Информационные процессы, Том 1, № 1, 2001, стр. 56–63.

ТЕОРИЯ И МЕТОДЫ ОБРАБОТКИ ИНФОРМАЦИИ

Алгоритм выделения регуляторных сигналов в последовательностях ДНК[1]

Л.В.Данилова*, К.Ю.Горбунов*, М.С.Гельфанд, В.А.Любецкий***

**Институт проблем передачи информации, Российская академия наук, Москва, Россия*
***Государственный научный центр ГосНИИГенетика, Москва, Россия*

Поступила в редколлегию 3.01.2001

Аннотация—Предложен алгоритм для выделения регуляторных сигналов в последовательностях ДНК. Сложность алгоритма близка к квадратичной. Приводятся результаты тестирования алгоритма на искусственных и природных последовательностях.

1. ВВЕДЕНИЕ

Задача выделения регуляторных сигналов является одной из классических задач вычислительной биологии. Она приобрела особую популярность в последние несколько лет после того, как началось проведение массовых экспериментов по анализу экспрессии генов (например, [1],[2], обзоры [3],[4]), с одной стороны, и были опубликованы полные геномы многих бактерий и некоторых эукариот, что сделало возможным сравнительный анализ процессов регуляции (в частности, [5],[6]; обзор [7]). Несмотря на всю важность, эта задача еще далека от ее решения. В частности, точные (а тем самым, и переборные) методы занимают столь большое время, что практически не осуществимы и не пригодны для решения практических задач.

Существующие подходы и алгоритмы выделения регуляторных сигналов из набора потенциальных регуляторных областей описаны в обзорах [8]–[11]. В настоящей работе предложен и тестирован *новый алгоритм для выделения сигнала из выборки невыравненных нуклеотидных последовательностей* (см. также: [12]).

2. АЛГОРИТМ

Постановка задачи. Дан набор из k нуклеотидных последовательностей, длины которых не фиксированы или примерно одинаковы (и в этом случае равны каждая какому-то числу n). *Системой* назовем набор слов фиксированной длины l, по одному слову из одной последовательности (или по несколько слов из одной последовательности — здесь для краткости будем говорить о первом случае); в систему включаются слова из какой-то заранее не фиксированной части исходных последовательностей; система должна состоять из как можно более попарно похожих друг на друга слов. Например, в смысле суммы попарных расстояний Хэмминга между словами, входящими в систему, или в смысле какой-то другой фиксированной метрики между этими словами, или, лучше сказать, в смысле максимизации какого-то фиксированного “качества системы”. Качество системы определяется, например, как сумма попарных “расстояний” между ее словами, вычисляемых с помощью функции $F(x, y)$, которая для двух слов x и y длины l отражает степень их похожести между собой (например, количество совпадающих букв в них). Интуитивно такая система понимается как *сигнал*; а слова, являющиеся представителями сигнала в исходных последовательностях, — как *регуляторные участки* (*сайты*). Проблема в том, что некоторые последовательности из исходной выборки могут

[1] Эта работа была частично поддержана грантами от Merck Genome Research Institute (244), INTAS (99-1476), РФФИ (99-04-48247 и 00-15-99362) и программой “Геном человека”.

не содержать регуляторных участков. Важно, что описанный алгоритм находит сигнал даже в том случае, когда он содержится в относительно небольшой части всех исходных последовательностей.

При необходимости можно учесть структурные особенности сигналов. Так, в частности, если есть основания полагать, что сигнал является (комплементарным) палиндромом (например, когда известно, что регуляторный белок связывается с ДНК как димер), то можно использовать, например, функцию $F(x,y) = S(x,y) + 0.5(max(S(x,pal(x)), S(x',pal(x'))) + max(S(y,pal(y)), S(y',pal(y'))))$, где $S(x,y)$ — количество совпадающих букв в словах x и y, а $\mathrm{pal}(x)$ — слово, полученное из x обращением с заменой каждой буквы на комплементарную и x' — слово x без последней буквы.

