УДК 519.764

П. А. Певзнер

ЭФФЕКТИВНОСТЬ ФИЛЬТРАЦИИ В СТАТИСТИЧЕСКИХ АЛГОРИТМАХ БЫСТРОГО ПОИСКА ГОМОЛОГИЙ

При поиске локальных гомологий (поиск гомологий в генетических банках, выбор оптимальных олигонуклеотидных зондов и т. п.) возникает проблема их «быстрого» поиска. Квадратичная трудоемкость алгоритмов динамического программирования заставляет прибегать к методам фильтрации, позволяющим быстро «отбраковать» последовательности с низким уровнем гомологии. В работе вводится понятие эффективности фильтрации и дается оценка эффективности некоторых фильтров, при этом показано, что в 1-граммном анализе эффективность фильтрации связана с потенциальным расширением исходного 4-буквенного алфавита.

Введение. Для определения гомологии между последовательностями ДНК в настоящее время используются методы динамического программирования типа алгоритмов Нидельмана — Вунша и Селлерса [1, 2]. Оценка трудоемкости таких алгоритмов — квадратичная, поэтому для поиска гомологий в случае длинных последовательностей приходится искать альтернативные подходы. Известны попытки решения задачи поиска гомологий в банках генетической информации на суперкомпьютерах [3—5], а также с использованием параллельных алгоритмов [6].

Перспективы значительного расширения баз данных нуклеотидных последовательностей, связанные с секвенированием генома человека, выдвигают задачу разработки методов «быстрого» поиска гомологий. В связи с тем, что попытки уменьшить вычислительную сложность алгоритмов, основанных на динамическом программировании, пока не привели к значительному снижению трудоемкости [7—12], в ряде работ [13—22] были предложены методы фильтрации для поиска гомологий по банку нуклеотидных последовательностей (аналогичные подходы предложены в работах [23—26]).

Возможность введения фильтрации связана с тем, что при поиске локальных гомологий в длинных последовательностях биологов, как правило, интересуют фрагменты с достаточно высоким пороговым уровнем гомологии (или, иными словами, фрагменты, расстояние между которыми мало), а все другие фрагменты отбрасываются. В данной работе считается, что задано пороговое значение расстояния b_0 между окнами и окна с расстоянием $b > b_0$ отбрасываются. Конечно, для выявления окон с $b > b_0$ можно $k_1 \cdot k_2$ раз применить какой-нибудь из алгоритмов глобального поиска гомологий (например, алгоритм Нидельмана — Вунша) с трудоемкостью n^2 (k_1, k_2 — длины анализируемых последовательностей, n — длина окна) и получить для каждой пары окон, начинающихся в позициях i и j, точные значения расстояния $b(i, j)$. Если при этом нас интересуют только позиции (i, j) с уровнем расстояния $b(i, j) \leqslant b_0$, составляющие, например, 0,01 % от числа всех пар (i, j), то 99,99 % работы проделывается впустую, так как информация о точном уровне гомологии при $b(i, j) > b_0$ не нужна.

Методы фильтрации позволяют отказаться от трудоемкого определения расстояния $b(i, j)$ между всеми окнами путем введения легко вычисляемого фильтра $f(i, j)$, принимающего два значения 0 и 1, при этом из $f(i, j) = 1$ следует, что $b(i, j) > b_0$. С введением фильтра f процедура поиска гомологий становится двухступенчатой: на первом этапе (фильтрация) вычисляется функция $f(i, j)$ (при этом пары окон, для

которых $f(i, j)=1$ исключаются из дальнейшего рассмотрения), а на втором проводится вычисление $b(i, j)$ для пар (i, j), удовлетворяющих условию $f(i, j)=0$. Эффективность фильтрации определяется долей отброшенных пар (i, j). Если бы оказалось, что для фильтра f из $f(i, j)=0$ следует $b(i, j) < b_0$, мы имели бы «идеальный» фильтр, однако ни l-граммный анализ, ни метод Миронова — Александрова [16] не предлагают «идеальных» фильтров. Возникает вопрос об эффективности фильтрации, который и рассматривается в настоящей работе. Приведенный анализ может быть использован при разработке методов «быстрого» поиска гомологий, построении dot-matrix и выборе оптимальных олигонуклеотидных зондов — необходимых составляющих современных пакетов программ анализа биополимеров.

