possible binding site as a matrix. When all of the sequences have been added to the analysis, the one with the highest information content is the best guess for the pattern of the binding site. The method works well on typical prokaryotic binding sites, and it is robust enough to work even with some erroneous data included.[20,21]

Summary

Matrices can provide realistic representations of protein/DNA specificity. In many cases simple mononucleotide-based matrices are adequate representations, but more complex matrices may be needed for other cases. Unlike simple consensus sequences, matrices allow for different penalties to be assessed for different changes to a binding site, a property that is essential for accurate description of a binding site pattern. When only a collection of binding site sequences is known, the best representation for the pattern is an information content formulation, based on both thermodynamic and statistical considerations. Quantitative data on relative binding affinities may be used to determine matrices that provide a best fit to the data. Matrix representations also provide an efficient method of aligning multiple sequences to identify binding site patterns that they have in common.

Acknowledgments

I thank Calvin Harley for sending me the *E. coli* promoter sequences on a floppy disk and George Hartzell for helping with the figures and Gerald Hertz for critical comments on the manuscript. Supported by National Institutes of Health Grant GM28755.

[21] G. Z. Hertz, G. W. Hartzell III, and G. D. Stormo, *Comput. Appl. Biosci.*, in press.

[14] Consensus Methods for DNA and Protein Sequence Alignment

By MICHAEL S. WATERMAN and ROBERT JONES

Introduction

The increasing body of nucleic acid sequence data has created interest among many scientists in computational approaches to macromolecular sequence analysis. Several international databases have been created in order to store the data in a useful format, both for archival and analysis purposes.[1] Both DNA and protein sequences databases are maintained. The value of simply having easy access to all membrane protein sequences, for example, is not to be underestimated. The quantity of data has naturally led to the development of computer approaches to sequence analysis.[2] The purpose of this chapter is to present some of the tools that we have created in order to analyze multiple sequences in a rigorous, efficient, and systematic way.

Much computer analysis of molecular sequences is directed toward discovery of biologically significant patterns. These patterns include homologous genes, RNA secondary structure, tRNA or structural RNAs, palindromes in DNA sequences, regulatory patterns in promoter regions, and protein structural patterns. Once the patterns have been located they can often be tested by experiment, as in the case of promoter elements. Evolutionary relationships, however, cannot be directly tested, and increasing emphasis is being attached to the discovery and interpretation of sequence evolution.

Sequence alignment is a popular approach to pattern analysis.[2] Computer alignments are often based on an explicit optimization function, rewarding matches and penalizing mismatches, insertions, and deletions. Sequence alignment often gives useful information about evolutionary or functional relationships between sequences. Our approach is based on what we refer to as consensus analysis.[2-4]

Consensus sequence analysis is usually performed by visual inspection of the sequences and by experiment. Of course, a protein binding site can only be verified by experiment, and analysis by "eye" can be biased. Thus, it is useful to have computer methods that can find consensus patterns best fitting explicitly stated criteria. Some algorithms have been developed along these lines,[2-4] and they are described here, along with some biological examples. Our earlier methods applied only to DNA; here we also describe recent extensions to protein sequences.

In 1970 Needleman and Wunsch[5] published an approach sequence comparison (alignment) using a dynamic programming algorithm. Their algorithm find maximum similarity between two sequences, where matches score positive weight and mismatches, insertions, and deletions

[1] C. Burks, J. W. Fickett, W. B. Goad, M. Kanehisa, F. I. Lewitter, W. P. Rindone. C. D. Swindell, C.-S. Tung, and H. S. Bilofsky, *CABIOS* **1,** 225 (1985).

[2] M. S. Waterman, ed., "Mathematical Methods for DNA Sequences." CRC Press, Boca Raton, Florida, 1988.

[3] M. S. Waterman, D. Galas, and R. Arratia, *Bull. Math. Biol.* **46,** 515 (1984).

[4] D. J. Galas, M. Eggert, and M. S. Waterman, *J. Mol. Biol.* **186,** 117 (1985).

[5] S. B. Needleman and C. Wunsch, *J. Mol. Biol.* **48,** 444 (1970).

score nonpositive weight. Mathematicians began to attempt to define a distance between sequences and so to construct a metric space. Sellers[6] obtained these results for single insertions and deletions, and later workers extended the work to multiple insertions and deletions.[7] While dynamic programming methods are very widespread in sequence analysis, there are severe restrictions in computation time with the extension of the dynamic programming methods to allow more than two sequences. A great many biological problems do involve more than two sequences. The consensus methods we have developed avoid the computational difficulties of dynamic programming by using a very different approach to sequence analysis.

The basis of the consensus method is an algorithm to find consensus words, with the degree of matching and alignment specified by the user of the program. We give the specifications of this method in the next section for DNA and protein sequences, along with examples. In this setting the consensus method finds patterns or words that are conserved in an unusual number of sequences. Then the basic method is extended, both for DNA and protein sequences, to an algorithm for sequence alignment. To illustrate the behavior of the algorithms, we have chosen two sequence sets, one DNA and the other protein. The DNA sequences are 19 promoters from the genome of vaccinia virus.[8] The protein sequence set is 16 proteins related to the *Escherichia coli ntrC* gene product.[9] We use these sequences to illustrate the use and power of the programs and the effect of varying certain parameters. We do not attempt to interpret the consensus pattern found in any biological context, but we invite anyone interested in these specific sequences and patterns to contact us for more detailed information.

Consensus Patterns

Now we give a general description of the consensus word algorithm. To begin, take a set of R sequences of length N

$$\begin{array}{cccc} a_{1,1} & a_{1,2} & \cdots & a_{1,N} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,N} \\ & & \vdots & \\ a_{R,1} & a_{R,2} & \cdots & a_{R,N} \end{array}$$

[6] P. Sellers, *SIAM J. Appl. Math.* **26,** 787 (1974).

[7] M. S. Waterman, T. F. Smith, and W. A. Beyer, *Adv. Math.* **20,** 367 (1976).

