Pergamon

0097-8485(95)00045-3

Computers Chem. Vol. 20, No. 1, pp. 103–118, 1996
Published by Elsevier Science Ltd
Printed in Great Britain

THE GENE IDENTIFICATION PROBLEM: AN OVERVIEW FOR DEVELOPERS

JAMES W. FICKETT

Theoretical Biology and Biophysics Group, MS K710, Los Alamos National Laboratory, Los Alamos, NM 87545, U.S.A.

(Received 1 December 1994)

Abstract—The *gene identification problem* is the problem of interpreting nucleotide sequences by computer, in order to provide tentative annotation on the location, structure, and functional class of protein-coding genes. This problem is of self-evident importance, and is far from being fully solved, particularly for higher eukaryotes. Thus it is not surprising that the number of algorithm and software developers working in the area is rapidly increasing. The present paper is an overview of the field, with an emphasis on eukaryotes, for such developers.

INTRODUCTION

In a rapidly moving field it is often easy to trace individual threads of work, but difficult to gain an overview. The first purpose of this review is to provide a concise directory to both standard and newer techniques, and so allow new developers to more quickly come to the point where they can make their own original contributions.

The second purpose is to give some perspective on the structure of the field and current research directions. This includes summarizing the high points of progress to date in each of several areas, evaluating what seem to be the most productive current lines of inquiry, and attempting to predict where the most useful developments will come from in the future. While large parts of this perspective are shared by many practitioners in the field, the overall analysis necessarily represents the personal views of the author.

A number of related reviews exist. A few of the more recent works on sequence analysis in general are Adams *et al.* (1994), Doolittle (1990), Gelfand (1995), Gindikin (1992), Gribskov & Devereux (1991), Griffin & Griffin (1994), Konopka (1994a) and Waterman (1989a, 1995). On-line bibliographies of publications relevant to analysis of nucleotide sequences are maintained by A. Bairoch (SEQANALREF; URL http://expasy.hcuge.ch on the World-Wide Web) and M. Gelfand (FANS-REF; ftp to imb.imb.ac.ru; in directory BIBLIO). Staden (1990) and Gelfand (1990b) give overviews of the gene identification problem. Fickett & Tung (1992) review recognizable statistical regularities in protein coding regions. Doolittle (1986) and Gish & States (1993) discuss the interpretation of similarity schemes in the context of gene identification.

The present review is primarily a guide to current techniques relevant to future development, rather than being a guide to current tools. The review is mostly restricted to published work, though unpublished developments may be mentioned briefly. In most sections, coverage is limited to techniques that are either widely used, or which seem to us to be particularly important for future developments. Although the number of papers cited is already large, there are doubtless many others that should have been included. Experimental approaches to gene identification are assuming an increasing importance. These will not be covered here, but the computational developer must stay abreast of the rapid developments in experimental techniques as well. For a recent overview see Church *et al.* (1994).

The paper begins with a definition of the problem. The main body of the paper consists of an overview of computational tools and techniques broken into six (somewhat arbitrary) categories:

- Sequence similarity search
- Statistical regularities in exons
- Signals: introduction
- Signals: basal gene biochemistry
- Signals: regulation of gene expression
- Gene syntax and integration of information.

In each of these categories the state of the field is summarized. In the last two sections, some higher level issues are considered.

DEFINITION OF THE PROBLEM

Sequence (old or new) to biochemistry

A framework for much of the work in computational analysis of nucleotide sequences may be had by seeing this work as directed towards the eventual

goal of automatic annotation: automatically producing a draft feature table that is as complete, accurate, and interesting as possible. Sequence "features", in the common usage of the term, include many kinds of information; the core problem in automatic annotation is to describe the sequence in functional terms. Concretely, this means to discover all biochemically active sites in a region of a DNA/RNA molecule, and describe the associated reactions and reaction products.

The ability to predict the biochemistry of a new sequence—one under design, say, by a pharmaceutical company—in a specific context, is of just as much interest as the ability to discover the function of naturally occurring genomes. One very important long term goal of nucleotide sequence analysis, then, is to generalize from the biochemistry of natural genomes to give rules for designing new genes and genomes.

The current gene identification problem

Although the identification of protein coding genes is clearly influenced by the knowledge of other significant features of the sequence, the difficulty of considering the automatic annotation problem as a logically integrated process has caused the gene identification problem usually to be considered independently of most other sequence analysis. Most of the rest of the paper will follow this tradition.

Eukaryotic gene regulation is complex and is just beginning to be understood. It still seems a rather difficult goal even to predict from sequence the course of the key biochemical reactions of gene expression: transcription, splicing and translation. At the present time the success of gene identification algorithms is measured in terms of the degree to which they correctly predict the amino acid sequence of protein products and, perhaps, some hint of product function.

COMPUTATIONAL TECHNIQUES

Overview

The sections that follow survey the various computational techniques relevant to gene identification. In the first five sections, methods for recognizing some particular aspect, or component, of genes, are covered. The last section then covers methods of integrating all the evidence and components into higher level statements about genes.

There are several higher level issues one should keep in mind. One is that the efficacy of many of the methods is still being debated, or in some cases, has not yet been challenged or tested. Each section will summarize what is known about the practical value of the techniques covered.

There is an emerging issue, possibly of fundamental importance, in the development of techniques for gene identification, which might best be expressed as the tension between template methods and lookup methods (termed "intrinsic" and "extrinsic" approaches in Borodovsky *et al.* (1994a, b). Template methods attempt to compose more or less concise and elegant descriptions of prototype objects, and then identify genes by matching to such prototypes. A good example is the use of consensus sequences in identifying promoter elements or splice sites. Lookup methods, on the other hand, attempt to identify a gene or gene component by finding a similar known object in available databases. An excellent example of a lookup method is searching for genes by trying to find a similarity between the sequence under analysis and the contents of the sequence databases.

Much of the work that comes out of a mathematical or computational background (including pattern recognition in particular) focuses on deriving prototype descriptions from the data. This approach often makes important contributions to our understanding, but usually leaves out important exceptions and ambiguities, most likely because genomes are not elegantly designed from scratch, but are a collection of contraptions honed by experience. Thus as the field has developed, and as molecular biological data have increased, lookup methods, which simply rely on what is, without attempting to summarize it neatly, have gained in importance.

Finally, it should be noted that the field as a whole is making a transition from studying primarily components of genes to studying genes and genomes in their entirety. Thus the issue of choosing an appropriate language in which to express and integrate the knowledge gained from the component calculations is one of the most active areas in computational gene identification.

Sequence similarity search

One of the oldest methods of gene identification, based on sequence conservation due to functional constraint, is to search for regions of similarity between the sequence under study (or its conceptual translation) and the sequences of known genes (or their protein products). A recent, large-scale example of the application of this method, clearly illustrating both its power and its difficulties, may be found in Robison *et al.* (1994).

A clear advantage to searching for genes by similarity is that, if a significant similarity is found, it is likely to yield clues as to the function, as well as the existence, of the new gene. In addition, if the search is carried out at the amino acid, rather than the nucleotide, level, the additional advantage may be had of lowered sensitivity to the "noise" of neutral mutations. The obvious disadvantage of this method is that when no homologues to the new gene are to be found in the databases, similarity search will yield little or no useful information.

The question naturally arises, then, of the likelihood that the databases will contain a homologue of a gene awaiting discovery. Seely *et al.* (1990), in an early attempt to answer this question, took one half of GenBank release 56 as a test set, introduced

"mutations", "introns" and "intergenic DNA" to make the test set resemble new genomic data, and searched for genes in this test set by comparing it to the remaining half of GenBank as a reference set. In this experiment, they found that approximately three-quarters of the genes could be clearly identified. Thus one might hope that the majority of new genes could be found by means of simple similarity searches in the database.

When the complete sequence of yeast chromosome III (Oliver *et al.*, 1992) was first reported, 26% of the putative protein products (conceptual translations of all open reading frames over 300 bp in length) were found to have significant similarity with some other known sequence. Similarly, in reporting analysis of three cosmid sequences from *C. elegans*, Sulston *et al.* (1992) state that roughly a third of the putative genes show clear homology to sequences already in the databases. Both of these estimates have rather large error bounds, as the list of tentative genes depends primarily on computational, not experimental, evidence. Yet these studies do seem to suggest that the conclusions of the Seely *et al.* study are perhaps too optimistic. Probably the disparity between the simulation study and the results of actual genomic sequencing are due to the biased nature of the databases. For example, both of the halves of GenBank used in the experiment of Seely *et al.* are much richer in highly expressed genes than is a eukaryotic genome *in toto*.

One overall lesson from a long line of work studying amino acid sequence motifs and blocks from related sets of proteins (cf. Gribskov *et al.*, 1987; Posfai *et al.*, 1989; Smith & Smith, 1990; Smith *et al.*, 1990; Henikoff & Henikoff, 1991; Bairoch, 1992; Ogiwara *et al.*, 1992), is that database searches seem to be much more sensitive if carried out with meaningful patterns such as motifs or profile matrices. When Bork *et al.* (1992a, b) studied the yeast chromosome III sequence using more permissive cut-off scores, multiple alignment methods, and motif searches, 42% of the putative genes were found to be similar to a known sequence or motif. Later, Koonin *et al.* (1994) revised the list of putative genes and again used the most recent and sensitive known algorithms, and found that 61% of the putative proteins exhibited significant similarities to known proteins or motifs. This increase is due in part to revisions in the list of putative proteins, in part to the databases becoming more complete, and in part to improvements in computational methods.

