *j*-ой нуклеотидной последовательности). Такова наша упрощенная постановка задачи и схема нашего алгоритма. Понятно, что рейтинги позволяют отобрать лучшие спирали по отдельности в каждой нуклеотидной последовательности, ранжировать их и выявить мусорные последовательности, как содержащие спирали с небольшим суммарным рейтингом.

Описание алгоритма

В соответствии со схемой алгоритма из предыдущего раздела он состоит из следующих этапов:

1. Отбор потенциальных спиралей в каждой из нуклеотидных последовательностей (последовательности индексируются *i* или *j*, спирали из одной последовательности индексируются *k* или *l*).

2. Для каждого *i* построение последовательности из плеч спиралей, отобранных в *i*-ой нуклеотидной последовательности.

3. Выравнивание сразу всех последовательностей плеч (множественное выравнивание) или отдельное выравнивание каждой пары последовательностей плеч.

4. Отбор и ранжирование спиралей в каждой *i*-ой нуклеотидной последовательности. На этом этапе может происходить отсев тех нуклеотидных последовательностей, в которых суммарный ранг всех отобранных спиралей ниже некоторого порога (алгоритм рассматривает такие последовательности как "мусорные").

5. Для каждой *i*-ой нуклеотидной последовательности построение из отобранных в ней спиралей в некотором смысле оптимальной ("консенсусной") вторичной структуры.

6. По набору так полученных вторичных структур построение консенсусной вторичной структуры. При практическом счете этап 5 часто бывает несущественным, а этап 6, в принципе, может быть заменен алгоритмом Зукера. Поэтому эти два этапа не описываются здесь, а приведены в [23].

7. В цикле повторение этапов 1–6 с изменением значений участвующих в них параметров, вплоть до достижения некоторого качества получаемых вторичных структур (или качества получаемой консенсусной структуры).

Теперь обсудим реализацию этапов 1–4.

1. Поиск потенциальных спиралей не представляет трудностей. Его можно проводить разными способами. В частности, использовать технику словарей (аналогичную BLAST, [24]), что позволяет искать спирали очень быстро, но с некоторыми возможными потерями, или использовать более аккуратный, но и более медленный алгоритм Смита–Ватермана. Мы подсчитывали энергию каждой спирали как сумму энергий всех ее пар комплементарных отрезков минус штрафы за выпячивания, внутренние петли и слишком длинную внешнюю петлю. При этом энергия пары комплементарных отрезков равнялась сумме чисел, по одному для каждой пары соседних пар склеенных нуклеотидов. Надо отметить, что эти числа зависят от температуры; мы рассчитывали энергию при температуре в 37°C. Для каждой возможной четверки параметров A, B, C, D (концов плеч) находится и хранится еще параметр – наибольшее значение E энергии всех спиралей с этими концами, а также – одна из спиралей с этим и значениями пяти параметров. После составления этого списка спирали в нем упорядочиваются по возрастанию энергии и список обрезается с конца до заданного размера (мы оставляли от 40 до 80% от всего списка).

Таким образом, алгоритм получает наборы потенциальных спиралей $F_i = \{h_k\}_i$ для каждой исходной нуклеотидной последовательности S_i.

2. Построение последовательности плеч спиралей. По каждому множеству F_i образуем множество F_i', состоящее из плеч всех спиралей, входящих в F_i. Точнее, F_i' состоит из всех пар вида $\langle k, d\rangle$, где переменная k нумерует все спирали из F_i, а дискретная переменная d один раз равна l, а один раз равна r для каждого k. Пара $\langle k, l\rangle$ обозначает левое плечо спирали k, а пара $\langle k, r\rangle$ – правое плечо той же спирали k. Каждое из множеств F_i' линейно упорядочивается, т.е. превращается в последовательность, которая выше была названа *i*-й *последовательностью плеч*, соответствующей исходной *i*-й последовательности нуклеотидов. Мы тестировали два варианта упорядочения: чаще по серединам всех плеч, но также и по первой координате у левого плеча и по последней координате у правого плеча.

Таким образом, индекс i нумерует все последовательности плеч (как и все исходные нуклеотидные последовательности, где i пробегает от 1 до n).

