Fastfinger: A study into the use of compressed residue pair separation matrices for protein sequence comparison

by B. Robson

Protein sequences are diverse in size and in content meaningful to researchers. They are rich in what seems to be "noise," or aspects of lesser interest that obscure clearer core features required to establish true relatedness and function. This paper represents part of a larger study that explores the possible efficient use and storage of "fingers" for protein sequence analysis, i.e., matrices of uniform size and shape that can "stand for" protein sequences by making more explicit the essential aspects of protein sequence pattern information. The essence of the study relates to data compression. Compression invokes an interesting alternative idea of pattern—the concept of "primeness" as in number theory is used to create the notion of an irreducible and potentially recurrent pattern element, and then this philosophy is mapped onto number theory by the unique factorization theorem, in order to define a novel measure of pattern difference. Other possible approaches are also discussed. Because compression and other approximations involve information loss, this is also a study of performance in the face of such loss. Because of the effects of this loss, no claims are made that encourage replacement of established sequence comparison methods, but the concept may have value in a number of applications within, and outside, molecular biology.

The interpretation of genes is based on comparison of the sequences of the proteins implied by those genes. Reference to known proteins serves to establish function when a sequence of related function is known, and a three-dimensional protein structure when a sequence of related structure is known. Protein sequences, however, are often inconvenient objects for routine manipulation in several respects. They are not regularly structured data objects, because they differ drastically in length and contain many features that obscure signals relating to a common function and even to a common folding configuration, at least to the eye of the human observer. The present study explores the possibility that matrices with fixed (e.g., 20×20) data structures may usefully represent sequences in some situations, and so be used to "finger" the full sequence, or related sequences, when required. For example, such finger matrices, or "fingers," might be "hard-wired," microcoded or otherwise precoded to stand for sequences, to allow extremely rapid preliminary identification of related protein structures and domains, given a new protein sequence. Although usually one will need to refer to actual sequences at some stage, key information might be pinpointed, and irrelevant information eliminated, to identify the types of sequence required. Although this implies, in simplest applications, avoidance and independence of alignment (a goal that has long attracted several laboratories), this is not the most fundamental feature of the method, and so this aspect will not be touched upon significantly here. Rather, the deeper challenge is that the notion of "relatedness" of sequences is variable. This study spans two major current approaches, lying at different extremes of the notion of relatedness, as follows.

The longest established software methods, sometimes colloquially described as the "gold standard"

0018-8670/01/$5.00 © 2001 IBM

methods[1–3] are linked with the concept of "homology." For more than ten years, the inference of homology between proteins, implying the occurrence of a common biological ancestor, has been based on the measurement of statistically significant sequence similarity. This is determined by comparison of identity or physicochemical similarity between amino acid residues throughout two or more whole proteins, or at least extensive regions of them, based in turn on a notion of optimal alignment of the sequences. For detailed work, such as modeling structures by homology, a typical approach has been to apply the approximate method BLAST[2] or PSIBLAST often followed by, say, a CLUSTALW alignment. FASTA[3] is becoming once more a very popular choice thanks to the emergence of powerful workstations. There is also still a preference among some researchers for tidying alignments by manual methods or semi-interactively. The "gold standard" techniques can be easily located by searching the Web (using the technique name), and are found on many servers, such as those of the European Bioinformatics Institute and the National Center for Biotechnology Information, and sometimes in large integrated packages such as the "Biology Workbench," from The National Center for Supercomputer Applications (University of Illinois at Champagne-Urbana). All of this broad family of techniques imply alignment, or require it at some stage. From that perspective, even for a given algorithm or overall protocol, there is no absolutely best way to align two weakly related sequences because there are adjustable parameters, to control factors such as the penalty for creating gaps, i.e., so as to better align two or more sequences elsewhere. (On servers there are, of course, hidden or recommended default settings.)

Alternative methods with a very different notion of relatedness are those that use a purely, or relatively, local definition of pattern. They do not usually involve alignment (although they can be invoked to help with alignment). Such local patterns or "motifs" may putatively code for features such as catalytic sites, binding regions concerned with intracellular and extracellular transport, or purely local architectural (conformational) motifs, which are perceived as relatively common recurrent themes when viewed over the space of all sequences as a whole. This includes many instances when the "gold standard" methods, by their own internal criteria, would not consider such patterns to be statistically significant. These local pattern methods are also commonly used and include PROSITE and BLOCKS, aspects of pfam, prints, MEME, and others, and the approaches such as the BIODICTIONARY at the IBM Computational Biology Center (based on local patterns and "seqlets" or "struclets"[4–7]). These methods may also be located on many Web servers. A principal purpose of the alternative methods is to predict the function of the protein product of a gene *irrespective* of overall homology (or lack of it). In some cases, as for some widely recurrent motifs that have common functions such as the glycosylation consensus, motif recognition is very useful to establish the overall biological role and life of the protein. However, motifs of that type are too broadly spread among many protein families to guarantee a common ancestor for extensive regions of the protein. Such examples make clear that in general in the local pattern approach, overall or extensive sequence homology is not necessarily expected, nor required, even in principle. Indeed, it might not be unreasonable to consider this type of approach as allowing and making use, *a priori* at least, of convergent evolution as much as divergent evolution.

These two types of approach are clearly very different in philosophy. It is well known that there is merit in comparing these and other techniques in a study on a new sequence, or in combining them in a general protocol for interrogation and annotation of novel sequences. The present study is one of a series to address the issue of whether it is possible or useful to develop a further approach, which potentially takes account of some aspects of both. Here we are particularly concerned with the need for simple techniques for compression of data that arise in considering one class of potential hybrid method. Practical considerations to do with that compression locally confine the extent of each pattern component (in this work, typically ten residues), but all parts of the sequence are considered as contributing potential pattern.

Theory

The use of a mathematical principle, known as the prime factorization theorem, has several possible applications in the field of data compression, beyond the application to sequence representation and comparison as presented here. In practice, however, there are inevitable information losses. The present study provides a simple vehicle for introducing the advantages and difficulties of the approach, which can to some extent be preempted by general theoretical consideration.

Representation of a sequence by a matrix of fixed form. One approach is to first represent the sequences in the form of tables of the frequency of occurrence of singlets, pairs, triplets, and so on of amino acids, in all specified relative positions, such as two residues apart in the sequence. This route seems promising as the basis for a universal approach because, in principle at least, it is possible to see how one might refine the method to describe unambiguously both local patterns, such as PROSITE motifs, and given a large enough matrix, overall sequences. A practical advantage for very high-speed searching is that such separation tables can certainly be calculated and stored in advance for an entire database (alternatively the matrix could be calculated at search time from the sequence data). As noted, an advantage for efficient storage and for parallel processing is the fixed dimensions of the matrix (e.g., 20×20), compared with sequences that vary considerably in length.

A conceptual starting point is a three-dimensional $20 \times 20 \times W$ "finger matrix." The dimensions of length 20 relate to the 20 types of biologically coded amino acid residues. Albeit with the caveat discussed later, this can be seen as extensible to $20 \times 20 \times 20$ for amino acid triplets, and to even higher dimensional matrices; if utility can be demonstrated for the lower information content inherent in the method, further study is justified for the more information-rich tables. W relates to the different separations used between the pairs. That is, W different separations are considered for relative distances along the sequence of 1, 2, 3, . . . W where W is the maximum separation or "window," which (in most cases in the present study) is 10 unless stated otherwise. Though this is a local range, in part chosen to avoid the loss of "phase" that occurs due to insertions and deletions, the samples are still drawn from throughout the entire sequence. Each element of such a matrix F is a frequency of occurrence of the form $N[R_i, R_{i+k}, k]$, where k is the number of times that an amino acid of type R_i is followed by an amino acid R_{i+k}, k residues in the C-terminal direction (i.e., to the right of any specified residue), where $k \leqslant W$. The basis for comparison of sequence is the distance between two such matrices, one of which will correspond to a sequence in the database, and one to the probe sequence. For example, one could score 0 every time a matrix element matches, and 1 otherwise, finally normalizing these scores by computing $1/(20 \times 20 \times W)$. The author considers all these as Fastfinger methodologies in the sense that a single matrix derived from a probe "fingers" another sequence as related by addressing its array. These more direct methods will be reported elsewhere, though the method EST%ID, described later for comparative purposes, is arguably of this general class.

A major departure explored in the present study deals with the difficulty that in practice these $20 \times 20 \times 20 \times \ldots$ matrices will be very large. It is convenient if these sequences can be represented by their precalculated, well-structured matrices in databases. However, while storage of data is not always an issue, multiplying by many orders of magnitude the storage requirements of data, on the scale of that available from the genome projects, is not attractive. Even the elements of the minimal ($20 \times 20 \times 10 = 4000$ elements) matrix explored here will most often be at least an order of magnitude larger than typical original sequences (say, 50–500 symbol elements). Hence, this study will serve as a useful example that is valid for the larger matrices, even if not a compelling case for compression in its own right.

Data compression. Various strategies of data compression are possible in principle and are based on removing storage space associated with empty, or nonrequired, information content. One common approach for images is based on the fact that the matrices describing them are typically sparse; however, in the present case, matrices are not in general very sparse, except for short sequences. Another approach, used here, is to remove information content less relevant to the pattern of interest, and to compress one or more dimensions on that basis. Here a method is introduced to reduce each data item in the (now typically) 20×20 array to a single scalar quantity, encoding all the separations that existed along the W-long separation dimension. There are also promising alternative representations of interest. For example, matrices compressed in different ways and with values more closely related to physicochemical properties of amino acid residues are also possible and are discussed later, and have particular value in threading.

How might one compress at least some of the information about a variety of sequence separations, and the counting of their occurrences, into a single number, so reducing the $20 \times 20 \times W$ matrix to a 20×20 matrix? One approach is to use prime numbers. The approach is interesting because, of itself, it suggests an alternative approach to thinking about pattern—as a form of "primeness."

Analogy by pattern factor (definition). If every possible nonempty, nonordered set $\{p_1\, p_2\, p_3 \ldots p_r\}$ of pattern factors $p_1\, p_2\, p_3 \ldots p_r$ in a set of data can be expressed as the function f returning value $\underline{f}$, which implies that set and no other set,

$$f(p_1\, p_2\, p_3 \ldots p_r) = \underline{f} \Leftrightarrow \{p_1\, p_2\, p_3 \ldots p_r\}, \qquad r \geqq 1 \tag{1}$$

then any two sets or subsets of data A, B with the same value of f, $\underline{f}_A = \underline{f}_B$ can be said to be "analogous" by their pattern factors $p_1\, p_2\, p_3 \ldots p_r$. The latter are considered irreducible, in that they cannot be reduced into simpler components encoding features, such as order, that are considered material.