Эффективность фильтрации. В большинстве работ по быстрому поиску гомологий отсутствуют оценки скорости, специфичности и чувствительности алгоритмов [27]. Впервые эмпирическое изучение этих характеристик, позволяющее выбрать параметры для быстрого поиска гомологий, проведено Лоуренсом и др. [27]. В настоящей работе вводится понятие эффективности фильтрации в методах быстрого поиска гомологий и приводятся аналитические формулы для ее вычисления. Полученные результаты позволяют, в частности, объяснить разные параметры, рекомедуемые [27] для поиска гомологий в нуклеотидных и аминокислотных последовательностях (программы DNASEARCH и PEPSEARCH соответственно).

Как в l-граммном анализе, так и в методе Миронова — Александрова, для построения фильтров вводится отображение h множества n-буквенных слов в R^m (принципиально другой подход предложен в [21, 22]). При этом $m=A^l$ в l-граммном анализе и $m=(i+1)\cdot A^2$ в методе Миронова — Александрова (i — параметр алгоритма, характеризующий максимальный размер «дырки» в рассматриваемых разнесенных динуклеотидах; A — размерность алфавита).

После введения отображения h сходство между словами S и T можно, как и в [16], определить через евклидово расстояние между точками $h(S)=(S_1, \ldots, S_m)$ и $h(T)=(t_1, \ldots, t_m)$:

$$d(S, T)=\sum_{i=1}^{m}(S_i-t_i)^2.$$

Расстояние $d(S, T)$ называют иногда статистическим расстоянием (в [28] аналогичный подход использован при введении автокорреляционной функции последовательности), подчеркивая его отличие от обычно используемого при сравнении текстов расстояния $b(S, T)$ — минимального числа элементарных операций (вставок, делеций и т. п.), необходимых для преобразования S в T.

Если S и T — случайные слова, то $d(S, T)$ — случайная величина. Можно рассмотреть функцию распределения $d(S, T)$, которая и будет характеризовать эффективность фильтрации. Действительно, если нас интересуют только пары слов с расстоянием ниже порогового, то можно ввести пороговое значение статистического расстояния d_0 и определить фильтр по правилу

$$f(S, T)=\begin{cases}0, \text{ при } d(S, T)\leqslant d_0;\\ 1, \text{ в противном случае.}\end{cases}$$

Эффективность фильтрации можно характеризовать вероятностью $P\{d(S, T)<=d_0\}$. Величина $P\{d(S, T)<=d_0\}$ определяет долю случаев, для которых фильтрация не приводит к отбраковке слов S и T. В этих случаях для анализа вопроса об отбраковке слов S и T приходится использовать алгоритмы динамического программирования. Эффективность фильтрации можно характеризовать, как принято в статистике, в терминах стандартных отклонений, т. е. с помощью величины

$$e=\frac{M-d_0}{D^{1/2}},$$

где M — математическое ожидание; D — дисперсия случайной величины $d(S, T)$ (для малых значений d_0 можно приближенно положить $e=M/D^{1/2}$; $M/D^{1/2}$ будем обозначать e_0).

Фильтрация по составу. Рассмотрим простейшую фильтрацию, когда в качестве h берется отображение множества n-слов в R^A:

$$S \xrightarrow{h} (s_1, \ldots, s_A).$$

Здесь S_i — число букв вида i в слове S; A — размерность алфавита. Определим эффективность фильтрации по составу для случая равновероятного появления букв. Для определения e необходимо получить значения математического ожидания $M(d(S, T))$ и дисперсии $D(d(S, T))$. Введем случайные величины:

$$z_i^a(S) = \begin{cases} 1, & \text{если на } i\text{-м месте в } S \text{ стоит буква } a; \\ 0, & \text{в противном случае} \end{cases}$$

и обозначим

$$x_i^a = z_i^a(S) - z_i^a(T).$$

Очевидно, что $d(S, T) = d(S, T) = \sum_a \left(\sum_{i=1}^{n} x_i^a\right)^2$, $p\{x_i^a = 1\} = p\{x_i^a = -1\} = \frac{1}{A} \cdot (1 - 1/A)$, $p\{x_i^a = 0\} = (1 - 1/A)^2 + 1/A^2$, $M(x_i^a x_i^a) = \frac{2}{A} \cdot (1 - 1/A)$, $M(x_i^a x_j^a) = 0$, если $i \neq j$, $M(x_i^a x_i^b) = -2/A^2$, если $a \neq b$.