3. Построение выравнивания двух последовательностей плеч. Здесь применялся следующий обычный алгоритм динамического программирования.

На множестве плеч спиралей применим рекурсивную формулу динамического программирования:

$$f(k, I) = \max\{f(k-1, I-1) + W(k, I),$$
$$f(k-1, I) - d, f(k, I-1) - d, 0\},$$

где f – качество выравнивания, оканчивающегося на паре плеч (k, I), $W(k, I)$ – вес соответствия плеч k и I (указанный чуть ниже), а d – штраф за делецию.

Теперь определим *функцию "похожести" (сходства)* $W(h_1, h_2)$ для любых двух спиралей h_1 и h_2 из различных множеств спиралей. Пусть A_i, D_i – внешние концы соответственно левого и правого плеч спирали h_i, а B_i, C_i – внутренние концы ее двух плеч. Для каждой из спиралей h_i образуем два слова: $S_{il} = [A_i - o_A, B_i + o_B]$ и $S_{ir} = [C_i - o_C, D_i + o_D]$, которые являются двумя плечами этой спирали с некоторыми их окрестностями. Здесь $-o$ и $+o$ означает: сдвинуться влево или соответственно вправо на o нуклеотидов. В алгоритме предусмотрена возможность удаления из самих плеч всех выпячиваний и внутренних петель, или, наоборот, удаление всех отрезков – склеенных участков этих плеч. Размеры o_A, o_B, o_C, o_D (влево и вправо) этих окрестностей являются параметрами алгоритма (обычно, их числовые значения не превышали 15). Полагаем

$$W(h_1, h_2) = W(S_{1l}, S_{2l}) + W(S_{1r}, S_{2r}),$$

где S – вес локального выравнивания двух указанных в скобках слов, вычисляемый, например, алгоритмом Смита–Ватермана [23]. Если подсчитанное таким образом сходство двух спиралей оказывается меньше некоторого порога W^*, то оно заменяется на большое отрицательное число (у нас равное числу –1000). Затем сходство $W(h_1, h_2)$ корректируется наложением штрафа за различие длин внешних петель спиралей h_1 и h_2 (а также, возможно, штрафов за длинные внешние петли, за различия длин плеч, призов и штрафов за наличие или отсутствие консервативных нуклеотидов в определенных местах около этих плеч и т.п.).

Описанный подход предполагает следующие уточнения. При выравнивании последовательностей плеч *функция сходства* $W(h_1, h_2)$ двух спиралей (играющая такую же роль, как обычные призы и штрафы при выравнивании двух нуклеотидных последовательностей) корректируется так, что разрешается выравнивание только левого плеча с левым, а правого с правым. Если при выравнивании двух последовательностей плеч оба плеча одной k-й спирали выравнились соответственно с двумя плечами какой-то r-й спирали (это, конечно, может быть и не так), то сами спирали k-ю и r-ю назовем *выравненными*. К весу $l_p(i)$ плеча добавляется приз, если происходит выравнивание не только этого плеча, но и его спирали.

Аналогично, если k-я спираль выравнилась с r-й, а r-я выравнилась с s-й (из какой-то третьей нуклеотидной последовательности), то в случае если при этом и k-я выравнилась с s-й, всем участвующим в этой ситуации плечам добавляется еще приз ("*правило треугольника*").

Наконец, существенной добавкой к изложенному алгоритму является следующее. Кроме множеств F_i' плеч рассмотрим и множества G_i консервативных блоков из той же i-й последовательности. Пусть теперь образована последовательность плеч и блоков из i-й последовательности, которую будем обозначать также F_i'. В этой последовательности отражено взаиморасположение уже как плеч, так и блоков. Будем называть ее *последовательностью плеч-боксов*. Действуем с этой последовательностью также, как и выше, естественно, разрешая выравниваться только блокам на блоки и плечам на плечи. Это приводит к одновременному выравниванию как плеч спиралей, так и консервативных блоков.

Идея изложенного выше алгоритма основана на том, что у консервативных структур гомологичные спирали часто имеют и сходные по последовательности плечи, поэтому выравнивание "последовательностей" спиралей можно заменить на выравнивание последовательностей плеч (или, еще лучше, на выравнивание последовательностей плеч-боксов).