The symbol $\Leftrightarrow$ signifies "maps to," one to one, without ambiguity in either direction. By this definition we may say that a pattern is analogous between two objects if it contains the same irreducible components, even though they are not necessarily in the same order. Here "analogous" is used to avoid "homologous."[8] This is an aspect of the local pattern concept in the sense that, for example, the order of PROSITE patterns found in a sequence is not significant.

How may it be guaranteed that a single scalar value $\underline{f}$ will uniquely and efficiently describe the pattern that gave rise to it? It is this key issue that corresponds to the properties of prime numbers and suggests their use.

Fundamental theorem of number theory. Every natural number n (>1) is a prime or can be expressed as the product of primes (prime factors) in the form

$$n = p_1\, p_2\, p_3 \ldots p_r, \qquad r \geqq 1 \tag{2}$$

and there is only one such expression as a product (decomposition into prime factors), if the order of the factors is not taken into consideration. (Theorem 4, Nagell.[9])

How does a prime number represent a separation? Let each p now be a function of the observed separation m between the specified residue at i and $i + m$ up to and including a maximum value for m, $m = W$ (i.e., a specified separation window W), and occurrence vs nonoccurrence in the window or sequence. Let such nonoccurrence be indicated by zero. The mapping from m to p is then here implemented by the function

$$\mathbf{\Lambda}(m) = \log(\mathrm{P}(m)) \Leftrightarrow m, \qquad \mathbf{\Lambda}(0) = 0 \tag{3}$$

where $\mathrm{P}(m)$ is the mth prime number in the series 2, 3, 5, 7, 11, 13, 17, . . . , and log is the natural logarithm.

How does one represent the number of times that each separation occurs? The notion of using "primality" to encode non-numeric concepts occurs in cryptography and in Goedel's theorem.[9,10] This latter is a corollary of Nagell's concise statement of the unique factorization theorem, the corollary being that Equation 2 also holds when two or more of the primes are identical, e.g., $p_1^a\, p_2^b\, p_3^c, \ldots$. It is the powers that are important in Goedel's theorem and arguably the prime numbers largely serve as arbitrary bases that guarantee uniqueness; in the present study, however, both the primes and their powers are important. The primes will code the separations, and the powers serve to count the number of times that a particular separation, encoded by each prime, occurs. If a particular separation occurs s times, then the corresponding prime factor p_x encoding separation x is raised to the power s. Colloquially, we might say that a term such as p_1 is introduced into Equation 2 every time a particular pair at a particular separation is observed. Consistently, $p_x = p_x^1$, implying exactly one occurrence, and a zeroth power $p_x^0 = 1$ means that there is no occurrence of the amino acid pair at the separation x, and a factor p_x does not appear explicitly in the list of factors.

The choice of taking the logarithm in Equation 3 is arguably a natural one, both for ease of manipulation and because an information-theoretic approach will be taken to combine matrices in future studies. The logarithmic value relates the concept of "recovery" of information by the factorization of primes from the viewpoint of the "Gauss conjecture" and the "one-prime" number theorem of Hadamard and Poussin.[9] Note that the usual formulation of this is in terms of the standard function $\pi(x)$ of number theory, which expresses the number of primes less than x. Here we use the inverse notation such that $m \cong \pi(\mathrm{P}(m))$, and hence from the one-prime number theorem[9] we obtain

$$\mathbf{\Lambda}(m) = \log(\mathrm{P}(m)) \cong \mathrm{P}(m)/m \geqslant \log(2) \tag{4}$$

Values occurring in this study are conveniently referred to in the information unit "nats" (natural units

based on the natural logarithm, analogous to bits, i.e., binary units, based on log base 2).

Implementation as the 20 × 20 finger matrix method. As implied, the $\Lambda(m)$ terms are added together to give the final term of a specific pair of amino acid residues, i.e., an element of a finger matrix characteristic of a protein sequence. Note that information is retained only as to the type of pair, say A–V, and all information about the various separations at which that pair occur is pooled into a single value by the following summation procedure:

$$\mathrm{F}(i, j) = \Sigma_d\, \Lambda(m_d) = \Sigma_d \log(\mathrm{P}(m_d)) \qquad (5)$$

The index d is to make it understood that the summations are performed over the log-prime.

Note that if we consider the $\mathrm{P}(m)/m$ as probability-like terms, then sums of these measures for each pair (i, j) that yields the 20×20 entries of the finger matrix $\mathrm{F}(i, j)$ have the status of applying successive logical "or" operations to those probability-like terms over all separations d encountered between the given symbols i, j.

Equation 5 summarized how to calculate the value of each element of the 20×20 array F, and the difference between two sequences can now be estimated without alignment by a procedure that involves subtracting the array for one sequence from that of the other. Let the two sequences be called A and B; the corresponding matrices are then F_A and F_B. The elements of the resulting 20×20 "difference matrix" are the "difference elements" $|\mathrm{F}_A(i, j) - \mathrm{F}_B(i, j)|$, i.e., the absolute values of the difference between corresponding elements. The corresponding matrix of such difference elements is the "difference matrix" F_{A-B}. For statistical purposes, note that we will only be interested in the properties of such a difference matrix, not the two matrices that gave rise to it.

Using such a matrix as a distance measure is possible but inconvenient. A corresponding scalar measure f of the difference between two sequences on this basis is simply the sum over all the terms of the difference matrix. This could later be re-expressed empirically in probability terms by statistical analysis, so in that sense, further scaling of the value is not critical. However, it is useful to set a standard and appropriate treatment of values, for working purposes, that returns values on an intuitive scale, say $\mathrm{f} = 0 \ldots 1$. Note that it is not strictly a (Euclidean) distance metric, as it does not fulfill the triangle inequality.

"Bad metric" self-editing principle and the statistics of F(i,j). An apparent primary difficulty with the method presented here is that there seems to be no immediately obvious relationship between the magnitude of the similarity measure and any concept of sequence similarity. Furthermore, the treatment of the components as a sum of logs, stated in limited precision, makes it impossible, in general, to actually recover the prime factors and deduce some measure of difference from the component separations. The first of these concerns arises because we are deliberately dealing with an alternative definition of pattern (though nonlinear correlation is discussed later in terms of BLAST searching). The second difficulty simply vanishes because the method does not require the user to recover the prime factors, merely to demonstrate a difference between their products. The difference between any two F, one such matrix for each sequence, provides an estimate of the information in favor of evidence that two sequences are different, i.e., in favor of *the null hypothesis that the two sequences have no relationship* at least within the definition of analogous pattern used here.

A natural difficulty arises in that, on some occasions, different products of primes can have similar, though never identical, values. Suppose we have the two numbers, 99 and 100, that factor as follows:

$$99 = 3 \times 3 \times 11$$

$$100 = 2 \times 2 \times 5 \times 5$$

These two products would be considered close to each other (indicating $|\mathrm{F}_A(i, j) - \mathrm{F}_B(i, j)| = |\ln(99/100)| = 0.010$ prior to rescaling), but it is clear that they represent quite different prime factors, and hence quite different patterns. This is indeed so, but this also well illustrates the "self-editing" principle inherent in the use of the prime factorization theorem. The difference is of an order smaller than typical differences between matrix elements, and it is therefore naturally weighted down. In short, such troublesome cases are naturally "damped out." It would be better said, however, that we are prepared to discard some information in return for the benefits of compression. The formal requirement is satisfied that information is lost, but not created. Another instance would be prime 29 (corresponding to

maximal separation of 10) and $30 = 2 \times 3 \times 5$ $|F_A(i, j) - F_B(i, j)| = |\ln(29/100)| = 0.034$.

Note the useful property that ambiguity arises and information is lost as the sizes of prime numbers increase. This is a proper trend, since autocorrelation information will tend to be lost in application because of the increasing chances of the separation spanning an insertion or deletion.

Has too much useful information been discarded overall? The minimum possible value when no corresponding pattern at all is observed in one sequence (i.e., where $\Lambda(m) = 0$) is $\log(2) = 0.693$, which corresponds to the difference between the log primes for separation distances of zero and of one. When neither corresponding element is zero and there are ten possible primes 2 . . . 29 corresponding to ten possible distances, and when just two such primes compared to two separations are compared, then the smallest possible nonzero contribution is that from a separation of 10 and a separation of 11. This is 0.231 (compared with 0.693 for separations 0–1, 0.405 for separations 1–2, and 0.511 for separations 2–3). The typical difference between products that are "coincidentally" of similar value is lower still.

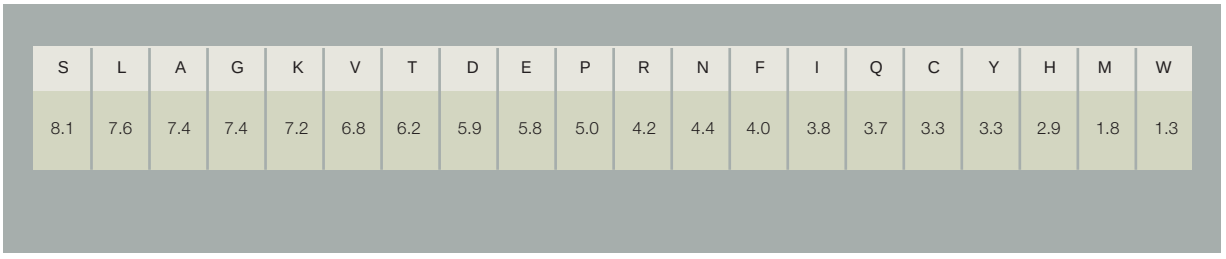
To show this more thoroughly, consider a simple Monte Carlo simulation. For one million tries, selecting at random one log prime number from the range log(2) . . . log(29), 18–36 percent will have the same value to the nearest nat. That is, placing samples in cells 0, 1, 2, . . . corresponding to the integer of the log prime number, no cell will contain less than 18 percent and no cell will contain more than 36 percent. It is self-evident that the distribution is skewed. Since as with squared velocities of gases there are no zero values, the property shares a general form with chi-squared, Maxwell-Boltzmann-like distributions (but corresponding to the absolute value, not the square). In such distributions, 18 percent corresponds to the zeroth cell and 36 percent to the maximum. Considering products of runs of N primes, the case of one prime corresponds to $N = 1$. For comparison, the possible ranges per cell for one million samples in cases $N = 2$ are 10–24 percent, $N = 3$: 1–17 percent, and $N = 10$: 1–5 percent. The average finger matrix elements (not difference matrix) $F(i, j)$ for log products of $N = 1 \ldots 10$ primes, randomly selected from the first to the tenth prime (average of one million random generations for each of $N = 1 \ldots 10$) are shown in Table 1.