[8] M. Mars and G. Beaud, *J. Mol. Biol.* **198,** 619 (1987).

[9] B. T. Nixon, C. W. Ronson, and F. M. Ausubel, *Proc. Natl. Acad. Sci. U.S.A.* **83,** 7850 (1986).

These sequences can be taken to be initially aligned on some biologically or statistically determined feature. The true alignment is, of course, unknown except approximately. Now we give an algorithm for locating consensus words of a given size. By way of comparison, the usual methods of sequence analysis align on single letters, that is words of length 1.

Of course, a concept basic to our algorithm is that of consensus word. The definition has been given in earlier work[2-4] and will be briefly reviewed here. First, take a fixed word size k and a word w of length k; there are 4^k such words in DNA and 20^k in proteins. Next, define the window width W which gives the width of sequence in which a consensus word can be found and thus defines the amount of shifting allowed in matching consensus words. The sequences starting at column $j + 1$ with window width W appear as

$$\begin{array}{cccc} a_{1,j+1} & a_{1,j+2} & \cdots & a_{1,j+W} \\ a_{2,j+1} & a_{2,j+2} & \cdots & a_{2,j+W} \\ & & \vdots & \\ a_{R,j+1} & a_{R,j+2} & \cdots & a_{R,j+W} \end{array}$$

To begin, we search the first sequence of the window for matches to our word w. An exact match to w is called a $d = 0$ neighbor while a 1-letter mismatch from w is called a $d = 1$ neighbor, and so on. For protein sequences, for example, it is desirable to distinguish the many types of $d = 1$ mismatches by different weightings based on amino acid similarity. It is possible to include insertions and deletions in this list of neighbors. We may decide, e.g., to limit the amount of mismatch to $d = 0, 1, 2$ and not find w in a portion of sequence unless it is within this neighborhood. Let $q_{w,d}$ equal the number of lines that the best occurrence of w is as a dth neighbor. Each of these occurrences receives weight λ_d. The score of word w in this window is

$$s_{j+1,j+W}(w) = \sum_d \lambda_d q_{w,d}$$

A best scoring word is word w^* satisfying

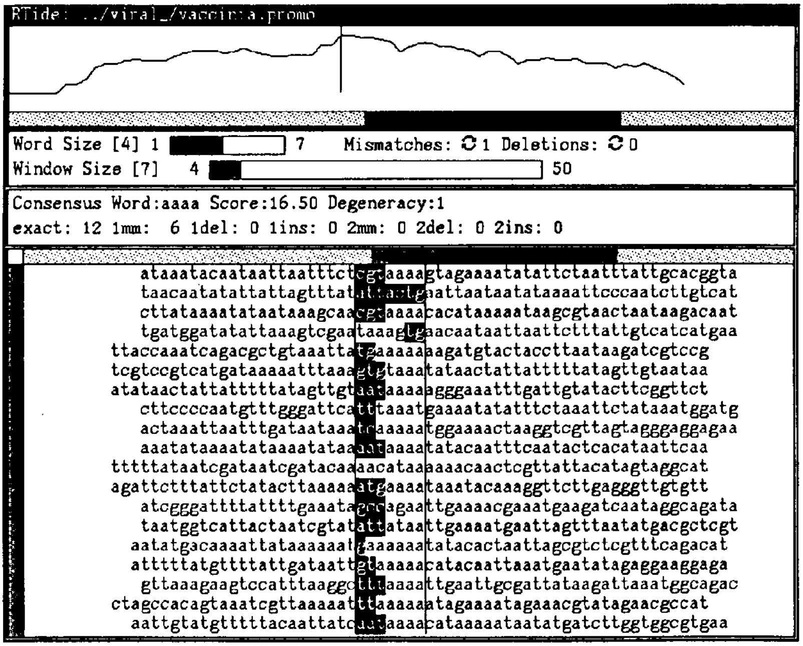
$$s_{j+1,j+W}(w^*) = \max_w s_{j+1,j+W}(w)$$

DNA Consensus Patterns

In the case of DNA all $d = 1$, $d = 2$, . . . mismatches are considered identical in weight. While more complex weighting schemes are easy to incorporate, we have found that it is adequate to score a word by the fraction of letters matching the consensus word. Thus, for a *d*-letter mismatch to a *k*-letter consensus word, $\lambda_d = 1 - d/k$.

To perform the computations, each word in the window is read and its neighborhood calculated. The possible words in a neighborhood are found by a simple combinatoric scheme; since all $d = 2$ mismatches receive the same weight it is only necessary to enumerate all $k(k-1)/2$ of these mismatches. When the portion of each sequence in the window has been examined, the best score each of the 4^k possible consensus words is retained.

Our consensus method for DNA sequences has been implemented in a program called RTIDE written in C and using the SunView window system. Figure 1 shows a typical screen display from this program. The aligned sequences are displayed at the bottom, the displayed consensus word and score of the word are shown above, along with parameters for the run. At top is a plot of the consensus word score against window position for the alignment. Peaks in this plot indicate regions of conservation that



```
RTide: ../viral_/vaccinia.promo

Word Size [4] 1                7     Mismatches: ↻1 Deletions: ↻0
Window Size [7]    4                                    50

Consensus Word:aaaa Score:16.50 Degeneracy:1
exact: 12 1mm:  6 1del: 0 1ins: 0 2mm: 0 2del: 0 2ins: 0

   ataaatacaataattaatttctcgtaaaagtagaaaatatattctaatttattgcacggta
   taacaatatattattagtttatattactgaattaataatataaaattcccaatcttgtcat
   cttataaaatataataaagcaacgtaaaacacataaaaataagcgtaactaataagacaat
   tgatggatatattaaagtcgaataaagtgaacaataattaattctttattgtcatcatgaa
ttaccaaatcagacgctgtaaattatgaaaaaaagatgtactaccttaataagatcgtccg
tcgtccgtcatgataaaaatttaaagtgtaaatataactattatttttatagttgtaataa
atataactattatttttatagttgtaataaaaagggaaatttgattgtatacttcggttct
   cttccccaatgtttgggattcatttaaatgaaaatatatttctaaattctataaatggatg
   actaaattaatttgataataaatcaaaaatggaaaactaaggtcgttagtagggaggagaa
   aaatataaaatataaaatataaaataaaatatacaatttcaatactcacataattcaa
tttttataatcgataatcgatacaaaacataaaaaacaactcgttattacatagtaggcat
agattctttattctatacttaaaaaatgaaaataaatacaaaggttcttgagggttgtgtt
   atcgggattttattttgaaatagccagaattgaaaacgaaatgaagatcaataggcagata
   taatggtcattactaatcgtatattataattgaaaatgaattagtttaatatgacgctcgt
  aatatgacaaaattataaaaaatgaaaaaatatacactaattagcgtctcgtttcagacat
  atttttatgttttattgataattgtaaaaacatacaattaaatgaatatagaggaaggaga
   gttaaagaagtccatttaaggctttaaaattgaattgcgattataagattaaatggcagac
ctagccacagtaaatcgttaaaaatttaaaaaatagaaaatagaaacgtatagaacgccat
  aattgtatgtttttacaattatcaataaaacataaaaataatatgatcttggtggcgtgaa
```

