МЕТОДЫ ОБРАБОТКИ СИМВОЛЬНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ И СИГНАЛОВ
(Вычислительные системы)
1989 год Выпуск 132

УДК 519.764

СЛОЖНОСТНЫЕ ПРОФИЛИ СИМВОЛЬНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

В.Д. Гусев

В в е д е н и е

В практике обработки сигналов и символьных последовательностей большое внимание уделяется не только получению интегральных характеристик этих последовательностей, но и выявлению их локальных свойств. Термин "локальный" обычно относят не ко всей последовательности, а к определенной ее части, выделяемой "окном" фиксированной длины (как правило, много меньшей, чем длина самой последовательности).

Режим обработки, при котором окно сдвигается вдоль последовательности с некоторым шагом, назовем режимом со скользящим окном (отсюда термины типа "скользящее среднее" и т.п.). При каждом положении окна будем вычислять некоторую числовую величину, характеризующую степень проявления определенного свойства δ во фрагменте, выделяемом окном. Упорядоченную совокупность таких величин, получаемых в режиме со скользящим окном, назовем профилем последовательности по свойству δ .

Выбор нужного свойства определяется содержательной стороной задачи. Данная работа в основном ориентирована на задачи дешифровочного типа, где в качестве исходных данных фигурируют слитные тексты с заранее неизвестными элементами структуры ("семантическими единицами"). Примерами таких текстов являются

первичные структуры ДНК-молекул и белков, нотные записи музыкальных произведений, последовательности слоев пород, проходимых при бурении скважин и т.п. Имеются достаточно веские основания для выбора в качестве свойства δ такой универсальной характеристики текста, как сложность. В первую очередь это объясняется тем, что многие функционально значимые структурные элементы в текстах характеризуются аномальным значением сложности, что и создает предпосылки для их формального обнаружения. При выборе подходящего определения сложности будем руководствоваться следующими соображениями, вытекающими из анализа специфики прикладных областей и результатов психологических измерений.

1. Определение сложности должно в явном виде апеллировать к понятию повтора, играющего фундаментальную роль в организации текстов различной языковой природы. Желательно учитывать весь спектр повторов, а не какой-то его отдельный срез.

2. Если последовательность построена по принципу периодического повторения какого-либо фрагмента, то сложность должна уменьшаться при уменьшении длины периода (т.е., к примеру, последовательность 01010101 должна быть менее сложной, чем 00110011).

3. Оба направления просмотра последовательности (слева-направо и справа-налево) должны быть равноправны. Это обеспечивает возможность обнаружения симметрий и сводящихся к ним структурных закономерностей.

4. Мера сложности должна учитывать наличие в тексте "синонимичных" фрагментов. Синонимия определяется по-разному для различных предметных областей. В простейшем случае два фрагмента можно считать синонимичными, если один получается из другого в результате простой перестановки ("переименования") элементов алфавита.

5. Определение сложности должно быть конструктивным, т.е. из него должен вытекать достаточно эффективный (даже для очень длинных последовательностей) алгоритм вычислений.

Требованиям 1,2,5 удовлетворяет определение сложности конечной последовательности, предложенное Лемпелем и Зивом [1]. Другое определение сложности, впервые вводимое в данной работе, является обобщением меры сложности из [1] в направлении учета требований 3 и 4.

Целью работы является описание и обоснование нового метода обнаружения структурных закономерностей в символьных последовательностях, в основу которого положено понятие сложностного профиля последовательности. Метод достаточно детально апробирован на генетических текстах [2,3] и двоичных последовательностях различной природы. Просматриваются возможности использования метода для анализа музыкальных произведений, текстов программ и т.п.

§1. Сложность конечной последовательности

Введем следующие обозначения: Σ - конечный алфавит; $|\Sigma|$ - число элементов алфавита; S - конечная последовательность, составленная из элементов Σ (текст); $N = |S|$ - длина S; $S[i]$ - элемент S, стоящий в i-й позиции (i-й по счету элемент S); $S[i:j]$ - фрагмент S, включающий элементы с i-го по j-й $(1 \le i \le j \le N)$; l-грамма - фрагмент текста, содержащий l подряд следующих символов; $S = QR$ - конкатенация (сцепление) последовательностей Q и R (если $|Q| = N_1$, $|R| = N_2$, то $|S| = N_1 + N_2$, причем $Q = S[1:N_1], R = S[N_1+1:N_1+N_2]$); $S = Q^k$ - последовательность, полученная k-кратным повторением фрагмента Q.

Пусть S - заданная последовательность. За меру ее сложности примем число шагов некоторого гипотетического процесса, по-

рождающего данную (известную заранее) последовательность. Предварительно зафиксируем множество допустимых ("порождающих") операций. Лемпель и Зив [1] предложили использовать в качестве "порождающих" операции генерации нового символа и копирования любого фрагмента из предыстории. Если по ходу процесса встречается символ, которого не было ранее, используется первая операция: при этом порождаемая последовательность удлиняется ровно на один элемент. Если же очередная цепочка символов встречалась ранее, ее можно скопировать: при этом порождаемая последовательность удлиняется не менее чем на один элемент. Таким образом, каждому шагу процесса ставится в соответствие фрагмент длиной от одного до нескольких символов. Историей формирования последовательности S назовем конкатенацию таких фрагментов:

$$H(S) = S[1:i_1]S[i_1+1:i_2]\ldots S[i_{k-1}+1:i_k]\ldots$$

$$\ldots S[i_{m-1}+1:N].$$

Здесь $S[i_{k-1}+1:i_k]$ - k-й компонент истории $(1 \le k \le m)$, $m = m_H(S)$ - число шагов процесса. Из всевозможных процессов порождения выбираем процесс с минимальным числом шагов. Это число будем рассматривать в качестве меры сложности последовательности S:

$$c_1(S) = \min_H \{m_H(S)\}. \qquad (1)$$

Минимальность числа шагов обеспечивается выбором для копирования при каждом применении этой операции такого фрагмента-прототипа из предыстории, который позволяет максимальным образом удлинить последовательность. Иначе говоря, если $S[i_{k-1}+1:i_k]$ - копируемый компонент истории $(k \in \{2, \ldots, m\})$, то его длина должна удовлетворять условию:

$$i_k - i_{k-1} = \max_{j \le i_{k-1}} \{l_j : S[i_{k-1}+1 : i_{k-1}+l_j] = S[j : j+l_j-1]\}. \quad (2)$$