Итак, рассмотренная в этом разделе задача, конечно, проще исходной постановки в терминах графов. Такое упрощение, конечно, имеет свой недостаток: в этом варианте исходной задачи труднее учитывать важное отношение "быть плечами одной спирали".

4. Ранжирование спиралей. Определим качество спирали h следующим образом. Например, пусть речь идет о попарном выравнивании 1-ой последовательности с каждой j-ой (где j пробегает от 2 до n). Припишем l-му плечу из 1-ой последовательности сумму весов, которые оно получает при каждом из этих выравниваний; эту сумму назовем весом l-го плеча. Пусть максимум весов плеч из 1-ой последовательности достигается на каком-то плече l_0, которое обозначим $l_0(1)$.

Заменяя число 1 на произвольный номер последовательности i, аналогично получим плечо $l_0(i)$. Естественно предполагать, что функция $l_0(i)$ дает хороших кандидатов во вторичные структуры в каждой i-й из исходных последовательностей. Более того, последовательности, для которых вес плеча $l_0(i)$ ниже некоторого порога, объявим мусорными и удалим из исходного набора нуклеотидных последовательностей. Аналогично находятся следующие после $l_0(i)$ выделенные плечи, их обозначим $l_1(i)$ и т.д. В каждой из остающихся (не мусорных) нуклеотидных последовательностей образуется некоторое фиксированное число "наилучших" плеч $l_0(i)$, $l_1(i)$, $l_2(i)$, …, $l_p(i)$. Таким образом, в каждой исходной последовательности

будут отобраны и ранжированы некоторые "наилучшие" спирали.

Описанный алгоритм характеризуется следующим набором параметров: максимальный размер внешней петли; минимальный размер внешней петли; максимальный размер внутренней петли; минимальная длина подряд склеенного отрезка в спирали; максимальное число GT-пар в одном отрезке; размеры окрестностей плеч, используемых для подсчета сходства спиралей; коэффициенты штрафов за длинные внешние петли и за различие их длин; максимальное число спиралей, оставляемых для выравнивания; порог сходства двух спиралей; минимальное число последовательностей, которые должны иметь сходную (с рассматриваемой) спираль; максимальное число спиралей, оставляемых после выравнивания для построения вторичной структуры; индикатор допустимости перекрытий в структуре; параметры выравнивания (приз за совпадение букв, штрафы за делецию и несовпадение букв); количество итераций выравнивания.

Алгоритм реализован в виде консольного приложения для платформы Windows 9x/NT/2000. Исполняемая версия программы и подробные результаты тестирования доступны на сайте http://www.iitp.ru / lyubetsky, а также в случае запроса по адресу gorbunov@iitp.ru.

РЕЗУЛЬТАТЫ СЧЕТА И ОБСУЖДЕНИЕ АЛГОРИТМА

Тестирование для случая тРНК

Сначала приведем результат тестирования нашего алгоритма на 18 фрагментах тРНК кишечной палочки *Escherichia coli*, каждый из которых длиной около 75 нуклеотидов. Эти фрагменты являются соответствующими генами и взяты из базы http://ncbi.nlm.nih.gov/. Ниже приводится табл. 1, в левой колонке которой расположены названия генов тРНК в полном геноме *E. coli*, а в каждой строке после длины фрагмента указаны найденные в нем нашим алгоритмом спирали вторичной структуры.

Таблица 1. Результаты тестирования алгоритма

Ген тРНК *E. coli*	Длина фрагмента	ACC	D	A	Ψ	Ложные спирали
alaV	76	+	+	+	+	2
alaX	76	+	+	+	+	1
cysT	74	+	–	+	+	2
aspU	77	+	–	+	+	2
gltU	76	+	–	+	+1, –2	1
pheV	76	+	+	+	–	1
glyT	75	+	–	–	+	3
glyV	76	+	+	+	+	1
glyU	74	+	+	+	+	1
hisR	77	+	+	+	+1, –2	0
IleV	77	+	+	+2, –1	+	0
IleX	76	+	+	+1, +7	–	1
lysT	76	+	+	+	–	1
leuQ	87	+	–	–2, +2	+	0
leuW	85	+	–	–	+	2
leuX	85	+	+	+	+	1
leuU	87	+	–	+	+	0
leuZ	87	+	–	+	+	2
%		100	56	89	83	

