Bulletin of Mathematical Biology Vol. 46, No. 4, pp. 515–527, 1984.
Printed in Great Britain

0092-8240/84$3.00 + 0.00
Pergamon Press Ltd.

PATTERN RECOGNITION IN SEVERAL SEQUENCES: CONSENSUS AND ALIGNMENT

■ M. S. WATERMAN* and R. ARRATIA†
Departments of Mathematics*† and of Molecular Biology,*
University of Southern California,
Los Angeles, CA 90089, U.S.A.

■ D. J. GALAS‡
Department of Molecular Biology,
University of Southern California,
Los Angeles, CA 90089, U.S.A.

The comparison of several sequences is central to many problems of molecular biology. Finding consensus patterns that define genetic control regions or that determine structural or functional themes are examples of these problems. Previously proposed methods, such as dynamic programming, are not adequate for solving problems of realistic size. This paper gives a new and practical solution for finding unknown patterns that occur imperfectly above a preset frequency. Algorithms for finding the patterns are given as well as estimates of statistical significance.

1. Introduction. In the mathematical analysis of macromolecular sequences one of the most developed areas is the comparison of sequences. Varied and powerful dynamic programming methods have been developed for the optimal alignment of two sequences, for the best fit of one sequence 'into' another and for determining the best matching segments of two sequences. Various methods for more rapid comparison of sequences have also been developed that are particularly useful for screening data bases. The subject of sequence comparison is reviewed elsewhere in this issue (Waterman, 1984).

The methods currently available for comparison of two sequences are not as useful when applied to several sequences. Dynamic programming methods, for example, take time and storage $O[(2n)^r]$ to compare r sequences of length n. No previously known methods are adequate for these problems. In this paper we address the problem of comparison of several sequences, which is of considerable biological interest, and explicitly approximate the probability that a pattern is held in common by at least a preset percentage of the sequences. We introduce practical techniques that solve the problem of finding these 'consensus' patterns for a set of sequences.

* This author supported by a grant from the System Development Foundation.
† This author supported by NSF grant MCS-8301960 and by a grant from the System Development Foundation.
‡ This author supported by NIH grant GM19036.

The problem considered here is, in general terms, that of finding unknown patterns (words over the alphabet) that occur imperfectly at or above a preset frequency. The specific problems addressed are those of finding:

(i) unknown patterns that occur in r sequences, $x_1, x_2, x_3 \ldots, x_r$,
(ii) known patterns that occur in $x_1, x_2, x_3 \ldots, x_r$,

and

(iii) alignments of $x_1, x_2 \ldots, x_r$.

Algorithms and estimates of statistical significance of the patterns found by these algorithms are presented in the next sections.

The problem of determining common patterns among sequences has been considered an important one since the first sequence data for proteins and nucleic acids became available. Among the patterns that have clear biological significance are those defining genetic control regions in DNA and those determining structural or functional themes in protein sequences and their respective DNA coding regions. Previous attempts to devise algorithms for the detection of such patterns in several sequences were beset by various difficulties.

In Sadler *et al.* (1983) regulatory pattern analysis is considered, using the tools of dynamic programming. Since the presumptive regulatory patterns are small and occur inexactly, algorithms to find long common matches between two sequences are not of much use. The paper concludes ". . . these tools are of limited value".

Several attempts have been made to study these problems using the concepts of finite automata and regular expressions. See Aho, Hopcroft and Ullman (1974). For example, Abarbanel *et al.* (1984) implement regular expression searches in a form convenient for use in molecular biology. However, in all such programs it is necessary to know the approximate identity of the pattern being sought. In the present paper, we consider the problem of finding patterns of which there is no prior knowledge.

Stormo *et al.* (1982) use a concept from artificial intelligence, the perceptron, to find translation initiation sites in *E. coli* in mRNA sequences. Minsky and Papert (1969) provide a detailed review of these concepts. These methods are closely related to one developed by R. A. Fisher, called linear discriminate analysis. See Gnanadesikan (1977) for a discussion of this statistical technique. These techniques may prove useful for several biological problems and should be more fully explored for nucleic acid and protein sequence data.