From this it can be seen that two $F(i, j)$ will tend to differ by about two nats (mean = 2.05 N nats) for each extra log prime. For minimal confusion between identical log products of primes, the spread for each number should be less than one nat. The standard deviation (s.d.), for $F(i, j)$ with the same number of primes, is in fact half $F(i, j)$ (s.d. = 1.08 N nats). The mean of values less than the mean is 2.701 and of those greater than the mean is 2.573. Correspondingly calculated s.d. is also shown and the renormalized sum of the component means as calculated is within 4 percent of the customary mean. The skew is not particularly severe, but $\langle 1.068 \rangle - 1.454$ does not imply the existence of a negative value, which is impossible.

Since distributions of amino acids are not random, it is useful to explore situations in actual context of use, or at least taking into account known distribution features. These are explored in the section on results, but some general observations follow. That useful information is carried, in the compression along W, can also be demonstrated empirically by comparing it with a simpler technique ("1DIST finger") that records only the shortest distance per element rather than compressing together in one number from many distances (see the section on results). Conversely, initial comparison with explicit (uncompressed) $20 \times 20 \times W$ matrices shows the information loss compared with the uncompressed case. The relative loss from the use of the $20 \times 20 \times W$ matrix is not so distinct for very short sequences, as there will be most commonly zero or one primes (the logs of single primes being distributed as previously mentioned). The proportion of ambiguous prime products is also smaller for both shorter sequences. In one typical study, using $W = 10$, three or four of the ten separations of specified types of pairs (e.g., A–S) that are one to ten apart will occur about once in a sequence, the rest zero, one, or two times, and about a third of the information is lost. In assessing the frequencies of pairs *a priori*, one should note that the relative abundance of the 20 types of amino acid residues are Zipf's law[11] (approximately exponentially) distributed, but they are not so extremely distributed that high and low occurrences are distinguished by orders of magnitude. Figure 1 shows the frequencies in vertebrates.

Also, as long as we do not partition these by, for example, function site or secondary structure, the pair distributions are in reasonable accord with these singlet occurrences. A type of pair at an unspecified single separation will tend to occur with a probability

Figure 1 Frequencies of amino acid residue types in vertebrates (percent)



S	L	A	G	K	V	T	D	E	P	R	N	F	I	Q	C	Y	H	M	W
8.1	7.6	7.4	7.4	7.2	6.8	6.2	5.9	5.8	5.0	4.2	4.4	4.0	3.8	3.7	3.3	3.3	2.9	1.8	1.3

Table 1 Average finger matrix elements for log products of *N* primes

N Primes	<mean> Plus or Minus Standard Deviation	Component Less Than Mean	Component Greater Than Mean
1 prime	< 2.053> +/–1.079	< 1.068> – 1.454	< 2.873> + 0.926
2 primes	< 4.108> +/– 2.158	< 2.865> – 2.175	< 5.289> + 1.642
3 primes	< 6.161> +/– 3.238	< 4.628> – 2.906	< 7.555> + 2.248
4 primes	< 8.213> +/– 4.316	< 6.455> – 3.549	< 9.823> + 2.817
5 primes	<10.267> +/– 5.397	< 8.325> – 4.173	<12.075> + 3.372
6 primes	<12.323> +/– 6.468	<10.205> – 4.763	<14.305> + 3.904
7 primes	<14.378> +/– 7.553	<12.101> – 5.343	<16.523> + 4.437
8 primes	<16.431> +/– 8.634	<14.003> – 5.922	<18.724> + 4.955
9 primes	<18.485> +/– 9.720	<15.917> – 6.475	<20.923> + 5.471
10 primes	<20.538> +/– 10.801	<17.837> – 7.026	<23.111> + 5.973

of 0.0002 for tryptophan–tryptophan pairs (W–W) to 0.007 for serine–serine pairs (S–S).

Method

The principal method explored in this study uses a 20×20 matrix of amino acid residue pairs, with single numbers for each element. Using the principles described in the section on theory, these single numbers actually stand for several quantities, namely the specific separations encountered for such a pair of residues. This preferred embodiment of the theory is described here. However, alternative approaches are possible that are, to varying extents, of similar spirit. These were compared in preliminary studies, and so are also described for comparison.

Scaling to produce final preferred method. There are several potential scaling options. One possible type of "normalization" for this purpose considers the ratio $2\,|F_A(i,j) - F_B(i,j)|/|F_A(i,j) + F_B(i,j)|$. This is not a true normalization, but may be identified as an indication of the deviation of the variance from that expected, conditional on statistics related to specific corresponding elements of the matrix. However, while these kinds of normalization are useful, it may be argued that it is an absolute measure of pattern difference that is of interest here,

and that any normalization or rescaling should preserve information about sequence length differences, different abundance of amino acid residues, etc., and any other features that may be construed, from one perspective at least, as "pattern difference." It is convenient to define a rescaling that would give an index, for most cases of interest, within the 0 . . . 1 interval, but that nonetheless is not normalized with respect to the factors just mentioned.

A more convenient normalization is to divide by the number of elements ($20 \times 20 = 400$) and then by the measure that is expected for maximum possible distance. The amino acid residues are Zipf's law (approximately exponentially) distributed. The frequencies of amino acid residues in vertebrates presented in Figure 1 are required for this calculation. A reasonable upper limit for the difference between two large random sequences is that the number of occurrences of every separation, averaged over all the ten separations, is about one (typically some zero, some more than one). Thus the value at each element is further divided by $\Sigma_{m=1\ldots W}\,\Lambda(m) = 22.590$, the sum of Λ values for separations 1, 2, 3, . . . 10 all occurring once. There is no reason *a priori* that a separation of one is "closer" in any physicochemical sense than is a separation of ten (that is, with the important caveat that an increased chance of an insertion or deletion at larger separations justifies a window limit, which is the reason for not considering larger separations). Note, however, the phrase "*a priori*": there are differences observed in practice and these are particularly significant when we distinguish different aspects of structure, such as secondary structure. Indeed the present method has been applied to the secondary structure letters "H," "E," and "C" as actually assigned from the experimental structures of the proteins, as opposed to the 20 residue types. This reveals rather similar degrees of relationship from the (secondary) structural viewpoint, since there is strong, if complex correlation between residue pair patterns and secondary structure, as will be shown. Most of all, we are concerned here with differences between sequences and, as noted, the increased chance of spanning an insertion or deletion increases with separation.

The final preferred formulation for the scalar difference measure, for present purposes, is thus

$$\mathrm{f} = \sum_i \sum_j \mathrm{F}_{A-B}/Z = \sum_i \sum_j |\mathrm{F}_A(i,j) - \mathrm{F}_B(i,j)|/Z \qquad (6)$$

Here $\mathrm{F}_{A-B} = |\mathrm{F}_A(i,j) - \mathrm{F}_B(i,j)|$ is the difference matrix for proteins A and B as described, and $Z = \Sigma_{m=1\ldots W}\,\Lambda(m) \times 400$.

Whereas normal values for comparison of very dissimilar sequences of similar length lie conveniently close to unity, the measure will continue to increase if we consider that, for example, new domains containing novel pattern features are progressively added to one or both features. At the same time, the limited capacity of the compression method to hold separation information (and so return significant values on taking the difference between two finger arrays) leads to hyperbolic curves with attenuating dependence on difference as sequences become very long.

Alternative embodiment: Implementation of weighted finger matrices. Although this is not explored in the present study, there are several reasons why the terms of the difference matrix might be differentially weighted. As noted in the section on theory, the amino acid residues do not occur with equal probability, and there is a case that the rarer ones should be more heavily weighted accordingly. This is on the argument that the rarer ones carry more information (by virtue of that rarity) and indeed, as with the catalytic role of H, perform specific functions that may earmark a relationship. The appropriate weighting for pairs can be calculated from the corresponding product, or more correctly, from express determination of the occurrence of the pairs. Also, one might choose to emphasize amino acid residues that possess some single specified property, such as hydrophobicity, or helix- or sheet-forming character. Related to this is the idea that the most general formulation of interest for present purposes is not confined to the 20×20 matrix but can handle pooled sets of amino acids (i.e., symbols representing a more generalized type of amino acid, such as acidic: aspartate and glutamate). The summation over the F_{A-B} cells of the amino acids that are to be pooled will then proceed before their absolute value is taken. As part of that procedure, a term $w(i,j)$ can be considered as normalizing the entries to allow for similar contribution of similar amino acid residues.

Thus in principle there is a more general treatment represented by

$$\mathrm{f}' = \sum_{i=1}^{20} \sum_{j=1}^{20} |w(i,j) \cdot (\mathrm{F}_A(i,j) - \mathrm{F}_B(i,j))| \qquad (7)$$

The value $w(i, j) = 1$ is used throughout this study. The use in Equation 7 of f' rather than f indicates that the quantity is not yet conveniently scaled.

Alternative implementation as the 20 × *W* finger matrix method. This alternative is presented briefly, in order to demonstrate that the 20 × 20 matrix approach (or reductions thereof by combining amino acids into sets) is not the only one possible. We can "permute" the dimensions and decide to retain details of separation and instead compress information about one of the amino acid residue types of a pair. Now logarithms of prime numbers can be assigned that relate to trends in physicochemical or evolutionary properties of amino acid residues. In practice, for symmetry, the specified amino acid is regarded at the center of each row at separation $m = 0$, the elements of each row representing $2W + 1$ separations ($m = -W, \ldots, +W$). Each element is no longer $\Lambda(m)$ summed over separations m, but $\Lambda(r)$ summed over 20 types of amino acid residue r.

The important feature is that this now provides an opportunity to order the $\Lambda(r)$ on r in a way that was not possible for $\Lambda(m)$ on m, that is, such that the value relates to amino acid type, not separation distance. The more similar are two amino acids, the more similar will be their $\Lambda(r)$ values. Following a notation analogous to that of the standard Perl computer language for hashed or associative arrays indexed by text strings, let $\Lambda\{'X'\} = \log(m)$ be the value associated with amino acid residue of type X, and let m and n in $\Lambda\{'X_1'\} = \log(m)$ and $\Lambda\{'X_2'\} = \log(n)$ be chosen to be similar values when X_1 and X_2 are similar amino acids.

