УДК: 519.24: 519.1
DOI: 10.25559/SITITO.15.201904.905-915

Сравнение основных алгоритмов поиска циклов в символьных последовательностях при наличии искажений

Г. Н. Жукова[1], Ю. Г. Сметанин[2], М. В. Ульянов[3,4*]

[1] Национальный исследовательский университет «Высшая школа экономики», г. Москва, Россия
101000, Россия, г. Москва, ул. Мясницкая, д. 20

[2] Федеральный исследовательский центр «Информатика и управление» Российской академии наук, г. Москва, Россия
119333, Россия, г. Москва, ул. Вавилова, д. 44, корп. 2

[3] Институт проблем управления им. В.А. Трапезникова Российской академии наук, г. Москва, Россия
117997, Россия, г. Москва, ул. Профсоюзная, д. 65

[4] Московский государственный университет имени М.В. Ломоносова, г. Москва, Россия
119991, Россия, г. Москва, ГСП-1, Ленинские горы, д. 1

[*] muljanov@mail.ru

Аннотация

В последние годы разработано множество новых алгоритмов для поиска повторяющихся фрагментов в символьных последовательностях с шумом, а также для определения длины периода. Интерес к этим алгоритмам обусловлен тем, что символьные последовательности, полученные на основе реальных данных, не являются периодическими в обычном смысле, а лишь до какой-то степени похожи на них. Для анализа скрытой периодичности в последовательностях, полученных из реальных данных, используется идея представления последовательности как результата внесения искажений в обычную периодическую последовательность.

Наличие искажений замены символа на какой-то другой, вставки или удаления символа не позволяет использовать традиционные алгоритмы поиска длины периода для анализа такой символьной последовательности. Более того, само понятие периодичности последовательности при наличии искажений требует отдельного определения, поскольку обычное понятие периода в этом случае неприменимо. В статье приводятся определения символьной и сегментной периодичности, а также скрытого повторяющегося фрагмента. Кратко описаны алгоритмы CONV, WARP, STNR и алгоритм группы исследователей под руководством Е.Короткова, указана вычислительная сложность алгоритмов CONV, WARP и STNR. Также приведены оценки значимости получаемых в результате работы этих методов значений длины периода.

Рассматриваемые алгоритмы сравниваются между собой по устойчивости к шуму разных типов (замена, вставка, удаление символа, а также их комбинация). Все методы позволяют выявить с некоторой долей уверенности скрытый период, если уровень шума менее 0.1. Алгоритм CONV позволяет определять скрытый период и при большем уровне шума, но при условии, что есть только шум замены. Остальные алгоритмы дают хорошие результаты и в случае шумов всех трех типов. Самым устойчивым к шуму алгоритмом оказался STNR.

Ключевые слова: символьная последовательность, скрытая периодичность, символьная периодичность, сегментная периодичность, шум замены, шум вставки, шум удаления.

Финансирование: данная работа была подготовлена при финансовой поддержке Российского фонда фундаментальных исследований в рамках научного проекта № 19-07-00151 «Комплексные научные и экспериментальные междисциплинарные исследования по моделированию поведения разливов нефти на море под воздействием гидрометеорологических условий».

Для цитирования: *Жукова Г. Н., Сметанин Ю. Г, Ульянов М. В.* Сравнение основных алгоритмов поиска циклов в символьных последовательностях при наличии искажений // Современные информационные технологии и ИТ-образование. 2019. Т. 15, № 4. С. 905-915. DOI: 10.25559/SITITO.15.201904.905-915

Comparison of Some Algorithms for Periodicity Detection in Symbolic Sequences in the Presence of Distortions

G. N. Zhukova[1], Yu. G. Smetanin[2], M. V. Ulyanov[3,4*]
[1] National Research University Higher School of Economics, Moscow, Russia
20 Myasnitskaya St., Moscow 101000, Russia
[2] Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, Moscow, Russia
44-2 Vavilova St., Moscow 119333, Russia
[3] V.A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences, Moscow, Russia
65 Profsoyuznaya St., Moscow 117997, Russia
[4] Lomonosov Moscow State University, Moscow, Russia
1, Leninskie gory, Moscow 119991, Russia
[*] muljanov@mail.ru

Abstract

In recent years, many new algorithms have been developed for detection of periodic patterns in symbolic sequences with noise, as well as for determining the length of a period. The interest in these algorithms is due to the fact that symbolic sequences obtained on the basis of real data are not periodic in the usual sense, but only to some extent similar to them. To analyze latent periodicity in sequences obtained from real data, the idea of representing a sequence as a result of introducing distortions into a regular periodic sequence is used.
The presence of distortions of replacing a symbol with some other one, insertion or deletion of a symbol makes it impossible to analyze such a symbolic sequence by the use of traditional algorithms for determining the length of a period. Moreover, the very concept of periodicity of a sequence in the presence of distortions is to be defined, since the usual concept of a period is not applicable in this case. The article provides definitions of symbol and segment periodicity, as well as the definition of an obscure periodic pattern. The CONV, WARP, STNR algorithms and the algorithm of a group of researchers led by E. Korotkov are briefly described; the computational complexity of the CONV, WARP, and STNR algorithms is given. Estimates of the confidence of a period value are also mentioned.
The considered algorithms are compared with each other in terms of resistance to noise of different types (replacement, insertion, deletion of a symbol, as well as their combination). All methods detect (with some degree of certainty) the latent period if the noise level is less than 0.1. The CONV algorithm detects the latent period at a higher noise level, but provided that there is only replacement noise. Other algorithms work well in the case of noise of all three types. The most noise resilient algorithm is STNR.

Keywords: symbolic sequence, latent periodicity, symbol periodicity, segment periodicity, replacement noise, insertion noise, deletion noise.

Funding: This work was prepared with the financial support of the Russian Foundation for Basic Research in the framework of scientific project No. 19-07-00151 "Complex Scientific and Experimental Interdisciplinary Studies of Modeling the Behavior of Oil Spills at Sea Under The Influence of Hydrometeorological Conditions."

For citation: Zhukova G.N., Smetanin Yu.G., Ulyanov M.V. Comparison of Some Algorithms for Periodicity Detection in Symbolic Sequences in the Presence of Distortions. *Sovremennye informacionnye tehnologii i IT-obrazovanie* = Modern Information Technologies and IT-Education. 2019; 15(4):905-915. DOI: 10.25559/SITITO.15.201904.905-915

Введение

Выявление скрытой периодичности в символьной последовательности привлекает внимание многих исследователей в области анализа данных. Символьные последовательности используются при решении ряда задач качественного анализа в прикладных областях, связанных с анализом больших данных реального мира. Такие данные целесообразно кодировать символами конечного алфавита, это позволяет отбросить лишние детали, не несущие полезной информации. Такое кодирование особенно актуально при анализе больших данных, поскольку в задачах, для решения которых достаточна лишь бинарная характеризация, высокая точность числовых представлений признаков приводит к неоправданно большим объемам информации и трудоемким вычислениям без повышения качества анализа.