In another vein, current efforts to sequence (at least fragments of) all transcribed sequences from a number of genomes (e.g. Adams *et al.*, 1991) concentrate much of the genomic information necessary for gene identification. Boguski *et al.* (1994) collected the 32 human disease gene sequences that have been positionally cloned to date and found that 85% of them showed homology to an entry in the dbEST collection (Boguski *et al.*, 1993) of expressed sequence tags. This is a small sample, but the indication still seems strong that cDNA sequence collections will be an important resource for gene identification. Note, however, that for most of the sequences in dbEST, the only information available is that they are transcribed; mapping and functional data will surely come, but are presently accumulating much more slowly than the sequences themselves.

How fast will the fraction of genes identifiable by similarity search go up? Green *et al.* (1993) [see also Claverie (1993a) and Green (1994)] compare recently determined sequences both to each other and to older sequences in the databases, and conclude that: (1) most *ancient conserved regions* (or ACRs, roughly defined as regions of protein sequences showing highly significant homologies across phyla) of the protein universe are already known and may be found in current databases; (2) roughly 20–50% of newly found genes contain an ancient conserved region that is represented in the databases [cf. also Borodovsky *et al.* (1994a)]; and (3) rarely expressed genes are less likely to contain an ancient conserved region than moderately or highly expressed ones.

Taken together, these results seem to suggest that on the order of one-half of all new genes may be discovered, and perhaps some functional information determined, on the basis of similarity to known sequences or motifs, and that this fraction will continue to rise. Due to the larger variety in non-ACR-containing proteins, however, the rise will likely be rather slow.

Sequencing errors, particularly frameshift errors, can be a serious problem for gene identification by similarity search. Gish & States (1993) discuss the effects of such errors, and the interpretation of BLASTX search results. Shavlik (1994) shows how to turn the difficulty to advantage, piecing together matches from different frames both to locate genes and to detect the sequence errors. Claverie (1992) also discusses practical aspects of similarity searching, in particular providing a means to eliminate the most common source of high scoring similarities *not* due to gene function, namely repeats.

Statistical regularities in exons

At the core of most gene recognition algorithms are one or more *coding measures*—functions which calculate, for any window of sequence, a number or vector that measures attributes correlated with protein coding function. Aggregate properties of such function values on coding regions thus form templates for exons in general. Common examples of coding measures include the codon usage vector, the base composition vector, and some type of Fourier transform of the sequence. These measures, which have a long and rich history, have been reviewed, synthesized, and uniformly evaluated in Fickett & Tung (1992) [cf. also Gelfand (1990b)]. The measures tested there are the following (for more details and full citations see the review; in the definitions that follow, the "test-codons" of an arbitrary sample window of

sequence are defined as the successive non-overlapping trinucleotides of the window, beginning with the first base).

Codon Usage Measure: The 64 element vector giving the frequencies, among the test-codons, of each of the 64 possible codons.

Hexamer-n Measure (for $n = 0, 1, 2$): The counts of all hexamers offset by n from the starting base of a test-codon. (The Hexamer-0 measure gives dicodon counts.)

Hexamer Measure: The frequency count in the window of all hexamers.

Open Reading Frame Measure: The length of the longest stretch of sense test-codons in the window.

Amino Acid Usage Measure: The 21-vector obtained by translating the sample window of sequence, beginning with the first base, according to the appropriate genetic code, and counting the frequencies of the 20 amino acids and "stop".

Diamino Acid Usage Measure: The 441-vector given by translating the window and counting all the (overlapping) dipeptides (including "stop" as an "amino acid").

Stability of Hydrophobicity Measure: First define the information value of a codon as $\Sigma_{j=1,3}[\Sigma_{i=1,nj}(p_i \times d_{ij})]/n_j$, where n_j is the number of sense mutations of the codon, p_i is the probability of the ith mutation, and d_{ij} is the difference in hydrophobicity caused by the mutation. The information value of a window, which we take as the Stability of Hydrophobicity Measure, is then the average information value of the test-codons in that window.

Composition Measure: $[f(b, i)]$, where for each base $b =$ A, C, G, T and each test-codon position $i = 1, 2, 3$, $f(b, 4i)$ is the frequency of b in position i.

Codon Prototype Measure: Let $p(b, i)$ be the probability of finding base b at position i in an actual codon. Let $q(b, i)$ be the probability of finding nucleotide b at position i in a trinucleotide that is not a codon. Consider p and q to be 4×3 matrices, with rows indexed by the bases $b =$ A, C, G, T. Let B be the matrix with element $(b, i) = p(b, i) - q(b, i)$. B can be considered a linear function on trinucleotides in an obvious way: each base b of a trinucleotide may be considered a column vector of a 3×4 matrix, with a 1 in the bth row. Then B of that trinucleotide is the dot product of B and the matrix representation of the trinucleotide. Elementary calculus shows that, up to a multiplicative constant, B is the matrix which maximizes the average of the difference B (codons) $- B$ (non-coding trinucleotides). We define the codon prototype measure to be the sum, over the window, of the dot product of B and the test-codons of the window.

Position Asymmetry Measure: Define $\mu(b) = \Sigma_i[f(b, i)]/3$ and $\text{assym}(b) = \Sigma_i[f(b, i) - \mu(b)]^2$. Then define the position asymmetry measure to be [assym(A), asymm(C), asymm(G), asymm(T)].

Entropy Measure: Given $f(b, i)$ as above, define $\text{entropy}(i) = \Sigma_b\{f(b, i)\ln[f(b, i)]\}$. If the three values of entropy (i) are significantly different a coding region is predicted, and the one with the largest difference from random is predicted to be third codon position. We define the Entropy Measure to be [entropy(1), entropy(2), entropy(3)].

Autocorrelation Measure: Let $\text{auto}(b, i)$ be the number of pairs of base b with i intervening bases. For the measure we correct for the number of such pairs expected on the basis of base composition alone, giving the matrix $[\text{auto}(b, i)/(\text{window\_length} - i - 1)(\text{frequency\_of\_b})^2]$, where $b =$ A, C, G, T and $i = 0, \ldots 9$.

Fourier Measure: Let the window be $2M$ long. Let $EQ(x, y)$ be the function which is 1 if $x = y$ and 0 otherwise. Define the nth Fourier coefficient (dropping the constant $1/4M^2$ for simplicity) by: $FC(n) = \Sigma_p\{\Sigma_m[EQ(\text{base } m, \text{ base } m - p)]\}\, e^{\pi inp/M}$. Then define the Fourier Measure to be $[FC(2M/2), FC(2M/3), \ldots, FC(2M/9]$ (i.e. the Fourier coefficients of the autocorrelation function for periods 2–9).

Period 9 Measure: Define $f(j) =$ frequency of R(j-other-bases)RYR and Period 9 Measure as the vector of values $[f(5), f(8), f(11)]$.

Dinucleotide Frame Measure: Make three frequency distributions of dinucleotides in the window: test-codon positions 1 & 2, positions 2 & 3 and positions 3 & 1. The indicator will be the three chi-squared values measuring bias of these distributions from the overall dinucleotide distribution of the training set (coding and noncoding).

Word Measure: Divide the window into successive, non-overlapping words of length 2, and also into words of length 3. The measure is the pair of chi-squared values comparing the frequency distributions of these words with the uniform distribution.

Run Measure: Lets $S_1, S_2, \ldots S_{14}$ be the non-trivial subsets of the set {A, C, G, T}. For each S_i construct a new sequence by replacing each base in S_i with 1 and replacing each base not in S_i with 0. Using this sequence define r_{ij} to be the number of runs of 1 of length j, for $j = 1, 2, 3, 4, 5$, and let r_{i6} be the number of runs of 1 of length greater than 5. The run measure will be the set of values $[r_{ij}]$.

Dinucleotide Bias Measure: Let $f(w)$, for any possible word w, be the frequency of w in the sample window. Now for each dinucleotide ab let $\text{bias}(ab) = [f(ab) - f(a)f(b)]/f(a)f(b)$. The Dinucleotide Bias Measure will be the bias values for the 16 dinucleotides.

Repeat Measure: Take all hexamers which occur, on average, more than twice every 4096 bases to be in the "repetitive" set. Using only the counts of these hexamers (324 in human, 247 in *E. coli*), in the coding and non-coding reference sets, gives the Repeat Measure.

In brief, the benchmark used is defined as follows. Homogeneous (fully coding or fully non-coding) windows of fixed size were taken from the international nucleotide sequence collection. The data corpus was

split in half, and the first part was used as a training set. Discriminant analysis (in two forms: classical linear discriminant analysis, which requires inversion of the covariance matrix, and Penrose discriminant analysis, which does not) was used to define a linear function of the measure which discriminates coding from non-coding. A threshold was then set to equalize the error rates on the coding and non-coding training sets. Then the performance of the algorithm so defined was evaluated on the other half of the data as test set. The average accuracy on the coding and non-coding parts of the test set was taken as the overall accuracy of the measure. The whole process was carried out both for a region-specific definition of coding and for a phase-specific definition.

There is a great deal of redundancy in the suite of measures proposed to date. In some cases two measures are sensing very similar things (e.g autocorrelation and Fourier). In many cases one measure is derivable from, or a specialization of, another (e.g. compositions can be derived from codon usage counts). Figure 1 shows which measures can be derived from others.

The tree in the right half of the figure contains most of the measures currently used. It is remarkable that, without exception, measures higher in this tree have higher accuracy than those below (and derived from) them. That is, in every case, if we derive an exon recognition function directly from a measure by using the Penrose discriminant, the result is higher accuracy than if we try to extract information from the measure in some clever way, and apply the Penrose discriminant procedure to the result.