The techniques we develop here are related to work of Parzen (1962), who proposes a method of estimating probability density functions. See Waterman and Whiteman (1978) for a discussion of the technique and its application to

experimental data. Queen *et al.* (1982) propose a method closely related to ours although their data analysis differs in critical ways which limit the utility of their procedure. In addition, Dumas and Ninio (1982) treat a sequence as a string of overlapping n-mers and Marliere (1982) analyzes tRNA sequences by computing a score for each overlapping n-mer. Our algorithms make use of similar ideas.

2. Basic Algorithm. In this section, an algorithm is presented which applies to the three problems described in the Introduction.

The data are a set of r sequences

$$X = \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \\ \boldsymbol{x}_r \end{bmatrix} = \begin{bmatrix} x_{11}x_{12} \ldots x_{1n_1} \\ x_{21}x_{22} \ldots x_{2n_2} \\ \ldots \\ x_{r1}x_{r2} \ldots x_{rn_r} \end{bmatrix}$$

where x_{ij} are members of a finite alphabet, such as {A, C, G, T}.

The analysis is based on the occurrence of k letter words, which may be ordered lexicographically (AA . . . AA, AA . . . AT, . . . , TT . . . TT) and put into correspondence with the integers 0, 1, . . . , $4^k - 1$. Since we are concerned with the occurrence of similar patterns, we must define sets of similar words which we will call neighborhoods. Neighborhoods of words are defined by functions, f, mapping a k-letter word into a set of k-letter words. For example, if w = AT and $f(w) = \{w' : w'$ is one mismatch from $w\}$ then $f(w) = f(\text{AT}) = \{\text{CT, GT, TT, AA, AC, AG}\}$. A neighborhood is determined from a list of such functions.

Basic to our analysis is an enumeration of the words and their neighbors. For a string $\boldsymbol{y} = y_1y_2 \ldots y_L$, of length L, define

$$q_{wd} = |\{m : w \in f_d(y_m y_{m+1} \ldots y_{m+k-1}),\ 1 \leqslant m \leqslant L + 1 - k\}|$$

where $|B|$ is the number of elements in the set B. In other words, q_{wd} is the number of times word w is an f_d neighbour of some k-letter words in $y_1 \ldots y_L$.

For example, let $k = 2$ and $\boldsymbol{y}$ = ACTAAA. Consider two neighborhood functions $f_0(\cdot)$, the exact match function, and $f_1(\cdot)$, the single-mismatch function. Then the matrix, Q, of neighbor occurrences [$Q = (q_{wd})$] is

$$Q = \begin{array}{c} \\ \text{AA} \\ \text{AC} \\ \text{AG} \\ \text{AT} \\ \text{CA} \\ \text{CC} \\ \text{CG} \\ \text{CT} \\ \text{GA} \\ \text{GC} \\ \text{GG} \\ \text{GT} \\ \text{TA} \\ \text{TC} \\ \text{TG} \\ \text{TT} \end{array} \begin{array}{c} \begin{array}{cc} f_0 & f_1 \end{array} \\ \left[\begin{array}{cc} 2 & 2 \\ 1 & 2 \\ 0 & 3 \\ 0 & 4 \\ 0 & 4 \\ 0 & 2 \\ 0 & 1 \\ 1 & 0 \\ 0 & 3 \\ 0 & 1 \\ 0 & 0 \\ 0 & 1 \\ 1 & 2 \\ 0 & 2 \\ 0 & 1 \\ 0 & 2 \end{array}\right] \end{array}.$$

CA, for example, does not occur exactly in y, but y does have four words that are one mismatch from CA.