Примечание. Колонки с 3-й по 6-ую соответствуют следующим спиралям: ACC – акцепторная спираль, D – спираль, A – антикодонная, Ψ – псевдоуридиновая спирали (рис. 2). Знак "+" показывает, что соответствующая спираль найдена точно. Два числа означают, что биологическая спираль найдена нашим алгоритмом с погрешностью, причем модули этих чисел указывают на величину погрешности на двух концах внешней петли, а знаки этих чисел – на направления сдвигов. Знак "–" означает, что алгоритм не нашел истинной спирали. В последней колонке указано число ложных спиралей, которые выдал наш алгоритм. В последней строке таблицы указан процент спиралей каждого вида, точно найденных алгоритмом.

Таблица 2. Результаты тестирования алгоритма при добавлении 9 случайных последовательностей

Ген	Длина	ACC	D	A	Ψ	Ложные спирали
alaV	76	+	+	+	+	0
alaX	76	+	+	+	+	1
cysT	74	+	–	+	+	1
aspU	77	–	–	–	+	1
gltU	76	+	–	+	+1, –2	1
pheV	76	+	+	+	–	1
glyT	75	+	–	–	+	2
glyV	76	+	+	+	+	0
glyU	74	+	–	+	+	0
hisR	77	+	+	+	+1, –2	0
IleV	77	+	+	–	+	0
IleX	76	+	+	+1, +7	–	1
lysT	76	+	+	+	–	1
leuQ	87	–	–	–2, +2	+	1
leuW	85	+	–	–	+	2
leuX	85	+	+	+	+	1
leuU	87	–	–	+	+	0
leuZ	87	+	–	+	+	1
%		83	50	78	83	

Примечание. Обозначения как в табл.1.

Одно замечание о ложных спиралях, выдаваемых алгоритмом. Их плечи, как правило, расположены поблизости от плеч биологических спиралей. При этом ложная спираль либо образует псевдоузел с биологической спиралью (и тогда, конечно, истинная не находится), либо ложная спираль является вариантом истинной (и тогда находятся обе спирали). В последнем случае вес истинной спирали часто (более, чем в 68% всех рассмотренных нами случаев) оказывается больше веса ложной.

В реальных задачах нет уверенности, что все представленные последовательности гарантированно имеют искомую вторичную структуру. Поэтому было проведено тестирование с целью выяснить, насколько устойчиво работает наш алгоритм в ситуации, когда исходная выборка разбавлена некоторым числом "мусорных" последовательностей. Для этого к нашей выборке добавлялись 1, 3, 5, 7, 9 и т.д. случайных бернуллиевских последовательностей, и к так расширенным выборкам применялся наш алгоритм. Аналогичным образом к исходным последовательностям добавлялись с обеих сторон случайные фланги. В итоге получены результаты, показывающие высокую устойчивость алгоритма как к добавлению мусорных последовательностей, так и к добавлению случайных полей (флангов) с обеих сторон. Результаты счета приведены на сайте http://www.iitp.ru/lyubetsky.

Таблица 3. Процент нахождения четырех спиралей тРНК в зависимости от числа добавленных случайных последовательностей

Число последовательностей	0	1	3	5	7	9
ACC-спираль	83.3	72.2	83.3	83.3	83.3	66.7
D-спираль	66.7	44.4	50	44.4	44.4	44.4
A-спираль	77.8	88.9	83.3	77.8	66.7	83.3
Ш-спираль	94.4	88.9	83.3	83.3	72.2	72.2
Средний %	80.6	73.6	75	72.2	66.7	66.7

Здесь в табл. 2–4 приведена малая часть этих результатов, а также общая статистика такого разбавления исходной выборки.