Для вычисления дисперсии оценим сначала $M(d(S, T)^2)$:

$$M(d(S, T)) = A \cdot M\left(\sum_{i=1}^{n} x_i^a\right)^2 = A \cdot M\left(\sum_{i=1}^{n} x_i^a x_i^a\right) = A \cdot n \cdot M(x_i^a x_i^a) = 2n(1 - 1/A).$$

Можно показать, что ненулевой вклад в $M(d(S, T)^2)$ вносят следующие пять составляющих:

$$M(d(S, T)^2) = M\left(\sum_a \left(\sum_{i=1}^{n} x_i^a\right)^2\right)^2 = M\left(\sum_{a,b} \sum_{\substack{i,j=1 \\ i \neq j}}^{n} x_i^a x_i^a x_j^b x_j^b\right) + 2M\left(\sum_{\substack{a,b \\ a \neq b}} \sum_{i,j=1}^{n} x_i^a x_j^a x_i^b x_j^b\right) +$$

$$+ 2M\left(\sum_a \sum_{\substack{i,j=1 \\ i \neq j}}^{n} x_i^a x_j^a x_i^a x_j^a\right) + M\left(\sum_{\substack{a,b \\ a \neq b}} \sum_{i=1}^{n} x_i^a x_i^a x_i^b x_i^b\right) + M\left(\sum_a \sum_{i=1}^{n} x_i^a x_i^a x_i^a x_i^a\right).$$

Детальный учет составляющих в последней формуле приводит к выражению:

$$M(d(S, T)^2) = A^2 \cdot 4n(n-1)\frac{1}{A^2} \cdot (1 - 1/A)^2 + 2A(A-1)n(n-1)\frac{4}{A^2} +$$

$$+ 2An(n-1)\frac{4}{A^2} \cdot (1 - 1/A)^2 + A(A-1)n\frac{2}{A^2} + A \cdot 2n \cdot \frac{1}{A}(1 - 1/A).$$

Отсюда следует, что

$$D(d(S, T)) = M(d(S, T)^2) - [M(d(S, T))]^2 = 4n(2n-1)\frac{1}{A}(1 - 1/A)$$

и

$$e_0 = \frac{M(d(S, T))}{D(d(S, T))^{1/2}} = (A/2)^{1/2} \cdot \left(\left(1 - \frac{1}{2n-1}\right)\left(1 - \frac{1}{A}\right)\right)^{1/2}.$$

Таким образом, эффективность фильтрации по составу растет при увеличении числа букв алфавита и ограничена величиной $(A/2)^{1/2}$ (при больших n $e_0 = 1{,}2$ для нуклеотидных последовательностей и $e_0 = 3{,}1$ для аминокислотных последовательностей). Оказывается, что как в l-граммном анализе, так и в методе Миронова — Александрова увеличение эффективности фильтрации происходит за счет потенциального расширения исходного алфавита, причем в первом случае в качестве букв рассматриваются все слова длины l, а во втором — разнесенные динуклеотиды [29]. В следующем разделе показано, что в l-граммном анализе эффективность фильтрации растет как корень квадратный от числа букв в расширенном алфавите, т. е. как $(A^l)^{1/2}$.

l-Граммная фильтрация. При l-граммной фильтрации все слова из l-букв (l-граммы) кодируются числами от 1 до A^l и в качестве отображения h рас-

сматривается:

$$S \xrightarrow{h} (s_1, \ldots, s_{A^l}),$$

где s_i — число l-грамм i-го вида в слове S. Подобно фильтрации по составу введем случайную величину

$$z_i^a(S) = \begin{cases} 1, & \text{если начиная с } i\text{-го места в } S \text{ стоит } l\text{-грамма } a \\ 0, & \text{в противном случае} \end{cases}$$

и обозначим:

$$x_i^a = z_i^a(S) = z_i^a(T).$$

В дальнейшем рассматриваются кольцевые слова S и T из n букв — это позволяет избежать рассмотрения граничных эффектов, возникающих в начале и конце слов. Очевидно, что

$$d(S, T) = \sum_a \left(\sum_{i=1} x_i^a \right)^2$$

и

$$p\{x_i^a = 1\} = p\{x_i^a = -1\} = \frac{1}{A^k}(1 - 1/A^k), \quad M(x_i^a \cdot x_i^a) = \frac{2}{A^k}(1 - 1/A^k),$$