Next we compute $\tilde{Q} = (\tilde{q}_{wd})$ where $\tilde{q}_{wd}$ is 1 if $d = \min\{l : q_{wl} \neq 0\}$, and $\tilde{q}_{wd}$ is 0 otherwise. For the above example,

$$\tilde{Q} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}.$$

The idea is to count only the best occurrence of a word w in the string y.

The search of the sequence set, X, will proceed by performing a search for the most frequently occurring word in the block from column j to

column $j + W - 1$. The window width, W, is a parameter set by the user. Too broad a search will give insignificant results, too narrow a search will usually not find a desired pattern. In Section 5 the statistical significance is assessed in detail. The sequences searched are

$$x_{1,j}x_{1,j+1} \cdots x_{1,j+W-1}$$

$$x_{2,j}x_{2,j+1} \cdots x_{2,j+W-1}$$

$$\cdots$$

$$x_{r,j}x_{r,j+1} \cdots x_{r,j+W-1}.$$

For each line i, $1 \leqslant i \leqslant r$, the $\tilde{Q} = \tilde{Q}(i)$ matrix above is calculated and a summation matrix

$$V = \sum_{i=1}^{r} \tilde{Q}(i)$$

is found. $V = (v_{wd})$ has the interpretation that v_{wd} is the number of lines for which the best occurrence of word w is as a dth neighbor.

Different occurrence scores can be calculated from V. First,

$$v_w = \sum_{d \geqslant 1} v_{wd}$$

is the number of lines in which any neighbor of word w occurs. A score weighted for the distance between the word and its neighbor is more appropriate. The general form is

$$s_w = \sum_{d \geqslant 1} \lambda_d v_{wd}.$$

A winning word w, the 'most common' pattern, satisfies

$$\max_{w'}(s_{w'}) = s_w \equiv s.$$

The scoring used in the programs discussed here is

$$\lambda_d = \frac{\text{Number of letters in common between } w \text{ and members of } f_d(w).}{k}$$

In particular, with this weighting

$$\lambda_{\text{exact}} = 1$$

and

$$\lambda_{d \text{ mismatches}} = 1 - \frac{d}{k}.$$

The algorithm begins with a set $f_0, f_1, \ldots$ of neighborhood functions and a

window width W. The winning word score is computed for $j = 1, 2, \ldots$. Estimates of statistical significance can be obtained (see Section 5 below) from the sequence probability distributions and the neighbors $f_0, f_1, \ldots$ and used to set W. For the problems of finding unknown and known patterns, specific algorithms are presented in Sections 3 and 4.

3. Search for Unknown Patterns. The assumption here is that the 'consensus' pattern among the set of sequences is unknown. This is the problem of most interest and the one which has attracted much attention from biologists because of the need to find significant sequence patterns that define specific functions among the rapidly expanding sequence data. The 'Pribnow box' from bacterial promoters or the ribosome binding site in bacteria, the 'Shine–Dalgarno sequence', are examples of such patterns that define part of the function of transcription initiation (Hawley and McClure, 1983) and translation initiation (Steitz and Jakes, 1975) respectively. In the first example, the pattern of this 'box' only becomes well-defined for a relatively large set of promoter sequences. That is to say, the 'shadow' of the consensus pattern is not very precise among the sequences. The approach to finding the pattern that casts this shadow that prescribes exhaustively comparing all subsequences of the set, requires an enormous number of operations even for short sequences and points to the need for efficient algorithms for pattern recognition of this kind. If there were 100 sequences and only two positions for each sequence, there would be $2^{100} \approx 1.26 \times 10^{30}$ possible overall configurations of the 100 sequences. Using this method to find patterns thus presents a hopeless task. The algorithm we present, on the other hand, here takes time approximately proportional to

$$(W - k + 1)\left(\sum_{i=1}^{r} n_i\right)\left(\sum_{d} |f_d|\right)4^k.$$

The search begins, as in Section 2, with a set of neighbors $f_0, f_1, \ldots$ and a window width W. The scores of all words are calculated at each window position and the best one determined. If desired, the sequences can be 'aligned' on a statistically significant word or pattern: a 'column' can be formed. Forming a column in this manner, on a word such as TATAAT in the bacterial promoters, may allow a second pattern, such as the TTGACA for the upstream, '—35', box, to be located much more easily.