FIG. 1. Typical screen display of the program RTIDE. The sequence alignment is shown in the bottom window. At top is a plot of maximum score against window position in which peaks represent conserved regions.

may be of biological interest; these patterns can then be examined in detail and sequences realigned on the consensus words in any particular window. Through cycles of realignment and analysis it is possible to identify and refine conserved sequence patterns. The nature of the method involves variation of several parameter settings during an analysis session. It is possible to adjust window width, word size, and degree of matching required as well as alignment of the sequences. To facilitate this interaction all features of the program are controlled through the mouse.

The effect of varying the word size on the consensus score is shown in Fig. 2. No mismatches are allowed in these runs, and the window size is increased along with word size to keep the number of words per window constant. There are 16 words of size 2, and the score is the maximum scoring word over all sequences. Since there are so few words, all window positions have a high score and no features are clearly resolved. At word size 3 the graph shows a few features, with the highest score indicated by the dashed line. At word size 4 the central peak is resolved from the background. As word size increases further this peak becomes smaller and

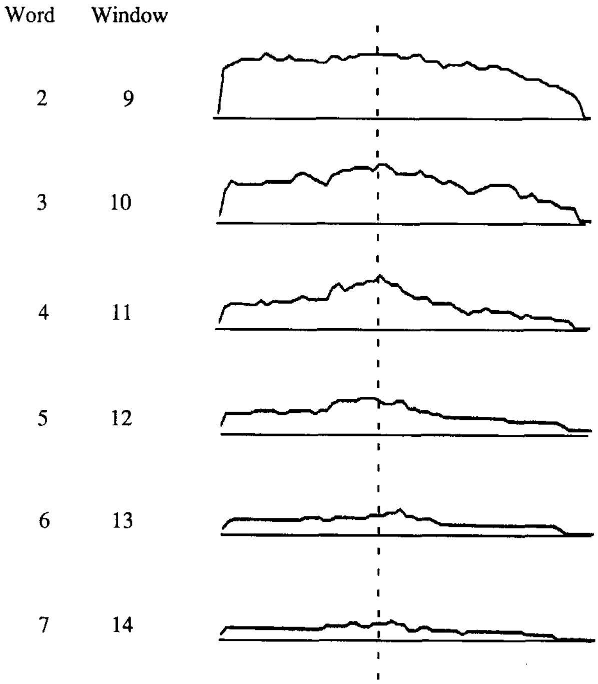


FIG. 2. Effect of varying word size from 2 to 7 for DNA sequence alignment. The window size is varied to maintain a fixed number of words per window. No mismatches are allowed.

merges into the background of nonconserved sequence. The word size at which a peak is most evident is a clear indication of the size of the feature that is conserved; in the present case that word size is 4 nucleotides.

In many cases the initial alignment in which sequences are supplied will not be optimal for a given conserved sequence feature. Varying the window size is a way to accommodate poorly aligned sequence sets. Figure 3 shows the effect of variation of window size from 4 to 20 for the DNA data set. Using a window equal to word size permits no misalignment and for our sequences gives a very low graph of scores. Widening the window brings more instances of a consensus word into a window. This is demonstrated in Fig. 3 by the appearance of a peak in the graphs, most clearly resolved at window sizes 10 and 12. Extending the window further may not bring any new instances of the consensus word, but it does increase the number of window positions that achieve a high score. This is shown by the plateau in the graphs for window sizes 16 and 20.

The effect of varying the neighborhood of words that can contribute to the score of a consensus word is shown in Fig. 4 for our DNA data set. We fix a window size of 12 and a word size of 6 and vary the number of mismatches permitted from 0 to 2, with no insertions or deletions. Requiring exact matches (0 mismatches) with the consensus word results in low scores with no distinct features. Permitting a single mismatch in general causes scores to increase, but the central conserved region emerges as a

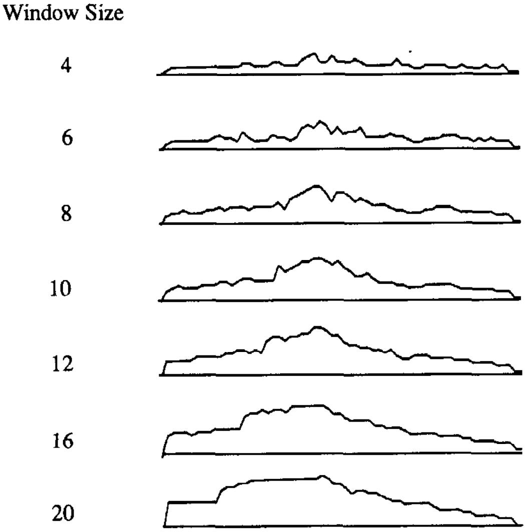


FIG. 3. Effect of varying window size from 4 to 20 for DNA sequence alignment. The word size is 4, and no mismatches are allowed.