где k — число букв в a, т. е. фактическая величина параметра l при l-граммной фильтрации. Значение $M(x_i^a \cdot x_j^a)$ при $i \neq j$ зависит от структуры самопересечений слова a. Зависимость вероятностных характеристик частот встречаемости слов от структуры их самопересечений изучена в [30, 31]. Мы будем говорить, что слово a допускает d-сдвиг, если $a_{i+d} = a_i$ при $i = 1$, $k - d$ (через a_i обозначена i-я буква слова a). Например, слово ATA допускает 2-сдвиг, а слово AAA — 1- и 2-сдвига. Очевидно, что $M(x_i^a \cdot x_j^a) = 0$, если расстояние между позициями i и j больше или равно k (имеется в виду расстояние между позициями на кольцевой молекуле), в этом случае x_i^a и x_j^a — независимые случайные величины (расстояние между позициями i и j обозначается $|j - i|$).

Можно показать, что в случае, когда a допускает d-сдвиг, и расстояние между i и j равно d:

$$M(x_i^a \cdot x_j^a) = 2/A^{k+d} - 2/A^{2k} = \frac{2}{A^k} \cdot (1/A^d - 1/A^k).$$

Если же a не допускает d-сдвига, то $M(x_i^a \cdot x_j^a) = -2/A^{2k}$. Так как все позиции от 1 до n равноправны, то

$$M(d(S, T)) = M \sum_a \sum_{i,j=1}^{n} x_i^a \cdot x_j^a = n \cdot \sum_a M(x_1^a \cdot x_1^a) + 2n \cdot \sum_a \sum_{t=1}^{k-1} (x_1^a \cdot x_{1+t}^a) =$$

$$= n \cdot A^k \cdot \left(\frac{2}{A^k} \left(1 - \frac{1}{A^k} \right) \right) + 2n \sum_{d=1}^{k-1} \left(\sum_{a \in K_d} (2/A^k \cdot (1/A^d - 1/A^k)) + \sum_{a \in \bar{K}_d} (-2/A^k \cdot 1/A^k) \right),$$

где K_d — множество слов, допускающих d-сдвиг, а $\bar{K}_d$ — множество всех остальных слов. Учитывая, что объединение K_d и $\bar{K}_d$ дает все множество k-буквенных слов, получим:

$$M(d(S, T)) = 2n \cdot \left(1 - \frac{1}{A^k} \right) + 2n \cdot \frac{2}{A^k} \sum_{d=1}^{k-1} \left(\sum_{a \in K_d} \frac{1}{A^d} - \sum_{a \in K_d} \frac{1}{A^k} - \sum_{a \in \bar{K}_d} \frac{1}{A^k} \right) =$$

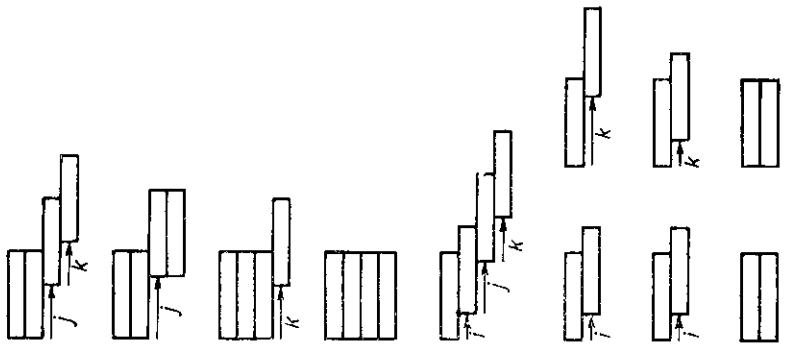
$$= 2n \cdot \left(1 - \frac{1}{A^k} \right) + 2n \cdot \frac{2}{A^k} \sum_{d=1}^{k-1} \left(\sum_{a \in K_d} \frac{1}{A^d} - \sum_a \frac{1}{A^k} \right).$$

Так как $|K_d| = A^d$, а общее число слов длины k равно A^k, то $\sum_{a \in K_d} \frac{1}{A^d} - \sum_a \frac{1}{A^k} = 0$ при любом d и $M(d(S, T)) = 2n \cdot \left(1 - \frac{1}{A^k} \right)$.