The method of assignment of values to the amino acid residues is empirical, by an optimization procedure. The starting configuration of values was the logarithms in the set of prime numbers that would best preserve the substitution distances (distances in terms of normalized number of accepted substitutions) of the 20 amino acids attained by multidimensional scaling, such that distances reflect the degree of dissimilarity in evolutionary terms (see French and Robson[12] for results and details). Though there is an element of arbitrariness, the final assignment of parameters is a complex balance in which the effects of multiple occurrences of the same pair at the same separation distance also have to be considered, in conjunction with the stress in departing from the dissimilarity of residues in terms of the number of accepted mutations. Starting from logs of prime numbers that are early in the series, e.g., log(2), log(3), log(5), or too high in the series, e.g., log(10079), log(10091), log(10093), allows less ready distinction. In the first case, multiple occurrences of low values become confused more readily with a very few occurrences of high values. Note that, in evaluating any 20 × *S* Fastfinger matrix, when a particular amino acid residue is observed for the nth time at the specified separation, say six, then the same value $\Lambda(m)$ for that amino acid residue will be added in for the nth time. In the second case, greater precision is used to distinguish types as $\log(N)$ and $\log(N + d)$ converge for large N and small d. The following set, although not guaranteed to be fully optimal assignments, nonetheless performed well when the 20 × *S* Fastfinger matrix was employed to identify distant sequences:

$\Lambda\{'W'\} = \log(211)$; $\Lambda\{'Y'\} = \log(199)$;
$\Lambda\{'F'\} = \log(197)$; $\Lambda\{'L'\} = \log(193)$;
$\Lambda\{'M'\} = \log(191)$; $\Lambda\{'I'\} = \log(181)$;
$\Lambda\{'V'\} = \log(179)$; $\Lambda\{'C'\} = \log(167)$;
$\Lambda\{'T'\} = \log(151)$; $\Lambda\{'S'\} = \log(149)$;
$\Lambda\{'A'\} = \log(139)$; $\Lambda\{'G'\} = \log(137)$;
$\Lambda\{'P'\} = \log(131)$; $\Lambda\{'N'\} = \log(109)$;
$\Lambda\{'Q'\} = \log(107)$; $\Lambda\{'E'\} = \log(103)$;
$\Lambda\{'D'\} = \log(101)$; $\Lambda\{'H'\} = \log(79)$;
$\Lambda\{'K'\} = \log(73)$; $\Lambda\{'R'\} = \log(71)$.

Shifted window *W*. We still make use of the same prime numbers 2 through 29, but use these to relate to larger separation such as $1 \ldots M$, $1 \ldots M + 1$, $1 \ldots M + 2$, ..., without loss of precision. Here one residue is envisaged as fixed and the others are in a window M locations away. This is called a "shifted," as opposed to a simply "widened," window. A trend in the measures when M increases, and when insertions and deletions are responsible for losing correlation, can help demonstrate the relative significance of the pattern agreement as normally measured up to $W = 10$. The trend can also help distinguish cases where there are different matching domains at different locations in two sequences.

Nonprime Fastfinger methods: Estimated percentage identity without alignment. Some studies (e.g., see the section on results, Table 4, later) required selection of sequences with approximately the same percentage residue identity. The following alternative nonalignment method was used to rapidly compare sequences to determine which might lie in the appropriate range, and this was then checked by a

more exact calculation, involving alignment. This method, EST%ID, replaces compression by use of primes with other kinds of approximations that sacrifice information. This is achieved by thinking of a more classical $L \times M$ table for comparing two sequences of length L and M, and estimating from all the elements what the identity percentage would be after alignment, solely from that table.

In brief, let $N(m)$ be the number of times that the same pair of amino acids is found at the same separation m, when every possible pair at separation m is compared in the two sequences. $N'(m)$ is the corresponding number of nonmatching comparisons. The full field of comparison is the $Lp \times Lt$ matrix for studying correlation of two sequences (probe and test) of lengths Lp and Lt. In practice, the summation is also over a range of values of m, say M, typical choices being with the first member amino acid residue of the pair at j and $m = j + 1 \ldots j + 10$, $m = j + 21 \ldots j + 30$, $m = j + 51 \ldots j + 60$, $m = j + 101 \ldots j + 110$, and $m = j + 201 \ldots j + 200$. That is, the $N(m)$ and $N'(m)$ are pooled over these ranges of m. These ranges can be arbitrarily defined as short, proximate, medium, long, and remote-range neighbors in the sequence. The summations over selective ranges ($m = M \ldots 10$) are called $N(M)$ and $N'(M)$ for the matching and nonmatching cases, respectively. The ratio $K = \Sigma_m N'(M)/\Sigma_m N(M)$ is the key quantity calculated. K is scaled to achieve a form of normalization by dividing the ratio K by $K_N =$ (number of matches expected for 100 percent sequences)/(number of mismatches expected for 100 percent sequences). This is done for pairs throughout the $Lp \times Lt$ matrix.

The scaling factor $1/K_N$ is determined as follows. The number of possible pairs available for comparison is $m = (Lp - W + 1)(Lt - W + 1)$ where W is the maximum separation of pairs considered. An approach to a perfect match for a large number of different amino acid types, without any spurious matches of pairs, would be one for each entry on the diagonal and zero for each entry off the diagonal, and hence $d = (Lp - W + 1) + (Lt - W + 1)/2$ for corresponding limit $Lp = Lt$. The limit is $(1 - \theta)(m - d)/(d + \theta \cdot (m - d))$ for the real case, where θ depends on the statistical distribution of amino acid residues and pairs. The measure $S = K \cdot (d + \theta \cdot (m - d))/(1 - \theta)(m - d)$ attempts to be the adequate scaling condition, including counting multiple separations when summation of $N'(M)$ and $N(M)$ is taken over "hits" between matching pairs for all ranges from $m = 1$ to $m = W$. θ is calibrated, as from the probe sequence, such that $S = K \cdot (d + \theta \cdot (m - d))/(1 - \theta)(m - d) = 1$. This is satisfied by taking θ as $\theta = K \cdot ((m - d) - d))/(K \cdot (m - d) - d + m))$. In practice, several θ_M are determined from the probe against itself, one for each range M. Hence $S_M = K \cdot (d + \theta_M \cdot (m - d))/(1 - \theta_M)(m - d)$ is determined for each range M. In practice, the measure S ranges from 0.4 to 1, or 4–100 percent. For convenience, this may be further rescaled as $S' = 20 + (80/60) \cdot (S - 40)$, approximating the 20 percent for random matches and 100 percent for complete matches encountered in aligning sequencing and scoring corresponding residues.

Use of public domain methods for comparison. The E-value (expectation value) of BLAST was used with standard settings (using gapped alignment, comparison matrix BLOSUM62,11,1 [0.85]).[13] The studies used three sequence data sets. First, for studies in percentage identity, the lysozymes plus α-lactalbumins, cytochromes, globins, and serine proteases were used for clearly similar sequences (in the range of 35–100 percent identity). The families were not chosen because Fastfinger performs particularly well (or badly) with them, but simply because they are large sample families, well researched for the construction of evolutionary trees, and well understood. Second, this was extended by a further set of 75 proteins that include some subtle but classically recognizable homologies by application of standard methods. This set is listed in, for example, Garnier and Robson.[14] This was well researched by the authors and their colleagues. It was the standard early training set to compare various classes of secondary structure prediction, some of which benefited from, and even exploited, homology between test sequence and one or more entries on the database, and some that sought to avoid that bias. This also aids interpretation of the correlation between residue pairs and secondary structure. Third, to ensure an unbiased component, the set of more than 200 nonhomologous proteins was used for the routine working set shown later in the results section, Table 3. A larger set of 500 has also been used, but pairs with significant homologies were not well purged; results were nonetheless similar.

In comparing percentage identity, pairs of sequences were first selected by the EST%ID method, followed by Smith-Waterman and CLUSTALW. A computer experiment to optimize the f score by reasonable variations in alignment demonstrated that the value of f is not sensitive to the choice of pairs by this method.

Results

In general, the method is not yet of the caliber of standard methods for routine sequence searching but, particularly in view of the losses of information due to compression, it was found to perform surprisingly well and may have merits in certain specific applications. Depending on hardware, algorithmic details of preliminary parallelization over several component tasks, and programming language, the method was found to be some 3–12 times faster in research codes for earmarking similar sequences when reading data blocks of fixed length and in parallel processing. This is largely because the matrices are all of the same length, while sequences are not; indeed other "tricks" using the same 20×20 data representation could theoretically achieve much higher speed gains, but they have not yet been demonstrated. As described in this section, there may also be useful behavior and even benefits in cases of more subtle sequence relationships, despite significant information losses due to compression, and these considerations form the major part of the present study.

Relation between scores and classical methods. Figure 2 gives a typical example result for a small database corresponding to that of 75 proteins from Garnier and Robson[14] (see previous section for discussion), again using window $W = 10$. In reporting results of simulations, FI ("f index") is equivalent to 100 times f as described in the theory and methods sections. Full output examples have been omitted for brevity, but it is worth noting that they include statistical analysis and various analyses as described in this paper; they also include use of H, E, and C (observed) secondary structure symbols as an alternative to amino acid residue assignments (discussed later). Again for brevity, straightforward nonmatching cases in which the sequence match is deemed insignificant by BLAST and FASTFINGER, with FI $>$ 65 or more, are excluded. This is with the exception of retaining a few cases that illustrate some feature of interest.

The arbitrary sample set well reflects the quantitative situation for false positives and negatives—the cause of a number of typical difficulties as well as curious features worthy of future study. Notably, it is impossible to avoid false positives or false negatives. In this example, all scores with 55 percent or less are also various types of proteases, with the exception of immunoglobulin (human myeloma chain 2), which has a BLAST E-value of 0.011 with respect to the probe (heavy chain gave 6.0). It is interesting that on such a BLAST score a protein might be considered as a possible template for homology modeling, in the absence of any template of more obvious structure. Note that several of the serine proteases did not score under the threshold of 55, for example, 2ALP1E Alpha lytic protease (FI = 61), 3RP21E Serine proteinase (FI = 65), 1TPO1E Beta trypsin (FI = 58), and 2EST1E Elastase (FI = 63). These actually illustrate the limitation that Fastfinger requires sequences to be roughly the same length as the template, since the lengths differ significantly. However, as is typically the case, the scores may still be regarded as low enough to be worthy of further inquiry. An example of such a further inquiry would be a "scan" of the matrix of the shorter sequence of length L against the matrix of progressive sections $1 \ldots L$, $2 \ldots L + 1, 3 \ldots L + 2, \ldots$ of the longer sequence. (This need not cover every extracted segment of probe sequence length, but, for example, be in jumps of ten residues: $L \ldots 1$, $L + 11$, $L + 21 \ldots$).

