© 1992 Oxford University Press Nucleic Acids Research, 1992, Vol. 20, No. 24 6441–6450

Assessment of protein coding measures

James W.Fickett and Chang-Shung Tung
Theoretical Biology and Biophysics Group, and Center for Human Genome Studies, Los Alamos National Laboratory, Los Alamos, NM 87545, USA

Received July 20, 1992; Revised and Accepted November 13, 1992

ABSTRACT

A number of methods for recognizing protein coding genes in DNA sequence have been published over the last 13 years, and new, more comprehensive algorithms, drawing on the repertoire of existing techniques, continue to be developed. To optimize continued development, it is valuable to systematically review and evaluate published techniques. At the core of most gene recognition algorithms is one or more *coding measures* — functions which produce, given any sample window of sequence, a number or vector intended to measure the degree to which a sample sequence resembles a window of 'typical' exonic DNA. In this paper we review and synthesize the underlying coding measures from published algorithms. A standardized benchmark is described, and each of the measures is evaluated according to this benchmark. Our main conclusion is that a very simple and obvious measure — counting oligomers — is more effective than any of the more sophisticated measures. Different measures contain different information. However there is a great deal of redundancy in the current suite of measures. We show that in future development of gene recognition algorithms, attention can probably be limited to six of the twenty or so measures proposed to date.

INTRODUCTION

A (protein-coding) gene may be defined as any pattern in a DNA sequence which results (under proper conditions) in the generation of a protein product. The problem of gene recognition is to define an algorithm which takes as input DNA sequence and produces as output a feature table describing the location and structure of the patterns making up any genes present in the sequence.

In practice, a systematic approach to building and evaluating gene recognition algorithms must depend heavily on information from many laboratories, uniformly presented, as, for example, in the GenBank™/EMBL/DDBJ international collection of nucleotide sequence data (1–3). In fact, almost the only form of the general gene recognition problem that is well defined and for which solutions are generally testable is the problem of automatically regenerating the annotation of such databases. Clearly this leaves out important aspects of the general problem; for example, there are doubtless genes that have not yet been discovered to lie in reported sequences. Also, data on the conditions under which genes are expressed is currently very sparse in the public databases. Nevertheless, even an algorithm which could reliably reproduce most of the database annotation, given the bare sequence, would be a significant advance over what is available today.

Important aspects of gene recognition methods have been reviewed in (4–9).

A natural overall approach to building gene recognition algorithms is to first construct component algorithms that recognize the major features of genes: statistical bias in exon sequence, the patterns at intron junctions, promoters, enhancers, etc., and then to build a combined algorithm that recognizes when all these component patterns occur in a pattern consistent with that present in a gene (c.f. 10, 11). A great many ideas have been suggested for recognition of the components of genes, but for a systematic approach to building a comprehensive recognition algorithm one thing is still missing, namely an objective comparison and evaluation of the competing recognition techniques that have been put forward over the last decade or so.

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 intended to measure the 'codingness' of the sequence. Common examples include the codon usage vector, the base composition vector, and some type of fourier transform of the sequence. In this paper we review, synthesize, and evaluate the coding measures from the published literature.

An exon recognition method includes both a coding measure and a decision method which deduces a 'coding' or 'noncoding' decision from each such vector. For the moment we ignore the variety of decision methods used in published algorithms. Narrowing attention to the coding measures themselves, we define a uniform benchmarking procedure to assess these alone. This systematic approach to one module of the general gene recognition problem reflects a focus on laying the foundation for the next generation of algorithms. However some of our results will also be useful to the consumer of the current generation of recognition algorithms.

Currently a number of investigators advocate combining several of the measures reviewed here to obtain higher accuracy. We demonstrate that indeed there is more information in the ensemble of the measures than in any one of them alone. However, there is a great deal of redundancy in the set of measures published to date, and we show that, for future development, attention can probably be narrowed to only a few of the measures.

In the following sections we present first, a review of the literature and a synthesis of the coding measures proposed to date, second, a uniform benchmark and resulting assessment of those measures, third, the ensemble results just mentioned, and fourth, a discussion of the results and suggestions for how they may be used.

SYNTHESIS OF PUBLISHED CODING MEASURES

In this section we survey the literature and describe the coding measures to be evaluated. The grouping of the measures in what follows is only for purposes of convenient exposition, and is not important to the results of this study.