In a test of these concepts on 59 sequences of bacterial promoter regions approximately 60 bases long, $k = 6$ was used. With neighborhood functions: f_0 = exact, f_1 = 1 mismatch, f_2 = 2 mismatches, window width $W = 12$ was used. For neighborhood functions: f_0 = exact, f_1 = 1 mismatch, f_2 = 2 mismatches, f_3 = 3 mismatches, $W = 9$ was used. We are also able to include

insertions and deletions, either separately or along with mismatches. For instance, we may use 1 mismatch and 1 insertion, or 1 mismatch and 1 deletion. With the above parameters the programmed algorithm easily found both the −10 consensus and the −35 consensus.

4. Search for Known Patterns. In this section a pattern of interest $y = y_1 y_2 \ldots y_m$ is assumed to be known. The Pribnow box, y = TATAAT, for example, is such a pattern in the example of bacterial promoters (Hawley and McClure, 1983). Similarly, a particular complete promoter sequence of length 60 might be chosen as a known pattern. In any case the algorithm outlined below would find the best 'shadow' of the pattern y in the set X of sequences. This problem is clearly a special case of the pattern recognition problem described in the previous section.

First choose a word w of length k from $y = y_1 \ldots y_m$. In the Pribnow box case this might be TATAAT with $k = 6$ but it also might be a k letter subword of a full (longer) sequence. Within a window of width W, the calculations of Section 2 are performed where

q_{w1} = number of exact occurrences of w in row i

q_{w2} = number of one mismatch occurrences of w in row i

. . . .

Only one line of Q, the wth row, is used in these calculations since we assume prior knowledge of the desired patterns.

5. Estimates of Statistical Significance. For ease of analysis, we analyze the score equal to the number of lines in which any neighbor occurs. (That is, $\lambda_d = 1.0$.)

Assume that, independently in every position on every line, each of the four letters, A, C, G, T appears with probability 1/4. For any word z of length k, the probability that the letters in k given positions spell z exactly is 4^{-k}. Let $F = \Sigma_d |f_d|$ be the total number of k-letter neighbors of a given word w. The probability that k random letters form a neighbor of w is $4^{-k}F$.

In Section 5.1 we present the essentials of our analysis and give some numerical examples. Then, in Section 5.2, more details are presented.

5.1 Survey of the analysis. Assume that w is a given pattern of length k, having F neighbors of length k. We use

$$\alpha = (W - k + 1)(F)4^{-k}$$

to approximate the probability that some neighbor of w occurs, on a given line, with a given position of the window of width W. Thus, if the data were random, for each word and window position j, one would expect

approximate matches to w on about αr of r lines. A fraction $\beta > \alpha$ is extremely unlikely.

Suppose we are looking for a pattern common to some preset fraction $\beta > \alpha$ of the r sequences. For word w and window position j, the probability that at least βr lines yield approximate matches to w can be estimated as

$$\exp[-rH(\beta, \alpha)], \text{ where } H(\beta, \alpha) = \beta \log\left(\frac{\beta}{\alpha}\right) + (1-\beta)\log\left(\frac{1-\beta}{1-\alpha}\right) > 0$$

is the entropy of β relative to α. Now there are n choices for the location j of the window, and, if the pattern w is unknown, there are 4^k choices for the word w. Then our estimates of significance level p are

$$\text{known pattern:} \quad p = n\exp[-rH(\beta, \alpha)]$$

$$\text{unknown pattern:} \quad p = n4^k\exp[-rH(\beta, \alpha)].$$

Thus p is an upper bound, for random data, on the probability that in some window position, an approximate match occurs on a fraction greater than or equal to β of the r lines. If these estimates exceed 1, we use 1 instead.