Заметим, что значительное число исследуемых в настоящее время процессов обладает циклическим характером, и достаточно часто исходные данные подвержены воздействию случайных искажений, вызванных внешними факторами. Эта особенность в аспекте символьного кодирования приводит к необходимости использования для адекватного описания таких процессов математической модели символьной последовательности с зашумленными циклами.

Отметим, что интерес к разработке методов исследования и моделей периодичностей во временных рядах в контексте искаженных шумом данных с каждым годом только растет, в первую очередь, из-за широкого перечня задач, которые могут быть решены или позволят улучшить существующие решения [1-22]. В этот перечень входит прогнозирование погоды, прогнозирование котировок на фондовых биржах, анализ различных особенностей клинических заболеваний, анализ кардиограмм, исследования в биоинформатике, например, поиск скрытых периодов в ДНК, а также задачи, связанные с анализом временных рядов [5,9-11,22].

Эффективность применения моделей циклов с шумами обусловлена возможностью решения задач по вероятностному прогнозированию символьных зашумленных последовательностей. Их решение позволит разработать эффективные методы прогнозирования и реконструкции данных в форме символьных кодов по фрагментарной, неполной и искаженной информации. Эти методы будут полезны в различных предметных областях, связанных с анализом длинных последовательностей, символьных описаний динамических процессов и временных рядов.

Для разработки таких моделей и методов необходимо решить задачу идентификации параметров в циклической зашумленной последовательности — задачу определения типа и уровня шума и задачу поиска периода зашумленного цикла. Современный базовый прием для решения задачи поиска циклов с шумами заключается в применении функций расстояния между последовательностью и ее сдвигами разной длины. Также существует ряд подходов, которые были разработаны исследователями в области анализа временных рядов и поиска информации — более подробно мы описываем эти подходы ниже. Очевидный интерес представляет исследование чувствительности различных методов поиска периодов в зашумленных символьных последовательностях к уровню шума. Именно эта задача и является предметом настоящей статьи.

Подходы к поиску периодичности в условиях шума

Далее в тексте авторы под циклом (простым циклом) в символьной последовательности понимают устойчиво повторяющееся слово конечного алфавита, а под периодом (длиной периода) — длину этого слова. При внесении шума в цикл мы уже не можем говорить об устойчивой повторяемости, а лишь о наличии некоторого вероятностного шаблона, описывающего цикл с шумом.

Задача поиска простого цикла в символьной последовательности есть задача нахождения повторяющегося шаблона, с помощью которого можно описать всю циклическую часть последовательности. Решение для периода и цикла, не содержащего шума, доставляет использование классических алгоритмов, таких как алгоритмы Флойда [23], Брента [24], Госпера [23] и Ниваши [23]. Сложность в нахождении периода возникает при добавлении различных видов шумов в циклическую последовательность, таких как вставка, замена и удаление символов, которые исключают возможность использования вышеперечисленных алгоритмов. Отметим, что шумы вставки и удаления искажают период, что вносит дополнительные сложности в анализ. Поэтому корректно говорить о задаче поиске шаблона цикла и о задаче поиска периодичности, а не о задаче поиска длины периода. При шуме замены мы имеем некоторый шаблон с длиной равной периоду, каждая реализация которого может содержать искажения некоторых символов с некоторыми вероятностями.

Для решения задачи поиска периодичности в контексте зашумленности данных, в настоящее время существует ряд подходов, которые широко применяются в различных областях. В статье будет рассмотрены наиболее известные алгоритмы, реализующие известные подходы.

Один из таких подходов для решения задачи поиска зашумленных циклов в ограниченной последовательности заключается в применении различных функций расстояния между последовательностью и ее сдвигами разной длины. Его реализация была предложена Mohamed G. Elfeky [25]. Алгоритм основан на применении свертки для перехода в пространство Фурье, сравнении последовательности со сдвинутой ее копией, и обратном переходе из Фурье пространства. Другой подход основан на построении структуры, основанной на суффиксах. Идея заключается в заполнении древовидной структуры разностями между позициями элементов и их дочерних вершин. Еще один подход использует динамическую модель цикла.

Объектами исследования являются наиболее известные алгоритмы, реализующие указанные выше подходы к поиску зашумленных циклов, а именно:

- динамическая модель цикла — алгоритм WARP;
- свертка — алгоритм CONV;
- дерево суффиксов — алгоритм STNR;
- метод, предложенный коллективом авторов под руководством Е.В. Короткова, далее называемый для краткости алгоритмом Короткова.

Более детальное описание этих алгоритмов будет дано ниже.

Задача исследования

Основная задача — определение на основе экспериментальных данных чувствительности основных алгоритмов поиска циклов с шумами к уровню шума. Мы рассматриваем следующие типы шумов:

Шум замены. Такой шум возникает в ситуации, когда при кодировании некоторого периода в исходном временном ряде значение отсчета попадает в другой сегмент, кодируемый другим символом, по сравнению с отсчетом, отстоящим на период ранее. И, хотя ожидаемым является близлежащий сегмент, мы можем наблюдать как резкие всплески значений, так и резкие падения, приводящие к замене символа, который стоял в этой позиции в предыдущем периоде, на любой символ кодирующего алфавита. Отметим, что шум этого типа не приводит к изменению длины периода.

Шум добавления. Возникновение такого типа шума, приводящего к изменению длины периода, связано с тем, что исходный процесс, подвергаемый символьному кодированию, возвращается на некотором периоде в условное «начальное» состояние позже, чем в предыдущем. Такое различие длин может характеризоваться добавлением не обязательно одного символа, а быть может и нескольких, сильно искажая длину периода.

Шум удаления. По аналогии с предыдущим типом шума, шум удаления возникает при условии, что исходный процесс, подвергаемый символьному кодированию, возвращается на некотором периоде в условное «начальное» состояние раньше, чем в предыдущем периоде.

Для детального анализа алгоритмов идентификации циклов в зашумленных периодических символьных последовательностях интерес представляет исследование способности этих методов определять цикличность в зависимости от уровня шума в каждом из описанных типов.

Под чувствительностью алгоритмов авторы в данной статье понимают их чувствительность к заданному уровню шума. Реально мы будем оценивать чувствительность характеристикой устойчивости — отношением правильно распознанных циклов с шумом данного уровня (пропорции шума) к общему числу экспериментов.