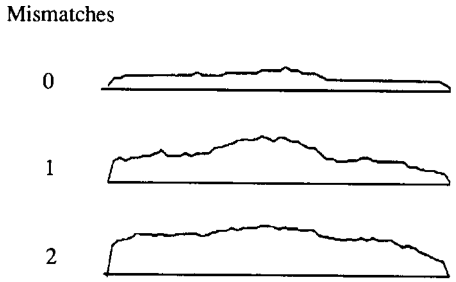


FIG. 4. Effect of varying the number of mismatches from 0 to 2 for DNA sequence alignment. The word size is 6, and the window size is 12.

distinct feature. When the neighborhood is increased to 2 mismatches, however, the background scores are almost equivalent to that of the conserved feature.

A very useful feature of our program is that once a consensus word has been identified, the sequences can be realigned on that word and the new alignment reevaluated for additional conserved features. Figure 5 shows the refinement of a consensus word using this technique. The graph of the scores after cycle 1 shows a maximum score of 14.25 at the position marked by the solid line. The sequences were realigned on the words that

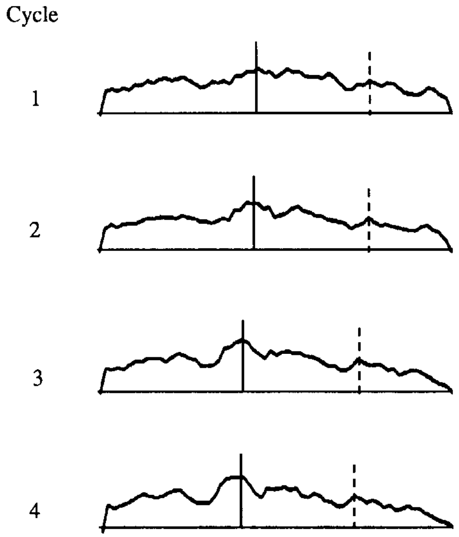


FIG. 5. Effect of realigning the sequences on a consensus word for DNA sequence alignment. The word size is 4, window size is 6, and 1 mismatch is allowed.

	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
A	8																			
C	6	8																		
D	3	3	8																	
E	1	2	6	8																
F	4	5	1	2	8															
G	7	6	2	1	4	8														
H	1	2	2	3	4	1	8													
I	4	5	1	2	5	4	2	8												
K	2	3	3	4	3	2	6	3	8											
L	4	5	1	2	5	4	2	7	3	8										
M	5	6	2	3	6	5	3	6	4	6	8									
N	4	5	5	4	3	4	2	3	3	3	4	8								
P	4	5	3	2	3	4	0	3	1	3	4	5	8							
Q	3	4	4	5	4	3	3	4	4	4	5	6	4	8						
R	1	2	4	5	2	1	5	2	6	2	3	4	2	5	8					
S	5	4	4	3	2	5	1	2	2	2	3	6	4	5	3	8				
T	5	6	4	3	4	5	3	4	4	4	5	6	4	5	3	5	8			
V	5	6	2	1	4	5	1	6	2	6	5	4	4	3	1	3	5	8		
W	3	4	2	3	6	3	5	4	4	4	5	4	2	5	3	3	5	3	8	
Y	3	4	2	3	6	3	5	4	4	4	5	4	2	5	3	3	5	3	7	8
	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y

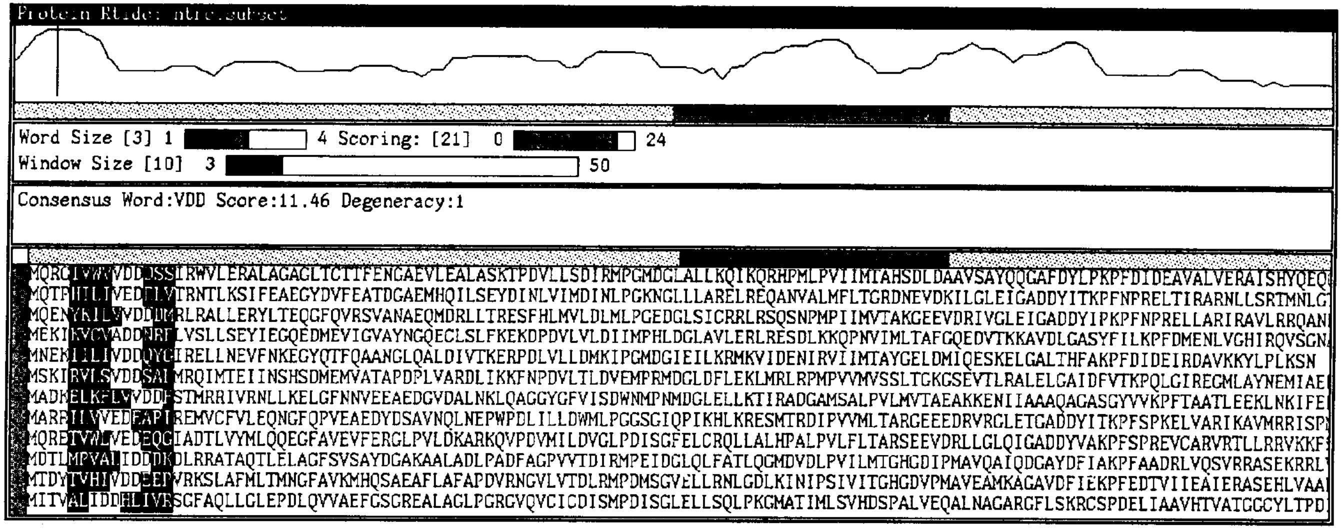
FIG. 6. Similarity matrix used in weighting sequence mismatches, based on the representation of amino acid similarity of Taylor.[10]

contributed to that score and the program run again. In the second cycle the marked peak is more clearly resolved and its score has risen to 15.25. An additional change is that the minor peak marked by the dashed line has become more evident. At the end of the third cycle the main peak has become more distinct and the score is 16.75, resulting from the realignment bringing more related words into the conserved window. The final cycle of scoring does not increase the maximum score and indicates the end of the process.