Номер позиции $j = j(k)$, с которой начинается копирование k-го компонента истории, назовем <u>указателем копирования</u> этого компонента. Условимся полагать $j(k) = 0$, если в позиции $i_{k-1}+1$ стоит символ, который не встречался ранее (копирование невозможно, нужно использовать операцию генерации нового символа). С учетом этого соглашения k-й компонент истории может быть записан в виде:

$$S[i_{k-1}+1 : i_k] = \begin{cases} S[j(k) : j(k)+l_{j(k)}-1] & \text{при } j(k) \neq 0, \\ S[i_{k-1}+1] & \text{при } j(k) = 0. \end{cases} \quad (3)$$

Соотношения (2) и (3) определяют ту единственную историю формирования последовательности, которая обеспечивает минимум выражения (1). Обозначим ее $H_1^*(S)$. Тогда

$$c_1(S) = m_{H_1^*}(S).$$

ПРИМЕР 1 (генетический текст, $\Sigma = \{A,T,G,C\}$, фрагмент генома бактериофага λ).

```
Позиции: 1...5... 10...15...20
         |   |    |    |    |
     S = CGACGAGACGAAAAAACCGGA;

H*1(S) = C·G·A·CGA·GACGA·AAAAA·CG·GA;     c1(S) = 8;
         ↑↑↑ ↑    ↑     ↑     ↑  ↑
j(k)   = 0;0;0; 1;   2;    11;   1;  2.
```

Здесь компоненты истории $H_1^*(S)$ отделены друг от друга точками. Следует отметить, что шестой компонент истории $S[12:16]$ копируется с фрагмента $S[11:15]$, т.е. процесс копирования начинается с элемента, предшествующего формируемому компоненту, а заканчивается на самом формируемом компоненте.

Приведенное определение сложности незначительно отличается от предложенного Лемпелем и Зивом. В [1] каждый акт копирования сопровождается генерацией нового символа. В нашем определении процесс генерации используется только при появлении нового символа алфавита. Возникающие в связи с этим различия носят скорее технический характер, чем принципиальный.

§2. Примеры сложных и простых последовательностей

Представляет интерес оценить, насколько рассматриваемое определение соответствует интуитивному представлению о сложности и как оно соотносится с другими определениями сложности.

В табл. 1-3 приведены примеры последовательностей, сложность которых оценена тем или иным способом. Одновременно приводятся оценки их сложности по мере c_1. Точками выделены компоненты истории $H_1^*(S)$.

В табл.1 выписаны 9 двоичных последовательностей длины 8, для которых в [4] получены оценки избыточности. Чем более избыточна последовательность, тем менее она сложна. Наличие периодической компоненты внутри последовательности повышает ее избыточность, причем тем сильнее, чем меньше период этой компоненты. Оценки избыточности дополнены данными о временах задержки откликов испытуемых, которым предъявлялись указанные последовательности для выяснения трудности их восприятия.

Из анализа табл.1 видно, что сложность по мере c_1 хорошо коррелирует и с оценками избыточности, и со сложностью восприятия. Расхождения на последовательностях №4 и №8 объясня -

Таблица 1

Оценка избыточности двоичных последовательностей и сложности восприятия их человеком

№	Последовательность	Оценка избыточности	Задержка отклика	Сложность по мере c_1
1	1·0·101010	0,875	11,5	3
2	1·0·0·1·1001	0,754	13	5
3	1·1·0·0·1100	0,754	13	5
4	1·111·0·000	0,709	18	4
5	1·1·0·10·00·1	0,671	21,4	6
6	1·1·0·000·11	0,66	18	5
7	1·0·00·10·1·1	0,629	21,4	6
8	1·11·0·000·1	0,619	18	5
9	1·0·1·101·0·0	0,619	103,9	6

Таблица 2

Данные о времени передачи информации муравьям

№	Путь по дереву	t_n	c_1	№	Путь по дереву	t_n	c_1
1	Л·ЛЛ	72	2	9	Л·Л·П	69	3
2	П·ПП	75	2	10	Л·П·Л·Л	100	4
3	Л·ЛЛЛЛ	84	2	11	П·Л·ЛЛ·П	120	4
4	П·ПППП	78	2	12	П·П·Л·ПЛ	150	4
5	Л·ЛЛЛЛЛ	90	2	13	П·Л·П·ПП·Л	180	5
6	П·ПППП	88	2	14	П·П·Л·ПП·П	220	5
7	Л·П·ЛПЛП	130	3	15	Л·П·Л·ЛПЛ	200	4
8	П·Л·ПЛПЛ	135	3				

Таблица 3

Примеры "неслучайных" (№ 1-4) и "случайных" (№ 5-8), по Кнуту, последовательностей

№	Последовательность	c_1
1	0•000000•1•111	4
2	0•1•010101010	3
3	0•000000•1•11•0	5
4	1•0•000000•1•10	5
5	0•0•1•00100100	4
6	0•00000•1•1111	4
7	1•0•1•11•0•01•00•0	8
8	1•1•0•0•10•11•110	7

ются теми особенностями процедуры копирования, которые пояснены в примере 1.

В табл.2 помещены данные о языке муравьев [5]. Последовательности 1-15 характеризуют путь в двоичном дереве-лабиринте от корня до узла, содержащего кормушку. Муравей, добравшийся до кормушки, передает информацию о ее местонахождении другим муравьям. Время передачи (t_n) измеряется и служит оценкой меры сложности пути (последовательности поворотов "левый - правый") по дереву.

Анализ табл.2 вновь демонстрирует хорошее соответствие между изменениями значений параметров t_n и c_1. Основное отличие заключается в том, что мера c_1 не реагирует на длину серий из однородных элементов (см.последовательности №1,3,5 или № 2,4,6), в то время как муравьи затрачивают определенное время на передачу информации о длине серии. Если ввести запрет на возможность копирования элементов из формируемого в данный момент компонента истории (см.пример 1), каждая серия будет представлена в $H_1^*(S)$ числом компонентов, логарифмически зависящим от ее длины.

В табл.3 приведены примеры "неслучайных" (см. [6, с.184]) и "случайных" двоичных последовательностей длины $N = 11$. "Случайными" считаются последовательности, удовлетворяющие свойству k-распределенности (равномерной распределенности частот k-грамм, $k \leq \log_2 N$), а "неслучайными" - последовательно-

сти с аномально высокой частотой вхождения какой-либо l-граммы. Так, при $N = 11$ к "неслучайным" будут отнесены все последовательности, содержащие 3-грамму с частотой $F \geq 5$ (таковыми, в частности, являются двоичные слова с сериями нулей или единиц длины, не меньшей чем 7).