Авторы рассматривают далее следующие четыре подхода и наиболее известные реализующие их алгоритмы:

1. Свертка последовательности с ней же, но записанной в обратном порядке. Благодаря использованию алгоритма быстрого преобразования Фурье авторам алгоритма CONV [25] удалось сократить сложность вычислений до $O(n\log n)$.
2. Динамическая модель цикла для нахождения как длины периода, так и повторяющейся последовательности символов, алгоритм WARP [26].
3. Подход дерева суффиксов. Реализация этого подхода — алгоритм STNR [27]. Подход основан на использовании алгоритма Укконена [28] для построения дерева суффиксов. На его основе можно получить реализацию, которая будет эффективна по трудозатратам.
4. Подход комбинацию вероятностного подхода и динамической модели, предложенную и реализованную коллективом авторов [29].

Мы хотим ответить на вопрос — как уровень (пропорция) шума влияет на устойчивость идентификации циклов выбранными четырьмя алгоритмами, что позволит сформулировать рекомендации по их применению.

Описание алгоритмов

Приведем краткое описание четырех алгоритмов поиска циклов в последовательностях с шумами.

CONV [25]

Алгоритм используется для определения символьной периодичности, т. е. такого числа p, что рассматриваемый символ встречается в большей части подпоследовательности исходной последовательности, состоящей из элементов с номерами вида $l+kp$, где l – первое вхождение символа в последовательности, k — натуральное число. Очевидно, алгоритм должен хорошо выявлять символьную периодичность при наличии только шума замены, хотя его авторы проводили эксперименты также при наличии шумов вставки и удаления.

Приведем авторское определение символьной периодичности, будем использовать обозначения:

$\pi_{p,l}(T) = e_l, e_{l+p}, e_{l+2p}, \ldots, e_{l+(m-1)p}$ – проекция временного ряда T (далее будем называть временной ряд символьной последовательностью) относительно периода p, начинающегося с позиции $0 \le l < p$

(e_l – элемент символьной последовательности T в позиции l), $m = (n-l)/p$, x – целая часть числа x, n – длина последовательности T;

$\mathcal{F}_2(s, \pi_{p,l}(T))$ – число символов s в проекции $\pi_{p,l}(T)$, имеющих хотя бы один символ s в соседней позиции в проекции $\pi_{p,l}(T)$.

Определение 1

Если последовательность T длины n содержит символ s, такой что существуют l и p, где $0 \le l < p$ и $\dfrac{\mathcal{F}_2(s, \pi_{p,l}(T))}{(n-l)/p - 1} \ge \tau$, где $0 \le \tau \le 1$, то символ s периодичен в последовательности T с периодом p начиная с позиции l относительно порога периодичности $\tau \le 1$.

Приведем определение скрытого периодического фрагмента (obscure periodic pattern).

Определение 2

Если последовательность T длины n содержит символ s, который периодичен *с периодом p начиная с позиции l относительно* произвольного *порога периодичности,* то периодический односимвольный фрагмент длины p, получаемый вставкой символа s в позицию l и произвольного символа, обозначаемого *, во всех остальных позициях, называется скрытым периодическим фрагментом.

Частота встречаемости (support) периодического односимвольного фрагмента, построенного в соответствии с определением, оценивается как $\dfrac{\mathcal{F}_2(s, \pi_{p,l}(T))}{(n-l)/p - 1}$. Таким образом, если фрагмент встречается не реже определенного порогового значения, он считается значимым.

Обозначая $S_{p,l}$ множество всех символов, периодических *с периодом p начиная с позиции l* относительно произвольного порога периодичности, а S^p - декартово произведение всех

$S_{p,l}$ в порядке возрастания l, т.е.

$S^p = \left(S_{p,0} \cup \{*\}\right) \times \left(S_{p,1} \cup \{*\}\right) \times \ldots \times \left(S_{p,p-1} \cup \{*\}\right)$,
авторы ставят в соответствие каждому упорядоченному элементу $\left(s_0, s_1, \ldots, s_{p-1}\right)$, принадлежащему S^p периодический фрагмент $s_0, s_1, \ldots, s_{p-1}$ где $s_i \in S_{p,i} \cup \{*\}$.
Кроме того, в статье рассматривается интервальная периодичность (segment periodicity), т. е. последовательность T длины n называется периодической с периодом p относительно порога периодичности τ если ее можно разделить на сегменты $T_0, T_1, .., T_i, \supset, T_N$ одинаковой длины p, где $T_i = e_{ip}, \ldots, e_{ip+p-1}$, $N = n/p - 1$, так чтобы $S\left(T_i, T_j\right) \geq \tau$, $\forall i, j = 0,1,\ldots,N$, $0 \leq \tau \leq 1$ (т.е. сегменты T_i и T_j «почти» одинаковы). Мера сходства $S(u,v)$ между сегментами одинаковой длины вычисляется на основе расстояния Хэмминга:

$$H(u,v) = \sum_{j=0}^{m-1} \begin{cases} 1, u_j \neq v_j \\ 0, u_j = v_j \end{cases}, S(u,v) = 1 - \frac{H(u,v)}{m},$$

где u и v два сегмента одинаковой длины m, u_j и v_j - символы в позиции j в соответствующем сегменте. расстояние Хэмминга, мера сходства. Функция сходства определена таким образом, что чем она больше, тем больше сходство, для одинаковых сегментов мера сходства равна 1.
Ключевым инструментом анализа в оригинальной статье авторов алгоритма является свертка двух последовательностей. Свертка вычисляется как последовательность чисел, i -й элемент которой равен:

$$\left(X \otimes Y\right)_i = \sum_{j=0}^{i} x_j y_{i-j}, i = 0,1,\ldots,n-1.$$

Пусть $X' = \left[x_0', x_1', \ldots, x_{n-1}'\right]$ - вектор из координат x_i вектора X, записанных в обратном порядке, т.е. $x_i' = x_{n-1-i}$. Вычисление свертки $X' \otimes Y$ позволяет получить некоторую меру сравнения сдвигов последовательности X с последовательностью Y, поскольку

$$\left(X' \otimes Y\right)_i' = \left(X' \otimes Y\right)_{n-1-i} = \sum_{j=0}^{n-1-i} x_j' y_{n-1-i-j} = \sum_{j=0}^{n-1-i} x_{n-1-j} y_{n-1-i-j},$$

$$\left(X' \otimes Y\right)_0' = x_0 y_0 + x_1 y_1 + \ldots + x_{n-1} y_{n-1}$$

$$\left(X' \otimes Y\right)_1' = x_1 y_0 + x_2 y_1 + \ldots + x_{n-1} y_{n-2}$$

...

$$\left(X' \otimes Y\right)_{n-1}' = x_{n-1} y_0$$

Авторами используется модифицированное определение свертки двух символьных последовательностей

$$\left(X \otimes Y\right)_i = \sum_{j=0}^{i} 2^j x_j y_{i-j}, i = 0,1,\ldots,n-1.$$