Для вычисления $M(d(S, T)^2)$ заметим:

$$M(d(S, T)^2) = M\left(\sum_a \left(\sum_{i=1}^{n} x_i^a\right)^2\right)^2 = M\left(\sum_{a,b} \sum_{n1,n2,n3,n4} x_{n1}^a x_{n2}^a x_{n3}^b x_{n4}^b\right).$$

Значение $M(x_{n1}^a x_{n2}^a x_{n3}^b x_{n4}^b)$ зависит от взаимного расположения позиций $n1$, $n2$, $n3$, $n4$ и структуры самопересечений слов a и b. Очевидно, что если среди позиций $n1$, $n2$, $n3$, $n4$ найдется хотя бы одна изолированная (позиция называется изолированной, если расстояние от нее до каждой из трех оставшихся позиций больше или равно k), то



Восемь типов конфигураций, рассматриваемых при оценке эффективности фильтрации (для каждой конфигурации представлены четыре слова, начинающиеся в позициях $n1$, $n2$, $n3$, $n4$). Конфигурация задается числами i, j, k, показывающими смещение $n1$, $n2$, $n3$, $n4$ относительно друг друга. Представленные на рисунке конфигурации здаются следующими параметрами (конфигурации 6—8 разбиваются на две пары неперекрывающихся слов): *1* — ровно одно из чисел i, j, k равно 0; *2* — $i=0$, $j>0$, $k=0$; *3* — $i=0$, $j=0$, $k>0$ или $i>0$, $j=0$, $k=0$; *4* — $i=0$, $j=0$, $k=0$; *5* — $i>0$, $j>0$, $k>0$; *6* — i, $k>0$ и $i\neq k$; *7* — i, $k>0$ и $i=k$; *8* — $i=k=0$

Eight configuration types considered upon estimating the filtration efficiency (for each configuration 4 words beginning in positions $n1$, $n2$, $n3$, $n4$ are presented). The configuration is given by numbers i, j, k, that reveal shifts of positions $n1$, $n2$, $n3$, $n4$ in respect to one another. The configurations presented are given by the following parameters (configurations 6-8 are subdivided to two pairs of non-overlapping words): *1* — exactly one of the numbers i, j, k is equal to zero; *2* — $i=0$, $j>0$, $k=0$; *3* — $i=0$, $j=0$, $k>0$ or $i>0$, $j=0$, $k=0$; *4* — $i=0$, $j=0$, $k=0$; *5* — $i>0$, $j>0$, $k>0$; *6* — i, $k>0$ and $i\neq k$; *7* — i, $k>0$ and $i=k$; *8* — $i=k=0$

$M(x_{n1}^a x_{n2}^a x_{n3}^b x_{n4}^b) = 0$. Следовательно, ненулевой вклад в $M(d(S, T))$ могут давать только четверки $n1$, $n2$, $n3$, $n4$, содержащие перекрывающиеся слова (такие четверки мы будем называть конфигурациями).

Введем 8 типов конфигураций F_1—F_8 (рисунок) и представим выражение $M(d(S, T))$ в виде:

$$M(d(S, T)^2) = \sum_{i=1}^{8} V_i,$$

где $V_i = M\left(\sum_{n1,n2,n3,n4 \in F_i} \sum_{a,b} x_{n1}^a x_{n2}^a x_{n3}^b x_{n4}^b\right)$.

Величина V_i существенно зависит от возможности самопересечений слов, входящих в конфигурацию. Детальный учет составляющих показывает, что:

$$D(d(S, T)) = \frac{8n^2}{A^k} G, \text{ где } G = 1 + \frac{2}{A}\left(\frac{1 - A^{-(k-1)}}{1 - A^{-1}}\right) - \frac{2k-1}{A^k} + O\left(\frac{1}{n}\right)$$

и

$$e_0 = \frac{A^{k/2}}{2^{1/2}} \frac{(1 - 1/A^k)}{G^{1/2}}.$$

Для $k = 2$ точная формула для $D(d(S, T))$ имеет вид:

$$D(S, T) = \frac{8n^2}{A^2} G, \quad \text{где} \quad G = 1 + \frac{2}{A} - \frac{3}{A^2} - \frac{1}{2n}\left(1 - \frac{6}{A} + \frac{5}{A^2}\right).$$