Protein Consensus Patterns

In the case of protein sequences various mismatches are weighted according to a matrix (Fig. 6), which is derived from Taylor.[10] The mismatches for each letter of a word are arranged according to weight, the nearest or smallest being first. Then in a systematic fashion we allocate mismatches until the limit or cutoff is reached. Then that letter is reduced to identity and the next letter is increased. The algorithm is similar to that of counting with the branch and bound feature we have described.

[10] W. R. Taylor, *J. Theor. Biol.* **119,** 205 (1986).



```
Word Size [3] 1        4 Scoring: [21]  0        24
Window Size [10]  3                   50
Consensus Word:VDD Score:11.46 Degeneracy:1
```

```
MQRGIVWVVDDDSSIRWVLERALAGAGLTCTIFENGAEVLEALASKTPDVLLSDIRMPGMDGLALLKQIKQRHPMLPVIIMTAHSDLDAAVSAYQQGAFDYLPKPFDIDEAVALVERAISHYQEQ
MQTFHILIVEDELVTRNTLKSIFEAEGYDVFEATDGAEMHQILSEYDINLVIMDINLPGKNGLLLARELREQANVALMFLTGRDNEVDKILGLEIGADDYITKPFNPRELTIRARNLLSRTMNLG
MQENYKILVVDDDMRLRALLERYLTEQGFQVRSVANAEQMDRLLTRESFHLMVLDLMLPGEDGLSICRRLRSQSNPMPIIMVTAKGEEVDRIVGLEIGADDYIPKPFNPRELLARIRAVLRRQAN
MEKIKVCVADDNRFLVSLLSEYIEGQEDMEVIGVAYNGQECLSLFKEKDPDVLVLDIIMPHLDGLAVLERLRESDLKKQPNVIMLTAFGQEDVTKKAVDLGASYFILKPFDMENLVGHIRQVSGN
MNEKIILIVDDQYGIRELLNEVFNKEGYQTFQAANGLQALDIVTKERPDLVLLDMKIPGMDGIEILKRMKVIDENIRVIIMTAYGELDMIQESKELGALTHFAKPFDIDEIRDAVKKYLPLKSN
MSKIRVLSVDDSAIMRQIMTEIINSHSDMEMVATAPDPLVARDLIKKFNPDVLTLDVEMPRMDGLDFLEKLMRLRPMPVVMVSSLTGKGSEVTLRALELGAIDFVTKPQLGIREGMLAYNEMIAE
MADKELKFLVVDDFSTMRRIVRNLLKELGFNNVEEAEDGVDALNKLQAGGYGFVISDWNMPNMDGLELLKTIRADGAMSALPVLMVTAEAKKENIIAAAQAGASGYVVKPFTAATLEEKLNKIFE
MARRIILVVEDEAPIREMVCFVLEQNGFQPVEAEDYDSAVNQLNEPWPDLILLDWMLPGGSGIQPIKHLKRESMTRDIPVVMLTARGEEEDRVRGLETGADDYITKPFSPKELVARIKAVMRRISP
MQRETVWLVEDEQGIADTLVYMLQQEGFAVEVFERGLPVLDKARKQVPDVMILDVGLPDISGFELCRQLLALHPALPVLFLTARSEEVDRLLGLQIGADDYVAKPFSPREVCARVRTLLRRVKKF
MDTLMPVALIDDDKDLRRATAQTLELAGFSVSAYDGAKAALADLPADFAGPVVTDIRMPEIDGLQLFATLQGMDVDLPVILMTGHGDIPMAVQAIQDGAYDFIAKPFAADRLVQSVRRASEKRRL
MTDYTVHIVDDEEPVRKSLAFMLTMNGFAVKMHQSAEAFLAFAPDVRNGVLVTDLRMPDMSGVELLRNLGDLKINIPSIVITGHGDVPMAVEAMKAGAVDFIEKPFEDTVIIEAIERASEHLVAA
MITVALIDDHLIVRSGFAQLLGLEPDLQVVAEFGSGREALAGLPGRGVQVCICDISMPDISGLELLSQLPKGMATIMLSVHDSPALVEQALNAGARGFLSKRCSPDELIAAVHTVATGGCYLTPD
```

FIG. 7. Typical screen display from the program PRTIDE.

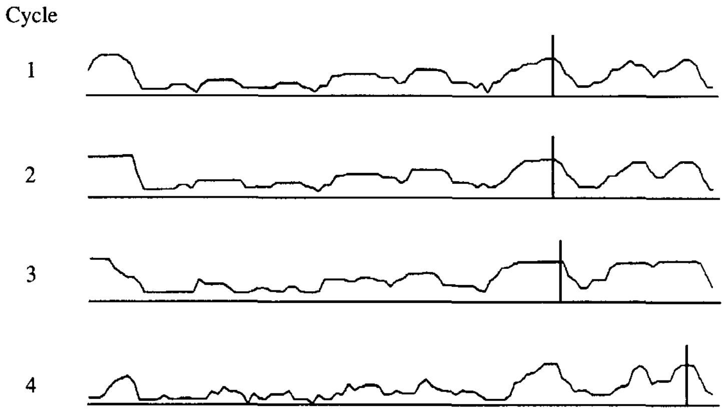


FIG. 8. Effect of realigning sequences on a consensus word over four cycles for the protein sequence data set. The word size is 3, window size is initially 10, and the neighborhood is set to 21, where 24 implies an exact match.

In the protein version of the consensus program, PRTIDE, the only substantial modification is in the definition of neighborhood. Whereas in the DNA version we specify the number of mismatches, insertions or deletions allowed, in the case of proteins we specify a similarity score that a word must attain when scored against a candidate consensus word in order to be included in the neighborhood. Figure 7 shows a typical display of this output.