Of the measures not in the main tree at the right of the figure, the period 9 measure and the word measure yield rather poor results, and the autocorrelation measure is essentially equivalent to the Fourier measure. The first main result of the review, then, is that of the measures tested, future algorithms should probably be based on Fourier, run, ORF and in-phase hexamer counts.

Combining several measures does improve accuracy. The highest score of any measure in the region-specific prediction of coding function on 108 base human windows was 76.6%. But Uberbacher kindly applied the Coding Recognition Module of GRAIL (Uberbacher & Mural, 1991) to the 108 base human test set (using only the first 100 bases of each window), and when a threshold was set to equalize sensitivity and specificity the resulting accuracy was 79%. For phase-specific discrimination we combined the six measures just discussed, again using classical linear discriminant analysis, and obtained 87.8% accuracy on human 108 base windows (compared to 84.9% for the most accurate individual measure). This last combination was also applied to human 54 base windows, giving 82.4% accuracy (compared to 80.7% accuracy for the highest individual measure).

The second main result is that a measure which seems to embody little biological understanding—counts of in-phase hexanucleotides—is in fact the most effective one. In-phase word count measures have a long history. The first use we know of the codon usage measure in a published algorithm is in Staden & McLachlan (1982). Separate word counts of different lengths for each phase were considered by Borodovsky *et al.* (1986a–c). These papers considered words of length 1, 2 and 3 (limited data were available at that time). More recently the same author (Borodovsky & McIninch, 1993) has extended his

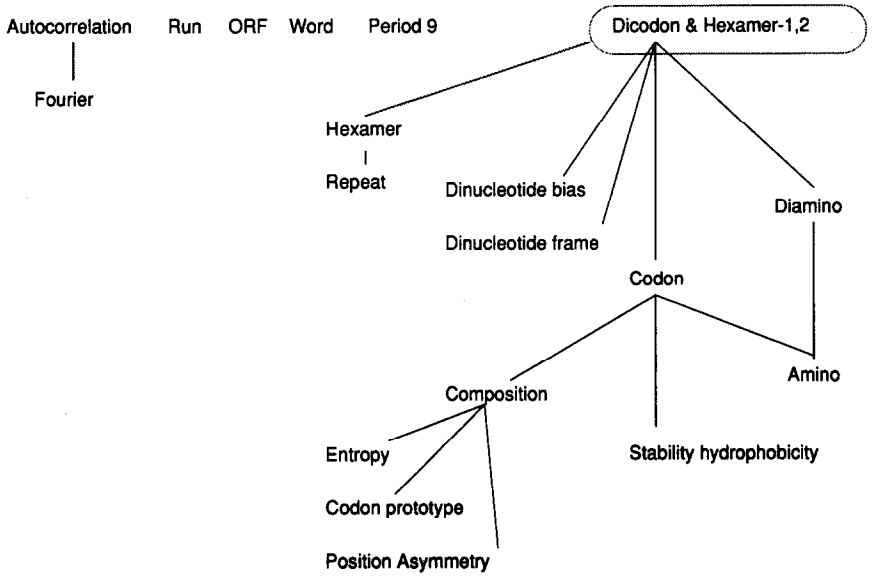

Fig. 1. Derivability of coding measures. Each measure is derivable from any measure above it and connected to it by a line. The dotted line shows that the Fourier measure is essentially equivalent to, though not formally derivable from, the autocorrelation measure.

work to include words of length 6. (Claverie *et al.*, 1990) was the first published use of in-phase hexamer count measures.

Since the time of the above survey, other measures have been proposed. Snyder & Stormo (1993) use the average complexity of octamers [measured by entropy in the sense of information theory; cf. Konopka & Owens (1990), which takes a somewhat different approach towards entropy than does Almagor (1985), reviewed in Fickett & Tung (1992)]. Solovyev & Lawrence (1993), extending the in-phase hexamer approach in a direction that takes on some characteristics of similarity search, report that in-phase octamers and nonamers give an even higher accuracy. (Ossadnik *et al.*, 1994) suggest a measure based on fluctuations in purine/pyrimidine window content (in a rather large window; >800 bp suggested by the authors). Often, when new coding measures are introduced, it is difficult to tell whether the measures are, in themselves, better or worse than existing ones, or whether, on the other hand, the context in which they are applied gives better performance. It would be interesting to apply the benchmark of Fickett & Tung (1992) to these new measures.

In a related vein, experimentalists often use the length of an open reading frame as primary evidence for the existence of a gene, particularly in organisms like yeast, where splicing is rare. In Fickett (1994, 1995) means are introduced for quantitative evaluation of the strength of such evidence.

We will likely continue to see incremental improvements in coding measures. First, Guigó & Fickett (1995) have shown that dependence of most measures on C + G content is high, and that mere base compositional differences can cause larger fluctuations in the values of coding measures than the differences between coding and non-coding regions. So tailoring the measures to differing base compositions may well improve accuracy. In this regard Xu *et al.* (1994) have adopted the strategy (not separately evaluated) of measuring hexamer counts for "high" and "low" CG content reference sets, and then using linear interpolation to make a set of counts intended to be appropriate for the CG content of the test sequence.

Second, it will probably be useful to systematically distinguish between several classes of sequence, rather than just "coding" and "non-coding". Konopka has long proposed a general framework of "functionally equivalent" classes of sequences [for a concise introduction see Konopka (1992)], and early showed that introns, in addition to lacking typical features of exons, also have features of their own, for example a tendency to show a two-base periodicity in the occurrence of certain oligonucleotides (Konopka & Smythers, 1987; Konopka *et al.*, 1987;). Guigó & Fickett (1995) show that intergenic DNA has very different statistical properties than gene flanking sequences. Krogh *et al.* (1994a) found it profitable to explicitly model intergenic DNA in *E. coli* (see below).

Finally, one wonders whether the many variables of some of the above coding measures (for example the 4096 variables of each hexamer measure) are all making important and independent contributions to discrimination. It might be, for example, that the signal-to-noise ratio of the measure could be improved by pruning out the less informative variables.

The means by which the information in a coding measure is reduced to a single score, or a yes/no answer, has varied greatly. In the case of in-phase hexamers, for example, Claverie *et al.* (1990) weight the observed count of each hexamer by the ratio of its frequency in coding regions to that in all DNA. Farber *et al.* (1992) use a neural net with 4096 inputs to derive a discriminant. Borodovsky & McIninch (1993) derive two non-homogeneous (frame-dependent) 5-step Markov models, one for the coding regions of each strand, and a homogeneous model for non-coding regions, calculate the probability of observing a window under each of the seven corresponding hypotheses, and then use Bayes' theorem to derive the posterior probability of each hypothesis given the window. (It is worth noting that in most algorithms, the method is applied separately to the two strands, and the results combined in a post-processing step. In the work of Borodovsky and McIninch, on the other hand, the seven relevant hypothesis—coding in each of six possible frames, or non-coding—are directly compared in one step.) Thomas & Skolnick (1994) consider seven classes of nucleotides: those in the three codon positions, those in intergenic regions, and those that are in introns breaking the coding sequence at each of the three possible codon positions. Assuming a one step Markov model for the state variable, and that the probability distribution of the bases at each position of the sequence depends only on the bases and states in the immediate vicinity, they use Bayes' theorem to make a maximum likelihood estimate of the state at each base of a given sequence. There is very limited information on which of these methods (or the many others that have been used with these measures, other measures or combinations of measures) is best. The general feeling among developers is that the differences are usually small, but comparative objective testing would be very valuable.

Signals: introduction

The coding measures considered above are all closely related to patterns of codon usage. In what has now become common usage, Staden (1990) termed the use of such measures "gene search by content". Of course codon usage is merely a side effect of the biochemistry of organisms. It will be more enlightening when we are able to recognize the locations in a genome where the gene expression machinery interacts with the nucleic acid, and so recognize the genes in a way parallel to the action of the cell. This approach Staden termed "gene search by signal".

Any portion of the DNA whose binding by another biochemical plays a key role in transcription is variously called a signal, a binding site, or a sequence element. Regions on a genome that correspond directly to regions on an mRNA or pre-mRNA with analogous function in splicing or translation are also referred to by the same terms.

The collection of all specific instances of some particular kind of signal, for instance, the set of all intron donor sites in human genes, will normally tend to be recognizably similar. In the early days of sequence analysis it was hoped that this similarity could be captured adequately by a *consensus sequence*. That is, one aligns all the specific sequences, and then takes the most commonly occurring base at each aligned position to form the consensus. Then, it was hoped, the actual sites would be differentiated from spurious sites simply by distance (e.g. number of bases different) from the consensus. This approach turned out to be too simple, though the consensus sequences at various sites are still useful for their mnemonic value.

It is now most common to summarize the commonalities in (that is, form a template for) a particular signal by recording the frequencies of each nucleotide at each aligned position, rather than simply recording the most frequent one. That is, the individual sequences are aligned, and the frequency of each base b at position i is tabulated as $f(b, i)$. Then a *position weight matrix* m is derived from f, most often by $m(b, i) = \log[f(b, i)/p(b)]$, where $p(b)$ is the genomic frequency of base b [reviewed in Stormo (1990)]. Any sequence to be tested for signal function is represented analogously, with $s(b, i) = 1$ if the ith base of the sequence is b, and 0 otherwise. Then the test value of a sequence is the dot product of these two matrices, $\Sigma_{b,i}[m(b, i) \times s(b, i)]$. (Because of the form of representation of the information, this approach is sometimes called, among computational biologists, the "matrix method".)

This approach is justified by several theoretical studies of protein–DNA binding [e.g. Berg & von Hippel (1988); von Hippel (1994) and references therein], and a number of experiments in which a DNA signal sequence is systematically varied and the activity of the variants measured [e.g. Mulligan *et al.* (1984), Takeda *et al.* (1989) and Barrick *et al.*, (1994)].