In many such studies, the range $55 <$ FI < 66 reasonably indicates "potentially interesting." These are marked "$<$" in Figure 2. It is evident that arbitrary increase of the threshold usefully catches these and several more proteases, but brings in a number of false positives.

Several of the positive hits (i.e., FI less than 55) are proteases, but not serine proteases. For example, 3TLN1M (Thermolysin, FI = 53) is a Zinc-dependent metalloproteinase, and 2ACT1M (Actinidin, FI = 55) is a sulfhydryl proteinase. Hence, some proteins with a proteolytic function but different folds do not behave very differently from proteins of weak sequence homology and the same fold. Should these be considered false positives? Rather than being a difficulty, the possibility that a common pattern actually exists (in the sense of pattern in the present method), and identifies function, would be interesting and useful. Another sulfhydryl proteinase, 1PPD1M (Papain sulfhydryl proteinase, FI = 64), differs significantly in length and was not initially considered a positive hit, but it does of course show up on the potentially interesting list (i.e., FI = 65 or less). This would make sense if there was either (1) a common ancient ancestor but an extensive differential change in protein architecture or (2) an effect of convergent evolution of pattern features essential for function; it might be that specific sets of separations of key residue types were favored by the common function. In addition, several immunoglobulins share pattern features. Be that as it may, Fastfinger "flags" the matches as worthy of further study.

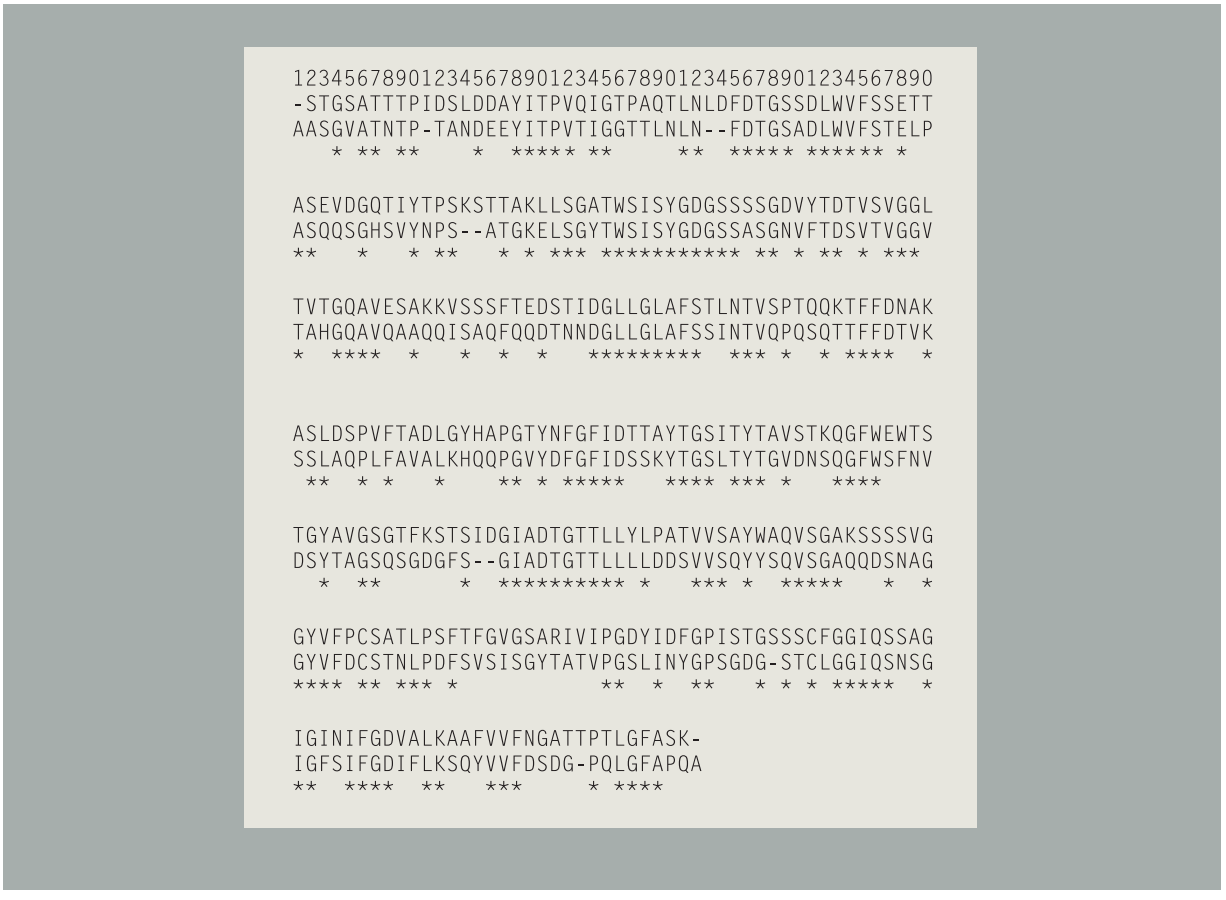
Figure 2 Fastfinger example result

```
UNK 330 4APE1E ACID PROTEINASE ENDOTHIAPEPSIN                          PROBE
SEQ 330 4APE1E ACID PROTEINASE ENDOTHIAPEPSIN                          FI=0 * BLAST=0
SEQ 323 2APP1E ACID PROTEINASE, PENICILLOPEPSIN [HYDROLASE: PROT       FI=36* BLAST=0.0
SEQ 229 2FB42E IMMUNOGLOBULIN FAB (HUMAN MYELOMA ) LIGHT CHAIN        FI=50X BLAST=0.011
SEQ 316 3TLN1M THERMOLYSIN [HYDROLASE: NEUTRAL METALLO-PROTEINAS      FI=53* BLAST=0.004
SEQ 307 5CPA1M CARBOXYPEPTIDASE A [C-TERMINAL AMINO ACID HYDROLA      FI=54* BLAST=0.14
SEQ 218 2ACT1M ACTINIDIN [HYDROLASE: SULFHYDRYL PROTEINASE] {KIW      FI=55* BLAST=0.23
SEQ 275 1SBT1M SUBTILISIN BPN' [HYDROLASE: SERINE PROTEINASE] {P      FI=55* BLAST=0.83
SEQ 181 2SGA1E PROTEINASE A (SGPA) [HYDROLASE: SERINE PROTEINASE      FI=56* BLAST=0.048
SEQ 220 1MCP1E IMMUNOGLOBULIN FAB IGG (MOUSE) CHAIN 1                 FI=56< BLAST=4.4
SEQ 222 1MCP2E IMMUNOGLOBULIN FAB IGG (MOUSE) CHAIN 2                 FI=57< BLAST=0.68
SEQ 223 1TPO1E BETA TRYPSIN (BOVINE) ORTHOROMBIC                      FI=58< BLAST=1.2
SEQ 222 4SBV1E SOUTHERN BEAN MOSAIC VIRUS COAT PROTEIN                FI=58< BLAST=1.8
SEQ 131 5CHA1E ALPHA CHYMOTRYPSIN A (BOS TAURUS)  CHAIN 1             FI=59< BLAST=0.42
SEQ 256 2CAB1E CARBONIC ANHYDRASE FORM B HUMAN ERYTHROCYTES           FI=60< BLAST=0.46
SEQ 114 2RHE1E BENCE JONES PROTEIN LAMBDA VARIABLE DOMAIN (HUMAN      FI=60< BLAST=3.3
SEQ 136 1ECD1H HEMOGLOBIN (ERYTHROCRUORIN, DEOXY) [OXYGEN TRANSP      FI=61< BLAST>100**
SEQ 198 2ALP1E ALPHA LYTIC PROTEASE [HYDROLASE: SERINE PROTEINAS      FI=61< BLAST=0.04
SEQ 149 2LHB1H HEMOGLOBIN V (CYANO,MET) SEA LAMPREY                   FI=63< BLAST=2.6
SEQ 240 2EST1E ELASTASE  PORCINE PANCREAS                             FI=63< BLAST=2.8
SEQ 212 1PPD1M PAPAIN SULFHYDRYL PROTEINASE (PAPAYA FRUIT LATEX)      FI=64< BLAST=0.13
SEQ 151 2SOD1E CU,ZN SUPEROXIDE DISMUTASE [OXIDOREDUCTASE: SUPER      FI=64< BLAST=0.19
SEQ 107 1REI1E BENCE-JONES IMMUNOGLOBULIN VARIABLE PORTION (REI)      FI=64< BLAST=0.96
SEQ 153 1LH11H LEGHEMOGLOBIN (ACETATE,MET) [OXYGEN TRANSPORT] {Y      FI=64< BLAST=1.6
SEQ 152 2PKA2M KALLIKREIN A (PORCINE PANCREAS) CHAIN 2                FI=64< BLAST=1.7
SEQ 162 3DFR1M DIHYDROFOLATE REDUCTASE [OXIDOREDUCTASE: NADPH/DO      FI=64< BLAST=1.8
SEQ 224 3RP21E SERINE PROTEINASE (RAT MAST CELL PROTEASE)             FI=65< BLAST=4.6
SEQ 184 2STV1E COAT PROTEIN OF SATELLITE TOBACCO NECROSIS VIRUS       FI=65< BLAST=6.3
SEQ 310 8ATC1M ASPARTATE TRANSCARBAMYLASE (E. COLI) CHAIN 1           FI=66  BLAST=0.56
SEQ 108 1CPV1H CALCIUM-BINDING PARVALBUMIN B [CALCIUM BINDING PR      FI=67  BLAST=55
SEQ 374 1LDE1M APO-LIVER ALCOHOL DEHYDROGENASE [OXIDOREDUCTASE:       FI=68  BLAST=0.18
SEQ 153 8ATC2M ASPARTATE TRANSCARBAMYLASE (E. COLI) CHAIN 2           FI=68  BLAST=1.5
SEQ 129 1AZA1E AZURIN ELECTRON TRANSPORT PROTEIN                      FI=69  BLAST=3.9
SEQ  80 2PKA1E KALLIKREIN A (PORCINE PANCREAS) CHAIN 1                FI=72  BLAST=0.66
SEQ  75 3ICB1H CALCIUM BINDING PROTEIN BOVINE INTESTINE VIT. D D      FI=74  BLAST>100**
SEQ  74 2ABX1M BUNGAROTOXIN BRAIDED KRAIT VENOM                       FI=75  BLAST=0.31
SEQ 170 3WGA1M LECTIN(AGGLUTININ) WHEAT GERM                          FI=80  BLAST=2.1
SEQ 498 8CAT1M CATALASE BEEF LIVER                                    FI=92  BLAST=0.54
```

Figure 3, a report for the principal match detected, gives some feeling for the relationship between the score for the case of approximately 50 percent identity.