In most practical aspects the DNA and protein programs are very similar. The larger alphabet of amino acids relative to nucleotides precludes use of words with more than 4 amino acids. Figure 8 shows an example of the protein version in which the sequences are realigned on a consensus word, with the result of resolving other conserved words. At top is shown the graph of scores for the protein data set in which the sequences are aligned at their left ends. The largest peak is found close to this end; aligning on the consensus word and rescoring creates the second plot in which three peaks near the right end of the plot have become more clearly resolved. Aligning on the largest of these and rescoring cause the formation of plateaus. This indicates too large a window size, and reducing the window size from 10 to 5 results in the final plot in which the three peaks are now well resolved.

Consensus Alignment

The idea of the algorithm builds on the previous section.[11] We align on consensus words, attempting to maximize the sum of the scores of the

[11] M. S. Waterman and M. Eggert, *Nucleic Acids Res.* **14,** 9095 (1986).

words. Before the practical algorithms are presented, a more general concept of alignment on words is presented.

We define a partial order on words. The words w_1 and w_2 satisfy $w_1 < w_2$ if the occurrences of w_1 in sequence i are to the left of the occurrences of w_2 in sequence i (and do not intersect) for $i = 1$ to R. It is not necessary for w_1 or w_2 to have occurrences in all sequences. Implicit in the definition is a window width W and neighborhood specification. The goal of an optimal alignment is to find words w_i which satisfy

$$\max\left\{\sum_{i\geq 1} s(w_i) : w_1 < w_2 < \ldots\right\}$$

(It is frequently desirable to require $s(w_i) \geq c$ for all i, where c is some cutoff value.) It is not possible to accomplish this goal in reasonable time, but it is possible to come quite close. We now define two practical algorithms.

Next, $w_1|w_2$ means that consensus words w_1 and w_2 can be found in nonoverlapping windows, each word satisfying as usual the window width and neighborhood constraints. The modified optimization problem is to satisfy

$$T = \max\left\{\sum_{i\geq 1} s(w_i) : w_1|w_2| \ldots\right\}$$

There is a straightforward recursion to find T. Let T_i be the maximum sum for the sequences from base 1 to base i:

$$\begin{array}{ccc} a_{1,1} & \cdots & a_{1,i} \\ a_{2,1} & \cdots & a_{2,i} \\ & \vdots & \\ a_{R,1} & \cdots & a_{R,i} \end{array}$$

Then T_i satisfies

$$T_i = \max\{T_j + s_{j+1,i} : i - W + 1 \leq j \leq i - k\}$$

and $T_{-W} = T_{-W+1} = \cdots = T_0 = T_1 = \cdots = T_{k-1} = 0$. Also, $s_{x,y} = 0$ if $y - x + 1 < k$. This algorithm runs in time approximately proportional to NW^2RB where B is the neighborhood size. Here the factor WRB accounts for the consensus word algorithm with a window width W. (This is an overestimate since the actual windows vary from k to W in width.)

If much shifting is necessary to match the sequences, T is an underestimate and misses some of the relevant matching. To overcome this problem, the definition of T is modified to

$$S_i = \max\{S_j + \hat{s}_{j+1,i}: i - W + 1 \leq j \leq i - k\}$$

where $\hat{s}_{j+1,i}$ is the largest scoring consensus word in the window from $j+1$ to i such that all occurrences of the consensus word are to the right of the consensus words for S_j. This algorithm is not guaranteed to be equal to the global maximum, but it is much more useful than T. Alignment for each case, DNA and protein, proceeds as just described with the modifications given next.

DNA Alignment

We have written a program, RALIGN, to align multiple DNA or RNA sequences.[11] The sequences are supplied in some initial alignment, which usually consists of the sequences being left justified. As in all our consensus methods, the parameters for window size, word size, and neighborhood are set. We require a consensus word to have a score at least equal to one-half the number of sequences before it can be used for alignment. This requirement is to eliminate the "junk" regions in alignments that are common with programs that optimize a total score.

To illustrate RALIGN for nucleic acids we take a set of 15 tRNA sequences from *Escherichia coli.* These sequences are difficult to align as their relationship is largely determined by conserved helices (secondary and tertiary structure), not the primary sequence itself. Analyses that fold tRNAs by minimum free energy are not too successful, usually folding about 50% of the tRNAs into the correct cloverleaf shape. An analysis based on consensus helices is successful for many structural RNAs, and a study of tRNA by consensus folding appears in Ref. 2. The only universal primary sequence patterns in tRNAs are the CCA at the acceptor arm and the GTTC in the TψC stem and loop. Our analyses (Fig. 9) find these invariant patterns along with other conserved words. Figure 9a has window size 7, word size 3, and up to 1 mismatch (total score equal to 179), while Fig. 9b has window size 8, word size 4, and up to 2 mismatches (total score equal to 127). The window is 7 in the first case and 8 in the second to allow shifts of 4 in each analysis. Notice that CCA is located at the 3′ end of all the sequences; in the case of Fig. 9 CCA is generally found overlapping the pattern CACC. In Fig. 9a,b CCA is not found by the program in all the sequences. In several cases an earlier, equally strong pattern is chosen in accord with the algorithm. Of course in Fig. 9b the object is to optimize a 4-letter consensus word, and this weakens the contribution of the 3-letter pattern CCA. GTTC is always located in Fig. 9b between consensus words GGTT and CGAA. In 14 of 15 sequences in Fig. 9a GTTC is located a

```
....gct.....tag.....cag...tgg..tag.....agc.......ctt...........cgg.....tcg......ggt...tcg....atc.....ctc..........cca
ggg gcta    tagct   cagc  tgg  gag     agcgcctg  ctttgcacg     caggagg tctgc    ggt   tcg    atcc    cgcatagct    ccacca
g   gcgcgt  taa     caaag cggt tatgt   agcgg     attgcaaatc    cgtc    tagtcc   ggt   tcga   ctccggaacgcgcct      cca
gga gcgg    tagtt   cag   tcggttagaat  acctg     cctgtcacg     cagggg  tcgcg    ggt   tcga   gtccc   gtccgtt      ccgcca
gtcccct     tcgtc   tag   aggcccaggac  accgcc    ctttcacgg     cggtaacagg       ggt   tcga   atc     ccctggggacg  cca
ggtggcta    tagct   cag   ttgg tag     agccctgg  attgtgattc    cagttg  tcgtg    ggt   tcga   atcccattag           ccacccca
ag  gcttg   tagct   cagg  tggt tag     agcgcacc  cctgataa      gggtgaggtcggt    ggt   tcaa   gtcca   ctcagg       cctacca
g   ggtcgt  tagct   cagt  tgg  tag     agcagttga ctttaatcaat   tgg     tcgca    ggt   tcga   atc     ctgcacgac    ccacca
g   gctacg  tagct   cagt  tggt tag     agcacatca ctcataatga    tgggg   tcaca    ggt   tcga   atccc   gtcgtag      ccacca
t   cctctg  tagtt   cag   tcgg tag     aacggcgga ctgttaat      ccgtatg tcact    ggt   tcga   gtcca   gtcagaggag   cca
tgg ggta    tcgc    caag  cgg  taaggc  accgg     attctgattc    cggcat  tccga    ggt   tcga   atc     ctcgtaccccag cca
    gcatccg tagct   cagc  tgg  tag     agtactcgg ctgcgaaccgag  cgg     tcgga    ggt   tcga   atc     ctcccggatgca cca
    gctgata tagct   cagt  tgg  tag     agcgcacc  cttggtaagggtg agg     tcggc    agt   tcga   atctgc  ctatcagca    cca
g   ggtgat  tagct   cagc  tgg  gag     agcacctcc cttacaagg     aggggg  tcggc    ggt   tcg    atccc   gtcatcac     ccacca
    gcgtccg tagct   cagt  tggt tag     agcaccac  cttgacatgg    tggggg  tcggt    ggt   tcga   gtcca   ctcggac      gcacca
aggggcg     tagtt   caat  tgg  tag     aacaccggt ctccaaaac     cgggtg  ttg      ggagt tcga   gtct    ctccgcccctg  cca
```