Коэффициент 2^j введен для того, чтобы вклад каждого совпадения был отличным от других.

$$\left(X' \otimes Y\right)_i = \sum_{j=0}^{i} 2^j x_j y_{i-j}.$$

Основная идея алгоритма поиска символьной периодичности заключается в следующем.
1. По имеющейся последовательности строится последовательность двоичных кодов символов этой последовательности. Затем полученная последовательность кодов записывается в обратном порядке.
2. Строится свертка преобразованных (по п.1) записанной в обратном порядке и исходной последовательностей двоичных кодов.
3. Записывается в обратном порядке свертка п.2
4. На основе анализа свертки п.3. находятся символьные периоды и их начальные позиции.
Для быстрого вычисления свертки используется быстрое преобразование Фурье:

$$X \otimes Y = FFT^{-1}\left(FFT(X) \cdot FFT(Y)\right),$$

в этом случае сложность $O(n \log n)$, в отличие от прямого вычисления свертки, сложность которого $O\left(n^2\right)$.
Пусть дана последовательность $T = e_0, e_1, \ldots, e_{n-1}$ длины n, где e_i – символы некоторого алфавита Σ мощности σ, тогда $\left|\left(s_i\right)\right| = 2^i$, где s_i – символ из алфавита Σ причем при необходимости слева к двоичному представлению добавляются нули так, чтобы все коды имели длину, равную σ. Вся последовательность кодируется следующим образом:
$\left|\left(T\right)\right| = \left|\left(e_0\right)\right|, \left|\left(e_1\right)\right|, \ldots, \left|\left(e_{n-1}\right)\right|$.

Преобразование исходной последовательности в последовательность двоичных представлений степеней двойки одинаковой длины используется для того, чтобы для совпадающих символов исходной последовательности произведение их двоичных кодов было не равно нулю, а для различных принимало нулевое значение ($010 \times 010 = 0 \times 0 + 1 \times 1 + 0 \times 0 = 1$, $010 \times 100 = 0 \times 1 + 1 \times 0 + 0 \times 0 = 0$).
Пусть $C(T) = \left(\left|\left(T\right)\right| ' \otimes \left|\left(T\right)\right|\right)$, $c_i(T)$ – i-я координата $C(T)$, тогда по значению $c_i(T)$ можно определить, какой символ привел к появлению этого значения и его соответствующие позиции.
В статье также предложен более простой алгоритм поиска длины периода, в этом случае не важно, из каких именно символов состоит повторяющийся фрагмент, поэтому для кодирования символов исходной последовательности используются корни мнимой единицы степени, равной мощности алфавита, а свертка вычисляется по формуле

$$\left(X \otimes Y\right)_i = \sum_{j=0}^{i} x_j y_{i-j}^{-1}, i = 0,1,\ldots,n-1.$$

Если – i-я координата этой свертки, разделенная на $n-p$, не меньше, чем заданное пороговое значение порога периодичности τ, то число i считается длиной потенциального периода. Для вычисления свертки опять используется быстрое преобразование Фурье, сложность алгоритма $O(n \log n)$,
WARP [26]
Алгоритм основан на использовании редакционного расстояния (расстояния Левенштейна) между двумя символьными последовательностями, основной инструмент - модифицированная матрица сравнения (матрица трансформаций) последовательности с ней же самой. Редакционное расстояние находится средствами динамического программирования, при этом строится матрица трансформаций, по которой можно восстановить последовательность редакционных предписа-

ний (замена, вставка или удаление символа), переводящую одну из последовательностей в другую. На диагонали такой матрицы сравнения авторы алгоритма [26] заменяют нули бесконечностью и для каждого сдвига последовательности на p от 1 до половины длины последовательности вычисляют редакционное расстояние до исходной последовательности по построенной матрице. Сдвиг, которому соответствует минимальное редакционное расстояние, считается кандидатом в периоды.

Предложена также online-версия алгоритма, позволяющая пересчитывать потенциальный период по мере поступления новых данных. Отличие в том, что рассматривается часть последовательности, попадающая в окно некоторой фиксированной ширины. Матрица трансформаций (матрица сдвигов) частично строится с использованием предыдущей матрицы, остальная часть пересчитывается на основе предыдущей матрицы и новых элементов последовательности, попавших в текущее окно.

Алгоритм обеспечивает большую точность в определении периода при наличии шумов замены, вставки и удаления как по отдельности, так и в различных комбинациях, по сравнению с RELAX, CONV и AWSOM, но требует большего объема памяти и дольше проводит вычисления (см. [26]).

Пусть $X=\left[x_0,x_1,\ldots,x_{n-1}\right]$ и $Y=\left[y_0,y_1,\ldots,y_{n-1}\right]$ две символьные последовательности конечной длины n. Обозначим $\tilde{X}$ последовательность, полученную из X удалением первого символа $\tilde{X}=\left[x_1,\ldots,x_{n-1}\right]$. Классическое определение редакционного расстояния DTW в этом случае

$$DTW\left(X,Y\right)=d\left(x_0,y_0\right)+min\begin{cases}DTW\left(X,\tilde{Y}\right)\\DTW\left(\tilde{X},Y\right),\\DTW\left(\tilde{X},\tilde{Y}\right)\end{cases}$$

где расстояние $d\left(x_{\mathrm{i}},y_{\mathrm{i}}\right)$ между двумя символами x_{i} и y_{i} равно 0, если символы одинаковые и 1 в противном случае:

$$d\left(x_{\mathrm{i}},y_{\mathrm{i}}\right)=\begin{cases}0,x_{\mathrm{i}}=y_{\mathrm{i}}\\1,x_{\mathrm{i}}\neq y_{\mathrm{i}}\end{cases}$$

Редакционное предписание (warping path) - непрерывный путь $M=m_0,m_1,\ldots,m_{K-1}$ от элемента (0,0) к элементу $\left(\mathrm{n}-1,\mathrm{n}-1\right)$ где m_k соответствует элементу $\left(i_{\mathrm{k}},j_{\mathrm{k}}\right)$, т.е. $m_k=d\left(x_{i_{\mathrm{k}}},y_{j_{\mathrm{k}}}\right)$. Редакционных предписаний, переводящих одну последовательность в другую экспоненциально много, каждому соответствует своя последовательность сдвигов (alignment). Стоимость трансформации, соответствующей пути M равна сумме расстояний $\sum_{k=0}^{K-1}m_k$.