Качественное описание алгоритма. Сначала образуем вспомогательный граф G, который остается фиксированным в процессе работы алгоритма. Граф G состоит из k вершин и всех ребер, которые возникают в процессе выполнения следующей процедуры. На первом шаге все вершины графа G разбиваются на две равные (с точностью до единицы, если k нечетное) части и между этими частями проводятся два ребра (A,B) и (C,D), не выходящие из одной вершины (пусть, скажем, A и C находятся в одной части, а B и D в другой). Любое из этих ребер (скажем, (A,B)) назовем *основным относительно этого разбиения*, а другое — *вспомогательным*. Также проводятся два *диагональных* ребра: (A,D) и (C,B). Каждую из двух полученных частей снова разбиваем на две (в том же смысле) равные части так, что A и C, как и B и D, находятся в разных частях этих разбиений. Основные ребра относительно уже этих разбиений определены однозначно: это (A,C) и (B,D),а вспомогательные ребра (по возможности, не выходящие из той же вершины, что и основные) выбираются произвольно. И так далее, каждую появившуюся в этой процедуре не одновершинную часть P разбиваем на две равные части так, чтобы основные ребра уже этих частей соединяли концы основного и вспомогательного ребер исходной части P. Процедура разбиений прекращается, когда все части P станут одновершинными; на самом деле, можно остановиться, когда эти части станут мелкими (из 1–3 вершин).

Внешний цикл алгоритма состоит во взаимно однозначном приписывании каждой вершине графа G одной из исходных последовательностей (одну из таких текущих расстановок последовательностей по вершинам графа G обозначим r) и последующего цикла сборки (см. ниже). Вопрос о том, как лучше выбирать совокупность таких приписываний (всех таких расстановок), является центральным. Суть дела в том, что качество системы s (иными словами, качество *сечения*s над G) определяется как сумма значений функции $F(x,y)$ по всем парам вершин графа G (т.е. как сумма по всем "ребрам" графа G, как если бы он был полным графом; это качество обозначим $H(s)$; здесь предполагается, что в систему входит по слову из каждой последовательности). И мы хотим приблизить это качество, беря вместо $H(s)$ величину $H(s,r)$, определяемую как сумма значений функции $F(x,y)$ по всем тем парам вершин графа G, которые в нем действительно соединены ребром. В сумме $H(s,r)$ гораздо меньше слагаемых, чем в сумме $H(s)$, и ее вычисление происходит гораздо быстрее.

За счет этого предлагаемый алгоритм решает исходную задачу за время **квадратичное от числа k исходных последовательностей и кубичное от их длины n.**

Конкретно, для данного r выполняется цикл (называемый *сборкой*), который мы опишем индукцией по глубине разбиений. Индуктивный шаг: пусть для двух частей P_1 и P_2 с основными ребрами соответственно $(,)$ и $(,D)$, полученных разбиением подграфа P с основным ребром $(,)$, уже определены два набора из t лучших сечений как продолжений с их основных ребер (точнее, для любых двух слов из последовательностей над и с качеством большим некоторого фиксированного *порога* определены t лучших продолжений соответственно на все множество P_1; и аналогично на все множество P_2). Тогда для любых слов (скажем, x и y) из последовательностей над и с качеством большим того же порога определим t лучших продолжений на все множество P (равное объединению множеств P_1 и P_2) следующим образом. В цикле рассмотрим все слова (скажем, x_1 и y_1) из последовательностей над C и D, для которых качество слов x и x_1 и y и y_1 выше этого порога (точнее, для которых определены наборы продолжений); и для x, x_1 и y, y_1 выберем продолжения на P_1 и P_2. Объединяя их подхо-

дящим образом, получим t лучших сечений над всем P. Если условие пороговости не может быть обеспечено, то соответствующее значение сечения над P считается по определению равным нулю; иными словами, из-за пороговости возникают частично определенные сечения над G. Кроме того, чтобы среди t лучших сечений, получаемых на различных шагах сборки, было меньше таких, которые не дают новых сигналов (по сравнению с уже *утвержденными* сечениями), проводится проверка каждого нового сечения s на существенность относительно списка S уже утвержденных сечений (с той же областью определения). Для этого, если на достаточно большой доле последовательностей сечение s не имеет новых по сравнению с S слов, предполагаем, что среди этих неновых слов из s значительную часть составляет сайты, и смотрим насколько близко к этой *сигнальной* совокупности каждое оставшееся слово из s. Если среди них не обнаружено близких к этой совокупности слов, то отвергаем s и переходим к следующему кандидату в список.