b

```
........tagc....tcag...tggt..agag....cacc..................ggtc.......ggtt....cgaa....tccc...........cacc
ggggcta tagc    tcagc  tggg  agag    cgcctgctttgcacgcagga  ggtctgc    ggtt    cgat    cccgcatagctc   cacca
ggcgcgt taac    aaag   cggttatg      tagcggattgcaaatccgtct agtcc      ggtt    cgac    tccggaacgcgc   ctcca
ggagcgg tagt    tcagt  cggtt agaa    tacctgcctgtcacgcaggg  ggtcgcg    ggtt    cgag    tcccgtccgttc   cgcca
gtcccct tcgt    ctag   aggc  ccagga  caccgccctttcacggc     ggtaacagg  ggtt    cgaa    tcccctgggga    cgcca
ggtggctatagc    tcagt  tggt  agag    ccctggattgtgattccagt  tgtcgtg    ggtt    cgaa    tcccattagc     cacccca
aggcttg tagc    tcagg  tggtt agagcg  cacccctgataagggtga    ggtcggt    ggtt    caag    tccactcaggcc   tacca
gggtcgt tagc    tcagt  tggt  agag    cagttgacttttaatcaatt  ggtcgca    ggtt    cgaa    tcctgcacgacc   cacca
ggctacg tagc    tcagt  tggtt agag    cacatcactcataatgatgg  ggtcaca    ggtt    cgaa    tcccgtcgtagc   cacca
tcctctg tagt    tcagt  cggt  agaa    cggcggactgttaatccgta  tgtcact    ggtt    cgag    tccagtcagag    gagcca
tgggg   tatcg   ccaag  cggt  aagg    caccggattctgattccggc  attccga    ggtt    cgaa    tcctcgtaccc    cagcca
gcatccg tagc    tcagc  tggt  agag    tactcggctgcgaaccgagc  ggtcgga    ggtt    cgaa    tcctcccggatg   cacca
gctgata tagc    tcagt  tggt  agagcg  caccccttggtaagggtga    ggtcggc    agtt    cgaatc  tgcctatcag     cacca
gggtgat tagc    tcagc  tggg  agag    cacctcccttacaaggaggg  ggtcggc    ggtt    cgat    cccgtcatcacc   cacca
gcgtccg tagc    tcagt  tggtt agag    caccaccttgacatggtggg  ggtcggt    ggtt    cgag    tccactcggacg   cacca
aggggcg tagt    tcaat  tggt  agaa    caccggtctccaaaaccg    ggtgttggg  agtt    cgag    tctctccgc      ccctgcca
```

FIG. 9. Multiple sequence alignment of 15 tRNA sequences by the program RALIGN. (a) The word size is 3 with up to 1 mismatch in a window of 7. (b) The word size is 4 with up to 2 mismatches in a window of 8.

between consensus words GGT and TCG. Note the failure in the last sequence where the alignment is

```
Consensus pattern:    . . . . ggt . . . tcg
Sequence 16:                  ggagt   tcg
```

This can be accounted for by the "greedy" nature of our algorithm. To allow more chances the consensus matching, whenever there are ties in scoring the algorithm chooses the 5′ or leftmost pattern. Therefore, gga is aligned rather than agt, the biologically correct alignment.

Protein Alignment

Figure 10 shows an example of the program PRALIGN applied to nine protein sequences. The sequences represent the amino-terminal 70 residues from a number of regulatory proteins related to the *E. coli ntrC* gene product. This region is fairly well conserved among the proteins and is believed to be responsible for interaction with proteins related to the *E. coli* gene product.

Figure 10 uses a word size of 3 and a window of 6. The neighborhood is limited to a similarity score of 18 or more. Since a perfect match receives a score of 8, an exact matching 3-letter word has score $24 = 3 \times 8$. As with Fig. 9, the top line of the alignment shows the conserved words identified at each position, and the instances of those words are shown in upper case in the sequence alignment below. With the neighborhood as specified, the conserved words are essentially those that contain conservative replacements from the consensus words. The majority of the amino acids in the sequences have been identified as part of conserved words and have been brought into alignment. The sequences are sufficiently dissimilar, however, that a number of short segments are not part of any conserved word.