Расстояние $DTW\left(X,Y\right)$ вычисляется как минимальная стоимость среди стоимостей всех возможных трансформаций последовательности X, переводящих ее в Y

$$DTW\left(X,Y\right)=\min\left\{\sum_{k=0}^{K-1}m_k\right\}$$

Этот минимальный путь может быть найден средствами динамического программирования следующим образом. Вначале вычисляются кумулятивные суммы

$$\gamma\left(\mathrm{i},\mathrm{j}\right)=d\left(x_{\mathrm{i}},y_{\mathrm{j}}\right)+min\left\{\gamma\left(\mathrm{i}-1,\mathrm{j}-1\right),\gamma\left(\mathrm{i}-1,\mathrm{j}\right),\gamma\left(\mathrm{i},\mathrm{j}-1\right)\right\}.$$

Расстояние $DTW\left(X,Y\right)$ находится как кумулятивная сумма элемента $\left(\mathrm{n}-1,\mathrm{n}-1\right)$

$$DTW\left(X,Y\right)=\gamma\left(\mathrm{n}-1,\mathrm{n}-1\right).$$

Временная и ресурсная сложность вычисления DTW порядка $O\left(n^2\right)$.

Авторы [26] предложили использовать матрицу трансформаций для поиска длины периода при наличии шумов. Вначале строится матрица трансформаций последовательности X в себя же. Эта матрица симметрична, ее диагональные элементы равны 0. Интересно то, что все параллельные главной диагонали последовательности элементов этой матрицы соответствуют сравнению последовательности с некоторым ее сдвигом. Так, последовательность, начинающаяся с элемента $\left(0,i\right)$ и заканчивающаяся в $\left(n-i-1,n-1\right)$ соответствует сравнению последовательности с ее сдвигом на i символов.

Пусть рассматривается последовательность длины n. ,огда для каждого сдвига на $p=1,2,\ldots,n/2$ вычисляется минимальный путь, начинающийся из элемента $\left(0,p\right)$. Стоимость трансформации, соответствующая пути трансформации M_p представляет собой расстояние DTW между последовательностью T и ее сдвигом $T^{(p)}$ и может быть вычислена динамически как кумулятивное расстояние $\gamma\left(n-p-1,n-1\right)$:

$$DTW\left(T,T^{(p)}\right)=\gamma\left(n-p-1,n-1\right)$$

Малое по величине значение $DTW\left(T,T^{(p)}\right)$ показывает довольно большое сходство между T и ее сдвигом $T^{(p)}$, поэтому p может быть длиной периода последовательности T. Максимальное возможное значение $DTW\left(T,T^{(p)}\right)$ равно $n-p$, поэтому значимость (support) величины p как длины периода предлагается оценивать так:

$$\frac{n-p-DTW\left(T,T^{(p)}\right)}{n-p}$$

Потенциальной длиной периода считается такое значение p, что значимость его не менее некоторого порогового значения τ.

Для предотвращения ситуации, когда в качестве наилучшей трансформации будет выбран сдвиг на то же самое число символов, но в противоположном направлении, авторы заменили диагональные нули в матрице трансформаций на бесконечность.

STNR [27]

Этот алгоритм основан на использовании дерева суффиксов и позиционируется как устойчивый к шумам трех видов (замена, вставка, удаление). Метод позволяет не только обнаруживать символьную или сегментную периодичность, но и почти-периодичность (partial periodicity) временных рядов. Авторы [27] сравнивают свой алгоритм с предложенными ранее AWSOME, CONV, WARP и показывают, что он работает более эффективно. Авторы не приводят четкого определения периодичности разных видов, но отмечают, что сегментная периодичность означает, что временной ряд можно представить в виде приблизительных повторений некоторого сегмента из этого ряда. Символьной периодичностью они называют ситуацию повторения некоторого фиксированного символа через одинаковые промежутки. Повторение целого фрагмента через некоторый интервал авторы называют почти-периодичностью (partial periodicity).

Для выявления периодичности вначале строится суффиксное дерево (suffix tree), используется алгоритм Укконена [28], работающий линейное время (относительно длины временного ряда). Алгоритм производит набор дуг, каждой из которых приписан номер входящей вершины и номер исходящей вершины, первый и последний индекс символа и значение.

После того, как построено и размечено дерево суффиксов, начинает работу алгоритм обхода дерева снизу-вверх. При обходе каждый лист передает свое значение родителю. Внутренним вершинам (не являющимся корнем или листом) после получения всех значений от своих дочерних вершин приписывается вектор из этих значений, называемый вектором вхождений (occurrence vector). Этот вектор показывает, в каких позициях встретился соответствующий фрагмент.

Следующим шагом по каждому вектору вхождений строится вектор различий (difference vector), его координаты представляют собой разность очередной и предыдущей координаты вектора вхождений $V = v_0, v_1, \ldots, v_{m-1}$. Длина вектора различий D на единицу меньше, чем длина вектора вхождений:

$$D = v_1 - v_0, v_2 - v_1, \ldots, v_{m-1} - v_{m-2}$$

Авторы отмечают, что вектор разностей содержит потенциальные периоды (длины периодов). Далее каждый из этих периодов (кроме тех, что начинаются после середины последовательности или имеют длину более половины длины последовательности) проверяется и вычисляется соответствующая значимость периода (periodic strength).

Значимость периода τ вычисляется по формуле

$$\tau(p, st) = \frac{count(p, st)}{\left\lceil \frac{n - st}{p} + y \right\rceil}, y = \begin{cases} 1, (n - st) \bmod p > \text{значения дуги} \\ 0, \text{ иначе} \end{cases}$$

она равна отношению частотой встречаемости фрагмента и максимального возможного числа случаев, когда встретился данный фрагмент.

Метод Короткова и др. [29]

Рассматриваемый метод был разработан для выявления скрытой периодичности в последовательностях аминокислот, образующих белки. Метод был апробирован авторами [29] и на символьных последовательностях, полученных на основе экономических данных. Метод позволил выявить длинные скрытые периоды в последовательностях аминокислот при наличии искажений типа вставки и удаления (искажение замены не рассматривалось).

Опишем кратко основные шаги алгоритма. Вначале строится множество случайных последовательностей путем случайного перемешивания символов изучаемой последовательности. Авторы использовали множество из 200 последовательностей. Затем генерируется множество W случайных матриц, размерность матриц $20 \times n$, где n – предполагаемая длина периода. Выравнивание последовательности и вычисление функции сходства производится средствами динамического программирования. Случайные матрицы затем трансформируются таким образом, чтобы вероятностное распределение функции сходства для всех случайных последовательностей было примерно одинаковым для всех полученных случайных матриц.