Overall, we may summarize the results of these studies as follows. The activity of a signal sequence is determined by the proportion of the time that the sequence is bound, which in turn depends on the abundance of the binding molecule (typically protein or RNA) and its binding specificity, that is, the degree to which the binding molecule "prefers" the signal sequence to pseudosites. In comparing the activity of different signal sequences for the same binding molecule, or in attempting to distinguish the signal sequences from pseudosites, we may take as constant all factors affecting the availability of the binding molecule (overall abundance, the frequency of pseudosites, and the average affinity of the pseudosites), and the deal simply with the binding energy of the binding molecule to the site at hand. The first major result from experiment is that this binding energy is often closely approximated by simply summing the contributions of the individual base positions, as if they were independent. This of course means that activity can be predicted reasonably well by some matrix calculation as described above, though it does not determine the form of the matrix.

If we assume that the $f(b, i)/p(b)$, as defined above, is representative of the ratio of bound to free reaction concentrations for base b in its interaction with a specific site on the binding molecule, then the logarithms in the position weight matrix are proportional to the free energies of binding for each base. This is one way of justifying the particular form of the position weight matrix. Alternatively, one may note that the sum in the dot product above is, from a statistical point of view, just the log likelihood ratio of the test sequence being found given: (1) the hypothesis that the sequence comes from a set in which the bases at position i have probability distribution $f(b, i)$, and (2) the hypothesis that the sequence comes from a set in which the bases occur with frequencies $p(b)$.

In many cases, the dot product of the position weight matrix with the sequence seems to be a relatively good predictor of signal activity. In Barrick *et al.* (1994), for example, 185 clones with randomized ribosome binding sites were selected, and for each the activity was measured and the binding site sequenced. A matrix was first determined by multiple linear regression. The regression matrix predicted actual activity with a correlation coefficient of 0.89 (when cases with alternate start codons were eliminated, this rose to 0.92). Further, when a position weight matrix was calculated from natural sites, the correlation coefficient between the two matrices was 0.88.

However, position weight matrices do not always work well, and it must be recognized that a number of simplifying assumptions underlie their use. The use of position weight matrices ignores the availability of the DNA or RNA (the effects of chromosome packaging and secondary structure), non-independence between bases (important, for example, in conformational changes due to base stacking), different versions or conformations of the binding molecule and interactions between multiple binding molecules.

Non-independence between bases may be taken into account by a relatively simple extension of the position weight matrix, namely using a larger matrix where columns correspond to the various possible oligomers at various positions, rather than to individual bases. One example of this approach may be seen in the work of Thomas and Skolnick already cited (the uniformity of their approach makes "retraining" the algorithm very easy). Another will be seen below in the work of Solovyev, Salamov and Lawrence. Of

course, the longer the oligomers, the more data are needed to reliably calculate the matrix.

The use of position weight matrices in recognizing key elements of eukaryotic genes, namely splice sites and promoter sequences, has to date led to relatively limited success. All of the above limitations of the method probably play a role here. However we would hazard the guess that the main factor is the cooperativity among multiple binding molecules. It is rare in eukaryotes, for example, for large numbers of genes to have precisely the same complement of proteins involved in the initiation of transcription. We will return to this point below.

Where applicable, the consensus and position weight matrix methods have the advantage of being relatively simple and well understood. Assessing the significance of search results has been treated in Waterman (1989a, b) for approximate matches to a consensus pattern, and in Claverie (1994a) and Goldstein & Waterman (1994) for searches using position weight matrices.

A wide variety of other methods, difficult to summarize in a limited space, have been proposed to recognize signal sequences in genomes. Most of these have not come into wide use, and the reader must be referred to Gelfand (1995) and the on-line bibliographies mentioned above for more details. One method which has seen fairly extensive use is that of neural networks. When the network has only one layer, it produces a linear discriminant function that is usually fairly close to the position weight matrix derived by the methods described above. However, when the network has multiple layers, with hidden units, the function encoded is more complex. The use of neural networks in the analysis of nucleotide (and amino acid) sequences was reviewed in Hirst & Sternberg (1992). The neural network algorithms reviewed showed better performance than more statistical approaches in a number of cases. However, it is not altogether clear whether the improvement was due to integration of several kinds of evidence (discussed below) or to the neural network means of integration.

One difficulty with neural nets, and in fact with machine learning methods in general, is the distance between the understanding in the machine and the understanding in the human expert. Most such algorithms are designed to begin from a randomized state, that is, without the benefit of any knowledge already gained by experiment or other methods. And, when the algorithm has finished the training state, it is typically rather difficult to retrieve the "understanding" that has been captured. In this regard, Shavlik *et al.* (1992) have made interesting progress by developing neural net methods that can start from an intelligible base of rules and, after training, can return a refined set of rules.

Many methods of sequence signal recognition require a set of sequences with functional sequence elements already precisely located and aligned. However, it is often the case that experimental work has only approximately located the sequence element, and that the best alignment is unclear. Thus several groups have developed methods to optimize the localization of the sequence elements, the alignment, and a weight matrix or other discriminant, simultaneously; see for example Cardon & Stormo (1992), Lawrence *et al.* (1993), Borodovsky & Peresetsky (1994), Krogh *et al.* (1994). These methods have to date been applied primarily to other problems, but show significant promise for the identification of eukaryotic signal sequences.

Signals: basal gene biochemistry

Gene signal recognition work to date has dealt with the problem of recognizing the signals common to essentially all genes. For example Bucher (1990) has defined weight matrices to partially characterize four elements common to most eukaryotic pol II promoters: the TATA-box, cap-signal, CCAAT- and GC-box. These were derived from the Eukaryotic Promoter Database (Bucher, 1988). In Cavener & Ray (1991) sequences flanking translational initiation and termination sites have been compiled and statistically analyzed for various eukaryotic taxonomic groups. The polyadenylation reaction is relatively well understood now (Wahle & Keller, 1992), and information on translation termination sites has been collected in the Translational Termination Signal Database (Brown *et al.*, 1993). Yada *et al.* (1994) use discriminant analysis to derive a position weight matrix to recognize the polyadenylation signal. All of this information is useful in helping to recognize the beginnings and ends of genes, however computational methods for such recognition are in their infancy, and will require significant further development to attain high reliability.

Consensus sequences for splice junctions have been recognized for many years (Breathnach & Chambon, 1981). A comprehensive collection of splice junctions and weight matrices, commonly referred to, may be found in Senapathy *et al.* (1990). Consensus sequences alone give rather unsatisfactory results. The best successes to date in predicting splice junctions come from integrating several kinds of evidence. Shapiro & Senapathy (1987) combine base frequency information at the splice site with a check for an open reading frame on the correct side, and an evaluation of a potential polypyrimidine tract near the acceptor. Including a requirement for related patterns [e.g. a branch point within a specified distance upstream of the acceptor, and no AG dinucleotide between these two sites (Oshima & Gotoh, 1988; Gelfand, 1989)] seems to improve accuracy. At true splice sites, coding measures should give values characteristic of coding regions on one side of the splice, and values characteristic of non-coding regions on the other. Thus in Nakata *et al.* (1985) and Brunak *et al.* (1991) information concerning splice sites *per se*, for example positional frequencies and binding energies, are combined with the values of coding measures on

either side of each potential splice site, to give improved splice site prediction. Solovyev *et al.* (1994b) give an excellent overview of the literature and a careful synthesis of existing techniques. They report what appears to be the most accurate algorithm for human sequences to date, using triplet counts [due to Mural *et al.* (1990)] at significant positions near the branch point and splice junctions, octamer counts on either side of the junction, counts of G, GG and GGG downstream of potential donor sites, and counts of T and C upstream of potential acceptor sites, all combined using linear discriminant analysis. Taking the sets of GT and AG dinucleotides as the set of all potential splice sites, Solovyev *et al.* report 96% sensitivity and 97% specificity for donors, and 96% sensitivity and 96% specificity for acceptors. (These methods are combined, using linear discriminant analysis, with oligonucleotide-based recognition methods for coding regions and the beginnings and ends of genes to produce an exon recognition algorithm FEX.)

In as many as 90% of the vertebrate mRNAs, the first AUG codon is the unique initiation site, and in the exceptional cases a number of factors have been elucidated that govern the probability of translation initiation at a particular ATG. These include neighboring nucleotides, leader length, distance to other ATGs, ORF length and secondary structure (Kozak, 1991).

Signals: regulation of gene expression

The complexity of gene regulation naturally increases greatly with the number of tissue and cell types in an organism. Thus, although some universal commonalities have been identified in the known genes of some prokaryotes, it would now appear unlikely that any simple characterization will be found for the gene promoters of *Homo sapiens* (or, probably, of any other differentiated metazoan species). Thus, although the regulation of eukaryotic gene expression has attracted relatively little attention to date from developers of gene identification algorithms, such algorithms will, in the future, almost certainly take into account the complex signals for transcription initiation of specific classes of genes.

Utilizing this sort of information will bring an added advantage, in that specific transcription elements provide important clues to gene function. This is an opportune time to begin making use of information on gene regulation, for a remarkable amount of information is now appearing, with new papers daily, on gene-specific, tissue-specific, stage-specific and stimulus-specific transcription signals.

Several collections of sequence elements for transcription factors have appeared, including the Transcription Factor Database (Ghosh, 1990), the collections in Locker & Buzard (1990) and Faisst & Meyer (1992), TRANSFAC (Knueppel *et al.*, 1994; Wingender 1994), TFDB (Mizushima & Hayashi 1994) and TRRD (Kel *et al.*, 1995). The first three of these are no longer maintained. These collections, in addition to incorporating the sequences of individual signal instances, sometimes include consensus sequences or weight matrices.