Several published measures are slight variants, or special cases, of others. Thus we have generalized and synthesized the measures. This has been done in such a way that for each published algorithm, one of the coding measures we test is at least as discriminatory as the one implicit in the published description of the algorithm.

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 (and counts of in-phase 'words')

Define the **Codon Usage Measure** to be the 64 element vector giving the frequencies, among the test-codons, of each of the 64 possible codons. Most coding measures are summarizing, in one way or another, the effects of unequal usage of codons. Thus the codon usage measure is in some sense the most fundamental of coding measures; it has been widely used.

Staden & McLachlan (12) calculate a probability of occurrence of the codon usage vector for each window. First the frequency of occurrence of each possible codon in some reference set of actual genes is noted. Then the product of these frequency values, over the test-codons of a sample windows, is taken as the probability of the window. To find coding regions the probabilities of successive windows, offset by a single nucleotide, are compared. (Because of the difficulty of choosing the correct reference set to compute the codon usage standard, Staden (13) suggests that one might begin with the average amino acid composition for all proteins and reverse translate, using synonymous codons equally, to get a universal codon usage standard. See the discussion of McCaldon and Argos' work (14) below.)

Gribskov, Devereux & Burgess (15; cf. also 16) use a likelihood ratio approach. The conditional probabilities of the sequence, given the translated amino acid sequence, under a coding and under a noncoding (random sequence) hypothesis, are compared. Thus set f(abc) = frequency in a reference set of coding regions of the codon abc, r(abc) = probability of occurrence of abc in random sequences (independently occurring nucleotides with the same frequencies as in the reference set), F(abc) = sum over synonymous codons of f(abc), R(abc) = sum over synonymous codons of r(abc). Then set preference(abc) = [f(abc)/F(abc)]/[r(abc)/R(abc)], and preference(window) = $\Pi(\text{preference(test-codons)})^{(1/(\text{window length}))}$. The reference set taken is one of highly expressed genes, and the suggestion is made that the preference function defined above is also correlated with expression level.

Hinds and Blake (17) set, for any trinucleotide abc, p(abc) = (frequency of abc in-phase in coding regions)/(frequency of abc overall). Then they score a window with the average p-values of the 'codons' in it, and smooth the score-versus-position curve over all windows.

Kolaskar & Reddy (18) combine tests for initiators and for codon bias. The idea of the codon bias test is basically to count the number of highly marked test-codons in an open reading frame, where markedness is defined as follows. If w is an arbitrary oligonucleotide let f(w,i) be the frequency of w in phase i of coding regions (of some reference set)—that is with the first base of w occurring in 'codon' position i. Now let abc be an arbitrary trinucleotide and define P_1(abc) = f(abc,1)/[f(ab,1)f(c,3)], P_2(abc) = f(abc,2)/[f(ab,2)f(c,1)], P_3(abc) = f(abc,3)/[f(ab,3)f(c,2)]. A trinucleotide is highly marked if P_1 is very different from 1 but P_2 and P_3 are near 1, i.e. if within coding regions the third base depends strongly on the first two, but outside of coding regions the third base is relatively independent.

Borodovsky et al. (19–21) construct four separate Markov models for DNA, one for noncoding regions, and three for the three phases of coding regions. Then they calculate the probability of each window of the sequence, based on each of the four models. Finally, a probability of each model, given the window, is calculated by Bayes' theorem. The Markov models used are of step lengths 1, 2, or 3. More recently, Borodovsky and McIninch have extended these results to Markov processes with step length up to 5 (*personal communication*). The Markov model of step length 5, considered the most accurate, is based on phase-specific counts for oligonucleotides of length 6. Thus for n=0, 1, 2 we define the **Hexamer-n Measure** to be the counts of all hexamers offset by n from the starting base of a test-codon.

Claverie and Bougueleret (22) (*cf.* also 23) use the differences in frequency of occurrence of all hexamers between intron and exon sequences. They define, for each possible hexamer w, p(w) = frequency of occurrence of w in exons, q(w) = frequency of occurrence of w in introns, and d(w) = p(w) /(p(w) + q(w)), or, later, d'(w) = p(w)−q(w). d(w) is plotted for each successive hexamer in the test sequence. In a later paper (Claverie, Sauvaget and Bougueleret, 24) d(w) is computed separately for hexamer counts in each of the three possible reading frames, and smoothed by averaging over a window. The phase-dependent methods depend on the Hexamer-n measures; define also the **Hexamer Measure** to be the frequency count in the window of all hexamers.