The Fastfinger score f correlates with percentage identity, although the relation is not exact because of the way in which f includes contributions from the insertion regions; these count as a component of pattern difference. In Figure 2, a score of $f = 0.36$ (FI = 36) conforms to a case of 53 percent sequence identity, a clearly homologous case. Table 2 shows the overall correlation in terms of curve fit to synthetic and real structures, and Table 3 shows the extent to which percentage identity can be deduced from f (without alignment) for real protein sequences. Both use a maximal separation window of $W = 10$. The form is less linear and more parabolic in the range 20–100 percent identity in circumstances

Figure 3 Report: Significant homology/identity hits (Frankenstein Internal Fastfinger 1.0) 4APE1E ACID PROTEINASE ENDOTHIAPEPSIN vs 2APP1E ACID PROTEINASE, PENICILLOPEPSIN (HYDROLASE: PROTEINASE) 53 percent residue identity



where there is greater information loss in transforming the sequence information into a matrix. Such information loss occurs when there are longer sequences, and the effect becomes significant for sequences of 500 residues or less. Information loss also occurs for alternative methods that deliberately discard some information from the outset. For example, 1DIST(1) is a simpler 20×20 matrix approach included for comparison. Only one of the separation distances is retained, and it holds the minimum separation encountered for a pair of residues (that is, if separations between residue pairs of the specified type are 3, 5, 7, 8, only 3 is retained). For clarity, data for the simpler method are not shown in Table 2, but the value fmax of the hyperbolic fitted function $f = \text{fmax}/(1 + P_{f=\text{fmax}/2}/P)$, where P is percentage identity, has been fitted to the mean and 90 percent density contour levels of the scatter plot of f vs percentage identity. The half maximal value $P_{f=\text{fmax}/2}$ varies from 60–80 percent in the different study cases. Table 3 does include results from 1DIST(1) because it emphasizes that Fastfinger does contain more information, corresponding to that specifically excluded from 1DIST(1), despite the compression process.

The relation between Fastfinger and log (base 10) of E-score of BLAST, which is routinely used in rapid scans for related sequences in large databases, is indicated in Figure 4. Values less than $f = 0.5$ (i.e., FI = 50 percent) are promising also by BLAST standards, and f values of 0.65 (FI = 65 percent) or less are worthy of further study. Note that only sequences within 30 percent comparable length were used. However, this selection applied only to total

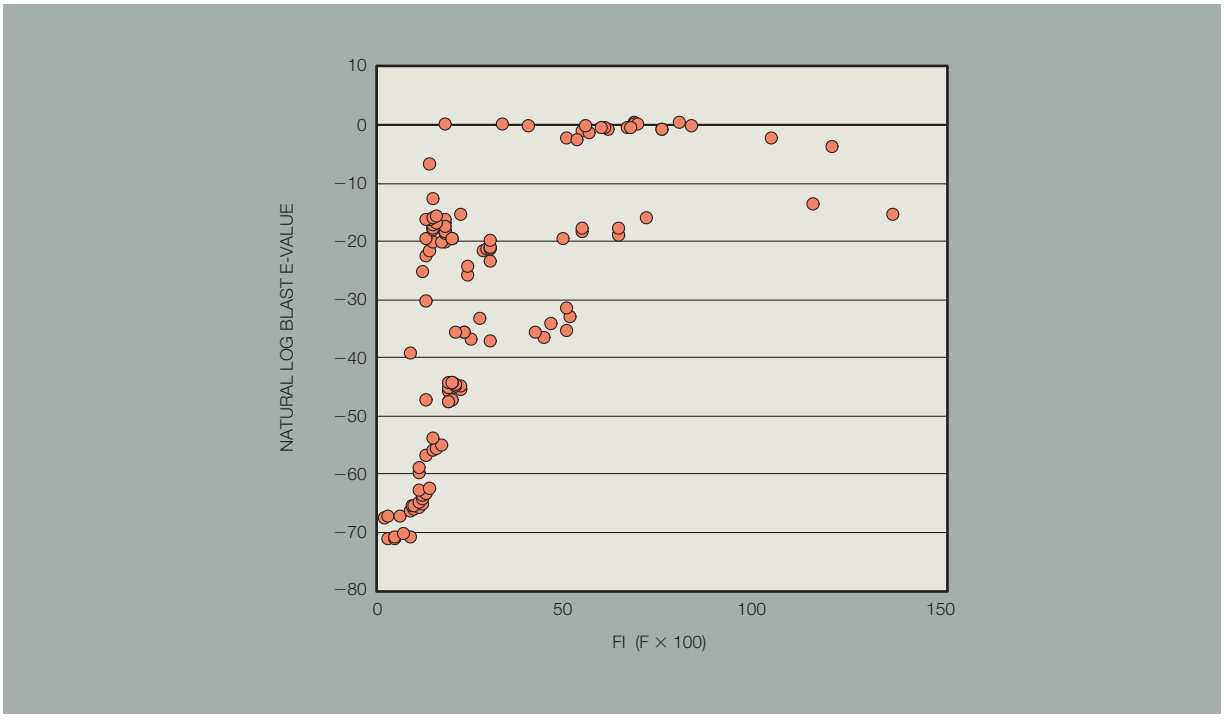
Table 2 Best-fit-curve properties of 20 x 20 preferred method (compressing information for multiple separations of a pair of residue types into a single scalar quantity). A high-quality fit requires a higher degree polynomial, but curves of f vs (100 – percentage identity) are approximately parabolic. They are also sufficiently close to linear f = slope x (100 – percentage identity) + intercept, for most purposes, except for high f and weak identity less than 20 percent, with only about 3 percent stress. This roughly linear form, with an intercept close to zero, is convenient: for sequences of similar length, multiplying f by 65 and subtracting from 100 percent will give a rough indication of identity. "Randomized sequences" means that one of a pair of identical sequences is randomly "mutated" to create identities from 100 to 20 percent.

Curve Fitted to Scatter Plot of f vs Percentage Nonidentity of Pairs of Sequences, in Region of Greater Than 20 Percent Identity	Slope	Intercept	fmax 1DIST(1) ($P_{f = fmax/2}$ Varies from 60 to 80 Percent)
Best-fit curve to f vs percentage nonidentity of sequence pairs	0.65	~0	0.9
Upper 90 percent contour of randomized sequences	1	0.13	0.95
Lower 90 percent contour of randomized sequences	0.55	0	0.85
Upper 90 percent contour of real sequences	0.75	0.26	0.95
Lower 90 percent contour of real sequences	0.5	~0	0.65

Table 3 How f varies with percentage identity, studied over many sequences, and expressed in terms of the percentage identity implied by the f measure. The relationship is not a one-to-one mapping, so one value of f can imply a range of percentage identities (and vice versa). The third column shows the performance of the standard recommended method (compressing information on multiple separations of a pair of residue types into a single scalar quantity). For comparison, the second column shows results for 1DIST(1) (which does not use this compression) and retains only the minimum separation encountered for a pair of residues. This reduced method is less informative for more distant sequences with poorer degrees of identity: the correlation with f is much poorer, and the ranges of percentage identity overlap broadly. Clearly more separation distance retains more information for the more distant relationships despite the compression process, which is based on the prime factorization theorem.

f	Percentage Identity of Sequences Implied: 1DIST(1)	Percentage Identity of Sequences Implied: 20 x 20 Method
0.65–1.00	0–60%	0–20%
0.35–0.45	55–87%	20–65%
0.25–0.35	70–90%	40–80%
0.0–0.15	90–100%	80–100%

Figure 4 Relationship between Fastfinger and BLAST, where natural log of the BLAST E-value is greater than −73



number of residues per sequence, and so sequences with insertions and deletions of very considerable length in different regions were allowed. Also, since for close homologues BLAST E-values have large algorithmic values, only points where the natural log of the BLAST E-value is greater than −73 are included. Fastfinger and BLAST methods essentially agree on the idea of close relatedness when $f < 10$, which is an important consideration for rapid identification of related sequences in a large database, including differences due to sequence errors, polymorphisms, or species variations. The divergence for higher values is not surprising, however, in view of the differences in philosophy. BLAST better distinguishes closely related proteins, while Fastfinger resolves sequences in the E-value "twilight zone" of around zero. Whether these are "false positives" in terms of biologically or physically meaningful resolutions remains to be determined, but they represent real differences in pattern, at least in the specific terms of this study, and hence similarities and differences so resolved are at least worthy of further study.

In a computer experiment, a similar study to that shown in Table 3 was done, except that for the matrices only three symbols, H, E, and C, of the observed secondary structure were used. The method has some resolving power, which relates to secondary structure threading and will be described elsewhere. For proteins of 40–60 percent customary sequence identity, 1DIST(1) obtained a secondary structure-based f of 0.5–0.75. The f value obtained for the same proteins from the recommended 20 × 20 matrix method was 0.3–0.43.

Contribution of "domains." The finger arrays are additive, such that a protein sequence that can be considered as composed of two parts, say AB, can be compared with two smaller proteins, or fragments of sequence, separately. This makes them well-suited to the study of domains, or segments found in different proteins in different orders and contexts.

First, it may be demonstrated that the method is insensitive to the order in which domains or any other segments may occur, provided that the length is greater than the window length W. While we are in-

Table 4 Effect of altering order of segments of sequence

	Previous f	f after "Mixing"
Unchanged	f = 0.000	f = 0.000
Reverse order of 160 residue segments	f = 0.000	f = 0.032
Reverse order of 80 residue segments	f = 0.000	f = 0.053
Reverse order of 20 residue segments	f = 0.000	f = 0.148
Reverse order of 20 residue segments, averaged over cases with 45–55 percentage sequence identity	<f> = 0.366	<f> = 0.384
Reverse order of 20 residue segments, averaged over cases with 15–25 percentage identity having a common fold	<f> = 0.559	<f> = 0.575

terested in a difference "penalty" measure and in measuring new pattern content, say due to one sequence having domains or loops not present in the other, the "shuffling" of existing patterns does not constitute new pattern. Table 4 confirms that the order does not greatly affect the f measure between two sequences. The shuffling method is shown in the table. "Reverse order of 80 segments," for example, means that the sequence is broken up into segments of length 80, then the segments are rejoined except that their order is reversed. In order to compare examples where f was initially nonzero, the study selected two different sequences and performed the shuffling operations on just one. The average of cases with 45–55 percent sequence identity were used, and also of 15–25 percent sequence identity, provided they were known to have a common fold.