Значение c_1 для "неслучайных" последовательностей меняется от 2 (для $S = 0^{11}$ или $S = 1^{11}$) до 5; для "случайных" - от 4 до 8. Слова № 5 и 6 удовлетворяют свойству k-распределенности, но по мере c_1 их следовало бы отнести к "неслучайным". С интуитивной точки зрения предпочтение в этой ситуации следовало бы отдать мере c_1 : в первом случае (слово №5) она реагирует на периодичность $(001)^3$, во втором (слово № 6) - на длинные серии нулей и единиц.

Рассмотренные примеры показывают, что мера c_1 достаточно хорошо согласуется и с интуитивными, и с формальными представлениями о сложности конечной последовательности.

§3. Алгоритм вычисления меры сложности

Вычисление меры c_1 требует знания полного спектра повторов различной длины, содержащихся в анализируемом тексте. Можно выделить два основных направления в технике получения таких спектров: хеширование [7] и применение конструкций типа "префиксное дерево", "суффиксное дерево", "направленный ациклический граф слова" [8-10]. Сопоставление этих двух направлений представляет самостоятельный интерес и не является целью данной работы. Ниже описан алгоритм, основанный на идеях хеширования*). К достоинствам его следует отнести логическую просто-

*) Этот алгоритм был реализован под руководством автора в дипломной работе студентки механико-математического факультета НГУ Запрягаевой Е.В. (1985 г.). В [1] алгоритм вычисления меры сложности не приводится.

ту и возможность использования для текстов большой длины (сопоставимой с объемом оперативной памяти). Однако по трудоемкости он несколько проигрывает алгоритму, где за основу берется префиксное дерево [11].

Обозначим i-ю по порядку следования l-грамму текста S через x_{il} ($x_{il} = S[i:i+l-1]$, $1 \le i \le N-l+1$). Фактически речь идет о нахождении для каждой позиции i максимально возможного значения l, такого, что $x_{il} = x_{jl}$ и $j<i$. Параметр l фиксирует длину копируемого участка, а параметр j является указателем копирования. Вектор значений l_i для позиций $1 \le i \le N$ обозначим через L, а вектор значений j_i - через J.

Кроме указанных векторов, введем еще две стандартные для процедур хеширования [7] структуры данных: основное расстановочное поле X_0 и дополнительное расстановочное поле $X_д$. Это упорядоченные множества записей, каждая из которых состоит из двух полей. В первом поле X_0 записывается информация о первом вхождении в S той l-граммы, за которой закреплена данная запись. Во втором поле содержится отсылка на следующий элемент списка l-грамм с тем же хеш-адресом. Второй и последующий элементы списка содержатся уже в $X_д$.

Векторы L и J определяются итеративно. На l-й итерации ($1 \le l \le l_{max}$, где l_{max} - длина максимального повтора в тексте S) фиксируются все повторы длины l и с помощью получаемой информации осуществляется текущая коррекция векторов L и J. Для единообразия вычисления компонент этих векторов с номерами $i > N-l+1$ (соответствующие позиции не могут служить началом l-грамм ввиду ограниченности длины текста) текст S дополняется пустым символом λ ($S[N+1] = \lambda$). Поскольку $\lambda \notin \Sigma$, любая l-грамма, заканчивающаяся этим символом, будет иметь единичную частоту, что обеспечивает корректность определения l_i и j_i для концевых элементов.

Шаг 0. Задание начальных значений элементов массивов L, X_0 и $X_д$. Полагаем $L[i]=\gamma$, $\gamma \notin [0,N+1]$, $1 \le i \le N+1$. Относительно компонент массивов X_0 и $X_д$ будем предполагать, что они принимают нулевые значения перед началом каждой итерации.

Шаг 1. Первая итерация $(l = 1)$. Просматриваем все элементы S, выделяем позиции $i_1, i_2, \ldots, i_k = N+1$ $(k \le |\Sigma| + 1)$, соответствующие первым вхождениям различных элементов алфавита в текст, корректируем вектор L и формируем вектор J:

$$L[i_1] = L[i_2] = \ldots = L[i_k] = 1;$$

$$J[i] = \begin{cases} 0, & \text{если } i \in \{i_1, i_2, \ldots, i_k\}, \\ i_m, \ 1 \le m < k, & \text{если } S[i] = S[i_m] \\ & \text{и } i \notin \{i_1, i_2, \ldots, i_k\}. \end{cases}$$

Шаг 2. Итерации с номерами $l = 2,3\ldots, l_{max}$.

На l-й итерации просматриваем текст S окном ширины l, сдвигаясь каждый раз от начала к концу на один символ. Для l-граммы x_{i1} вычисляем хеш-адрес $h(x_{i1})$ (см., например, [7]) и анализируем содержимое элемента массива X_0 с адресом $h(x_{i1})$. Если данная l-грамма уже встречалась в S раньше (а информация об этом как раз и содержится по указанному адресу в X_0 либо в связанном с данным адресом списке наложений из $X_д$), то значение $L[i]$ еще не определено: мы знаем, что существует $x_{j1} = x_{i1}$, где $j < i$, но пока неясно, является ли потенциально допустимый для копирования участок $S[j:j+l-1]$ максимальным из возможных для i-й позиции. Этот вопрос будет решен на последующих итерациях. Пока же осуществляется коррекция соответствующего элемента вектора J: $J[i] = j$, где j - позиция, указанная в записи из X_0 с

номером $h(x_{i1})$ или в связанном с ней списке наложений из $X_д$.

Если 1-грамма x_{i1} встречается впервые, т.е. список с заголовком, находящимся по адресу $h(x_{i1})$, пуст либо не содержит x_{i1}, то информация об x_{i1} заносится соответственно в X_0 либо в $X_д$. Далее проверяется значение $L[i]$. Если $L[i] = \gamma$, полагаем $L[i] = l-1$ и сохраняем значение $J[i]$. Можно утверждать, что i-е компоненты векторов L и J определены окончательно.