Conclusion

The programs are available from Waterman. The programs RTIDE and PRTIDE are both written in C and run on a SUN using the SunView windows system. The programs RALIGN and PRALIGN are written in C and do not require the special graphics interface.

Several generalizations of these ideas are possible. One of the most obvious is to apply the methods to single rather than multiple sequences. Our programs for single sequences find the maximal nonoverlapping repeat pattern; as for the programs described in this chapter there are two programs, one for DNA and one for proteins. Elsewhere we have reported methods to find consensus palindromes in DNA, both for multiple and

```
mqr.ktl.lvd..dnc...irq...lve..cln..qeg...fqv..qav...ena...ecl....lnk.....pdv...lll...dim...mpt...
MQRgivw VVDd DSS   IRW   VLEr ALAgagltcttf          ENGa  EVLeal ASKt    PDV   LLS   DIR   MPG
MQTpHIL IVEdelv    TRNtlksi        FEAeg YDVf EAT   DGAemhqi     LSEydin       LVIm  DIN   LPG
MQEnyki LVVd DDMr  LRA   LLEr YLT  EQG   FQV  RSVana      EQMdrl LTResfh       LMVl  DLM   LPGedg
MEKiKVC VAD  DNRelvsl    LSEy      IEGqedMEV  IGVa  YNGq  ECLs   LFKekd  PDV   LVL   DII   MPH
MNE KIIlIVD  DQYg  IREl  LNE  VFN  KEG   YQTf QAAngl      QALdi  VTKer   PDL   VLL   DMK   IPG
MSKiRVL SVD  DSAl  MRQi  MTE  IINshsdm        EMVatap     DPLvardLIKkfn  PDV   LTLdve      MPR
mad KELkflvv DDFstmrr    IVRnllkelgf     NNV  EEAe  DGVda LNKlqaggyg     FVIsdwnm    PNMdglel
MAR RIL VVE  DEAp  IREmvcf    VLE  QNG   FQPveaedy   DSAvnq       LNEpw   PDL   ILL   DWM   LPG
MQR ETVwLVEd EQGiadt     LVY  MLQ  QEG   FAVevf     ERGlpvldkarkqv       PDV   MIL   DVG   LPD


cng...lql...lrr....lkn....nip..vmm....ltv....hge....dei...glq..iga...dfa....spf..cpk...eic..akv.kgl
MDG   LAL   LKQ    IKQrhp MLP  VII    MTA    HSDldaavs    AYQ  QGAf  DYLp   KPFdid     EAV  ALVeRAIs
KNG   LLLare       LREqa  NVA  LMF    LTGrdnev      DKIl  GLE  IGAd  DYIt   KPF  NPR   ELTirARNllsr
      LSI   CRR    LRSqsn PMP  IIM    VTA    KGEev  DRIv  GLE  IGAd  DYIp   KPF  NPR   ELL  ARI RAVl
LDGlav      LER    LREsdlkkqpnv       IMLtafgq      EDVtkkav   DLGa  SYFilkpfdmenlvghir
MDG   IEI   LKR    MKVide NIR  VII    MTA    YGEl   DMIqeske   LGAlt HFA    KPFdid     EIRdavk  KYLplksn
MDGldf      LEKlmrlr      PMP  VVMvssltgkg          SEV   TLRal       ELGaidf      VTKp  QLGir    EGMlayn
      LKTirad      GAMsa  LPVlmvTAEakkeni           IAAaqaga   SGY   VVKpfta     ATLeekl
GSG   IQPikhLKResmtr      DIP  VVM    LTA    RGEee  DRVr  GLE  TGAd  DYIt   KPF  SPK   ELV  ARI KAVmr
ISG   FEL   CRQlla LHP    ALP  VLF    LTA    RSEev  DRLl  GLQ  IGAd  DYVa   KPF  SPR   EVC  ARV RTLlr
```

FIG. 10. Multiple sequence alignment of nine protein sequences using the program PRALIGN. The word size is 3 in a window of 6 positions. The neighborhood is limited to words with a similarity score of at least 18, where a score of 24 represents an exact match.

single sequences.[2] In addition, these methods can also be applied to consensus secondary structure. Finally, although we have not done so, it is possible to include other ideas of consensus such as gap length to improve the alignments.

Acknowledgments

This research was supported by grants from the System Development Foundation, the National Science Foundation, and the National Institutes of Health.

[15] *k*-Tuple Frequency Analysis: From Intron/Exon Discrimination to T-Cell Epitope Mapping

By JEAN-MICHEL CLAVERIE, ISABELLE SAUVAGET, and LYDIE BOUGUELERET

Introduction

To determine the function of a gene or a protein from mere inspection of its sequence is one of the ultimate goals of research in sequence analysis. In principle, the solution of the protein folding problem, among others, stands as a prerequisite to this accomplishment. Meanwhile, shortcuts have been found, and several methods, mostly based on the recognition of conserved features in genes or proteins with similar functions, can be used to give at least a partial answer to this general problem. For instance, a straightforward way to assign a function to a given sequence is to run exhaustive homology searches on available data banks. As a rule of thumb, an overall 30% amino acid identity (over several hundred residues) between two protein sequences is taken as strongly suggestive of a similar three-dimensional fold and, thus, function.[1,2]

Alternatively, the simultaneous consideration of a set of divergent sequences of known similar function is used to establish characteristic consensus patterns which can be searched for in unassigned candidate sequences. These patterns can be defined on a small number of positions, be highly degenerate, and yet retain a good discriminative power. Such functional signatures usually correspond to key residues in the active (or binding) site of proteins or key nucleotides to be recognized within genes

[1] C. Chothia and A. M. Lesk, *EMBO J.* 5,823 (1986).

[2] R. F. Doolittle, *in* "Of URFs and ORFs." University Science Books, Mill Valley, CA, 1986.