Внешний цикл, состоящий в расстановке последовательностей по вершинам графа G и последующей сборке, работает, по крайней мере, до тех пор, пока любая пара последовательностей хотя бы раз не соединится ребром в графе G. Процедура расстановки устроена так, чтобы на каждой итерации по r покрыть ребрами графа G больше пар последовательностей, не покрытых на предыдущих итерациях этого цикла. Это обеспечивает разумное количество итераций этого цикла при достаточном разнообразии распределений последовательностей по вершинам графа G. Последнее важно для получения достаточно представительной статистики на следующем, последнем цикле работы алгоритма.

А именно, каждой позиции в каждой последовательности (содержащей, скажем, букву i) ставится в соответствие число, которое отражает меру того, что буква i входит в искомый сайт. Второй вариант состоит в том, чтобы сопоставлять такое число каждому подслову, отражая тем самым меру того, что это подслово входит в искомый сайт (далее рассмотрим этот вариант). Это число равно сумме качеств по всем полученным сечениям, которые включают данное подслово. Здесь под качеством понимается качество не всего сечения, а качество слова из сечения, которое содержит данное подслово, по отношению ко всему сечению, т.е. сумма значений $F(u, x)$, где u — упомянутое слово, а x пробегает все остальные слова этого сечения. Таким образом, подслова, входящие в сайт, будут помечены в исходных последовательностях числами, которые заметно больше чисел, стоящих в других позициях этих же последовательностей.

Счет, проведенный на многих примерах, показал, что в подавляющем большинстве случаев алгоритм находил хотя бы одно сигнальное слово в каждой из содержащих его последовательностей. Чтобы проверить, не содержит ли некоторая последовательность других сигнальных слов применялся следующий прием. Вместо каждой буквы найденного в данной последовательности слова ставился знак “звездочка”, т.е. буква, которую программа считала отличной от всех букв и даже от самой себя. После этого алгоритм находил в последовательности еще одно сигнальное слово (если оно там было) и т.д. В редких случаях алгоритм не находил в какой-то последовательности ни одного сигнального слова (хотя оно там было). Причиной этого был “мусор”, т.е. совокупность очень похожих друг на друга, но не сигнальных слов. Этот мусор выделялся сравнительно большими числами и его легко было увидеть (особенно, с учетом биологических соображений). Тогда “мусорные” слова забивались звездочками, программа запускалась снова и находила ранее не найденные сигнальные слова.

Алгоритм был реализован в виде программы, которая использовалась в основном для поиска сигнала в наборе генетических последовательностей с характерной длиной от 100 до 200 нуклеотидов. В одном частном случае сигнал представлял собой систему отдельных слов с характерной длиной от 15 до 30. В этом варианте программа была написана на языке Object Pascal с помощью среды программирования Delphi. На ее вход подавался текстовый файл, содержащий набор генетических последовательностей, а на ее выходе создавался файл, содержащий функцию, которая каждому подслову сопоставляет степень достоверности того, что оно является сигналом (входит в сигнал). С помощью этой программы при пороге равном половине длины сигнала на компьютере Pentium–

Celeron 300 MHz с оперативной памятью 64 MB обработка 19 последовательностей длины 200 заняло 12 минут, а 9 последовательностей той же длины — 1,2 минуты.

3. РЕЗУЛЬТАТЫ И ОБСУЖДЕНИЕ

Искусственная выборка. Для определения характеристик алгоритма в контролируемых условиях было произведено его тестирование на искусственной выборке. Были сгенерированы последовательности длины 200 в четырехбуквенном алфавите и в каждую из них подставлено одно и тоже слово длины 16. Далее в каждом из этих слов случайным образом “портилось” по несколько букв (имитация ослабления сигнала), а также добавлялись последовательности, не содержащие сигнала (имитация загрязнения выборки).

Результаты тестирования приведены в таблице 1.

Таблица 1. Результаты тестирования на искусственных последовательностях. Приведено количество правильно найденных сайтов, каждый символ соответствует одному независимому тесту (X обозначает 10). Верхняя таблица: выборка из 10 последовательностей. Нижняя таблица: выборка из 5 последовательностей. Строки: добавлено последовательностей (от 0–10 и от 0–5 соответственно). Столбцы: заменено букв (от 0–6 и от 0–4 соответственно).