В отличие от всех описанных выше методов, метод [29] использует априорное задание длины периода или хотя бы некоторого диапазона возможных длин периода (в этом случае возможен простой перебор всех возможных длин из диапазона). Каждая матрица рассматривается как точка в пространстве размерности $20n$. Случайные матрицы с мерой различия (с каждой из уже включенных в множество W матриц) не менее некоторого заданного значения включаются в множество W. Мера различия между матрицами M_1 и M_2 с элементами, соответственно $m_1(i, j)$ и $m_2(i, j)$ вычисляется по формуле

$$I(M_1, M_2) = \sum_{j=1}^{n} I_j(M_1, M_2),$$

$$I_j(M_1, M_2) = \sum_{i=1}^{20} \left[m_1 \ln m_1 + m_2 \ln m_2 - (m_1 + m_2) \ln(m_1 + m_2) \right] +$$

$$+ \left[\sum_{i=1}^{20} (m_1 + m_2) \right] \ln \left[\sum_{i=1}^{20} (m_1 + m_2) \right] - \left[\sum_{i=1}^{20} m_1 \right] \ln \left[\sum_{i=1}^{20} m_1 \right] -$$

$$- \left[\sum_{i=1}^{20} m_2 \right] \ln \left[\sum_{i=1}^{20} m_2 \right]$$

Для простоты обозначений вместо $m_1(i, j)$ и $m_2(i, j)$ в формулах используются m_1 и m_2 соответственно.

Таким образом, в множество W включаются случайные матрицы с мерой различия со всеми остальными матрицами этого множества не менее 1. Всего было построено 10^8 матриц для каждого значения длины периода.

Далее изучаемая последовательность выравнивается с использованием построенного множества случайных матриц W, при этом используется мера сходства (относительно модифицированной матрицы $m'(i, j)$:

$$F(i, j) = max \begin{cases} 0 \\ F(i-1, j-1) + m'(s(i), k) \\ F(i, j-1) - d \\ F(i-1, j) - d \end{cases} \quad (1)$$

где $s(i)$ элемент последовательности S, d - штраф за вставку или удаление в последовательности S. Индексы i и j изменяются от 1 до n и $k = j - n\frac{j - 0,1}{n}$.

Одновременно с построением функции сходства $F(i, j)$ вычисляются элементы обратной матрицы переноса F'. Каждый элемент $F'(i, j)$ матрицы F' равен числу элементов матрицы F, для которых достигается максимум в правой части выражения (1). Затем в матрице F находится максимальный элемент и соответствующая последовательность редакционных предписаний.

Далее для выравнивания последовательностей используются модифицированные матрицы m', для построения которых по матрицам m вычисляются величины A и B

$$A = \sum_{i=1}^{20} \sum_{j=1}^{n} m(i, j)^2$$

$$B = \sum_{i=1}^{20} \sum_{j=1}^{n} m(i, j)\, p(i)\, f(j),$$

где $f(j) = 1/n$, $p(i) = n(i)/N$, $n(i)$ – число аминокислот типа i в последовательности S, N – общее число аминокислот в последовательности S.

Для выравнивания последовательностей строятся матрицы m' с одинаковыми значениями величины A (для фиксированной длины периода n) и функция распределения максимального элемента матрицы F почти одинакова для всех матриц с n столбцами. Рассматривая матрицы m и m' как точки в пространстве размерности $20n$, авторы [7] находят матрицы m' из условий, что величины A и B для искомой матрицы m' принимают фиксированные значения, и из всех таких матриц выбирается такая m', которая соответствует точке в пространстве размерности $20n$, ближайшей к той, которой соответствует матрица m.

Далее из всех матриц m', построенных на основе матриц из W, выбирается матрица $\max(m')$ с максимальным значением функции сходства F. Авторы [29] отмечают, что несмотря на то, что в W включено большое число матриц, матрица $\max(m')$ может не соответствовать максимально возможному значению $\max F$ функции сходства F для последовательности S и периода n. Чтобы преодолеть это затруднение, строится еще 10^7 матриц, отличающихся от $\max(m')$ со значением меры различия менее 1 (но большое 0). Окончательно выбирается матрица с наибольшим значением $\max F$.

Вышеописанная процедура повторяется для всех случайных последовательностей множества Q., результате находятся среднее значение и дисперсия $\max F$ для каждой величины периода n и последовательности аминокислот S. Далее строится мера периодичности:

$$Z(n) = \frac{\max F(n) - \max \bar{F}(n)}{\sqrt{D(\max F(n))}}.$$

На графике зависимости $Z(n)$, построенном по последовательности со скрытой периодичностью, наблюдаются пики при значениях n, кратных периоду. Для случайных непериодических последовательностей такой эффект не наблюдается. Предложенный авторами метод был использован ими при решении ряда практических задач [30-34].

Результаты

Вначале приведем кратное описание результатов сравнения методов CONV, WARP и STNR, проведенного авторами этих алгоритмов.

Авторы CONV и WARP сравнивают их друг с другом и с ранее разработанными другими исследователями алгоритмами RELAX и AWSOM. В случае только шума замены RELAX, CONV и WARP примерно одинаково устойчивы к шуму в диапазоне до 0.5 (доля замененных символов) AWSOM. С шумом только вставки и замены AWSOM и WARP работают лучше RELAX и намного лучше CONV, а с комбинацией шумов всех трех видов WARP справляется лучше CONV, RELAX и AWSOM. Устойчивость алгоритма к шуму измеряется надежностью (confidence), вычисляемой по формуле

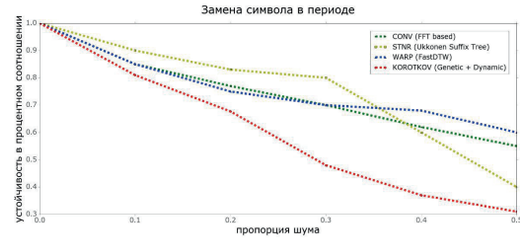
$$\frac{n - p - DTW(T, T^{(p)})}{n - p},$$

где $DTW(T, T^{(p)})$ - редакционное расстояние между последовательностью T и ее сдвигом $T^{(p)}$ на p символов вправо, n – длина T.

Алгоритм STNR был разработан позже перечисленных выше методов, в сравнении с ними он показывает большую устойчивость к шумам всех типов вместе и каждого в отдельности при относительно небольшом уровне шума (до 0.2 в случае только шума замены и до 0.4 в случае комбинации все типов шума).

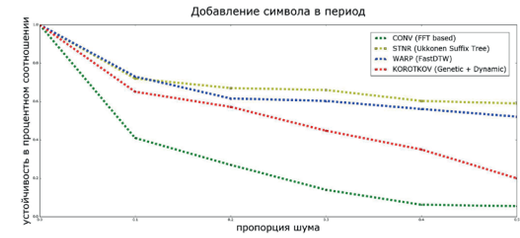
Авторы [29] не сравнивали свой алгоритм с CONV, WARP и STNR, такое сравнение было проведено Н. Ендальцевым, которому авторы выражают благодарность за предоставленные данные. Результаты сравнения четырех рассматриваемых в работе алгоритмов представлены на рис. 1-3.