Two examples are presented in Table 1 for patterns of length $k = 6$, with $r = 59$ sequences of length $n = 60$. For the first example, the neighborhood is 0, 1 or 2 mismatches, and $F = 1 + 18 + 135 = 154$. The second example has a neighborhood of 0, 1, 2 or 3 mismatches, and $F = 1 + 18 + 135 + 540 = 694$.

TABLE I

Estimates of Statistical Significance for Patterns of Length $k = 6$ in $r = 59$ Sequences with 60 Bases

F	W	α	β	$H(\beta, a)$	e^{-rH}	Known pattern p	Unknown pattern p
154	12	0.263	0.75	0.515	6.3×10^{-14}	3.8×10^{-12}	1.5×10^{-8}
154	14	0.338	0.75	0.354	8.7×10^{-10}	5.2×10^{-8}	2.1×10^{-4}
154	16	0.414	0.75	0.233	1.1×10^{-6}	6.3×10^{-5}	2.6×10^{-1}
694	7	0.339	0.75	0.353	9.2×10^{-10}	5.5×10^{-8}	2.3×10^{-4}
694	8	0.508	0.75	0.123	7.2×10^{-4}	4.3×10^{-2}	1.0

5.2 Details of the analysis. Within a window of length W, there are $l = W - k + 1$ places for a block of k consecutive letters. For each of these l choices, consider the event, of probability $a = 4^{-k}F$, that some neighbor of word w occurs at that position. Regardless of the dependence of these l events, al is an upper bound on the probability of their union. We will use

$al = 4^{-k}F(W - k + 1)$ as our estimate of the probability α that some neighbor of w occurs elsewhere in the window.

If the l events were independent, the probability of their union is $1 - (1 - a)^l$, which is closely approximated by al whenever $al < 1$. An exact bound is $(1 - 1/e)\min(al, 1) \leqslant 1 - (1 - a)^l \leqslant \min(al, 1)$, for $l = 1, 2, \ldots$ and any $0 \leqslant a \leqslant 1$. The l events here are dependent, in a complex way that we cannot analyze, and furthermore, the dependence varies with the word w. For example, the events for two adjacent positions, $\{x_1x_2 \ldots x_k$ is a neighbor of $w\}$ and $\{x_2x_3 \ldots x_{k+1}$ is a neighbour of $w\}$, are positively correlated if w = AAAAAA, and negatively correlated if w = ACGTAC, using the neighborhoods of one or two mismatches. For further discussion of this dependence see Waterman (1983) and Breen *et al.* (1985).

Consider a fixed window position j. Independently, on each of r lines, there is the event of probability α that some neighbor of w occurs. The probability that exactly m of these events takes place is

$$\binom{r}{m}\alpha^m(1-\alpha)^{r-m}.$$

For a prescribed threshold β between 0 and 1, the probability that some neighbor of w occurs, within the window, on at least βr of the r lines, is

$$\sum_{\beta r \leqslant m \leqslant r}\binom{r}{m}\alpha^m(1-\alpha)^{r-m}.$$

As long as $\beta > \alpha$ and r is large, a good approximation for this sum is the large deviation estimate, P (at least βr successes in r independent trials with individual success probability $\alpha < \beta) \approx e^{-rH(\beta,\alpha)}$. Here $H(\beta, \alpha)$ is the relative entropy, $H(\beta, \alpha) = \beta\log\left(\frac{\beta}{\alpha}\right) + (1 - \beta)\log\left(\frac{1-\beta}{1-\alpha}\right)$. Note that when $\beta = 1$, the probability of finding a neighbor of w on all r lines is α^r, and $H(\beta, \alpha)$ reduces to $-\log \alpha$, so that the approximation is exact: for $\beta = 1$, $e^{-rH(\beta,\alpha)} = e^{r\log\alpha} = \alpha^r$.