	0, 1, 2	3	4	5	6
0	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	1061011555	1220021000
1	XXXXXXXXXX	XXXXXXXXXX	XXXXX9XXXX	0063	-
2	XXXXXXXXXX	XXXXXXXXXX	XXXX97XX7X	0053	-
3	XXXXXXXXXX	XXXXXXXXXX	X9X478XX9X	0065	-
4	XXXXXXXXXX	XXXXXXXXXX	799286X96X	1000	-
5	XXXXXXXXXX	XXXXXXXXXX	3498168858	0020	-
6	XXXXXXXXXX	XXXXXXXXXX	05X0477052	0000	-
7	XXXXXXXXXX	XXXXXXXXXX	0330053404	0000	-
8	XXXXXXXXXX	XXXXXXXXXX	2240004025	0000	-
9	XXXXXXXXXX	XXXXXXXX9X	2650427021	0000	-
10	XXXXXXXXXX	XXXXXXXXXX	8423552057	-	-

	0, 1	2	3	4
0	5	5555555555	5555555555	554530255
1	5	5555555555	5555555555	255413144
2	5	5555555555	5555555555	514330023
3	5	5555555555	5555453541	005000030
4	5	5555555555	5535255554	003020011
5	5	4555555555	3403535443	000300013

Сайты устойчиво находятся при внесении в сигнал до 2 ошибок при выборке из 5 последовательностей, до 3 ошибок — из 10 последовательностей, а также, когда количество лишних последовательностей (не содержащих сайты) составляло до 50% выборки. Случай 10 последовательностей был исследован более подробно. При ошибках в 4 позициях сигнала результат зависит от чистоты выборки: приемлемые результаты получаются при количестве лишних последовательностей до 3–4 (в большинстве тестов сайты определяются точно практически во всех последовательностях). При дальнейшем загрязнении выборки сигнал может не быть обнаружен, доля таких тестов повышается с увеличением лишних последовательностей. При более слабом сигнале сайты обнаруживаются только в отдельных тестах.

Природные выборки. По аналогичной схеме были рассмотрены три выборки последовательностей, содержащих регуляторные сайты *Escherichia coli*. Процедура загрязнения выборки лишними

последовательностями имитировалась следующим образом: маскировался лучший из сайтов, полученных на очередном этапе тестирования (нуклеотиды, составляющие этот сайт заменялись буквой ∗). Тем самым, не только появлялись последовательности, не содержащие сайтов, но и постепенно ослаблялся сам сигнал.

Последовательности регуляторных сайтов были взяты из базы данных dpinteract [13], а фрагменты последовательностей, содержащие эти сайты, были извлечены из полного генома *E. coli* при помощи программы GenomeExplorer [14].

Пуриновый регулон. Выборка регуляторных областей генов, регулируемых пуриновым репрессором PurR, состояла из 19 последовательностей длиной 200 нуклеотидов, содержащих 21 сайт длиной 16 нуклеотидов. Две последовательности содержали по два сайта, остальные — по одному. Результаты тестирования приведены в таблице 2. Первые ошибки появляются при маскировке 8–9 сайтов, однако даже если выборка более чем наполовину состояла из последовательностей, не содержащих сайты, большинство сайтов опознавалось правильно.

Таблица 2. Выделение сайтов связывания PurR. Первая строка: количество последовательностей, не содержащих сайтов, и общее количество замаскированных сайтов (обозначены звездочками). Вторая строка: C — вес истинного сайта, M — максимальный вес ложного сайта. Жирный шрифт — пропущенный истинный сайт; лучший ложный сайт той же или большей достоверности, чем лучший истинный сайт (кроме случая нулевого веса).