Таким образом, при наличии шумов замены, вставки и удаления с общей долей шума до 0.4 рекомендуется для поиска скрытой периодичности использовать алгоритм STNR.



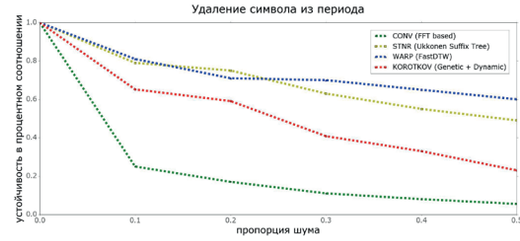
Р и с. 1. Результаты исследования алгоритмов по устойчивости к шуму замены

Fi g. 1. Research results for replacement noise immunity algorithms



Р и с. 2. Результаты исследования алгоритмов по устойчивости к шуму добавления

F i g. 2. Research results for insertion noise immunity algorithms



Р и с. 3. Результаты исследования алгоритмов по устойчивости к шуму удаления

F i g. 3. Research results for deletion noise immunity algorithms

References

[1] Hou W., Pan Q., Peng Q., He M. A new method to analyze protein sequence similarity using Dynamic Time Warping. *Genomics.* 2017; 109(2):123-130. (In Eng.) DOI: 10.1016/j.ygeno.2016.12.002

[2] Parthasarathy S., Mehta S., Srinivasan S. Robust Periodicity Detection Algorithms. In: *Proceedings of the 15th ACM*

international conference on Information and knowledge management (CIKM '06). Association for Computing Machinery, New York, NY, USA, pp. 874-875. (In Eng.) DOI: 10.1145/1183614.1183774

[3] Salvador S., Chan P. Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis.* 2007; 11(5):561-580. (In Eng.) DOI: 10.3233/IDA-2007-11508

[4] Otunba R., Lin J. APT: Approximate Period Detection in Time Series. In: *Proceedings of the 26th International Conference on Software Engineering & Knowledge Engineering (SEKE).* Vancouver, Canada, July 1-3, 2014, pp. 490-494. Available at: https://ksiresearchorg.ipage.com/seke/seke14paper/seke14paper_9.pdf (accessed 10.08.2019). (In Eng.)

[5] Lin J., Keogh E., Lonardi S., Chiu B. A Symbolic Representation of Time Series, with Implications for Streaming Algorithms. In: *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery (DMKD'03).* Association for Computing Machinery, New York, NY, USA, 2003, pp. 2-11. (In Eng.) DOI: 10.1145/882082.882086

[6] Pommerening K. Finding the Period of a Periodic Sequence. 2009, pp. 1-6. Available at: https://www.staff.uni-mainz.de/pommeren/MathMisc/Periods.pdf (accessed 28.07.2019). (In Eng.)

[7] Vlachos M., Yu P., Castelli V. On Periodicity Detection and Structural Periodic Similarity. In: *Proceedings of the 2005 SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics, 2005, pp. 449-460. (In Eng.) DOI: 10.1137/1.9781611972757.40

[8] Otunba R., Lin J., Senin P. MBPD: Motif-Based Period Detection. In: Peng WC. et al. (eds*) Pacific-Asia Conference on Knowledge Discovery and Data Mining. Trends and Applications in Knowledge Discovery and Data Mining. PAKDD 2014.* Lecture Notes in Computer Science, vol. 8643. Springer, Cham, 2014, pp. 793-804. (In Eng.) DOI: 10.1007/978-3-319-13186-3_71

[9] Rao K.P., Gayathri M. Noise Resilient Periodicity Mining in Time Series Data Bases. *International Journal of Computer Science and Network Security.* 2014; 14(7):41-44. Available at: http://paper.ijcsns.org/07_book/201407/20140707.pdf (accessed 28.07.2019). (In Eng.)

[10] JishaKrishnan, Chitharanjan K. Periodicity detection algorithms in time series databases-a survey. *International Journal of Computer Science & Engineering Technology*. 2013; 4(1):22-28. Available at: http://www.ijcset.com/docs/IJCSET13-04-01-013.pdf (accessed 28.07.2019). (In Eng.)

[11] Sujatha B., Pandian S. C. Noise Removal in Distributed Time Series Database Using Predominant Pattern Distribution Model. *IOSR Journal of Engineering*. 2013; 3(2):06-13. (In Eng.) DOI: 10.9790/3021-03220613

[12] Ma Sh., Hellerstein J.L. Mining Partially Periodic Event Patterns with Unknown Periods. In: *Proceedings 17th International Conference on Data Engineering.* Heidelberg, Germany, 2001, pp. 205-214. (In Eng.) DOI: 10.1109/ICDE.2001.914829

[13] Sujatha B., Pandian S. C. Multiplex Tree Pruning for Periodic Pattern Mining. *International Journal of Soft Computing.* 2014; 9(1):37-43. (In Eng.) DOI: 10.36478/ijscomp.2014.37.43

[14] Grossi R., Italiano G.F. Suffix trees and their applications in string algorithms. In: *Proceedings of the 1st South American Workshop on String Processing.* 1993, pp. 57-76. (In Eng.)

[15] Chanda A.K., Ahmed C.F., Samiullah Md., Leung C.K. A new framework for mining weighted periodic patterns in time series databases. *Expert Systems with Applications*. 2017; 79:207-224. (In Eng.) DOI: 10.1016/j.eswa.2017.02.028

[16] Chanda A.K., Saha S., Nishi M.A., Samiullah Md., Ahmed C.F. et al. An efficient approach to mine flexible periodic patterns in time series databases. *Engineering Applications of Artificial Intelligence.* 2015; 44:46-63. (In Eng.) DOI: 10.1016/j.engappai.2015.04.014

[17] Yuan Q., Zhang W., Zhang C., Geng X., Cong G., Han J. PRED: Periodic Region Detection for Mobility Modeling of Social Media Users. In: *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (WSDM'17).* Association for Computing Machinery, New York, NY, USA, 2017, pp. 263-272. (In Eng.) DOI: 10.1145/3018661.3018680

[18] Yuan Q., Shang J., Cao X., Zhang C., Geng X., Han J. Detecting Multiple Periods and Periodic Patterns in Event Time Sequences. In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM '17).* Association for Computing Machinery, New York, NY, USA, 2017, pp. 617-626. (In Eng.) DOI: 10.1145/3132847.3133027

[19] Han J., Dong G., Yin Y. Efficient mining of partial periodic patterns in time series database. In: *Proceedings 15th International Conference on Data Engineering (Cat. No.99CB36337)*, Sydney, NSW, Australia, 1999, pp. 106-115. (In Eng.) DOI: 10.1109/ICDE.1999.754913

[20] Berndt D.J., Clifford J. Using Dynamic Time Warping to Find Patterns in Time Series. In: *KDD-94:AAAI-94 Workshop on Knowledge Discovery in Databases*. 1994; 10(16):359-370. Available at: https://aaai.org/Papers/Workshops/1994/WS-94-03/WS94-03-031.pdf (accessed 28.07.2019). (In Eng.)