Начнем со случая, когда та же самая "чистая" выборка тРНК, что и выше в табл. 1, была нагружена девятью случайными последовательностями. В результате выборка увеличилась в 1.5 раза. По данным табл. 2 и аналогичным результатам, приведенным на сайте http://www.iitp.ru/lyubetsky, видно, что средний процент всех найденных алгоритмом спиралей с ростом числа случайных последовательностей медленно убывает с 82 до 73.5%. Кроме того, видно, что D-спираль нахо-

Таблица 4. Результаты тестирования алгоритма при добавлении случайных флангов по 40 нуклеотидов с каждой стороны

Ген	Длина	ACC	D	A	Ψ	Ложные спирали
AlaV	76	+	+	+	+	2
AlaX	76	+	–	+	+	2
Cyst	74	+	–	+	+	2
AspU	77	–	–	–	+	1
GltU	76	–	–	+	+	3
PheV	76	+	+	+	–	3
GlyT	75	+	–	–	+	4
GlyV	76	+	+	+	+	2
GlyU	74	+	–	+	+	2
HisR	77	+	–	+	+	2
IleV	77	–	–	+	+	1
IleX	76	+	+	+	+	0
LysT	76	+	+	+	+	0
LeuQ	87	–	–	–	+	5
LeuW	85	–	–	+	+	1
LeuX	85	–	–	+	+	2
LeuU	87	–	–	+	+	2
LeuZ	87	–	–	+	+	3
%		56	28	83	94	

Примечание. Обозначения как в табл. 1.

дится с большим трудом, что естественно объясняется ее малой длиной и низкой консервативностью соответствующего нуклеотидного участка.

Отметим, что алгоритм не использовал никакой специфики структуры тРНК, как и ниже – специфики структур RFN, T-боксов и S-боксов.

Приведем проценты нахождения каждой из четырех спиралей тРНК в зависимости от числа добавленных случайных последовательностей (проценты указаны по возрастанию числа случайных последовательностей, равному соответственно 0, 1, 3, 5, 7, 9), табл. 3.

Немонотонность уменьшения этого показателя объясняется тем, что количество (ложных) спиралей, возникающих при добавлении случайных последовательностей или флангов может сильно колебаться.

В реальных ситуациях, разумеется, неизвестна точная локализация искомой вторичной структуры. Поэтому проведено тестирование на той же, что выше, "чистой" выборке тРНК, с добавленными к каждой из ее последовательностей биологическими флангами с длинами от 10 до 40 и более нуклеотидов с каждой стороны. При этом алгоритм нашел дополнительные биологически значимые спирали (от 1 до 3), которые принадлежат предыдущему или последующему генам тРНК. Это неудивительно, учитывая, что гены тРНК образуют кластеры. Эти результаты полностью вынесены на сайте http://www.iitp.ru/lyubetsky.

Ситуация, когда вторичная структура тандемно повторяется в геноме несколько раз (кроме случая тРНК) встречается крайне редко. Поэтому была проведена другая серия тестов, при которой каждая последовательность из той же исходной выборки обрамлялась флангами из случайной бернуллиевской последовательности с длинами от 10 до 40 и более нуклеотидов справа и слева. В табл. 4 такой результат приведен для случая флангов по 40 нуклеотидов каждый (другие результаты см. на том же сайте).

Точность предсказания и в этом случае медленно убывает с ростом размеров флангов: по всем найденным алгоритмом спиралям в среднем от 82 до 65%.

Случай RFN-структуры

Рассмотренный выше случай тРНК является обычной тестовой ситуацией для анализа вторичных структур. Однако реальные структуры часто бывают гораздо бóльшими по длине, сложнее по числу и расположению спиралей, гораздо менее консервативными. Поэтому в качестве следую-