Итерации продолжаются до тех пор, пока все компоненты вектора L не будут определены, т.е. не останется значений, равных γ. Ниже приведен пример изменения векторов L и J в процессе итераций для последовательности S = ATATGGCATGTTAλ:

	Матрица значений L														Матрица значений J													
	A	T	A	T	G	G	C	A	T	G	T	T	A	λ	A	T	A	T	G	G	C	A	T	G	T	T	A	λ
l=0	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ														
l=1	1	1	γ	γ	1	γ	1	γ	γ	γ	γ	γ	γ	1	0	0	1	2	0	5	0	1	2	5	2	2	1	0
l=2	1	1	γ	1	1	1	1	γ	γ	1	1	γ	1	1	0	0	1	2	0	5	0	1	4	5	2	2	1	0
l=3	1	1	2	1	1	1	1	γ	2	1	1	2	1	1	0	0	1	2	0	5	0	3	4	5	2	2	1	0
l=4	1	1	2	1	1	1	1	3	2	1	1	2	1	1	0	0	1	2	0	5	0	3	4	5	2	2	1	0

Шаг 3. Разбиение S на компоненты сложности с использованием вектора L.

Пусть выделены первые $k-1$ компонентов истории $H_1^*(S)$. Тем самым определено начало k-го компонента - позиция $i_{k-1}+1$. Определяем по вектору L длину k-го компонента $L[i_{k-1}+1]$, вычисляем начало $(k+1)$-го компонента $i_k+1 = i_{k-1}+1 + L[i_{k-1}+1]$ и повторяем процесс.

Корректность описанного алгоритма вытекает из следующих соображений.

Этап $l = 1$ очевиден: впервые появляющиеся в S элементы алфавита (при движении от начала к концу) всегда образуют

отдельные компоненты истории длины l (используется операция генерации символа). Поэтому соответствующие компоненты вектора L принимают значение 1, а аналогичные компоненты вектора J - значение 0. Остальные элементы S будут получаться копированием с уже порожденных элементов, но длины участков копирования еще не определены. Указатели копирования определяют, с какого из порожденных элементов могут быть скопированы остальные, если на следующей итерации выяснится, что для некоторых i $L[i] = 1$.

Рассмотрим теперь l-ю итерацию $(l \geq 2)$. Если $L[i] \neq \gamma$, это означает, что на предыдущих итерациях длина участка копирования для i-й позиции уже выявлена и измениться она не может. Если $L[i] = \gamma$, это означает, что по ходу $(l-1)$-й итерации $L[i]$ еще не было определено, хотя для $(l-1)$-граммы $x_{i(l-1)}$ существует как минимум один прототип $x_{j(l-1)}$ такой, что $x_{i(l-1)} = x_{j(l-1)}$ и $j < i$. Ввиду этого можно утверждать, что $L[i] \geq l-1$.

На l-й итерации решается вопрос, может ли действительно $L[i]$ превысить $l-1$. Если l-грамма x_{il} совпала с x_{jl} (или с каким-либо другим прототипом $x_{j'l}$, таким, что $j < j' < i$), то да. Значение $L[i]$ сохраняется $(L[i] = \gamma)$. В противном случае l-грамма x_{il} появляется впервые, она не может быть скопирована ни с одного из предшествующих участков, т.е. $L[i] < l$. Сопоставляя с предыдущим выводом относительно $L[i]$, получаем $L[i] = l-1$, а текущее значение указателя копирования после $(l-1)$-й итерации уже указывает нам позицию j такую, что $x_{i(l-1)} = x_{j(l-1)}$ и $j < i$.

Длина максимального повтора ограничена длиной текста $(l_{max} \leq N-1)$. Поскольку с ростом l фиксируются повторы все большей длины, на некотором этапе все компоненты в L станут отличными от γ , т.е. все участки копирования будут найдены.

Оценим трудоемкость алгоритма в среднем для случайных последовательностей (таковыми в первом приближении можно считать генетические тексты). Число итераций определяется длиной максимального повтора. Для случайных последовательностей с равновероятной встречаемостью элементов алфавита $(p = 1/|\Sigma|)$ средняя длина максимального повтора составляет $O(\ln N/\ln|\Sigma|)$ [12]. На каждой итерации $(N-l+2)$ раза вычисляется функция расстановки $h(x_{i1})$ и столько же раз осуществляется поиск по списку наложений со средней трудоемкостью, равной $1+\alpha/2$, где α - коэффициент загрузки основного расстановочного поля X_0 (как правило, $\alpha < 1$ [7]). Отсюда трудоемкость алгоритма в среднем составляет $O(N \cdot \frac{\ln N}{\ln|\Sigma|})$ операций типа вычисления функции расстановки.

ЗАМЕЧАНИЕ 1. Трудоемкость вычисления функции расстановки можно сделать близкой к константе. Зависимость от l нивелируется использованием рекуррентной схемы хеширования.

ЗАМЕЧАНИЕ 2. Число итераций можно уменьшить, отказавшись при некотором значении $l < l_{max}$ от хеширования. Это оправдано, когда число недоопределенных компонент вектора L (т.е. таких, что $L[i] = \gamma$) уже невелико. Дальнейшие вычисления сводятся к расширению копируемого участка и его возможных прототипов, информация о которых хранится в X_0 и $X_д$ до тех пор, пока они не начнут отличаться.

4. Обобщение меры сложности c_1

Целесообразность обобщения меры c_1 возникает в приложениях, где появляются новые типы допустимых операций (например, операция "поступенного" заполнения интервалов в музыкальных текстах), проводится анализ последовательности как в прямом, так и в обратном направлениях (генетические тексты, доде-кафонная музыка), используется расширенная трактовка понятия

повтора. Проиллюстрируем один из возможных вариантов обобщения меры c_1, учитывающий указанные моменты.

Пусть $X = x_1 x_2 \ldots x_N$ - последовательность, составленная из элементов Σ; $X^R = x_N x_{N-1} \ldots x_1$ - та же последовательность, прочитанная в обратном направлении; $f : \Sigma \to \Sigma$ - взаимно однозначное отображение алфавита Σ самого на себя ("переименование" элементов алфавита); $f(X) = f(x_1)f(x_2) \ldots \ldots f(x_N)$ - фрагмент X, элементы которого переименованы в соответствии с f: $(f(X))^R = f(x_N) \cdots f(x_2)f(x_1)$ - результат последовательного применения преобразований f и R к X.