	0	0	0	1	1	2	2	3	3	4	4	5	5	6	6	7	7	8	8	9	9	10	10	11	11	12	11	13
Оперон	C	M	C	M	C	M	C	M	C	M	C	M	C	M	C	M	C	M	C	M	C	M	C	M	C	M	C	M
PurR-1	88	8	*	8	*	8	*	14	*	7	*	6	*	6	*	6	*	13	*	7	*	**12**	*	**12**	*	**7**	*	**7**
PurR-2	0		72		63		41		27		24		24		23		23		15		**7**		**7**		**7**		**7**	
PurEK	88	0	88	0	*	12	*	7	*	18	*	12	*	12	*	19	*	12	*	7	*	7	*	7	*	7	*	7
CvpApurF	88	0	88	0	76	0	*	22	*	22	*	21	*	14	*	14	*	13	*	7	*	7	*	7	*	7	*	7
PurC	87	0	80	0	70	0	59	0	29	0	18	6	18	6	22	6	24	7	15	7	15	7	8	7	8	7	**7**	7
PurMN	88	0	88	0	70	0	60	0	30	0	28	0	*	22	*	42	*	13	*	13	*	13	*	6	*	7	*	12
PurL	88	0	88	0	75	0	60	6	*	24	*	24	*	28	*	20	*	13	*	7	*	12	*	12	*	12	*	12
PurB	80	0	80	0	59	7	45	6	27	6	17	6	17	6	16	6	16	7	13	7	13	7	8	7	8	7	**6**	**11**
GuaBA	79	0	80	0	67	0	54	0	27	0	17	7	17	6	23	14	24	7	*	14	*	14	*	14	*	14	*	21
PurHD	88	0	88	0	70	0	59	0	30	0	27	7	27	7	24	7	24	7	**7**	**13**	8	7	8	6	7	7	**0**	**14**
GlyA	80	0	80	0	70	0	59	0	29	0	26	6	26	8	25	7	17	7	8	7	8	7	8	7	**8**	**12**	**0**	**8**
PyrD	88	0	88	0	70	0	49	0	29	0	26	0	26	0	*	12	*	12	*	7	*	7	*	7	*	7	*	11
PrsA	88	0	88	0	70	0	60	0	30	0	*	28	*	21	*	19	*	7	*	12	*	7	*	7	*	7	*	12
GlnB	80	0	80	0	63	0	53	0	27	0	16	8	15	8	14	7	7	6	13	6	13	6	13	6	*	6	*	7
PurA-1	80	0	80	0	63	0	53	0	27	0	24	6	24	6	31	5	22	**6**	**7**	7	**7**	7	11	7	11	7	*	7
PurA-2	0		0		0		0		0		0		0		7		**6**		**6**		**6**		**0**		**0**		**0**	
CodBA	88	0	88	0	75	0	60	0	30	0	27	0	26	0	32	6	17	8	15	8	*	8	*	10	*	8	*	8
PyrC	80	0	80	0	69	0	59	0	29	0	18	5	18	5	9	7	24	7	15	6	**7**	**7**	7	6	**7**	**7**	**7**	**7**
PurT	88	0	88	0	76	0	61	0	30	0	27	0	27	0	26	5	*	7	*	12	*	12	*	12	*	7	*	12
GcvTHP	72	0	72	0	63	0	45	7	26	7	15	0	14	7	14	6	15	7	15	6	15	6	7	6	7	6	**7**	**7**
SpeAB	70	8	70	8	54	5	45	6	18	5	17	8	17	8	23	5	16	7	15	7	15	7	*	7	*	11	*	13

Аргининовый регулон. Выборка регуляторных областей генов, регулируемых аргининовым репрессором ArgR, состояла из 9 последовательностей длиной 200 нуклеотидов, содержащих 19 сайтов длиной 18 нуклеотидов. Одна последовательность содержала три сайта, остальные — по два. Результаты тестирования приведены в таблице 3. Аргининовый бокс — слабый сигнал, и специфичность регуляции осуществляется за счет кооперативного узнавания мультимерными комплексами молекул репрессора пар сайтов, расположенных на фиксированном расстоянии друг от друга [15]. Тем не менее, сайты связывания аргининового репрессора находятся достаточно уверенно даже при маскировке значительного числа лучших сайтов: первые потери обнаруживаются при маскировке 5 сайтов, причем в одной из последовательностей оказываются замаскированными оба сайта.

Регулон катаболитной репрессии. Выборка регуляторных областей генов, регулируемых белком CRP, состояла из 31 последовательности длиной 200 нуклеотидов, содержащих 48 сайтов длиной 22 нуклеотидов. В 16 последовательностях содержался один сайт, в остальных — от двух до четырех. Выборка сайтов связывания CRP содержит множество слабых сайтов, многие из которых не были найдены даже в первоначальной постановке (таблица 4), поэтому в этом случае тесты с маскировкой сайтов не проводились. Следует отметить, что взаимодействия CRP с регуляторными участками сложны и включают динамические переключения с одних сайтов на другие [16]. Поэтому нельзя исключать, что некоторые из ошибочно опознанных сайтов на самом деле функциональны.

Таблица 3. Выделение сайтов связывания ArgR. Обозначения такие же, как в таблице 2. Последняя пара столбцов: в каждой последовательности замаскирован лучший сайт.