Second, appropriate behavior with respect to comparison with parts from several other sequences can be demonstrated. The operation of subtraction can be performed more than once, and the absolute value of the difference for each element is taken when the operations are completed. In the method described previously, the matrix $F_{A-B} = F_A - F_B$. In this use, the operation of subtraction also indicates that the absolute value of the difference of the two corresponding elements is taken. In the same notation, the difference between probe sequence A and two proteins or protein fragments B and C might be represented by $F_{A-B-C} = F_A - F_B - F_C$. This is demonstrated for the case where $B = C$ by preparing a dimer sequence A, which is the same sequence extended once by a copy of itself. The above theoretical considerations would predict that the score for F_{A-B-C} would be close to zero in such a case. Table 5 shows the final scores. Here 4APE1E is an endothiapepsin and 2APP1E is a penicillopepsin. These are acid proteases and weakly related.

Note that a low score of 0.017 is obtained if F_B is subtracted twice, reflecting the fact that two copies of sequence B are found in the probe protein sequence dimer A. The value is not exactly zero because new pattern components appear in the region where the two sequence copies are spliced together. Note that these proteins, prior to the artificial doubling in length, already naturally consist of two weakly related domains—the first half and second half of 4APE1E relate to each other with a score of 0.276 and the first half also relates to the first and second halves of 2APP1E, of corresponding length, with scores of 0.219 and 0.2843, respectively. The method can be readily extended to simultaneous comparison with more than two proteins or protein fragments by generalizing to $F_{A-B-C-D-\ldots}$. The general problem is to find the solution of the coefficients (i.e., weights) of the component terms that can be determined by optimization or by successive testing of each test protein or protein segments, with respect to their f score.

Statistical properties of finger matrices. Of particular interest is the underlying cause of the spread of

Table 5 Example of identification of domains and repeated domains

	Target = 4APE1E	Target = 4APE1E Dimer	Target = 2APP1E
F[4APE1D dimer] – F[target]	0.825	0	0.998
F[24APP1D dimer] – 2 x F[target]	0.017	1.633	0.743

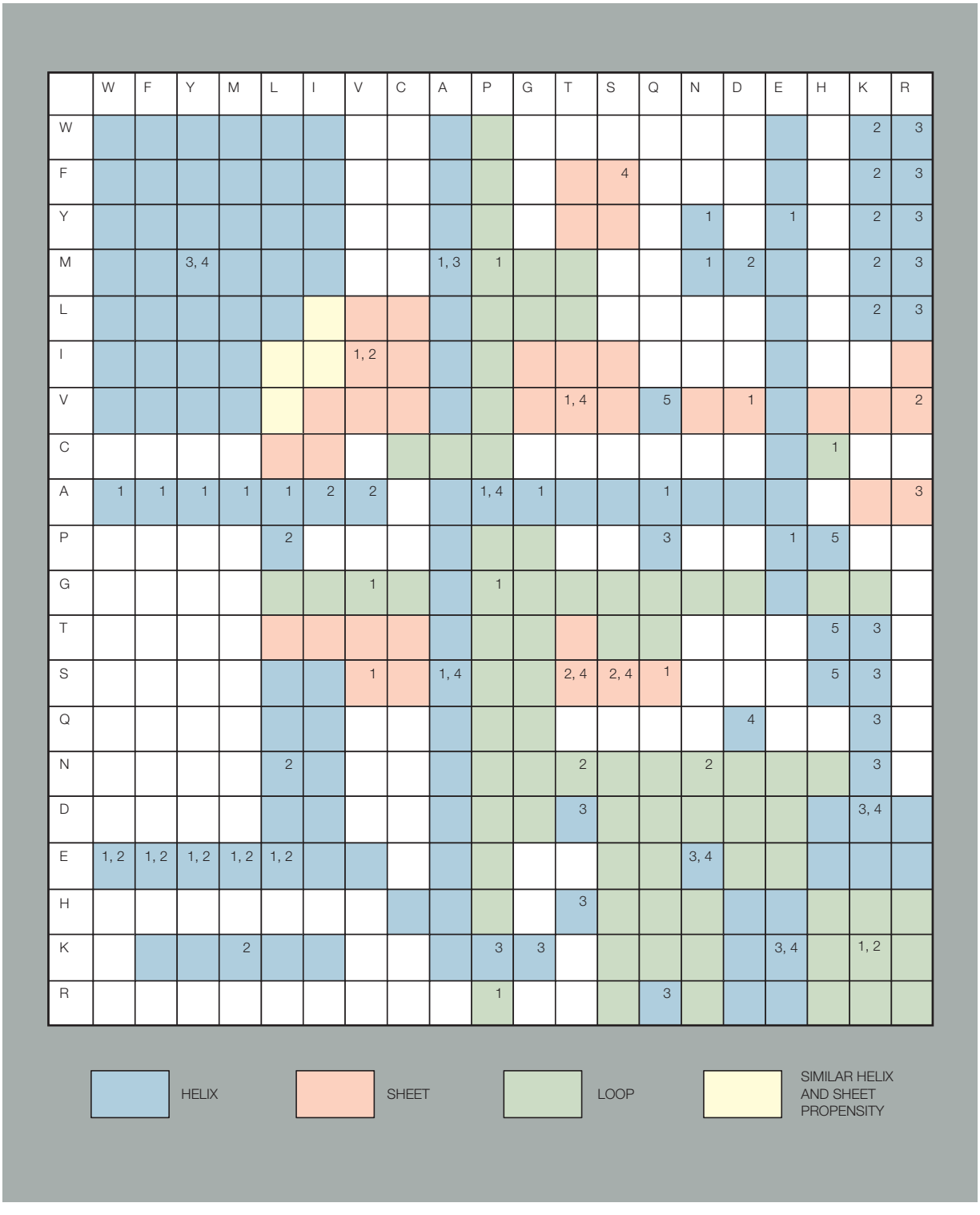
$\mathfrak{f}$ values for sequences of similar BLAST E-scores. While differences in the number of residues of each sequence, including insertions and deletions, play a role, the effect remains for sequences of very similar length and no insertions or deletions. Preliminary results are presented in Table 6 for the data of Figure 4. Let the experimental value of an element in a difference matrix be v, and $N(v)$ the number of elements within a small cell of that value. Although seemingly at first an exponential distribution, decreasing with increasing value v of the difference elements, a peak is apparent in the region of $2 \leqslant v \leqslant 5$.

However, these features vary significantly in different specific instances, and particularly so for the less related sequences. One possibility suggested by observations of a limited sequence set was that, for the more distant relationships, statistical distributions of the terms of the difference matrix F_{A-B} change when the two proteins are, or are not, of comparable secondary structure. For example, pleated-sheet-rich proteins may have different sequences and even rather different folds, but a distribution more closely resembling the case $0.333 < \mathfrak{f} < 0.666$ than $\mathfrak{f} > 0.666$ can be obtained. A similar secondary structure tends to make the two proteins look more similar in terms of their $\mathfrak{f}$ score. This is interesting, because specific pairs at various separations may be better predictors of secondary structure than single residues, and it suggests a reason why the finger matrices might be genuinely picking up meaningful variations.

Studies relating proteins of similar and different secondary structure are available;[15] they support the above hypothesis but are preliminary. The required correlation can, however, be demonstrated directly as follows. The idea is that, for example, a leucine–leucine pair in a protein, especially at separations i, $i + 3$ and i, $i + 4$, is more likely to be associated with a helical region in the source sequence. A matrix derived from a sequence that is rich in such pairs, and others indicative of helix, is likely to represent a sequence rich in helical secondary-structure content. Difference matrices for segments of sequence that have low values of $\mathfrak{f}$ are likely to imply that the two proteins are of similar secondary structure, and of course a low $\mathfrak{f}$ value overall will indicate a likely common secondary structure and tertiary structure. For example, even if a leucine–leucine interaction was absent in one matrix, other pairs with similar helix-forming propensity are still likely to be retained.

Figure 5 shows a 20×20 finger matrix for a general probe (or test database) sequence, indicating *a priori* the pair separation that is most likely to occur, given the secondary structure conformation (helix, sheet, or loop) from which it occurred. Conversely, it may be considered as showing the most likely secondary structure if a single pair at a known separation is encountered. The matrix is obtained by constructing matrices for the sequences considered in the above studies, without compression, and counting the frequencies of the occurrences. In practice, particular information measures[12,16] are used because these compensate for excess or deficiency of certain secondary structure types in a sample, evaluated as $I(X: \sim X; A, B) = \ln(n[X, A, B]/n[\sim X, A, B]) - \ln(n[X]/n[\sim X])$, where $n[X, A, B]$ is the number of times that a pair of amino acid residues A, B occurs in secondary state X (say helix), and $n[\sim X, A, B]$ is the number of times that pair A, B occurs in state other than X (say nonhelix, i.e., sheet or loop), such measures being conditional on the particular sep-

Figure 5 Finger matrix (20 × 20) for a test database sequence



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

Table 6 Example distributions of the number of elements N(v) in a difference matrix of 400 elements, which are associated with particular ranges of difference value v. Arbitrary examples are chosen from the ranges f < 0.333, 0.333 < f < 0.666, and f > 0.666, roughly indicating plausible homology, marginal evidence of homology, and no evidence of homology.

Value (v)	N(v) f < 0.333	N(v) 0.333 < f < 0.666	N(v) f > 0.666
0	135	80	17
1	28	27	14
2	34	38	19
3	41	23	21
4	20	14	14
5	18	21	13
6	13	15	17
7	12	16	13
8	11	14	14
9	9	11	16
10	7	0	13
11	5	9	11
12	6	8	12
13	5	8	9
14	4	6	11
15	3	6	9
16	2	4	7

aration (say, A at i and B at $i + 3$).[17] Though Robson and Garnier[14] originally used a more rigorous Bayesian estimation method used for finite n, both a rigorous approach, and simple use of terms n [ ] (i.e., the number of occurrences of events as directly counted) gave similar results in this study.