Если X_1 и X_2 - два произвольных фрагмента текста S, будем называть пару (X_1, X_2) прямым повтором, если $X_1 = X_2$; симметричным повтором, если $X_2 = X_1^R$; прямым f-повтором, если $X_2 = f(X_1)$; симметричным f-повтором, если $X_2 = (f(X_1))^R$. Введем следующие допустимые операции: а) операцию генерации нового символа; б) 4 типа операций копирования, каждая из которых ведет к образованию одного из перечисленных выше типов повторов. Факт наличия повтора типа "p" (p = 1-4), образуемого двумя фрагментами длины l, начинающимися в позициях i и j, условимся записывать в виде: $S[i:i+l-1] = S^{(p)}[j:j+l-1]$. Тогда аналог формулы (2) для длины k-го копируемого компонента истории будет выглядеть следующим образом:

$$i_k - i_{k-1} = \max_{j \le i_{k-1}} \{ \max_p l_j^{(p)} : S[i_{k-1}+1 : i_{k-1}+l_j^{(p)}] = S[j : j+l_j^{(p)}-1] \}. \qquad (4)$$

Соответственно сам k-й компонент может быть записан в виде:

$$S[i_{k-1}+1:i_k] = \begin{cases} S[j(k):j(k)+1^{(p(k))}_{j(k)}-1] & \text{при } j(k) \neq 0, \\ S[i_{k-1}+1] & \text{при } j(k) = 0, \end{cases} \quad (5)$$

где $j(k)$ и $p(k)$ - значения параметров j и p, обеспечивающие максимум выражения (4). Историю формирования S в соответствии с (4) и (5) обозначим $H_2^*(S)$. Тогда

$$c_2(S) = m_{H_2^*}(S). \quad (6)$$

Продемонстрируем возможность использования меры (6) для анализа генетических текстов. Отображение f в данном случае фиксирует возможность комплементарного взаимодействия нуклеотидов: $f(A) = T$, $f(T) = A$, $f(G) = C$, $f(C) = G$. При этом наличие симметричных f-повторов может свидетельствовать о существовании палиндромно-шпилечных структур, несущих важную функциональную нагрузку, а прямые f-повторы, возможно, соответствуют участкам так называемой "параллельной ДНК" [13]. Поскольку существование последних проблематично, ограничимся частным случаем меры (6) – $c_2^!$, исключив операцию копирования, приводящую к образованию прямых f-повторов.

ПРИМЕР 2 (геном вируса гриппа А, штамм PR, сегмент полимеразы 1, поз. 1411, N = 30)

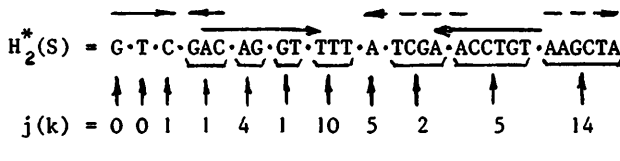
S = GTCGACAGGTTTTATCGAACCTGTAAGCTA;

поз. 1 5 10 15 20 25 30

$H_1^*(S)$ = G·T·C·G·A·C·A·G·GT·TTT·A·TCGA·AC·C·T·GT·AA·G·CT·A;

j(k) = 0 0 0 1 0 3 5 1 1 10 5 2 5 3 2 1 18 1 21 5;

$c_1(S)$ = 20;



$c'_2(S) = 11.$

Здесь $p = 0$ соответствует генерации нового символа, $p = 1$ - прямому повтору, $p = 2$ - симметричному повтору, $p = 4$ - симметричному f-повтору. Отметим,что новый элемент алфавита теперь может быть получен не только при помощи операции генерации, но (иногда) и с помощью операций переименования и копирования (см. компоненты №3 и 4 истории $H^*_2(S)$). Более того, вначале делается попытка получить очередной компонент истории с помощью всевозможных операций копирования (при этом длина компонента может оказаться больше единицы - см., например,компонент №4) и лишь при неудаче используется операция генерации.

Сопоставление $H^*_1(S)$ и $H^*_2(S)$ в приведенном примере показывает, что по мере c_1 фрагмент S является достаточно сложным, тогда как по мере c_2 он имеет аномально низкую сложность из-за наличия шпилечной структуры (см.стрелки сверху, непрерывная линия) и симметричного повтора (пунктирные стрелки).

5. Сложностной профиль текста

Сложность текста в целом представляет определенный интерес с позиций классификации, однако эта интегральная характеристика мало что дает в плане выявления локальных структурных особенностей текста. Гораздо бо́льшую информацию о тексте можно получить в режиме обработки со скользящим окном размера D.

Сложностным профилем текста S (или профилем текста по мере C) назовем последовательность значений

$$P(S,D) = c_1 c_2 \dots c_{N-D+1}, \qquad (7)$$

где c_i - сложность фрагмента $S[i:i+D-1]$, $1 \le i \le N-D+1$, N - длина S. При больших N эта информация может оказаться уже избыточной и возникает проблема ее компактизации. Отметим три возможных подхода к получению компактного представления профиля.

Первый подход связан с введением небольшого числа градаций сложности. Можно, например, условно разбить все фрагменты на простые (п), средней сложности (с) и большой сложности (б), выделить на профиле однородные зоны и закодировать каждую из них серией элементов типа "п", "с" или "б", длина которой пропорциональна размеру зоны, отнесенному к размеру окна анализа. Такое представление обеспечивает сжатие информации примерно в D раз и сохраняет картину распределения простых и сложных участков по длине текста.

Второй подход предполагает построение гистограммы значений c_i. Гистограммы разных текстов удобно сравнивать визуально. Они могут различаться по таким параметрам, как $c_{min} = \min_i c_i$, $1 \le i \le N-D+1$, $c_{max} = \max_i c_i$, $\bar{c} = (\sum_i c_i)/(N-D+1)$, высота, степень асимметрии и т.п. Поскольку гистограммы можно строить для разных значений D, получаем большой набор наглядно трактуемых классификационных параметров.

Третий подход связан с представлением профиля в виде последовательности экстремальных значений. Будем считать значение сложности экстремальным (аномальным), если

$$|c - \bar{c}| \ge 3s, \qquad (8)$$

где c - любое значение из последовательности (7), $\bar{c} = (\sum_i c_i)/(N-D+1)$ - среднее значение сложности всех фрагментов текста, $s^2 = \sum_i (c_i - \bar{c})^2/(N-D)$ - несмещенная оценка дис-

персии значений сложности. Для унимодальных распределений вероятность выхода значений сложности за указанные выше границы мала (для нормальных - очень мала). Основной интерес для приложений представляют аномальные фрагменты с минимальной сложностью, поскольку именно они определяют многие существенные структурные особенности текстов. Более того, суперсложные длинные фрагменты требуют специального конструирования, и они нетипичны даже для случайных текстов.