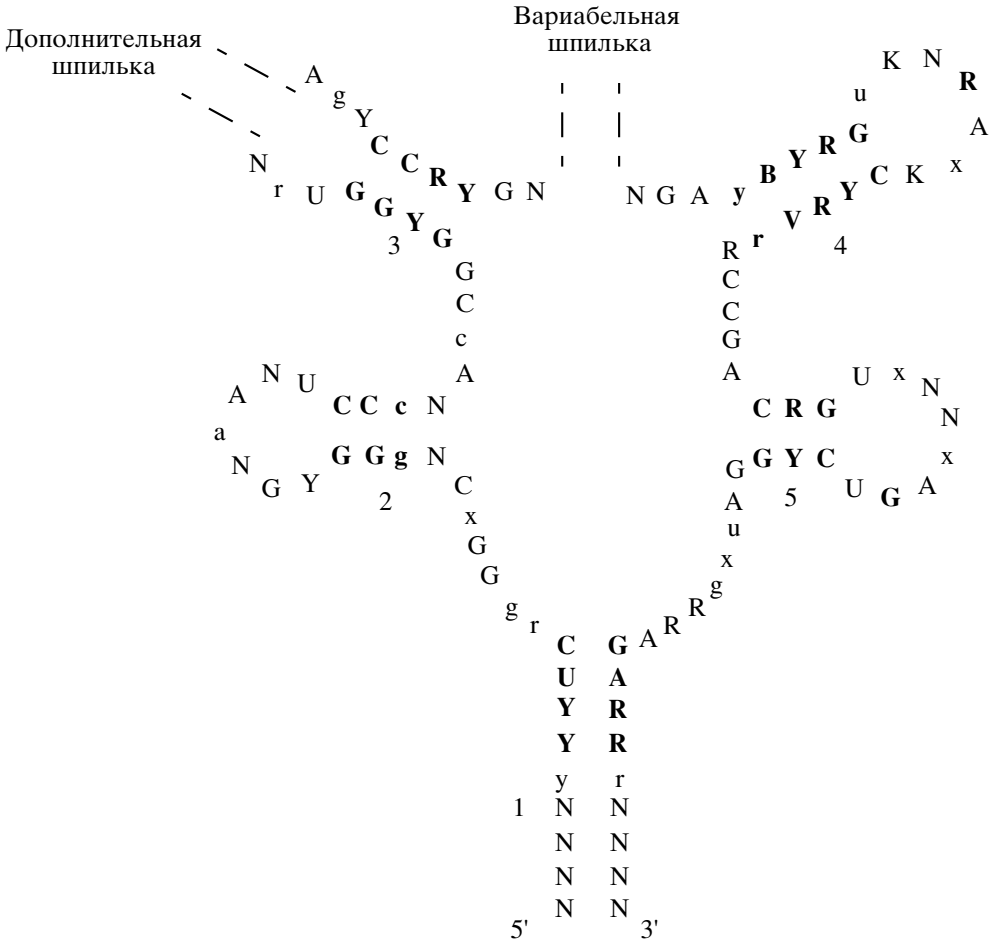

Рис. 3. RFN-элемент. Далее ссылки относятся к приведенным здесь с нумерацией по часовой стрелке пяти спиралям этого элемента.

щего примера посмотрим, как наш алгоритм может предсказывать RFN-элемент.

RFN-структура [1] регулирует экспрессию генов биосинтеза и транспорта рибофлавина. Она состоит из черенка и четырех спиралей (рис. 3; другие спирали обычно не столь консервативны и нами не искались). В отличии от случая тРНК RFN-элемент содержит гораздо более разнообразные по размеру петли (в основном, от 5 до 50), что влияет на работу алгоритма. Длины последовательностей варьировали от 119 до 170.

Ниже приведены два результата работы нашего алгоритма на одной и той же выборке из 39 RFN-фрагментов. При этом учитывались нуклеотиды из окрестностей как левого, так и правого плеч каждой спирали (по 10 нуклеотидов с каждой стороны от плеча). В первом тестировании (второй столбец табл. 5) отсутствовало всякое ограничение на длину внешней петли.

Во втором тестировании (третий столбец той же таблицы) в алгоритме было задано ограничение на длину внешней петли сверху, равное 35. Естественно, что при таком ограничении черенок не мог быть найден и не искался. Однако улучшается качество работы алгоритма за счет значительного уменьшения числа рассматриваемых спиралей, резко уменьшается время его работы, что позволяет обрабатывать выборки гораздо большего размера.

В колонках табл. 5 представлен процент предсказания спиралей. Знак “+” означает практически точное предсказание спирали с возможным сдвигом по каждому плечу на не более чем, половину его длины. Знак “±” означает, что найденная спираль отличается от правильной не более, чем на 6 нуклеотидов. Знак “–” означает, что спираль не найдена или найдена с большим сдвигом. Заметим, однако, что расположение спиралей в RFN-структуре известно лишь приближенно, так что сдвиги указываются относительно *предполагаемого* биологического ответа, и в каких-то случаях наш ответ, отнесенный здесь к числу неточных, может оказаться правильным. Этот вопрос требует отдельного биохимического анализа.