It is not clear at this point to what extent consensus sequences or weight matrices can differentiate true from false transcription elements. This remains a research area, as does the problem of how best to use the transcription element information in gene identification algorithms. One promising approach is reported in Prestridge (1995): in the calibration step, consensus sequences are used to recognize putative transcription factor binding sites in a training set of promoter and non-promoter regions, and ratios of densities for putative binding sites in promoters and non-promoters are recorded for all transcription factors in TFD. In application, the density ratios of putative transcription factor binding sites (again recognized by means of consensus sequences) are summed, and this score is combined with the Bucher weight matrix score of any putative TATA box. When the score threshold is set so that 70% of promoters are recognized correctly, one false positive is recorded about once every 5600 bases. [An earlier paper (Prestridge & Burks, 1993), found that the simple density of putative transcription elements is not discriminatory.]

Gene syntax and integration of information

It is well known that gene expression *in vivo* involves considerable interaction and interdependence among various components of the transcription and translation machinery. Examples include coordinate binding of multiple transcription factors and mutations in a 5′ splice site resulting in the skipping of an upstream 3′ site. Thus it is not surprising that programs incorporating some overall model of gene structure give increased accuracy even for the recognition of individual gene components. In the case of intron splice sites, the integrated methods discussed above give roughly a factor of 10 improvement over recognition by consensus or matrix methods. Another example is seen in Einstein *et al.* (1992), where it is shown that 60% of exons under 50 bp missed by the original GRAIL e-mail server may be detected by a logical analysis of splicing and frame.

A number of programs have appeared in the last few years that are integrated in the sense of taking gene structure into account to predict exons [SORFIND (Hutchinson & Hayden, 1992, 1993); FEX (Solovyev *et al.*, 1994a, b)] or genes [GM (Fields & Soderlund, 1990; Soderlund *et al.*, 1992); the Gelfand program (Gelfand, 1990a; Gelfand & Roytberg, 1993); GeneID (Guigo *et al.*, 1992; Knudsen *et al.*, 1993); GenViewer (Milanesi *et al.*, 1993); GeneParser (Snyder & Stormo, 1993); GRAIL II (Uberbacher *et al.*, 1993; Xu *et al.*, 1994); GenLang (Dong & Searls 1994) [cf. also (Searls, 1992)]; and the program of Krogh *et al.* (1994)].

(There are other gene prediction algorithms not yet published. In one prominent case, the analysis of the *C. elegans* genomic sequencing group [cf. Wilson *et al.* (1994)] makes use of an algorithm GeneFinder developed by P. Green.)

The goal. In writing any new algorithm, the single most important decision is often, of course, the choice of a precise goal. Today, the goal of a gene identification algorithm is usually taken as obvious. Though there are minor differences, mostly associated with whether or not only optimal solutions are shown, developers take the essential goal to be the assembly of all components of a gene and the reporting of an integral gene to the user.

In the long run it will be important to extend this goal to meet the practical needs of more complex situations. Current algorithms typically expect to find all components of each gene and, sometimes, of only one gene. In practice, however, a sequence presented for analysis may have no genes, partial genes (particularly in the case of very large genes, such as human dystrophin, which is over 2 Mb long), multiple genes, genes embedded in the introns of other genes [cf. Levinson *et al.* (1990)], or genes with multiple expression patterns. Unusual mechanisms such as genome rearrangements (as in the immunoglobulins), trans-splitting and RNA editing (as in some organellar genes) and the use of unusual tRNA species, are rarely dealt with. Thus it will be necessary to develop algorithms that can produce a feature table of relevant gene features in whatever combinations they happen to occur.

In addition, it is now widely recognized that an important part of the goal must be to recognize when a small change in sequence will result in a large change in function. This is important for recognizing non-functional alleles of "disease genes", pseudogenes, and genes in first pass sequence data [cf. Claverie (1993b), Krogh *et al.* (1994) and Fields (1994)].

Kinds of integration. Gene identification algorithms typically begin by attempting to evaluate possible component objects or aspects of genes, proceed to integrate these into exons, and finally integrate the exons into genes. At both the exon level and the gene level there are two very different kinds of integration involved. The first is primarily biological, taking into account the syntax of genes, for example typical spacing of components and the partitioning of the primary transcript into alternate exons and introns. The second is primarily logical and statistical, taking into account the relative importance of different kinds of evidence, and the combining of scores into overall measures of optimality in gene models. We will take these up in turn.

Syntactical integration. All integrated gene identification programs make use of the high level syntax of genes resulting from our basic understanding of transcription, splicing, and translation. Taking "exon" in the coding sense, rules similar to the following are normally used:

- The first coding exon begins with the start codon and ends with a donor site (or the stop codon, if there are no internal exons).
- Any internal exons begin with acceptor sites and end with donor sites.
- The last exon begins with an acceptor site (or the start codon) and ends with the stop codon.
- The primary transcript consists of the transcription initiation site, a 5′UTR, alternating exons and introns, the 3′UTR, and the transcription termination site.
- When the introns are excised and the combined exons read in frame, no internal stop codons are found.

In addition to this syntax of order, there is also some information on distance, as for example appears in known size distributions for exons and introns [cf. Naora & Deacon (1982), Hawkins (1988) and Smith (1988)].

Although this basic syntax is clear enough, biology is of course far more complex, and less well understood, than these simple rules would imply. Such facets of gene syntax as alternative splicing, overlapping genes and promoter structure remain beyond the reach of the current generation of algorithms.

In many of the algorithms available today, the rules of gene syntax are implicit in the structure of the algorithm, but no "gene grammar" is explicitly listed. Two groups have, however, taken a more linguistic approach, making an explicit grammar the foundation of the algorithm.

Searls suggested, some years ago, that a linguistic approach to the analysis of features in DNA sequences could be beneficial [for an overview, see Searls (1992)]. This approach is first applied to the identification of protein coding genes in Dong & Searls (1994), where a formal, definite clause grammar of genes is described. Partial scores are passed up the parse tree, and combined by rules stored as part of the grammar. A training procedure is used to alter the score combination rules in order to optimize accuracy. Standard parse tools are used to find correct and high scoring parses of a sequence.

Krogh *et al.* (1994) use a Hidden Markov Model (HMM) to integrate gene components into overall gene models for *E. coli* sequences. In essence, this means that they construct a probabilistic finite automation that assigns a probability to every possible parse of a sequence into promoter, start, coding, stop and intergenic regions. The Expectation Maximization algorithm is used to estimate the parameters of the HMM. Then the Viterbi algorithm is used to find the most probable parse of the sequence.

Logical integration. A variety of evidence is typically employed in computer searches for protein coding genes. One of the critical choices in algorithm

design is the choice of method for combining these different types of evidence.

Gelfand (1990a) was the first to explicitly discuss the question of providing a natural framework for the integration of coding measures, matrix scores of signals and overall syntactical requirements. The approach chosen was basically statistical. To avoid dependence of score on the length of the gene, raw scores are taken as the average donor score, the average acceptor score, and the average TESTCODE window score (Fickett, 1982) over the exons. Then all scores are put on the same scale by expressing them in standard deviation units about the means of their observed distributions. The sum of these normalized scores is the score for the gene.

Several other authors have also taken a fundamentally probabilistic/statistical approach. The discriminant analysis approach of Solovyev *et al.* (1994a, b) is of course statistical. Stormo & Haussler (1994) suggest a general probabilistic framework in the situation where one is partitioning a sequence into two classes of intervals (e.g. exons and introns), has a number of scores for each possible classification of each possible interval, and is combining these scores as a linear weighted sum. They suggest interpreting the scores and the sum as log probabilities. They then give efficient algorithms for scaling the scores so that the probabilities will sum to one, for calculating the probabilities, for choosing the weights in order to maximize the probability of given ("training set") sequence parses, and for finding the top ranked optimal and suboptimal parses. [Compare also States & Gish (1994), where codon bias is integrated into BLAST searches using a likelihood approach.]

A particular advantage of the HMM approach of Krogh *et al.* (1994) is that it naturally provides a joint probability distribution over sequences and parses of those sequences. The HMM thus provides a very natural vehicle for considering the possibility of introducing a sequence correction to get a more probable parse.

The salient advantage of taking a probabilistic point of view is that it may be possible to assign a natural meaning to the scores. It would seem to be very desirable to apply the probabilistic point of view consistently to the sequence interpretation problem, in a way that allowed one to provide answers for such questions as "how likely is it that at least one exon of this predicted gene is completely correct?", "how likely is it that the correct gene and this predicted gene have at least 90% of the translated protein in common?" or, "how likely is it that this is in fact the most commonly used translation initiation site?".

Applying probabilistic notions consistently is, however, very difficult because of our limited knowledge. Most authors, therefore, have taken what might be termed a machine learning approach, in which scores of various aspects of putative genes are meaningless numbers, and the rules for combining these numbers may therefore be manipulated at will to improve the accuracy of prediction. The advantage of this point of view, successfully exploited by a number of investigators, is that purely empirical machine learning techniques may be used to improve the algorithms by which scores are combined. Thus for example both Guigó *et al.* (1992) and Snyder & Stormo (1993) use a neural net to revise the weights by which different atomic measures are combined, Dong & Searls (1994) use an *ad hoc* training procedure to revise the score-combining rules associated with each node of the parse tree, and Salzberg (1995) uses a decision tree algorithm to combine information from several coding measures. In these cases it is reported that machine learning algorithms combine information in a way that significantly improves performance.