Принципиальным моментом (применительно к длинным текстам) является наличие быстрого алгоритма вычисления сложностного профиля. Эффективность используемого нами алгоритма [11] обусловлена тем, что при переходе от i-го к $(i+1)$-му фрагменту сложность не пересчитывается заново, а лишь корректируется (используется "зацепленность" соседних фрагментов). Нетривиальность коррекции связана с тем, что при сдвиге окна изменяются не только начальный и конечный компоненты истории, но и некоторые внутренние. Трудоемкость алгоритма вычисления сложностного профиля составляет $O(N \cdot |\Sigma| \cdot \log D/\Sigma)$, дополнительные затраты памяти — $O(D)$.

Размеры аномальных по сложности зон могут меняться в широком диапазоне. Укажем два подхода к проблеме автоматического выявления размера аномальных зон.

Первый подход предполагает m-кратную обработку текста с помощью набора окон увеличивающейся длины $(D_1 < D_2 < \ldots < D_m)$. При этом для получения достаточно высокой разрешающей способности диапазон $D_1 - D_m$ должен покрываться большим числом окон, что невыгодно с вычислительной точки зрения. Если же m мало, всегда существует опасность, что при размере окна D_i фрагмент еще не выделяется как аномальный, а при размере $D_{i+1} > D_i$ - уже не выделяется как аномальный.

Во втором подходе фиксируются лишь границы интересующего диапазона: $D_{min} - D_{max}$. Анализ ведется при $D = D_{max}$,

но по ходу накапливается вся необходимая информация для принятия решения относительно любого другого окна размера $D_{min} < D < D_{max}$. Поскольку число возможных окон равно $D_{max} - D_{min} + 1$, формируются векторы значений $\bar{c}$ и s указанной размерности, которые корректируются при каждом сдвиге. Основой для реализации такого подхода являются две предпосылки: 1) компоненты истории $H^*(S,D)$, полученные для окна размера D, составляют начальный отрезок истории $H^*(S,D')$ для любого другого окна с размером $D' > D$, являющегося расширением (вправо) исходного; 2) параметр $\bar{c}$ и,что особенно важно в смысле экономии памяти, параметр s могут вычисляться рекуррентно для каждого значения D.

Главной привлекательной особенностью описанной методики является возможность выявления в рамках единого подхода широкого класса различных структурных закономерностей. Проиллюстрируем на примере генетических текстов [3] наиболее интересные из них.

6. Классификация структурных закономерностей

В приводимых ниже примерах используется следующая система обозначений:

- аномальный фрагмент выделяется из контекста двумя вертикальными чертами; компоненты сложности внутри него разделены точками;

- надчеркиванием и подчеркиванием выделяются наиболее характерные участки аномального фрагмента (короткие периодичности, повторы, функционально значимые зоны в знаках пунктуации, осуществляющих управление основными генетическими процессами); более длинные периодичности и повторы выделяются скобками;

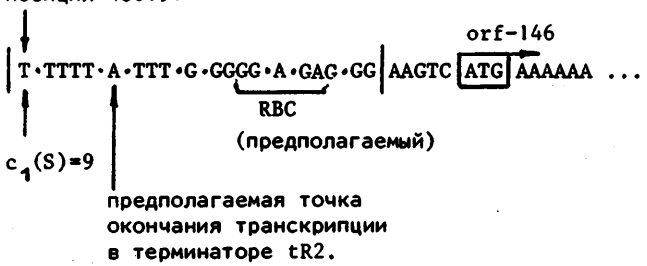
- палиндромно-шпилечные структуры указываются стрелками сверху, направленными навстречу друг другу; симметрии - стрелками, направленными в противоположные стороны.

К числу наиболее характерных закономерностей относятся:

а) серии повторяющихся однородных фрагментов (периодичности). Если повторяющийся фрагмент состоит из одного элемента $a \in \Sigma$, такие серии часто называют поли-а-участками.

ПРИМЕР 3 (геном бактериофага λ, $D = 20$)

Позиция 40619:

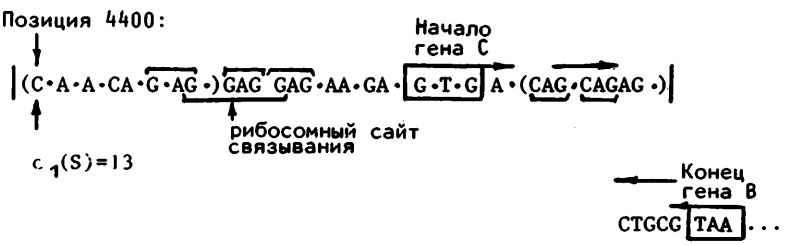


В приведенном примере имеем ярко выраженные поли-Т-, поли-G- и поли-А-участки. Фрагмент насыщен управляющими элементами. Поли-Т-область соответствует предполагаемому терминатору транскрипции tR2, поли-G-область - рибосомному сайту связывания (RBS) для предполагаемого гена (orf-146), поли-А-область характерна для начала многих генов. Корреляция в расположении аномальных фрагментов и знаков пунктуации - одна из важных предпосылок для использования сложностных профилей при анализе генетических текстов.

Серии из повторяющихся фрагментов длины 2 и 3 часто входят в состав различных знаков пунктуации: в сайты рестрикции (CG,CG , AT,AT), энхансеры (TG,TG,G), сайты рекомбинации (GC TGG,TGG-фаг λ) и т.п.

ПРИМЕР 4 (геном бактериофага λ, $D = 30$)

Позиция 4400:



Рибосомный сайт связывания в данном примере образован трехкратным повторением фрагмента GAG.

Серии из повторяющихся фрагментов большей длины уже, как правило, зашумлены (наблюдаются замены символов, а иногда - короткие вставки и делеции). Длинные периодичности связаны часто с дублированием знаков пунктуации либо с проявлением регуляр - ности на уровне белков, кодируемых ДНК.

б) Аномально низкая частота использования отдельных элементов алфавита.

ПРИМЕР 5 (геном бактериофага λ , $D = 20$)

Позиция 44925

$$H_1^*(S) = |C \cdot CC \cdot A \cdot G \cdot CA \cdot A \cdot CAGCA \cdot CAAC \cdot CCA \cdot | AAC...$$

$$c_1(S) = 9$$

В этом C,A-богатом фрагменте почти не представлены элементы алфавита T и G . Столь сильная неравномерность в нуклеотидном составе часто сопровождается повторами.