Результаты и обсуждение. Несмотря на значительные усилия в течение последних двух десятилетий при поиске гомологии двух последовательностей до сих пор неизвестны алгоритмы с линейной (или хотя бы значительно более низкой, чем квадратичная) трудоемкостью. В связи с отсутствием линейных по вычислительной сложности алгоритмов было бы целесообразно разработать алгоритмы, трудоемкость которых является линейной в среднем. Эта задача даже для случая максимальной общей подпоследовательности до сих пор не решена. Представляется существенным для разработки такого алгоритма использовать методы фильтрации, применяющиеся при быстром поиске гомологий по банкам генетической информации. Настоящая работа является попыткой решения проблемы именно в этом направлении, хотя полученные в ней оценки для эффективности фильтрации еще не позволяют утвердительно ответить на вопрос о существовании линейного в среднем алгоритма для проблемы максимальной общей подпоследовательности. Следует также отметить, что в работе получены только оценки параметров функции распределения статистического расстояния, а вопрос о виде этой функции (и возможности ее аппроксимации) остается открытым.

Автор выражает признательность Н. Н. Александрову, А. М. Леонтовичу, А. А. Миронову и А. В. Финкельштейну за обсуждение алгоритмов быстрого поиска гомологий и В. Г. Тимковскому — за обсуждение проблем вычислительной сложности задачи поиска максимальной общей подпоследовательности.

FILTRATION EFFICIENCY IN RAPID HOMOLOGY SEARCH STATISTICAL ALGORITHMS

P. A. Pevzner

Mathematical Laboratory of the Institute for Genetics of Microorganisms, Moscow, USSR

Summary

Upon searching local homologies in long sequences (homology search in nucleotide and amino acid sequences banks, selection of optimal oligonucleotide probes etc.) the necessity of a «rapid» homology search becomes acute. Quadratic complexity of the dynamic programming algorithms (Needleman—Wunsch and Sellers type) forces the employment of filtration methods, that permits one to reject the sequences with a low homology level (among the filtration methods the l—tuple analysis and the statistical method of Mironov—Alexandrov were used). But theoretical substantiations of such algorithms have not been made yet. The present work introduces the notion of filtration efficiency and the efficiency of several filters is given. It was shown that in the l—tuple analysis the filtration efficiency is associated with the potential extension of the original four—letter alphabet. The formulas that allow choosing the filtration parameters are presented.

СПИСОК ЛИТЕРАТУРЫ

1. *Needleman S. B., Wunch C. D.* A general method applicable to the search for similarities in the amino acids sequences of two proteins // J. Mol. Biol.— 1970.— 48, N 2.— P. 443—453.
2. *Sellers P. H.* On the theory and computations of evolutionary distance // SIAM J. Appl. Math.— 1974.— 26, N 4.— P. 787—793.
3. *Smith T. F., Waterman M. S., Burks C.* The statistical distribution of nucleic acids similarities // Nucl. Acids Res.— 1985.— 13, N 2.— P. 645—656.
4. *Parallel* computing'85/A. Lyall, C. Hill, J. F. Collins, A. F. W. Coulson / Eds M. Felmeier et al.— North-Holland, 1986.— P. 235—258.
5. *Gotoh O., Tagashira Y.* Sequence search on supercomputer // Nucl. Acids Res.— 1986.— 14, N 1.— P. 57—64.
6. *Collins J. F., Coulson A. F. W.* Applications of parallel processing algorithms for DNA sequence analysis // Ibid.— 1984.— 12, N 1.— P. 181—192.