Таблица 5. Проценты качества предсказания пяти типов спиралей RFN-элемента

Номер спирали	Верхняя граница для внешней петли спирали без ограничения (равна 170), %			Верхняя граница для внешней петли спирали равна 35, %		
	+	±	–	+	±	–
1	82	8	10	0	0	0
2	64	23	13	33	62	5
3	72	7	21	62	10	28
4	77	0	23	77	5	18
5	0	41	59	28	49	23

Примечание. Нумерация спиралей соответствует рис. 3. Знак "+" означает практически точное предсказание спирали с возможным сдвигом по каждому плечу на не более, чем половину его длины. Знак "±" означает, что найденная спираль отличается от правильной не более, чем на 6 нуклеотидов. Знак "–" означает, что спираль не найдена или найдена с большим сдвигом. Заметим, что расположение спиралей в RFN-структуре известно лишь приближенно, так что сдвиги указываются относительно предполагаемого биологического ответа, и в каких-то случаях наш ответ, отнесенный здесь к числу неточных, может оказаться правильным.

В случаях, когда склеенные отрезки спиралей структуры имеют в основном длины не менее 4 (как, например, в случаях структур T-бокс или S-бокс), разумно увеличить такой параметр алгоритма как минимальную длину подряд склеенных нуклеотидов в каждой спирали, участвующей в выравнивании. Это приводит к значительному сокращению числа рассматриваемых спиралей. Результаты приведены на том же сайте.

Заметим еще, что часто можно отличить истинные спирали от ложных по качеству (суммарному весу при всех выравниваниях), которые они набрали. Это систематически проявляется в примере, который приведен на том же сайте; там также приведены сходные результаты тестирования для случаев структур T- и S-боксов.

Повторим еще раз, что везде применялся один и тот же алгоритм, не использующий специфики каждого из рассмотренных случаев.

Кроме хорошего качества предсказания, наш алгоритм показал хорошее быстродействие. Так, на персональном компьютере Penteum-4 (2.4 ГГц) тестирование на указанных выше тРНК файлах занимало от 1 до 3 с, а на указанных RFN файлах – от 40 до 60 мин.

СПИСОК ЛИТЕРАТУРЫ

1. Vitreschak A.G., Rodionov D.A., Mironov A.A., Gelfand M.S. 2002. Regulation of riboflavin biosynthesis and transport genes in bacteria by transcriptional and translational attenuation. *Nucleic Acids Res.* **30**, 3141–3151.
2. Rodionov D.A., Vitreschak A.G., Mironov A.A., Gelfand M.S. 2002. Comparative genomics of thiamin biosynthesis in procaryotes. New genes and regulatory mechanisms. *J. Biol. Chem.* **277**, 48949–48959.
3. Vitreschak A.G. Computer analysis of regulation of genes, encoding aminoacyl-tRNA synthetases and amino acid biosynthetic proteins in Gram positive bacteria: T-box RNA regulatory element. Prediction of regulation of new genes, including amino acid transporters. In: *The Proceedings of International School "Artifical Intelligence and Heuristic Methods for Bioinformatics"*. 2001, October 1–11. Italy, San-Miniato. 63.
4. Grundy F.J., Henkin T.M. 1998. The S box regulon: a new global transcription termination control system for methionine and cysteine biosynthesis genes in gram-positive bacteria. *Mol. Microbiol.* **30**, 737–749.
5. Туманян В.Г., Сотникова Л.Е., Холопов А.Е. 1966. Об определении вторичной структуры РНК по последовательности нуклеотидов. *Докл. АН СССР.* **166**, 1465–1468.
6. Nussinov R., Jacobson A.B. 1980. Fast Algorithm for predicting the secondary structure of single-stranded RNA. *Proc. Natl. Acad. Sci. USA.* **77**, 6309–6313.
7. Zuker M., Stiegler P. 1981. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Res.* **9**, 133–148.
8. Zuker M. 1989. Computer prediction of RNA structure. *Meth. Enzymol.* **180**, 262–288.
9. Zuker M. 1991. Suboptimal sequence alignment in molecular biology. Alignment with error analysis. *J. Mol. Biol.* **221**, 403–420.
10. Mironov A.A., Dyakonova L.P., Kister A.E. 1985. A theoretical analysis of the kinetics of RNA secondary structure formation. *J. Biomol. Struct. Dynam.* **2**, 953–962.
11. Миронов А.А., Кистер А.Э. 1985. Теоретический анализ структурных перестроек в процессе образования вторичных структур РНК. *Молекуляр. биология.* **23**, 61–71.
12. Mironov A.A., Lebedev V.F. 1993. A kinetic model of RNA folding. *Biosystems.* **30**, 49–56.
13. Woese C.R., Magrum L.J., Gupta R., Siegel R.B., Stahl D.A., Kop J., Crawford N., Brosius J., Gutell R., Hogan J.J., Noller H.F. 1980. Secondary structure model for bacterial 16S ribosomal RNA: phylogenetic, enzymatic and chemical evidence. *Nucleic Acids Res.* **8**, 2275–2293.
14. Tahi F., Gouy M., Regnier M. 2002. Related Articles, Books, Link Out Automatic RNA secondary structure prediction with a comparative approach. *Computer Chem.* **26**, 521–530.