в) Резкое преобладание частоты какой-либо 1-граммы над остальными. Сама 1-грамма может не образовывать периодичности и зачастую диспергирована по всей длине фрагмента. Выявление закономерностей подобного рода говорит о том,что мера c_1 достаточно чувствительна к нарушению свойства k-распределенности (см. п.2).

ПРИМЕР 6 (геном вируса гриппа А, штамм PR , сегмент полимеразы 2, $D = 30$)

Позиция 606:

$$H_1^*(S) = |C \cdot G \cdot A \cdot GA\,GAG \cdot GAGA \cdot AGAGA \cdot C \cdot AA \cdot T \cdot T \cdot GAAGA \cdot AAG| ...$$

$$c_2(S) = 12$$

Частота вхождения биграммы GA в аномальный фрагмент равна 9 (пурино-богатый тракт). Функциональная значимость пуриновых

трактов отмечалась неоднократно, в частности, их широкое распространение в знаках пунктуации. Интересно отметить,что незначительные пиримидиновые вкрапления (C и T) фигурируют лишь в составе комплементарного палиндрома (CAATTG).

г) Разнесенные повторы. Образцом последних могут служить фрагменты из примера 4, заключенные в круглые скобки. Разнесенные повторы иногда выступают в качестве сохранившихся в процессе эволюции фрагментов (ядер), входящих в состав длинных "зашумленных" периодичностей. В других случаях разнесенные повторы могут фланкировать функционально значимый фрагмент-вставку или, наоборот, образовывать вставку в составе функционально значимого фрагмента.

ПРИМЕР 7 (геном вируса гриппа А, штамм PR, сегмент полимеразы 1, $D = 50$, мера c_2').

Позиция 834

$$H_2^*(S) = |G\cdot T\cdot TG\cdot C\cdot \underset{1}{CA}\cdot GTTG\cdot \underset{1}{GA}\cdot GGCAA\cdot TGA\cdot GA\cdot AGAA\cdot A(G\cdot \underset{3}{CAA}\cdot$$

$$\underset{1}{AGTTG})(G\cdot \underset{3}{CAAA}\cdot TG\cdot TTG\cdot)TAA|\ldots;\ c_2'(S) = 18$$

Здесь разнесенный повтор образован двумя вхождениями фрагмента 1 (AGTTGG). Первое из них участвует в формировании петли в шпилечной структуре, задаваемой симметричным f-повтором (2). Второе - фланкировано фрагментами CAA, образующими прямой повтор (3).

д) Симметричные повторы и f-повторы. Примеры 2 и 7 демонстрируют обе возможности. Роль палиндромно-шпилечных структур в регуляции основных генетических процессов обсуждалась неоднократно и не вызывает сомнений.

Сделаем два замечания в связи с приведенной классификацией.

1. Как правило, описанные выше элементарные закономерности встречаются не по отдельности, а в комбинации друг с другом.

2. Введение симметричного и f-повторов, а также меры c_2 ориентировано не только на генетические приложения. К примеру, понятие f-повтора может быть использовано для выявления такого метода варьирования в музыкальных произведениях, как перенос музыкального фрагмента (по высоте) на несколько ступеней вверх или вниз. Понятие симметричного повтора может быть использовано для выявления схемы построения мелодии, получившей название "принцип дополняющего ответа" [14]. Аналогичные примеры можно привести и для других языковых систем.

7. Возможности использования

7.1. Сжатие текстов. Мера сложности, введенная в [1], ориентирована именно на этот традиционный класс приложений. Мы не будем останавливаться на данном вопросе, поскольку он не имеет прямого отношения к сложностному профилю. Упомянем лишь, что использование меры c_2 в определенных ситуациях может способствовать повышению коэффициента сжатия текста.

7.2. Классификация текстов. Сопоставлять тексты непосредственно по значениям c_1 или c_2 бессмысленно, поскольку эти величины зависят от длин текстов. Ничего не дает и нормирование c_1 и c_2 к длине текста, поскольку число компонентов истории $H^*(S)$ не является линейной функцией от N.

В связи с этим представляет интерес такая характеристика профиля, как $\bar{c} = (\sum_i c_i)/(N-D+1)$. В эксперименте с 30 геномами разной длины [2] выявились два обнадеживающих фактора: а) значения $\bar{c}$ для родственных геномов оказались близки (в смысле правила "ближайшего соседа"), несмотря на существенное (порой) различие в длинах; б) указанный эффект, как правило,

устойчиво наблюдался при всех значениях D (размер окна менялся в диапазоне от 20 до 150 символов).

7.3. Выявление функционально значимых фрагментов текста. Выше уже упоминалось о корреляции в расположении минимальных по сложности фрагментов и знаков пунктуации. Максимальные по сложности фрагменты также представляют интерес в ряде приложений (например,в теории связи).В текстах музыкальных произведений основная эмоциональная нагрузка также ложится на фрагменты максимальной сложности.

Некоторые тексты являются конкатенацией двух, трех и более похожих фрагментов (димер, тример и т.д.). Когда размер повторяющегося фрагмента сопоставим с размером текста и велика степень зашумления (вставки, замены, делеции), обнаружить подобную "k-мерность" ($k = 2,3,...$) в структуре текста довольно трудно. Для этой цели можно использовать сложностные профили с большим размером окна ($D \sim N/k$, $k = 1,2,3$).

Низкочастотная фильтрация сложностного профиля также может выявить функционально значимые зоны в тексте: участки типа "плато", переходные фрагменты, участки с колебательным характером изменения сложности и т.п.

7.4. Выявление гомологий по структурному сходству. Задача поиска гомологичных (похожих) фрагментов в больших текстах является достаточно трудной в вычислительном отношении. Разработанные для этой цели алгоритмы динамического программирования имеют трудоемкость $O(N^2)$ и,что более критично, требуют квадратичных же затрат памяти. Сложностные профили не ориентированы на решение этой задачи в полном объеме, однако с их помощью удается получать интересные частные решения без больших вычислительных затрат.