In Fichant and Gautier (25) codon usage vectors (ignoring stop codons) in a sliding window are thought of as points in 61-dimensional space. Correspondence analysis (CA; see, e.g., 26), applied to the set of vectors from one contiguous sequence, is used to make a projection on a one-dimensional subspace which (usually) differentiates between coding and noncoding. (The assumption is that codon usage is even more uniform within a gene than within a genome.) Then two things are used to make a coding/noncoding judgment. First, the variance of CA scores of three windows offset by one tells whether the windows cover genuine coding regions. Second, the window with the lowest CA score gives the correct frame.

Lapedes et al. (27) explore several neural net approaches (for an introduction to neural nets see 28). They find that a neural net with one layer of hidden neurons, looking at a raw window of sequence, uses the hidden layer to summarize codon usage data. This confirms that the codon usage vector is a very natural coding measure. Their final results use a one layer net (essentially a perceptron; cf. 29) on codon usage vectors in 64-space. Farber, Lapedes, and Sirotkin (30) extend these results, using a neural net with all di-test-codon frequencies as input. Thus we also define the **Dicodon Usage Measure** (equivalent to the Hexamer-0 Measure), with the frequencies of all in-frame hexamers from the sample window.

Methods related to the encoded amino acid sequence

A very natural measure, used in some form by most investigators, is simply the presence or absence of in-frame stop codons. We define the **Open Reading Frame Measure** as the length of the longest stretch of sense test-codons in the window.

McCaldon & Argos (14) use an approach closely related to those based on observed codon usage. They begin by grouping the PIR database into a new set of superfamilies, with the property that no two superfamilies share highly similar sequences. This grouping is used to make frequency tables for oligopeptides which they believe to be representative of proteins in general. From this they make a two step Markov model for amino acid sequences, and thence a two step Markov model for codon choice in protein coding regions. Then for any window on a DNA sequence, they calculate the probability of occurrence of that window, based on the Markov model. As in (12, 13), this probability is compared for a triplet of overlapping windows, offset by 1.

The McCaldon and Argos model assumes equal codon usage within a family of synonymous codons, so that all of the information actually used is in oligopeptide frequency counts. It would be natural to define coding measures for the amino acid frequencies (corresponding to Staden's method) and the tripeptide frequencies (for McCaldon and Argos' method). Since McCaldon and Argos indicate that similar results were obtained with a Bernoulli process (corresponding to the use of the single amino acid frequencies), we test here the mono- and di-amino acid frequency counts as coding measures.

Define the **Amino Acid Usage Measure** to be 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'. Define the **Diamino Acid Usage Measure** to be the 441-vector given by translating the window and counting all the (overlapping) dipeptides (including 'stop' as an 'amino acid').

Tramontano and Macchiato (31) select hydrophobicity as a significant measure of protein function. They suggest that a mutation in a genuine codon is likely to result in a smaller change in hydrophobicity (in the encoded amino acid) than is a mutation in a trinucleotide of a noncoding region (in its corresponding amino acid). So they define the information value of a codon as $\Sigma_{j=1,3}[\Sigma_{i=1,nj}(p_i*d_{ij})]/n_j$, where n_j is the number of sense mutations of the codon, p_i is the probability of the i^{th} mutation, and d_{ij} is the difference in hydrophobicity caused by the mutation (see the paper for tabulated values for d_{ij}). The information value of a window is then the average information value of the test-codons in that window. We define the **Stability of Hydrophobicity Measure** as this average. The authors show that this measure is little affected by the existence of overlapping coding regions.

Tramontano and Macchiato make a prediction based on the assumption that an unknown sequence is drawn from two populations of equal size, each having a normal distribution of the information value indicator, one with mean and standard deviation matching the coding regions in their database, one with mean and standard deviation matching the noncoding regions. Moody and Fristensky (32) refine the model by allowing the two populations from which the sample comes to be of different sizes. They again use the same stability of hydrophobicity measure.

Base compositional bias between codon positions

The next few methods are based, directly or indirectly, on the asymmetry of the base composition in the three codon positions. As a basis for all of these, define the **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,i) is the frequency of b in position i.