Finally, consider the shifting window position. At each $n - W + 1 \approx n$ possible window positions, the event that some neighbor of w occurs within that window on at least βr of the r lines is approximately $e^{-rH(\beta,\alpha)}$. The estimate of significance $ne^{-rH(\beta,\alpha)}$ is an upper bound on the probability that, for a given word w, some window position reveals an approximate match to w on at least fraction β of the lines.

The bound above can be used to choose a window size W. Pick a significance level p, e.g. 0.01 or 0.001. To get $ne^{-rH(\beta,\alpha)} \leqslant p$ we need

$$H(\beta, \alpha) \geqslant \frac{1}{r}\log\left(\frac{n}{p}\right);$$

so we let

$$\varepsilon = \frac{1}{r}\log\left(\frac{n}{p}\right).$$

For $\beta = 1$, this says $-\log \alpha \geqslant \varepsilon = \frac{1}{r}\log\left(\frac{n}{p}\right)$, i.e. $\alpha \leqslant e^{-\varepsilon} = \left(\frac{p}{n}\right)^{1/r}$. Now $\alpha \leqslant 4^{-k}F(W-k+1)$ so we solve for W:

$$\left(\frac{p}{n}\right)^{1/r} = 4^{-k}F(W-k+1),$$

$$W = \left(\frac{p}{n}\right)^{1/r}(4^k/F) + (k-1).$$

For $\beta < 1$, we need approximations to solve $H(\beta, \alpha) = \varepsilon$. Here ε is small, so α will be slightly less than β; we let $\alpha = \beta(1-\delta)$. Using $\log(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} \ldots$,

$$H(\beta, \alpha) = H(\beta, \beta - \beta\delta) = \beta\log\left(\frac{\beta}{\beta - \beta\delta}\right) + (1-\beta)\log\left(\frac{1-\beta}{1-\beta+\beta\delta}\right)$$

$$= \beta\log\left(\frac{1}{1-\delta}\right) + (1-\beta)\log\left[1 + \left(\frac{\beta}{1-\beta}\right)\delta\right]^{-1}$$

$$\approx \beta\left(\delta + \frac{\delta^2}{2}\right) + (1-\beta)\left[-\left(\frac{\beta\delta}{1-\beta}\right) + \left(\frac{\beta\delta}{1-\beta}\right)^2\Big/2\right] = \frac{\beta\delta^2}{2(1-\beta)}.$$

Thus to solve $H(\beta, \alpha) = \varepsilon$, we take $\alpha = (1-\delta)\beta$ with

$$\delta = \left(\frac{2(1-\beta)}{\beta}\varepsilon\right)^{1/2}.$$

6. Aligning Sequences. Obviously most of the sequences must already be approximately aligned. This may be done by finding long matches in all sequences, or by using prior knowledge of biological functions such as the beginning of coding regions.

The algorithm and statistical significance estimates have already been described. The approximate alignments can be improved, although not drastically altered. The algorithm begins at some aligned positions and searches until statistically significant matches are found.

Alignments on features other than matches are possible. For example, the Noller–Woese (1981) method utilizes both matches and helical regions to perform their phylogenetic analysis of ribosomal RNA secondary structure. In Noller, Waterman and Woese (in preparation) these methods are given a rigorous basis. The double-stranded regions are found by positioning a window and then searching by moving another window across the approximately aligned sequences until a region with significant base pairing is located.