Orthogonal to the choice of a probabilistic or a machine learning approach to the interpretation of scores, there is also the issue of organizing one's evidence. Most gene identification algorithms recursively construct gene models from partial subassemblies. For instance, atomic components may be scored first, then exons constructed and scored, and finally genes assembled from exons and a final score assigned. Further, most evidence gathered by gene identification algorithms fits neatly into this recursive hierarchy. Thus Dong & Searls (1994) elegantly summarize the basic approach of most investigators by attaching the scoring rules directly to nodes of the gene parse tree.

Unfortunately, however, not all of the evidence that one needs to take into account is directly related to a subassembly of the gene. For example, if the translated protein from a candidate gene contains a region similar to a known protein motif, and this region corresponds to parts of each of two exons, it is not obvious how this should affect either the scores of the exons or the score of the gene overall. Dong and Searls solve this problem by specifying a grammar in which not all components of a parse are components of the gene; for example, one parse component is the average exon quality. Another common approach is to append postprocessor rules to the main algorithm. Thus GRAIL (Uberbacher *et al.*, 1993; Xu *et al.*, 1994) incorporates a number of heuristic rules for finding the boundaries of exons, and Krogh *et al.* (1994) complete independent analyses of the complementary DNA strands, and then combine them by means of a small set of rules.

Efficient computation. The number of possible genes to construct, score, and rank, even in a sequence of a few kilobases, is quite large. Snyder & Stormo (1993) and, independently, Gelfand & Roytberg (1993), introduce dynamic programming algorithms to efficiently find optimally scoring solutions. Guigó *et al.* (1992) introduce the idea of exon equivalence—using one exon to represent a class of roughly equivalent exons—as an alternative (and possibly coordinate approach).

Despite significant advances in sequencing technology, it still takes longer to produce a sequence than it does to submit it to the analysis of even the slowest gene identification algorithms. What may be an even more serious bottleneck is the human attention required to interpret and integrate the output from the several kinds of important computational analyses. Thus in addition to efficient computation, significant attention should be devoted to the problem of building algorithms to truly integrate all the evidence for gene location and function, and to give accurate answers to biologically meaningful questions.

Summary. As will be clear from even this short overview, the area of whole-gene recognition is moving rapidly, with advances being made on several fronts. Divergent, and sometimes even conflicting, innovations are being made by different groups. Particular techniques are rarely evaluated in isolation, and each pair of programs usually differs in many aspects. Thus there is no one best program, nor is there likely to be one soon.

Accuracies of the above programs are somewhat difficult to compare, as benchmark sets and testing methodology are not yet standardized. Roughly speaking, the accuracy of most of the above programs is reported so: when a new (not seen before by the algorithm) sequence is chosen that contains all of one gene and its flanking regions (and no other genes or partial genes), and this sequence is presented to the algorithm, the predicted gene will typically largely overlap the known gene, in such a way that about 85–90% of the predicted coding bases are in the known gene, and about 85–90% of the known coding bases will be in the predicted gene. That is, the predicted gene will look very much like the known one, but there will usually be significant differences as well.

There are, however, hints that this performance may not extend to genes typical of the genome (rather than of the database). For example Lopez *et al.* (1994) report that when GRAIL is used on long, recently determined sequences, the accuracy is significantly lower than on the original test set. It is quite possible that similar results will be found for other tools.

THE DEVELOPMENT PROCESS

This paper is concerned primarily with algorithm design. However it is important to mention briefly some closely related issues.

Data

It is beyond the scope of the nucleotide sequence databases to maintain a reflection of current biological understanding in the features recorded on all known sequences. Thus the algorithm developer must be aware that annotation in the databases is often incomplete and sometimes incorrect.

One solution to this difficulty is to take a set of a few tens of sequences, verify the annotation in detail for this set, and then use it for algorithm development and evaluation. The advantage of this approach is, of course, that one can be personally assured of the quality of the data. A disadvantage is that the variety in such a set is rather limited, and algorithms developed in this way may not generalize well to new data.

Another solution is to accept the databases as they are, perhaps removing some large classes of entries likely to confuse one's study (for example, entries with no annotation, STS sequences or duplicates) and take the incompleteness of annotation into account in interpreting results.

A compromise between these two approaches is to take advantage of one of a number of specialized, curated databases of intermediate size. One such, of particular relevance to the development of gene identification algorithms, is the collection of Functionally Equivalent Sequence sets (including, for example, a number of specialized collections of exons and introns) described in Konopka (1994a).

Evaluation

When only a few techniques had been developed for gene identification, it was often sufficient to demonstrate the value of a new technique in a few special cases. However, extensive benchmarking is now widely appreciated, and an innovative technique that is objectively shown to be of value in a large number of cases also stands a better chance to be widely adopted. It is also increasingly important to know the performance of new techniques not only on the "mainstream" genes common in the public databases, but on genes with unusual base composition, on rarely expressed genes, and on single pass, error-prone sequences.

The evaluation of integrated algorithms is complex because there is no one best interpretation of the question, "how correct is this prediction?". Guigó *et al.* (1992) made an important advance by suggesting that accuracy of integrated algorithms be evaluated on a nucleotide basis. They report the counts of three classes of nucleotides: those in the known coding region and the predicted coding region; those in the known, but not the predicted coding region; and those in the predicted, but not the known coding region. These numbers are combined in the set-theoretic correlation coefficient (Cramer, 1946; Matthews, 1975) between the set of true coding nucleotides and the set of predicted coding nucleotides. Since the correlation coefficient depends not only on the algorithm, but also the data set, developers should always give the raw numbers as well as the summary coefficient. Evaluation is also difficult because there is as yet no consensus on the form of the algorithm output, and different forms (e.g. a set of coding regions, a set of exons, a single most likely gene, a ranked list of possible genes, etc.) are not completely comparable.

Performance of algorithms is, of course, in part dependent on the quality and contiguity of the

sequences presented. Claverie (1994c) evaluates the performance of GRAIL when raw, single sequencing runs are analyzed, and suggests that it is unlikely for the use of first pass, fragmented data, in itself, to lead to failed detection of genes. Kamb *et al.* (1955) evaluate XPOUND (Thomas & Skolnick, 1994) on 400 bp sequences containing coding segments of various known sizes and positions. They conclude that in performing 100 sequencing reactions on randomly selected fragments of a P1 clone, followed by XPOUND analysis, between 50 and 75% (depending on the types of sequencing errors) of all genes present in the P1 would be detected.

Benchmarking is also a significant issue for users, who need to know not only how good the algorithm is, but how to interpret a particular score. In the case of SORFIND, which predicts internal exons, Hutchinson & Hayden (1992, 1993) divide the range of the output score into four ranges, and for each report the actual frequency with which the algorithm correctly reports exons in that score range. The situation becomes more complicated when the output consists of genes (or feature tables) rather than exons. By considering many suboptimal solutions, Snyder & Stormo (1992) attempt to give the user a feel for which parts of a predicted gene are most likely to be correct.

Singh & Krawetz (1994) compare the performance of four coding measures and the GRAIL e-mail server on one *E. coli* and four human genes. This sort of objective, third-party, comparative performance measurement is very valuable and unfortunately rare. It is to be hoped that further, and more comprehensive, studies will appear.

Communication of results

Since a large number of reasonably good techniques are already in existence, every developer must be aware that in order for an important innovation to spread, it needs to be described clearly, in enough detail that other investigators can easily duplicate the work. This has, of course, become more difficult as algorithms have grown more complicated. Yet the developer who is able to completely specify the algorithm in print will find others much more willing to adopt proposed techniques.

Interface

It is a remarkable fact about the field of gene identification today that many, perhaps most, of the best algorithms are not widely available. This is first of all simply because many developers have not had the time to develop an intuitive interface for those whose primary business is experimental biology. Indeed, one of the most important factors in the widespread use of GM and GRAIL is the effort that its developers have put into interface development and community education.

A second limitation on availability is less obvious but no less real. This is that most algorithms today are organism specific, in implementation even if not in concept. To overcome this problem research on the degree of generality of various techniques is needed. For example, are in-phase hexamer counts, the single most useful coding measure, fairly stable only within species? Or can discriminant vectors for this measure be meaningfully calibrated for all mammals, or even for some wider group, in one step? If most techniques are highly specific to relatively small parts of the taxonomic tree (similar remarks apply to classes of genes), then a way needs to be found to allow the typical computational support person in larger biological laboratories to tailor existing algorithms to a particular context.

SUMMARY

There has been a great deal of progress in gene identification methods in the last few years. At least in the case of sequence data from mammals, *C. elegans* and *E. coli*, the older coding region identification methods have given way to methods that can suggest the overall structure of genes. And for all organisms, computational methods are sufficiently accurate that they give practical help in many projects of biological and medical import.

Yet there is still room for significant improvement. Many of the better algorithms are not widely available. Investigators studying organisms other than those mentioned above may find that only the older algorithms are available to them. For the more advanced algorithms, it is still the case that predicted genes, while largely overlapping expressed natural genes, are typically incorrect in a number of details. Further, it is not clear that current algorithms, developed on the very atypical gene sample available in current databases, will perform as well on genes more typical of the biological universe as a whole. Essentially all current algorithms depend heavily on codon usage bias, but it has been shown that this bias is less informative in genes with low-level expression (McLachlan *et al.*, 1984; Sharp *et al.*, 1988; States & Gish, 1994).