[21] Grassly N.C., Fraser C. Seasonal infectious disease epidemiology. *Proceedings of the Royal Society of London B: Biological Sciences.* 2006; 273(1600):2541-2550. (In Eng.) DOI: 10.1098/rspb.2006.3604

[22] Xylogiannopoulos K.F., Karampelas P., Alhajj R. Analyzing very large time series using suffix arrays. Applied Intelligence. 2014; 41(3):941-955. (In Eng.) DOI: 10.1007/s10489-014-0553-x

[23] Nesterenko A.Yu. Cycle detection algorithms and their applications. *Fundamentalnaya i prikladnaya matematika*. 2010; 16(6):109-122. Available at: https://elibrary.ru/item.asp?id=20285258& (accessed 28.07.2019). (In Russ., abstract in Eng.)

[24] Brent R.P. An improved Monte Carlo factorization algorithm. *BIT Numerical Mathematics*. 1980; 20(2):176-184. (In Eng.) DOI: 10.1007/BF01933190

[25] Elfeky M.G., Aref W.G., Elmagarmid A.K. Periodicity detection in time series databases. *IEEE Transactions on Knowledge and Data Engineering.* 2005; 17(7):875-887. (In Eng.) DOI: 10.1109/TKDE.2005.114

[26] Elfeky M.G., Aref W.G., Elmagarmid A.K. WARP: time warping for periodicity detection. In: *Fifth IEEE International Conference on Data Mining (ICDM'05)*, Houston, TX, 2005, pp. 8. (In Eng.) DOI: 10.1109/ICDM.2005.152

[27] Rasheed F., Alhajj R. STNR: A suffix tree based noise resilient algorithm for periodicity detection in time series data-

bases. *Applied Intelligence*. 2010; 32(3):267-278. (In Eng.) DOI: 10.1007/s10489-008-0144-9

[28] Ukkonen E. On-line construction of suffix trees. *Algorithmica*. 1995; 14(3):249-260. (In Eng.) DOI: 10.1007/BF01206331

[29] Korotkov E.V., Korotkova M.A. Developing New Mathematical Method for Search of the Time Series Periodicity with Deletions and Insertions. *Journal of Physics: Conference Series.* 2017; 788(1):012019. (In Eng.) DOI: 10.1088/1742-6596/788/1/012019

[30] Frenkel F.E., Korotkova M.A., Korotkov E.V. Database of Periodic DNA Regions in Major Genomes. *BioMed Research International.* 2017; 2017:7949287. (In Eng.) DOI: 10.1155/2017/7949287

[31] Korotkov E.V. et al. Latent Periodicity Regions in Amino Acid Sequences. *Molekuliarnaia Biologiia* = Molecular Biology. 1999; 33(4):611-617. (In Eng.)

[32] Chaley M.B., Korotkov E.V., Skryabin K.G. Method Revealing Latent Periodicity of the Nucleotide Sequences Modified for a Case of Small Samples. *DNA Research.* 1999; 6(3):153-163. (In Eng.) DOI: 10.1093/dnares/6.3.153

[33] Korotkova M.A., Korotkov E.V., Rudenko V.M. Latent periodicity of protein sequences. *Journal of Molecular Modeling.* 1999; 5(6):103-115. (In Eng.) DOI: 10.1007/s008940050122

[34] Korotkov E.V., Korotkova M.A. Latent periodicity of DNA sequences from some human gene regions. *DNA Sequence.* 1995; 5(6):353-358. (In Eng.) DOI: 10.3109/10425179509020866

Поступила 28.07.2019; принята к публикации 05.10.2019;
опубликована онлайн 23.12.2019.
Submitted 28.07.2019; revised 05.10.2019;
published online 23.12.2019.

Об авторах:

Жукова Галина Николаевна, доцент Департамента программной инженерии, Факультет компьютерных наук, Национальный исследовательский университет «Высшая школа экономики» (101000, Россия, г. Москва, ул. Мясницкая, д. 20), кандидат физико-математических наук, доцент, ORCID: http://orcid.org/0000-0003-1835-7422, gzhukova@hse.ru

Сметанин Юрий Геннадиевич, главный научный сотрудник, Вычислительный центр им. А.А. Дородницына РАН, Федеральный исследовательский центр «Информатика и управление» Российской академии наук (119333, Россия, г. Москва, ул. Вавилова, д. 44, корп. 2), доктор физико-математических наук, ORCID: http://orcid.org/0000-0003-0242-6972, ysmetanin@rambler.ru

Ульянов Михаил Васильевич, ведущий научный сотрудник, Институт проблем управления им. В.А. Трапезникова Российской академии наук (117997, Россия, г. Москва, ул. Профсоюзная, д. 65), профессор кафедры алгоритмических языков, факультет вычислительной математики и кибернетики, Московский государственный университет имени М.В. Ломоносова (119991, Россия, г. Москва, ГСП-1, Ленинские горы, д. 1), доктор технических наук, профессор, ORCID: http://orcid.org/0000-0002-5784-9836, muljanov@mail.ru

Все авторы прочитали и одобрили окончательный вариант рукописи.

About the authors:

Galina N. Zhukova, Associate Professor of the School of Software Engineering, Faculty of Computer Science, National Research University Higher School of Economics (20 Myasnitskaya St., Moscow 101000, Russia), Ph.D. (Phys.-Math.), Associate Professor, ORCID: http://orcid.org/0000-0003-1835-7422, gzhukova@hse.ru

Yuri G. Smetanin, Chief researcher of the Dorodnicyn Computing Centre of RAS, Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences (44-2 Vavilov St., Moscow 119333, Russia), Dr. Sci. (Phys.-Math.), ORCID: http://orcid.org/0000-0003-0242-6972, ysmetanin@rambler.ru

Mikhail V. Uljanov, Leading scientist researcher, V.A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences (65 Profsoyuznaya St., Moscow 117997, Russia), Professor of the Department of Algorithmic Languages, Faculty of Computational Mathematics and Cybernetics, Lomonosov Moscow State University (1, Leninskie gory, Moscow 119991, Russia), Dr.Sci. (Engineering), Professor, ORCID: http://orcid.org/0000-0002-5784-9836, muljanov@mail.ru

All authors have read and approved the final manuscript.