	0 0		0 1		0 2		0 3		0 4		1 5		1 6		1 7		1 8		0 9	
оперон	C	M	C	M	C	M	C	M	C	M	C	M	C	M	C	M	C	M	C	M
argR-1	33	0	36	0	*	0	*	8	*	8	*	0	*	**13**	*	**18**	*	**9**	*	18
argR-2	0		0		24		34		18		8		**0**		**8**		**8**		**8**	
argA-1	33	0	32	0	19	0	19	0	17	8	9	7	9	8	**0**	**16**	**0**	**16**	*	**16**
argA-2	0		0		9		26		17		9		9		**9**		**0**		**9**	
argCBH-1	0	0	**2**	**6**	18	0	37	7	27	7	16	8	**8**	**8**	**8**	**8**	**0**	**9**	**10**	**18**
argCBH-2	39		*		*		*		*		*		*		*		*		*	
argD-1	0	0	0	0	0	0	0	0	8	7	**7**	**7**	**7**	**8**	**8**	**8**	**8**	**7**	**8**	**16**
argD-2	35		35		30		51		*		*		*		*		*		*	
argE-1	0	0	0	0	9	0	10	0	10	0	8	0	**0**	8	**0**	**9**	**0**	**9**	**10**	**18**
argE-2	39		27		22		43		38		8		9		**9**		**8**		*	
argF-1	36	0	36	0	31	0	21	0	19	0	**6**	**14**	10	8	*	**17**	*	**8**	*	**18**
argF-2	0		0		0		30		27		**6**		**0**		**7**		**7**		**10**	
argG-1	11	0	0	0	10	0	10	0	19	0	**8**	8	8	7	**9**	**16**	**0**	**9**	**10**	**16**
argG-2	0		0		0		0		0		**0**		8		**0**		**0**		**0**	
argG-3	24		36		22		42		29		10		*		*		*		*	
argI-1	36	0	35	0	31	0	*	0	*	0	*	15	*	8	*	16	*	8	*	**16**
argI-2	0		0		0		47		44		*		*		*		*		**10**	
carAB-1	30	0	16	0	24	0	18	0	19	0	9	8	9	8	17	8	*	**8**	*	**16**
carAB-2	0		7		0		9		8		**8**		**8**		**0**		**0**		**8**	

4. ЗАКЛЮЧЕНИЕ

Тестирование показало практическую применимость предложенного алгоритма. В настоящее время он используется в практике анализа регуляторных взаимодействий; в частности, с его помощью анализируются неохарактеризованные регулоны сахарного метаболизма гамма-протеобактерий, а также — начат систематический анализ регуляции в пирококках. Дальнейшее развитие алгоритма будет проводиться в нескольких направлениях. Актуальной является проблема повышения устойчивости к шуму, в особенности, к загрязнению выборки. Предполагается модифицировать алгоритм таким образом, чтобы он явно принимал во внимание возможность загрязнения, а ожидаемое количество лишних последовательностей являлось параметром, выставляемым исходя из априорных соображений или настраиваемым автоматически. Необходимо также принять во внимание статистические особенности исследуемых геномов, в частности, неравномерность нуклеотидного состава и особенности частот олигонуклеотидов (слово, не являющееся случайным по отношению ко всему геному вряд ли является специфическим регуляторным сигналом). Следует аккуратно исследовать возможности использования различных типов симметрий (палиндромы, прямые повторы и т.п.). Наконец, алгоритм должен более специфически учитывать возможность появления в последовательности нескольких сайтов, а в выборке — нескольких различных сигналов.

Мы благодарны А.А. Миронову за помощь в работе с GenomeExplorer и ценное обсуждение. Эта работа была частично поддержана грантами от Merck Genome Research Institute (244), INTAS (99-1476), РФФИ (99-04-48247 и 00-15-99362) и программой “Геном человека”.

Таблица 4. Выделение сайтов связывания CRP. Обозначения такие же, как в таблице 2.