Perhaps the single greatest opportunity in the development of gene identification algorithms is to include more detailed biological knowledge, relying less on techniques that attempt to provide a single elegant description valid for all cases. The description of (say) human genes inherent in any of the current gene recognition programs could be written down in a few pages. Given the extent to which evolution is opportunistic and haphazard, and given the prevalence of exceptions to essentially all general principles in molecular biology and biochemistry, it seems most unlikely that essential aspects of any genome will be described in such simple terms. Greater emphasis should probably be placed, then, on lookup methods over template methods; more richness is needed in the modeling of eukaryotic gene regulation; and, in general, a trend may be expected

towards gene identification algorithms becoming interfaces, with a general model of gene syntax, to a large number of databases of specific facts. First steps in this direction may be found in Borodovsky *et al.* (1994a, b), Claverie (1994b) and States & Gish (1994).

The single most important area where specific aspects of genes are important, even to discover the coding regions, is control of gene expression. Further, control of gene expression is very closely connected to product function. Thus, in addition to providing greater accuracy, bringing gene identification algorithms close to models of underlying biological mechanisms will also bring them closer to answering what is, in the end, the more important questions: not just "Where are the genes in this sequence?", but "How do they determine the biochemistry of the cell?".

Acknowledgements—This work was supported by NCHGR, and was carried out under the auspices of DOE. Parts of this work were carried out at the Aspen Center for Physics and the Telluride Academy. I would like to thank the many people who made suggestions for improving the paper, particularly M. Borodovsky, P. Bucher, J.-M. Claverie, M. Gelfand, R. Guigó, A. Konopka, S. Salzberg, D. Searls and V. Solovyev.

REFERENCES

Adams M. D., Kelley J. M., Gocayne J. D., Dubnick M., Polymeropoulos M. H., Xiao H., Merril C. R., Wu A., Olde B., Moreno R. F., Kerlavage A. R., McCombie W. R. & Venter J. C. (1991) *Science* **252,** 1651–1656.

Adams M. D., Fields C. & Venter J. C. (Eds) (1994) *Automated DNA Sequencing and Analysis.* Academic Press, San Diego, CA.

Almagor H. (1985) *J. Theor. Biol.* **117,** 127.

Altman R., Brutlag D., Karp P., Lathrop R. & Searls D. (Eds) (1994) *Proc. Second International Conference on Intelligent Systems for Molecular Biology.* AAAI Press, Menlo Park, CA.

Altschul S. F., Gish W., Miller W., Myers E. W. & Lipman D. J. (1990) *J. Mol. Biol.* **215,** 403.

Altschul S. F., Boguski M. S., Gish W. & Wootton J. C. (1994) *Nature Genet.* **6,** 119.

Bairoch A. (1992) *Nucl. Acids Res.* **20,** 2013.

Barrick D., Villanueba K., Childs J., Kalil R., Schneider T. D., Lawrence C. E., Gold L., & Stormo G. D. (1994) *Nucl. Acids Res.* **22,** 1287.

Berg O. G. & von Hippel P. H. (1988) *Trends Biochem. Sci.* **13,** 207.

Boguski M. S., Lowe T. M. J. & Tolstoshev C. M. (1993) *Nature Genet.* **4,** 332.

Boguski M. S., Tolstoshev C. M., Bassett D. E. Jr (1994) *Science* **265,** 1993 (see also http://www.ncbi.nlm.nih.gov/dbEST/dbEST_genes/ for supplementary data).

Bork P., Ouzounis C., Sander C., Scharf M., Schneider R. & Sonnhammer E. (1992a) *Nature* **358,** 287.

Bork P., Ouzounis C., Sander C., Scharf M., Schneider R. & Sonnhammer E. (1992b) *Prot. Sci.* **1,** 1677.

Borodovsky M. & McIninch J. (1993) *Proc. International Conference on Open Problems in Computational Biology* (Edited by A. Konopka). Published as *Computers Chem.*, **17,** 123.

Borodovsky M. & Peresetsky A. (1994) In Konopka (1994b), p. 259.

Borodovsky M. Y., Sprizhitskii Y. A., Golovanov E. I. & Aleksandrov A. A. (1986a) *Molek. Biol* **20,** 1014.

Borodovsky M. Y., Sprizhitskii Y. A., Golovanov E. I. & Aleksandrov A. A. (1986b) *Molek. Biol.* **20,** 1024.

Borodovsky M. Y., Sprizhitskii Y. A., Golovanov E. I. & Aleksandrov A. A. (1986c) *Molek. Biol.* **20,** 1390.

Borodovsky M., Koonin E. V. & Rudd K. E. (1994a) *Trends Biochem. Sci.* **19,** 309.

Borodovsky M., Rudd K. E. & Koonin E. V. (1994b) *Nucl. Acids Res.* **22,** 4756.

Breathnach R. & Chambon P. (1981) *Ann. Rev. Biochem.* **50,** 349.

Brown C. M., Dalphin M. E., Stockwell P. A. & Tate W. P. (1993) *Nucl. Acids Res.* **21,** 3119–3123.

Brunak S., Engelbrecht J. & Knudsen S. (1991) *J. Mol. Biol.* **220,** 49.

Bucher P. (1988) *EMBL Nucleotide Sequence Data Library* **17,** Postfach 10.2209, D-6900, Heidelberg.

Bucher P. (1990) *J. Mol. Biol.* **212,** 563.

Cardon L. R. & Stormo G. D. (1992) *J. Mol. Biol.* **223,** 159.

Cavener D. R. & Ray S. C. (1991) *Nucl Acids Res.* **19,** 3185.

Church D. M., Stotler C. J., Rutter J. L., Murrell J. R., Trofatter J. A. & Buckler A. J. (1994) *Nature Genetics* **6,** 98.

Claverie J.-M. (1992) *Genomics* **12,** 838.

Claverie J.-M. (1993a) *Nature* **364,** 19.

Claverie J.-M. (1993b) *J. Mol. Biol.* **234,** 1140.

Claverie J.-M. (1994a) In Konopka (1994b), p. 287.

Claverie J.-M. (1994b) In Adams *et al.* (1994), p. 267 Chap. 36.

Claverie J.-M. (1994c) *Genomics* **23,** 575.

Claverie J.-M., Sauvaget I. & Bougueleret L. (1990) In Doolittle (1990), p. 237.

Cramer H. (1946) *Mathematical Methods of Statistics.* Princeton University Press, Princeton.

Dong S. & Searls D. B. (1994) *Genomics* **23,** 540.

Doolittle R. F. (1986) *Of URFs and ORFs.* University Science Books, Mill Valley, CA.

Doolittle R. F. (Ed.) (1990) *Molecular Evolution: Computer Analysis of Protein and Nucleic Acid Sequences.* Vol. 183 (special issue) of *Methods in Enzymology.*

Einstein J. R., Mural R. J., Guan X. & Uberbacher E. C. (1992) Oak Ridge National Laboratory Report TM-12174.

Faisst S. & Meyer S. (1991) *Nucl. Acids Res.* **20,** 3.

Farber R. B., Lapedes A. S. & Sirotkin K. M. (1992) *J. Mol. Biol.* **226,** 471.

Fickett J. W. (1982) *Nucl. Acids Res.* **10,** 5303.

Fickett J. W. (1994) In Konopka (1994b) p. 203.

Fickett J. W. (1995) *J. Comp. Biol.* **2,** 117.

Fickett J. W. & Tung C.-S. (1992) *Nucl. Acids Res.* **20,** 6441.

Fickett J. W., Torney D. C. & Wolf D. R. (1992) *Genomics* **13,** 1056.

Fields C. A. (1994) In Adams, Fields & Venter (1994), p. 321.

Fields C. A. & Soderlund C. A. (1990) *Comp. Appl. Biosci.* **6,** 263.

Gelfand M. S. (1989) *Nucl. Acids Res.* **17,** 6369.

Gelfand M. S. (1990a) *Nucl. Acids Res.* **18,** 5865.

Gelfand M. S. (1990b) *Biotechnol. Software* **7,** 3.

Gelfand M. S. (1992) In Gindikin (1992), p. 87.

Gelfand M. S. (1995) *J. Comp. Biol.* **2,** 87.

Gelfand M. S. & Roytberg M. A. (1993) *Bio Syst.* **30,** 173.

Ghosh D. (1990) *Nucl. Acids Res.* **18,** 1749.

Gindikin S. (Ed) (1992) *Mathematical Methods of Analysis of Biopolymer Sequences (DIMACS Series in Discrete Mathematics and Theoretical Computer Science, V8).* American Mathematical Society, Providence, RI.

Gish W. & States D. J. (1993) *Nature Genet.* **3,** 266.

Goldstein L. & Waterman M. S. (1994) *J. Comp. Biol.* **1,** 93–104.

Green P. (1994) *Curr. Opin. Struct. Biol.* **4,** 404.