7. *Goad W. B., Kanehisa M. I.* Pattern recognition in nucleic acids sequences. 1. A general method for finding local homologies and symmetries // Ibid.— 1982.— **10**, N 1.— P. 247—263.
8. *Masek W. J., Paterson M. S.* How to compute string-edit distance quickly // Time warps, string edits and macromolecules: the theory and practice of sequence comparison.— Addison-Wesley, 1983.— P. 337—349.
9. *Ukkonen E.* On approximate string matching : Proc. Int. Conf. Found. Comp. Theor. Lecture notes in computer // Science.— 1983.— **158**.— P. 487—496.
10. *Ройтберг М. А.* Алгоритм определения гомологии первичных структур.— Пущино, 1984.— 24 с.— (Препринт / АН СССР. НИВЦ АН СССР).
11. *Hsu W. J., Du M. V.* New algorithms for LCS problem // J. Comput. Syst. Sci.— 1984.— **29**, N 2.— P. 133—152.
12. *Kumar S. K., Rangan C. P.* A linear space algorithm for the LCS problem // Acta inf.— 1987.— **24**.— P. 353—362.
13. *Гусев В. Д., Куличков В. А., Титкова Т. Н.* Анализ генетических текстов. 1. *L*-граммные характеристики // Вычислительные системы.— 1980.— **83**, № 1.— С. 11—33.
14. *Fondrat C., Dessen P., Le Beux P.* Principle of codification for quick comparison with the entire biomolecule databanks // Nuck. Acids Res.— 1986.— **14**, N 1.— P. 197—204.
15. *Колчанов Н. А., Соловьев В. В., Жарких А. А.* Контекстные методы теоретического анализа генетических макромолекул (ДНК, РНК и белков // Итоги науки и техники.— М. : ВИНИТИ, 1985.— С. 6— 34.— (Молекуляр. биология; Т. 21).
16. *Mironov A. A., Alexandrov N. N.* Statistical method for rapid homology search // Nucl. Acids Res.— 1988.— **16**, N 11.— P. 5169—5174.
17. *Соловьев В. В., Рогозин И. Б.* Быстрый поиск гомологий по банкам нуклеотидных и аминокислотных последовательностей // Теорет. исследования и банки данных по молекуляр. биологии и генетике.— Новосибирск, 1988.— С. 40.
18. *Жарких А. А., Ржецкий А. Ю., Родин С. Н.* Оценка гомологии последовательностей по частотам олигонуклеотидов. I. Ограничения на состав нуклеотидов в ДНК.— Новосибирск, 1988.— 25 с.— (Препринт / АН СССР. Ин-т цитологии и генетики).
19. *Жарких А. А., Ржецкий А. Ю.* Оценка гомологии последовательностей по частотам олигонуклеотидов. II. Анализ распределения замен в нуклеотидных последовательностях. П-статистика.— Новосибирск, 1988.— 26 с.— (Препринт / АН СССР. Ин-т цитологии и генетики).
20. *Жарких А. А., Ржецкий А. Ю.* Определение гомологии последовательностей по частотам олигонуклеотидов // Теорет. исследования и банки данных по молекуляр. биологии и генетике.— Новосибирск, 1988.— С. 40.
21. *Срелец В. Б., Шиндялов И. Н.* Быстрый поиск сходства между аминокислотными последовательностями // Там же.— С. 53.
22. *Стрелец В. Б., Шиндялов И. Н.* Быстрые методы поиска сходства между аминокислотными последовательностями.— Новосибирск, 1988.— 25 с.— (Препринт / АН СССР. Ин-т цитологии и генетики).
23. *Wilbur W. J., Lipman D. J.* Rapid similarity searches on nucleic acids and protein data banks // Proc. Nat. Acad. Sci. USA.— 1983.— **80**, N 2.— P. 726—730.
24. *Kanehisa M.* Use of statistical criteria for screening potential homologies in nucleic acids sequences // Nucl. Acids Res.— 1984.— **12**, N 1.— P. 203—214.
25. *Lipman D. J., Pearson W. R.* Rapid and sensitive similarity searches // Science.— 1985.— **227**, N 6.— P. 1435—1440.
26. *Blaisdell B. E.* A measure of the similarity of sets of sequences not requiring sequence alignment // Proc. Nat. Acad. Sci. USA.— 1986.— **83**, N 10.— P. 5155—5159.
27. *Lawrence C. B., Goldman D. A., Hood R. T.* Optimized homology searches of the gene and protein sequence data banks // Bull. Math. Biol.— 1986.— **48**, N 5/6.— P. 569—583.
28. *Zhurkin V. B.* Periodicity in DNA primary structure is defained by secondary structure of the coded protein // Nucl. Acids Res.— 1981.— **9**, N 5.— P. 1963—1971.
29. *Певзнер П. А.* Эффективность фильтрации в методах быстрого поиска гомологий // Теорет. исследования и банки данных по молекуляр. биологии и генетике.— Новосибирск, 1988.— С. 63—64.
30. *Guibas L. J., Odlyzko A. M.* String overlaps, pattern matching and nontransitive games // J. Combin. Theory. A.— 1981.— **23**.— P. 183—208.
31. *Pevzner P. A., Borodovsky M. Yu., Mironov A. A.* Linguistics of nucleotide sequences 1: the significance of deviations from mean statistical characteristics and prediction of the frequencies of occurence of words // J. Biomol. Struct. and Dyn.— **6** N 5.— 1989.— P. 1013—1026.

ВНИИ генетики и селекции пром. микроорганизмов,
Москва

Получено 06.06.90