Numbers in Figure 5 indicate major separations m (row residue at i, column residue at $i + m$) and are specified more than once in a continuous zone only when needed to resolve ambiguity. Color does not relate to any notion of strength of effect but distinguishes the secondary structure states preferred, as shown in the color key in Figure 5. For simplicity, "ambiguous" regions of fine balance are in white, though there is sometimes still a strong preference to two states only, here indicated by pale yellow. For example, in this sample there was a fine balance between helix (3, 4) and sheet (1, 2) for squares at (row-column) L-I, I-L, V-L, and even I-I (though the latter has a rather clearer bias to sheet), but occurrences in loop in all cases are very rare.

Conclusions

This study explores whether a method of compressing information, based on the prime factorization principle, is of potential value in comparing sequences. This approach cannot yet be recommended to replace existing methods, and any competitive challenges to the established methods are likely to be, at best, through substantial modifications of these simple approaches. However, the relatively crude compression methods used here clearly retain useful information. Indeed, it is noteworthy that the information content of the method is sufficient to state the sequence comparison conclusions in Table 7.

The biggest difficulty is that a contribution to the Fastfinger measure arises simply because of any length differences between the two sequences being compared. An original intent was to proceed as follows. Let two sequences be of lengths L and M. The contribution arising from the difference in length may then be estimated by $k|L - M|$ where constant of

Table 7 Fastfinger results and likelihood of homology

FI (= 100 x f)	Conclusion
FI = 0 ... 19%	Homologous
FI = 20 ... 39%	Very likely homologous, probable overall significant sequence identity
FI = 40 ... 49%	Possibly homologous, likely regions of significant sequence identity
FI = 50 ... 65%	Possible local similarities or possible conformational relationship without good sequence relationship; worthy of further study
FI > 65%	Unlikely to be related. Reject.

proportionality k is calibrated as a constant in advance, or more specifically, estimated from the statistical properties of the log properties of prime numbers of the sequences involved in the comparison. The correction method imagines that the insertion regions *do* contain residues, and the net effect of the appropriate types and separation characteristics that would then occur is predicted. To put it another way, the approach would make use of the idea that an insertion contributes the log product of primes in the normal way, and the corresponding deletion contributes zero, and seek to compensate for that effect. When the approach is so expressed, however, it becomes questionable whether such a compensation is justified within the philosophy of the method, because all pattern differences, irrespective of whether they can be associated with insertions and deletions, are scored on the basis that a nonexistent separation in the other sequence contributes zero, without any such "compensation." Further, from any perspective, this method of compensation uses the idea that one can focus on a specific region of matching in which an insertion or deletion event occurs; and for sequences with weak identity, it may not be possible to locate such regions unambiguously, even in principle. In the present study such compensation has not been used, and indeed it is remarkable that relatedness can be detected despite difficulties due to length differences, at least to a point where a need for further study can be identified. To achieve this requires that we accept a number of false positive matches, and conversely it requires that care is taken, and further study indicated, when sequences differ greatly in length.

Fastfinger represents a relatively unusual class of method, which, it may be argued, encompasses aspects of general homology-based methods and local pattern-motif methods. Nonetheless, the novel philosophy of the method poses some difficulties for comparison with established methods. To overcome some of the difficulties, one might address the concept of "utility." Utility can of course differ from application to application, as between function identification as opposed to an interest in common ancestry, or the issue of the extent to which sequences have a common fold. A further method, Frankenstein, has been developed to test very rapidly the performance of Fastfinger and related variant methods in the specific domain of protein modeling. For example, the difficulty in modeling by homology appears to be a reasonably smooth function of the similarity measures returned by Fastfinger. Details will be reported elsewhere, but one observation that is worth noting is that the amount of computation required in modeling rises roughly linearly with increasing difference between sequences, at least when f is small.

Despite these cautionary notes, it is fair to say that the method has proven of worth, even in its present simple form. Notably, although it is too early to make a general "statistical" statement about the incidences where there has been utility in regard to identifying

proteins of related function or fold compared with other methods, the importance of discovering functional and conformational relationships in difficult cases, even when many false positives are generated, is such that even occasional discovery by an extra pattern comparison measure justifies its use. Hence, for detection of potential cryptic relationships, the "trick" is not to use the method in isolation but as an adjunct to other methods.

As indicated by Figure 2, BLAST E-scores and percentage identity (or its estimation as described in the methods section) are usually reported together. The method modestly extends the armory of tools, and in view of this, one may envisage implementing a miniature "expert system" not only to compare the techniques on their merits for each case, but also to invoke each at its most appropriate level for filtering out candidates. Last but not least, note that it is remarkably simple to code the Fastfinger method, compared with most of the established techniques, and so to supplement, and explore the merits in conjunction with, one's own favorite methods.

Cited references and notes

1. M. S. Waterman, *Introduction to Computational Biology*, Chapman & Hall, London (1988).
2. S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic Local Alignment Search Tool," *Journal of Molecular Biology* **215,** No. 3, 403–410 (1990).
3. W. R. Pearson and D. J. Lipman, "Improved Tools for Biological Sequence Comparison," *Proceedings of the National Academy of Sciences (USA)* **85,** No. 8, 2444–2448 (1988).
4. I. Rigoutsos, A. Floratos, C. Ouzounis, V. Gao, and L. Parida, "Dictionary Building via Unsupervised Hierarchical Motif Discovery in the Sequence Space of Natural Proteins," *Proteins: Structure, Function, and Genetics* **37,** No. 2, 264–267 (1999).
5. A. Califano and I. Rigoutsos, "FLASH: A Fast Look-up Algorithm for String Homology," *Proceedings, First International Conference on Intelligent Systems for Molecular Biology*, Bethesda, MD (July 7–9, 1993).
6. G. Stolovitsky and A. Califano, "Discrete Applied Mathematics Series," P. Penver, Editor, submitted.
7. I. Rigoutsos and A. Floratos, "Motif Discovery Without Alignment or Enumeration," *Proceedings, Second Annual ACM International Conference on Computational Molecular Biology*, New York (March 22–25, 1998).
8. Several referees and colleagues objected to my particular uses of the words "homology" and "homologous" in the original manuscript, as differing from that adopted in molecular biology for the past ten years or so. I have adjusted the present text accordingly, but I have never been comfortable with the more recent usage that relates to the idea of common biological origin and departs from the original definition. (For discussion, see B. Robson and J. Garnier, *Introduction to Proteins and Protein Engineering*, First Edition, Elsevier Press, Amsterdam (1984), pp. 235–238, and also in relation to the use of the term "conservative substitution" as discussed by French and Robson.[12]) First, the modern definition used within molecular biology does not correspond to the mathematical definition. The latter relates more to the idea of correspondence, and makes no stipulation of a (hypothetical) common origin back in time. It seems quite reasonable to speak of sequences of symbols as being homologous, irrespective of the hypothesis that there is a common origin of the genes for the proteins that they represent. The use of "homology" as revealed through finger matrices makes particular sense, because mathematical uses of the word, having to do with correspondence between vertices of an object, can be shown to be much closer to the matter of the correspondences between matrices of pair-wise residue symbol separation distances. Second, at least one referee held to the definition that homology is a binary state: the hypothesis is either true or false, so homology, if one uses percentage notation, would be either 0 or 100 percent. Another expressed the view that since homology expresses common ancestry, "100 percent homology" is simply not meaningful. However, even leaving aside the argument that "percentage homology" is a reasonable shorthand for "estimates of the probability (on a percentage basis) of a hypothesis about homology being true," it is arguable that the hypothesis of common ancestry is not usefully treated as a binary state. Rather, there are multiple subhypotheses about states representing gradations, relating to the tree structure of evolution (over which gradations, we can also distribute our held degrees of belief). The foreleg of a horse is homologous to the wing of a bird in the sense that they are believed to be of common origin, but it does not seem unreasonable to state that it is "more homologous" to the foreleg of a cow, both in terms of contemporary points of correspondence, and in terms of the hypothesis of a more recent common origin.
9. T. Nagell, *Introduction to Number Theory*, John Wiley & Sons, Inc., New York (1951).
10. E. Nagel and J. R. Newman, *Goedel's Proof*, New York University Press, New York (1958).
11. R. Perline, "Zipf's Law, the Central Limit Theorem, and the Random Division of the Unit Interval," *Physical Review E* **54,** No. 1, 220–223 (1996).
12. S. French and B. Robson, "What Is a Conservative Substitution?" *Journal of Molecular Evolution* **19,** 171–175 (1983).
13. S. Henikoff and J. Henikoff, "Amino Acid Substitution Matrices from Protein Blocks," *Proceedings of the National Academy of Sciences (USA)* **89,** No. 22, 10915–10919 (1992).
14. J. Garnier and B. Robson, "The GOR Method for Predicting Secondary Structures in Proteins," *Prediction of Protein Conformation and the Principles of Protein Conformation*, G. D. Fasman, Editor, Plenum Publishers, New York (1989), Chapter 10, pp. 417–465.
15. Robson, unpublished manuscript.
16. S. D. Silvey, *Statistical Inference*, Penguin Books, London (1970).
17. I(X: $\sim X$; A, B) may be read as "the information provided by A and B as to whether X will or will not occur," or "the information that the joint event A and B carries about X as opposed to any other possibility than X."

Accepted for publication March 23, 2001.

Barry Robson *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598 (electronic mail: robsonb@us.ibm.com).* Dr. Robson is Strategic Advisor to IBM's Computational Biology Center. He was an "early player" in bioinformatics, protein modeling, and computer-

aided drug design and is a coinventor of several successful therapeutics and diagnostics, including the Proteus/Protherics/Enfer diagnostic for mad cow disease. He was awarded a Ph.D. degree from the University of Newcastle upon Tyne for experimental and theoretical studies in protein folding in 1972, a D.Sc. degree from the University of Manchester for computational biochemistry in 1984, and the title of Distinguished Engineer from IBM for contributions to bioinformatics in 1998. He is also Professorial Lecturer at Mount Sinai Medical School, New York. Dr. Robson sat on the board of five biopharmaceutical companies and was the scientific founder of the Proteus group of pharmaceutical companies, where he served as Science Director for nine years, and The Dirac Foundation at the Royal Veterinary College, London, where he served as its Chief Executive Officer and Chairman. He was Visiting Scholar and lecturer in bioinformatics at Stanford University Medical School in California, where he also assisted California companies in start-up ventures, specializing in industrialization of academic research and development, notably as Chief Science Officer at Gryphon Sciences and as Principal Scientist at MDL Information Systems. Dr. Robson is author of some 190 papers, books, and patents and was a *Nature* "News and Views" correspondent on proteins for five years. His current research is in exploration of possible novel algorithms as bioinformatics tools, protein modeling, and linkage of genomics to the electronic patient record for more personalized drug design.