Green P., Lipman D., Hillier L., Waterston R., States D. & Claverie J.-M. (1993) *Science* **259,** 1711.
Gribskov M. & Devereux J. (1991) *Sequence Analysis Primer.* Stockton Press, New York.
Gribskov M., McLachan A. D. & Eisenberg D. (1987) *Proc. Nat. Acad. Sci. U.S.A.* **84,** 4355.
Griffin A. & Griffin H. G. (1994) *Computer Analysis of Sequence Data* (2 Vol). Humana Press, Totwa, NJ.
Guigó R., Knudsen S., Drake N. & Smith T. (1992) *J. Mol. Biol.* **226,** 141.
Guigó R. & Fickett J. W. (1995) Manuscript submitted.
Hawkins J. D. (1988) *Nucl. Acids Res.* **16,** 9893.
Henikoff S. & Henikoff J. G. (1991) *Nucl. Acids Res.* **19,** 6565.
Hirst J. D. & Sternberg M. J. E. (1992) *Biochemistry* **31,** 7211.
Hutchinson G. B. & Hayden M. R. (1992) *Nucl. Acids Res.* **20,** 3453.
Hutchinson G. B. & Hayden M. R. (1993) In Lim *et al.* (1993), p. 513.
Jurka J., Walichiewicz J. & Milosavljevic A. (1992) *J. Mol. Evol.* **35,** 286.
Kamb A., Wang C., Thomas A., DeHoff B. S., Norris F. H., Richardson K., Rine J., Skolnick M. & Rosteck P. R. Jr (1995) *Computers Biomed. Res.* **28,** 140.
Kel O. V., Romachenko A. G., Kel A. E., Naumochkin A. & Kolchanov N. A. (1995) In *Proc. 28th Annual Hawaii Int. Conf. on System Sciences (HICSS)*, Wailea, Hawaii.
Knudsen S., Guigo R. & Smith T. (1993) In Lim *et al.* (1993), p. 545.
Knueppel R., Dietze P., Lehnberg W., Frech K. & Wingender E. (1994) *J. Comp. Biol.* **1,** 191.
Konopka A. K. (1992) In Lim *et al.* (1992), p. 69.
Konopka A. K. (1994a) *Biocomputing: Informatics and Genome Projects.* Academic Press, New York.
Konopka A. K. (Ed.) (1994b) *Proc. Third Int. Conf. on Open Problems in Computational Biology, Computers Chem.* **18**(3).
Konopka A. K. & Owens J. (1990) *Gene Anal. Techn. Appl.* **7,** 35.
Konopka A. K. & Smythers G. W. (1987) *Comp. Applic. Biosci.* **3,** 193.
Konopka A. K., Smythers G. W., Owens J. & Maizel J. W. Jr (1987) *Gene Anal. Techn.* **4,** 63.
Koonin E. V., Bork P. & Sander C. (1994) *EMBO J.* **13,** 493.
Kozak M. (1991) *J. Cell Biol.* **115,** 887.
Krogh A., Mian I. S. & Haussler D. (1994) *Nucl. Acids Res.* **22,** 4768.
Krogh A., Brown M., Mian I. S., Sjoelander K. & Haussler D. (1994) *J. Mol. Biol.* **235,** 1501.
Lawrence C. E., Altschul S. F., Boguski M. S., Liu J. S., Neuwald A. & Wootton J. C. (1993) *Science* **262,** 208.
Levinson B., Kenwrick S., Lakich D., Hammonds G. Jr & Gitishier J. (1990) *Genomics* **7,** 1–11.
Lim H. A., Fickett J. W., Cantor C. R. & Robbins R. J. (1993) *Proc. Second Int. Conf. on Bioinformatics, Supercomputing, and Complex Genome Analysis.* World Scientific, Singapore.
Locker J. & Buzard G. (1990) *J. DNA Sequencing Mapping* **1,** 3.
Lopez R., Larsen F. & Prydz H. (1994) *Genomics* **24,** 133.
Matthews B. W. (1975) *Biochem. Biophys. Acta* **405,** 442.
McKeown M. (1992) *Annu. Rev. Cell. Biol.* **8,** 133.
McLachlan A. D., Staden R. & Boswell D. R. (1984) *Nucl. Acids Res.* **12,** 9567.
Milanesi L., Kolchanov N. A., Rogozin I. B., Ischenko I. V., Kel A. E., Orlov Y. L., Marenko M. P. & Vezzoni P. (1993) In Lim *et al.* (1993), p. 573.
Mitchell P. J. & Tjian R. (1989) *Science* **245,** 371.
Mizushima H. & Hayashi K. (1994) *Proceedings of Genome Informatics Workshop 1994.* Universal Academy Press, Tokyo.
Mulligan M. E., Hawley D. K., Entriken R. & McClure W. R. (1984) *Nucl. Acids Res.* **12,** 789.
Mural R. J., Mann R. C. & Uberbacher E. C. (1990) *The First Int. Conf. on Electrophoresis, Supercomputing and the Human Genome* (Edited by Cantor C. R. and Lim H. A.), p. 164. World Scientific, London.
Nakata K., Kanehisa M. & DeLisi C. (1985) *Nucl. Acids Res.* **13,** 5327.
Naora H. & Deacon N. J. (1982) *Proc Nat. Acad. Sci. U.S.A.* **78,** 6196.
Ogiwara A., Uchiyama I., Seto Y. & Kanehisa M. (1992) *Protein Engng* **5,** 479.
Oshima Y. & Gotoh Y. (1988) *J. Mol. Biol.* **195,** 247.
Oliver *et al.* (1992) *Nature* **357,** 38.
Ossadnik S. M., Buldyrev S. V., Goldberger A. L., Havlin S., Mantegna R. N., Peng, C.-K., Simons M. & Stanley H. E. (1994) *Biophys. J.* **67,** 64.
Posfai J., Bhagwat A. S., Posfai G. & Roberts R. J. (1989) *Nucl. Acids Res.* **17,** 2421.
Prestridge D. S. (1995) *J. Mol. Biol.* to appear.
Prestridge D. S. & Burks C. (1993) *Hum. Mol. Genet.* **2,** 1449.
Robison K., Gilbert W. & Church G. M. (1994) *Nature Genet.* **7,** 205.
Salzberg S. (1995) *Comp. Biol.*, to appear.
Searls D. B. (1992) *Am. Sci.* **80,** 579.
Seely O. Jr, Feng D.-F., Smith D. W., Sulzbach D. & Doolittle R. F. (1990) *Genomics* **8,** 71.
Senapathy P., Shapiro M. B. & Harris N. L. (1990) In Doolittle (1990), p. 252.
Shapiro M. B. & Senapathy P. (1987) *Nucl. Acids Res.* **15,** 7155.
Sharp P. M., Cowe E., Desmond G. H., Shields D. C., Wolfe K. H. & Wright F. (1988) *Nucl. Acids Res.* **16,** 8207.
Shavlik J. W. (1994) In Adams *et al.* (1994), p. 280.
Shavlik J. W., Towell G. G. & Noordewier M. O. (1992) *Int. J. Genome Res.* **1,** 81.
Singh G. B. & Krawetz S. A. (1994) *Int. J. Genome Res.* **1,** 321.
Smith M. W. (1988) *J. Mol. Evol.* **27,** 45.
Smith H.O., Annau T. M. & Chanresegaran S. (1990) *Proc. Nat. Acad. Sci. U.S.A.* **87,** 826.
Smith R. F. & Smith T. F. (1990) *Proc. Nat. Acad. Sci. U.S.A.* **87,** 118.
Snyder E. E. & Stormo G. D. (1993) *Nucl. Acids Res.* **21,** 607.
Soderlund C., Shanmugam P., White O. & Fields C. (1992) *Proc. 25th Hawaii Int. Conf. System Sciences, V.* (Edited by Milutinovic V. and Shriver B.), p. 653. IEEE Computer Society Press, Los Alamitos, CA.
Solovyev V. V. & Lawrence C. B. (1993) *The First International Conference on Intelligent Systems for Molecular Biology* (Edited by Hunter L., Searls D. & Shavlik J.). AAAI Press, Menlo Park, CA.
Solovyev, V. V., Salamov A. A., & Lawrence C. B. (1994a) In Altman *et al.* (1994), p. 354.
Solovyev V. V., Salamov A. A. & Lawrence C. B. (1994b) *Nucl. Acids Res.*, to appear.
Staden R. & McLachlan A. D. (1982) *Nucl. Acids Res.* **10,** 141.
Staden R. (1990) In Doolittle (1990), pp. 163.
States D. J. & Gish W. (1994) *J. Comp. Biol.* **1,** 39.
Stormo G. D. (1990) In Doolittle (1990), pp. 211.
Stormo G. D. & Haussler D. (1994) In Altman *et al.* (1994), p. 369.
Sulston J., Du Z., Thomas K., Wilson R., Hillier L., Staden R., Halloran N., Green P., Thierry-Mieg J., Qin L., Dear S., Coulson A., Craxton M., Durbin R., Berks M., Metzstein M., Hawkins T., Ainscough R. & Waterston R. (1992) *Nature* **356,** 37.

Takeda Y., Sarai A. & Rivera V. M. (1989) *Proc. Nat. Acad. Sci. U.S.A.* **86,** 439.
Thomas A. & Skolnick M. H. (1994) *IMA J. Math. Appl. Med. & Biol.* **11,** 149.
Uberbacher E. & Mural R. J. (1991) *Proc. Natl Acad. Sci. U.S.A.* **88,** 11,261.
Uberbacher E. C., Einstein J. R., Guan X. & Mural R. J. (1993) In Lim *et al.* (1993), p. 465.
von Hippel P. H. (1994) *Science* **263,** 796.
Wahle E. & Keller W. (1992) *Annu. Rev. Biochem.* **61,** 419.
Waterman M. (Ed.) (1989a) *Mathematical methods for DNA Sequences.* CRC Press, Boca Raton, FL.
Waterman M. (1989b) In Waterman (1989a), p. 93.
Waterman M. (1995) *Introduction to Computational Biology: Maps, Sequences and Genomes.* Chapman & Hall.
Wilson R., Ainscough R., Anderson K., Baynes C., Berks M., Burton J., Connell M., Boonfield J. & Copsey T. (1994) *Nature* **368,** 32.
Wingender E. (1994) TRANSFAC database. 31 October posting to human-genome-program net.bio.net.
Xu Y., Einstein J. R., Mural R. J., Shah M. & Uberbacher E. C. (1994) In Altman *et al.* (1994), p. 376.
Yada T., Ishikawa M., Totoki Y. & Okubo K. (1994) Institute for New Generation Computer Technology, Technical Report TR-876.