15. Hofacker I.L., Fekete M., Stadler P.F. 2002. Secondary structure prediction for aligned RNA sequences. *J. Mol. Biol*. **319**, 1059–1066.
16. Gorodkin J., Stricklin S.L., Stormo G.D. 2001. Discovering common stem-loop motifs in unaligned RNA sequences. *Nucleic Acids Res*. **29**, 2135–2144.
17. Akmaev V.R., Kelley S.T., Stormo G.D. 2000. Phylogenetically enhanced statistical tools for RNA structure prediction. *Bioinformatics*. **16**, 501–512.
18. Chen J.H., Le S.Y., Maizel J.V. 2000. Prediction of common secondary structures of RNAs: a genetic algorithm approach. *Nucleic Acids Res*. **28**, 991–999.
19. Titov I.I., Ivanisenko V.A., Kolchanov N.A. 2000. FITNESS-A WWW-resource for RNA folding simulation based on genetic algorithm with local optimization. *Comput. Technol*. **5**, 48–56.
20. Миронов А.А., Дьяконова Л.П., Кистер А.Э. 1984. Теоретический анализ кинетики образования вторичных структур РНК. *Докл. АН СССР*. **259**, 725–728.
21. Gorbunov K.Yu., Lyubetsky V.A. An algorithm for searching common secondary structures in a set of RNA sequences. In *The Proceedings of the third international conference of bioinformatics of genome regulation and structure, BGRS'2002*. 2002, July 14–20, Novosibirsk, Russia, **3**, 21–23.
22. Горбунов К.Ю., Любецкая Е.В., Любецкий В.А. 2001. О двух алгоритмах поиска альтернативной вторичной структуры РНК. *Информационные процессы* (http://www.jip.ru/). **1**, 178–187.
23. Горбунов К.Ю., Любецкий В.А. 2002. Алгоритм поиска консервативных вторичных структур в наборе фрагментов РНК. *Информационные процессы* (http://www.jip.ru/). **2**, 55–58.
24. Altschul S.F., Gish W., Miller W., Myers E.W., Lipman D.J. 1990. Basic local alignment search tool. *J. Mol. Biol*. **215**, 403–410.

Search for Conserved Secondary Structures of RNA

K. Yu Gorbunov[1], A. A. Mironov[2], and V. A. Lyubetsky[1]

[1]Institute for Information Transmission Problems, Russian Academy of Sciences, Moscow, 101447 Russia; E-mail: gorbunov@iitp.ru

[2]State Research Center GosNIIGenetika, Moscow, 113545 Russia

Abstract—We suggest a new algorithm to search a given set of the RNA sequences for conserved secondary structures. The algorithm is based on alignment of the sequences for potential helical strands. This procedure can be used to search for new structured RNAs and new regulatory elements. It is efficient for the genome-scale analysis. The results of various tests run with this algorithm are shown.