7. Application to Biological Problems. Since DNA and protein sequence information has been available, various short patterns have been identified as having particular functional significance. In several presumptive regulatory DNA sequences, for example, a candidate 'consensus' sequence, specifying a particular biological function, has been identified by simple inspection of several examples of DNA sequences known to determine such a function. The 'Pribnow box' of bacterial promoter sequences (Pribnow, 1975) and the 'Goldberg–Hogness' box of eukaryotic polymerase II promoter sequences are examples of such feature extraction by inspection (Schaller *et al.*, 1975; Goldberg, 1979; Breathnoch and Chambon, 1981). The difficulties with this process are evident: there is no unambiguous definition of a 'consensus' sequence; the subjective nature of the process introduces arbitrary, unstated choices (such as alignment of the "boxes"); the features that are evident from the comparison of single letters may not be the most important features of the functional pattern and it is not clear to what extent the observed features are statistically significant. There are other potential problems, some of which have been addressed by previous attempts at pattern recognition, as briefly discussed in the introduction. The general nature of these problems has been discussed previously (Sadler *et al.*, 1983; Smith *et al.*, 1981).

The method presented here overcomes many of the difficulties referred to above by providing: a clear and explicit definition of a 'consensus' pattern; an algorithm for finding such patterns among many sequences and an analysis of the statistical significance of these patterns. Furthermore, the present method provides a general tool that can be used to detect more subtle patterns. We need not confine our attention to the standard alphabet {A, C, G, T} and single positions in the sequence. Since the method is equally applicable to any set of strings of symbols, we may map sequences in the standard alphabet into sequences in a sub-alphabet, or into an alphabet (or sub-alphabet) of dinucleotides, trinucleotides, etc. and search for patterns

in these non-standard alphabets. It has been suggested, for example, that important features of the DNA-protein recognition are in the array of functional groups in the grooves of B-form DNA (see Matthews *et al.*, 1982, for example) or in the set of sequence-specific 'twist' and 'roll' angles, etc. that modify the relatively uniform structure of the DNA molecule (Dickerson, 1983, Dickerson *et al.*, 1982; Anderson *et al.*, 1982). These features would be manifest in patterns in one of the sub-alphabet sequences mentioned above.

Regulatory signals in DNA sequences are particularly amenable to analysis by the proposed method. Initially, we have given particular attention to the bacterial promoter sequences, since they represent an extensive and well-studied set of sequences with known function (Hawley and McClure, 1983). The important patterns (in the standard alphabet) are reasonably well-determined, so this set is an ideal test case. The known patterns in the -10 and -35 regions are easily found by our programs. The detailed results of this study will be reported elsewhere. Among the functional sequences of particular interest for further study are: the eukaryotic promoter sequence for polymerases I, II and III, the mRNA capping site, the poly-A addition site, enhancer sequences, the splicing sites for *pol*II transcripts, ribosome binding sites in prokaryotes and eukaryotes, the binding sites for various proteins (CRP and various repressor proteins), hormone receptor sites, common features in sequences surrounding mutational 'hotspots' and several others. To be useful the present method only requires that we have several examples of sequences with closely similar functions. It is worth noting here that many sets of functional sequences exhibit a wide range of functional activities among the members of the set (promoters of various strengths, for example), and that the present method is easily modifiable to take this into account. It is simply a matter of using weights, indicating the activities, in the algorithm to extract the patterns from various sequences. In this manner a weighted, 'consensus' promoter, for example, can be specified. In general, the more that is known about the function of a sequence the more information can be extracted.

LITERATURE

Arbanel, R. M., P. R. Wienecke, E. Mansfield, D. A. Jaffe and D. L. Brutlag. 1984. "Rapid Searches for Computer Patterns in Biological Molecules." *Nucl. Acids. Res.* **12**, 263–280.

Aho, V. A., J. E. Hopcroft and J. D. Ullman. 1974. *The Design and Analysis of Computer Algorithms*. Menlo Park, CA: Addison-Wesley.

Anderson, W. F., Y. Takeda, D. H. Ahlendorf and B. W. Matthews. 1982. "Proposed Helix Super-Secondary Structure Associated with Protein–DNA Recognition." *J. Mol. Biol.* **159**, 745–751.

Breathnoch, R. and P. Chambon. 1981. "Organization Expression of Split Genes Coding for Proteins." *A. Rev. Biochem.* **50**, 344–383.

Breen, S., M. S. Waterman and N. Zhang. 1984. "Renewal Theory for Several Patterns." *J. Appl. Prob.* (in press).