Суть подхода заключается в следующем. Выделяются фрагменты с минимальной сложностью, удовлетворяющие соотношению (8) или более "мягкому" критерию. Каждый из них характеризуется оп-

ределенной структурной закономерностью. Далее сравниваются друг с другом (и расширяются по мере необходимости в обе стороны) лишь фрагменты с одинаковыми структурными особенностями. Поскольку число фрагментов с фиксированной структурной особенностью (например, с периодичностью $(CAG)^3$ или явно выраженным CT-преобладанием и т.п.), как правило, невелико, возможно сопоставление их друг с другом любым методом, включая и метод динамического программирования. Примеры использования такого подхода приведены в [3].

7.5. Поиск фрагментов, тождественных с точностью до переименования элементов алфавита. В определении меры c_2 фигурировало априорно задаваемое отображение f. Обычно оно определяется исходя из специфики предметной области, но всегда остается вопрос: а не существует ли другого варианта переименования элементов алфавита, позволяющего выявить какие-то характерные структурные особенности текста? Для ответа на этот вопрос разработан алгоритм поиска всевозможных f-повторов, где вид отображения (способ переименования) не фиксируется заранее.

В основу алгоритма положено следующее соображение: если взаимно однозначным образом переименовать элементы алфавита, то сложность любого фрагмента не изменится. Более того, сохранятся длины компонентов истории и указатели копирования. Это дает возможность осуществить иерархическую таксономию всех фрагментов текста по указанным параметрам и тем самым быстро получить все f-повторы фиксированной длины D.

ПРИМЕР 8.

а) Аналог комплементарному палиндрому с другим типом "комплементарности" (A → C, C → A, T → G, G → T), геном ASVY73, D = 18, позиция 3320:

$$S = \overrightarrow{CTTCGATGA}\,|\,\overleftarrow{CTGCTAGGA}$$

Легко убедиться, что $S = (f(S))^R$, где f - указанное выше отображение;

б) прямой f-повтор длины 15, где f-отображение, заменяющее C на T, а T на C (геном HBVADYW):

Позиция 2078: GCTTTCCCCCATTGT,

Позиция 2152: GTCCCTTTTTACCGC.

Неизвестно, соответствует ли указанным структурным особенностям какая-либо функциональная нагрузка. Отметим, однако, что в частном случае с помощью данной методики можно выявлять и обычные шпилечно-палиндромные структуры, функциональная значимость которых не вызывает сомнений.

З а к л ю ч е н и е

Предложен и эмпирически обоснован новый метод обнаружения структурных закономерностей в символьных последовательностях большой длины. В его основу положено понятие сложностного профиля последовательности. Используемые в работе определения сложности апеллируют к понятию повтора, играющего фундаментальную роль в организации текстов различной природы. Апробация метода на генетических текстах и двоичных последовательностях, возникающих в теории связи, позволяет характеризовать его как весьма универсальный инструмент отыскания закономерностей локального типа, хотя довольно часто с его помощью удается обнаруживать ассоциативные связи и между сколь угодно отдаленными фрагментами.

Основные результаты:

- введена мера сложности конечной последовательности, в которой использована обобщенная трактовка понятия повтора;

- разработан алгоритм вычисления меры сложности, основанный на идеях хеширования;

- предложена и экспериментально подтверждена методика выявления структурных закономерностей с помощью понятия сложностного профиля текста;

- проведена классификация структурных закономерностей, выделяемых с помощью сложностного профиля;

- указаны возможности нестандартного использования сложностного профиля в задачах анализа слитного неструктурированного текста.

Литература

1. LEMPEL A., ZIV J. On the Complexity of Finite Sequences //IEEE Trans. on Inf.Th. - 1976. - Vol.IT-22,N 1. - P. 75-81.

2. ГУСЕВ В.Д., КУЛИЧКОВ В.А., ЧУПАХИНА О.М. Сложностные профили - новый метод обнаружения структурных закономерностей в первичных структурах НК-молекул и белков //3 Всесоюз. совещание "Теоретические исследования и банки данных по молекулярной биологии и генетике", Новосибирск, июль 1988 г.: Тез.докл. - Новосибирск, 1988. - С. 115-116.

3. ГУСЕВ В.Д., КУЛИЧКОВ В.А., ЧУПАХИНА О.М. Сложностной анализ генетических текстов (на примере фага λ). - Новосибирск, 1989. - 48 с. -(Препринт/АН СССР. Сиб. отд-ние. Институт математики, № 20).

4. JERMANN W.H. Redundancy in Deterministic Sequences // IEEE Trans. on Syst. sci. and Cybernetics. - 1970. - Vol.SSC-6, N 4.

5. РЕЗНИКОВА Ж.И., РЯБКО Б.Я. Анализ языка муравьев методами теории информации //Пробл.перед.инф.-1986.-Т.XXII, вып.3. - С. 103-108.

6. КНУТ Д. Искусство программирования для ЭВМ.Т.2. -М.: Мир, 1977.

7. ГУСЕВ В.Д., КОСАРЕВ Ю.Г., ТИТКОВА Т.Н. О задаче поиска повторяющихся отрезков текста //Вычислительные системы. - Новосибирск, 1975. - Вып. 62. - С. 49-71.

8. WEINER P. Linear Pattern Matching Algorithms //Conf.records IEEE 14th Annual Symposium on Switching and Automata Theory. - 1973. -P. 1-11.

9. Mc CREIGHT E.M. A Space-economical Suffix Tree Construction Algorithm //J.Association of Comput. Machin. - 1976. -Vol. 23, N 2. -P. 262-272.

10. Building the Minimal DFA for the Set of All Subwords of a Word On-line in Linear Time /A.Blumer, J.Blumer, A.Ehrenfeucht, et al.//Lect.Notes in Comput.Sci. - 1984. - Vol.172. -P.109-118.

11. ЧУПАХИНА О.М. Алгоритм построения сложностного профиля символьных последовательностей. -Настоящий сборник.-С.64-91.

12. ЗУБКОВ А.М., МИХАЙЛОВ В.Г. Предельные распределения случайных величин, связанных с длинными повторениями в последовательности независимых испытаний //Теория вероятностей и ее применения. - 1974. - Т.XIX, № 1. - С. 173-181.

13. Параллельная ДНК - возможность существования /Н.А.Чуриков, В.Б.Чернов, Ю.Б.Голова, Ю.Д.Нечипуренко //Докл. АН СССР. - 1988. - Т. 303, № 5. -С. 1254-1258.

14. МАЗЕЛЬ Л. О мелодии. - М.: Музыка, 1952. - 300 с.

Поступила в ред.-изд.отд.
8 сентября 1989 года