		C	M
1	aldB	**8**	**15**
2	ansB	**0**	**18**
3	araB-1	17	8
	araB-2	**7**	
4	cdd-1	**0**	7
	cdd-2	28	
5	crp-1	**0**	14
	crp-2	16	
6	cya	27	9
7	cytR-1	17	16
	cytR-2	**7**	
8	dadAX-1	43	8
	dadAX-2	**8**	
9	deoP-1	**9**	**17**
	deop-2	**8**	
10	fur	26	13
11	gal	**8**	**21**
12	glpACB-1	26	8
	glpACB-2	**8**	
	glpACB-3	**8**	
13	glpD	17	8
14	glpFK-1	**0**	15
	glpFK-2	32	
15	gut	34	14
16	ilvB	**10**	**17**
17	lac-1	9	7
	lac-2	24	
18	malEpKp-1	8	6
	malEpKp-2	9	
	malEpKp-3	9	
	malEpKp-4	**0**	
19	malt	18	16
20	melR	**16**	**30**
21	mtl	**17**	**21**
22	nupG-1	20	9
	nupG-2	19	
23	ompA	**16**	**16**
24	ompR	24	16
25	ptsH-1	**9**	**26**
	ptsH-2	**16**	
26	rhaS	19	8
27	rot-1	**8**	8
	rot-2	27	
28	tdcA	28	8
29	tnaL	28	8
30	tsx-1	**10**	**26**
	tsx-2	**7**	
31	uxuAB	23	8

СПИСОК ЛИТЕРАТУРЫ

1. Spellman P.T., *et al.* Comprehensive Identification of Cell Cycle-Regulated Genes of the Yeast *Saccahromyces cerevisiae* by Microarray Hybridization. *Mol. Biol. Cell.,* 1998, vol. 9, pp. 3273–3297.

2. Roth F.R. *et al.* Finding DNA Regulatory Motifs within Unaligned Non-coding Sequences Clustered by Whole-Genome mRNA Quantification. *Nature Biotechnol.,* 1998, vol. 16, pp. 939–945.

3. Bassett D.E. Jr. *et al.* Gene Expression Informatics—It's All in Your Mine. *Nature Genet.,* 1999, vol. 21, pp. 51–55.

4. Bucher P. Regulatory Elements and Expression Profiles. *Curr. Opin. Struct. Biol.,* 1999, vol. 9, pp. 400–407.

5. Gelfand M.S., Koonin E.V., Mironov A.A. Prediction of Transcription Regulatory Sites in Archaea by a Comparative Genomic Approach. *Nucleic Acids Res.,* 2000, vol. 28, pp. 695–705.

6. McGuire A.M., Hughes J.D., Church G.M. Conservation of DNA Regulatory Motifs and Discovery of New Motifs in Microbial Genomes. *Genome Res.,* 2000, vol. 10, pp. 744–757.

7. Gelfand M.S. Recognition of Regulatory Sites by Genomic Comparison. *Res. Microbiol.,* 1999, vol. 150, pp. 755–771.

8. Gelfand M.S. Prediction of Function in DNA Sequence Analysis. *J. Comput. Biol.,* 1995, vol. 2, pp. 87–115.

9. Frech K., Quandt K., Werner T. Software for the Analysis of DNA Sequence Elements of Transcription. *Comput. Appl. Biosci.,* 1997, vol. 13, pp. 89–97.

10. Duret L., Bucher P. Searching for Regulatoy Elements in Human Noncoding Regions. *Curr. Opin. Struct. Biol.,* 1997, vol. 7, pp. 399–406.

11. Fickett J.W., Wasserman W.W. Discovery and Modeling of Transcriptional Regulatory Regions. *Curr. Opin. Biotechnol.,* 2000, vol. 11, pp. 19–24.

12. К.Ю. Горбунов и др. Об алгоритмах выявления регуляторного сигнала и построения эволюционного дерева. *Труды конференции "Проблемы управления и моделирования в сложных системах",* Самара, 2000, стр. 130–137.

13. Robison K., McGuire A.M., Church G.M. A Comprehensive Library of DNA-binding Site Matrices for 55 Proteins Applied to the Complete *Escherichia coli* K-12 Genome. *J. Mol. Biol.,* 1998, vol. 284, pp. 241–254.

14. Миронов А.А., Винокурова Н.П., Гельфанд М.С. Программное обеспечение анализа бактериальных геномов. *Молекулярная биология,* 2000, том 34, стр. 253–264.

15. Maas W.K. The Arginine Repressor of *Escherichia coli*. *Microbiol. Rev.,* 1994, vol. 58, pp. 631–640.

16. Busby S., Kolb A. The CAP Modulon. In: *Regulation of Gene Expression in Escherichia coli.* Lin E.C.C., Lynch A.S., Eds., E.G.Landes Co., 1995, ch. 12, pp. 255–279.

Статью представил к публикации член редколлегии Н.А. Кузнецов