Dickerson, R. E., H. R. Brew, B. N. Conner, R. M. Wing, A. V. Frantini and M. L. Kopha. 1982. "The Anatomy of A-B-, C and Z-DNA." *Science* **216**, 475–485.

Dickerson, R. E. 1983. "Base Sequence and Helix Structure Variation in B DNA." *J. Mol. Biol.* **166**, 419–441.

Dumas, J. P. and J. Ninio. 1982. "Efficient Algorithms for Folding and Comparing Nucleic Acid Sequences." *Nucl. Acids Res.* **80**, 197–206.

Gnanadesikan, R. 1977. *Methods for Statistical Data Analysis of Multivariate Observations.* New York: John Wiley.

Goldberg, M. L. 1979. Ph.D. thesis, Stanford University.

Hawley, D. K. and W. R. McClure. 1983. "Compilation and Analysis of *Escherichia Coli* Promoter DNA Sequences." *Nucl. Acids Res.* **11**, 2237–2255.

Marliere, P. 1982. "The Fossil Organization of Transfer-RNA Sequences." Unpublished manuscript.

Matthews, B. W., D. H. Ahlendorf, W. F. Anderson and Y. Takeda. 1982. "Structure of the DNA-binding Region of *Lac* Repressor Inferred from its Homology with *Cro* Repressor." *Proc. natn. Acad. Sci. U.S.A.* **79**, 1428–1432.

Minsky, M. and S. Papert. 1969. In "Perceptrons." MIT Press, Cambridge, MA.

Noller, H. F. and C. R. Woese. 1981. "Secondary Structure of 16S Ribosomal RNA." *Science* **212**, 403–410.

Parzen, E. 1962. "On the Estimation of Probability Density Functions and Mode." *Ann. Math. Statist.* **33**, 1065–1076.

Pribnow, D. 1975. "Bacteriophage T7 Early Promoters: Nucleotide Sequences of Two RNA Polymerase Binding Sites." *J. Mol. Biol.* **99**, 419–443.

Queen, C. M., N. Wegman and L. T. Korn. 1982. "Improvements to a Program for DNA Analysis: A Procedure to Find Homologies Among Many Sequences." *Nucl. Acids Res.* **10**, 449–456.

Sadler, J. R., M. S. Waterman and T. F. Smith. 1983. "Regulatory Pattern Identification in Nucleic Acid Sequences." *Nucl. Acids Res.* **11**, 2221–2231.

Schaller, H., C. Gray and K. Herrmann. 1975. "Nucleotide Sequence of an RNA Polymerase Binding Site from the DNA of Bacteriophage fd." *PNAS* **72**, 737–741.

Smith, T. F., M. S. Waterman and W. M. Fitch. 1981. "Comparative Biosequence Metrics." *J. Mol. Biol.* **18**, 38–46.

Steitz, J. A. and K. Jakes. 1975. "How Ribosomes Select Initiator Regions in mRNA: Base Pair Formation Between the 3′ Terminus of 16S rRNA and the mRNA During Initiation of Protein Synthesis in *E. coli*." *Proc. natn. Acad. Sci. U.S.A.* **72**, 4734–4738.

Stormo, G. D., T. D. Schneider, L. Gold and A. Ehrenfeucht. 1982. "Use of the 'Perceptron' Algorithm to Distinguish Translational Initiation Sites in *E. coli*." *Nucl. Acids Res.* **10**, 2997–3011.

Waterman, M. S. and D. E. Whiteman. 1978. "Estimation of Probability Densities by Empirical Density Functions." *Int. J. Math. Educ. Sci. Technol.* **9**, 127–137.

Waterman, M. S. 1983. "Frequences of Restriction Sites." *Nucl. Acids. Res.* **11**, 8951–8956.

Waterman, M. S. 1984. "General Methods of Sequence Comparison." *Bull. math. Biol.*