Shepherd (33) notes that the most frequently used codons are of the form RNY. He tests for the existence and frame of a coding region by measuring the number of differences between the sequence and the pattern RNYRNY...RNY. In fact, a number of investigators have noticed prototypical properties of codons, e.g. that they are often of the form RNY or WWS, or that a certain base is more common in one position than in another. The following measure is one natural generalization of all of these observations.

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 b^{th} 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(noncoding 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. (This is very close to the 'Frame Bias Matrix' measure of Mural, Mann, and Uberbacher (34)).

Fickett (35) takes eight measurements on a window. Four of them are simply the frequencies of the bases. The other four measure the asymmetry of the base composition in the three codon positions. That is, with f(b,i) as above, define assym(b) = max(f(b,1),f(b,2),f(b,3))/[1+min(f(b,1),f(b,2),f(b,3))]. Each of these is used to make an estimate of coding likelihood, and the separate estimates are all combined using a linear weighted sum. (Staden (13) uses the following variant of the asymmetry measure. For each base he calculates $\mu(b) = \Sigma_i(f(b,i))/3$ and $diff(b) = \Sigma_i(|f(b,i)-\mu(b)|)$. His measure is then $\Sigma_b(diff(b))$.)

Both Fickett (35) and Staden (13) are giving ad hoc measures of how much f(b,i) varies with i. For the **Position Asymmetry Measure** we use a standard measure of the spread of data points, (a multiple of) the sample variance. Define $\mu(b) = \Sigma_i(f(b,i))/3$ and $asymm(b) = \Sigma_i(f(b,i)-\mu(b))^2$ Then define the position asymmetry measure to be [asymm(A),asymm(C),asymm(G), asymm(T)].

Bibb, Findlay and Johnson (36) calculate C+G content at each of the three test-codon positions. Staden (13), in a variant of Shepherd's (33) method, finds base compositions of each position in a prototypical codon by reverse translation from a prototypical protein. That is, he calculates a prototypical amino acid frequency distribution by averaging over all proteins, derives from this a prototypical codon frequency distribution by assuming equal use of synonymous codons, and from this calculates a prototypical base composition for each codon position. The correlation between this and the base composition for each codon position in the window is used as a discriminator. The codon prototype measure, defined above, covers these cases as well.

Almagor (37) measures bias from random of f(b,i) as a function of b, using entropy in the sense of information theory. That is, given f(b,i) as above, define 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)].

Trifonov (38) compiled frequency tables for each base in each codon position for 130 species, and found that the frequency of G in first codon position is universally greater that the frequency of G in second codon position. The codon prototype measure is again implied.

Imperfect periodicity in base occurrences

Michel (39) bases an exon recognition algorithm on the T autocorrelation function. Let T_i be the number of pairs of T nucleotides separated by i bases, L the length of the window, and $D_i = T_i/L$. Linear discriminant analysis is used to show that D_2, D_3, D_4, D_7, D_8, D_9 are the most telling six of the first ten D_i's. Linear discriminant analysis is again applied to these six to make a coding/noncoding decision.

It seems likely that the autocorrelation functions of all four bases could contribute significant information to the coding decision. Thus we define the **Autocorrelation Measure** as follows. Let 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 [auto(b,i)/(window__length-i−1) (frequency__of__b)^2], where b=A,C,G,T and i=0,...9.

Silverman and Linsker (40) define a Fourier transform on a DNA sequence which depends only on the overall patterns of periodicity in the sequence and not on which bases are involved in the periodic patterns. The vertices of a regular tetrahedron centered at the origin in 3-spac e are labeled with A, C, G, and T. Then the function f(m) is defined to be the vector from the origin to the vertex labeled with the mth base of the sequence. If the window length is 2M then the Fourier decomposition of f(m) may be written as $f(m) = \Sigma_n[C_n\, e^{\pi inm/M}]$. The scalar transform is $g(n) = |C_n|^2$. This turns out to be the same as $g(n) = (1/(4M^2))\Sigma_p[\Sigma_m(f(m)\cdot f(m-p)]e^{\pi inp/M}$, i.e. the ordinary Fourier transform of a natural 3-dimensional autocorrelation function. Silverman and Linsker point out that the periodicity of 3 in coding regions appears as a peak at g(2M/3).

An approach essentially equivalent to that of Silverman and Linsker is to represent the four bases as the four basis vectors in 4-space. This latter representation results in a somewhat cleaner analysis (and is easier to generalize to symbol strings over different size alphabets). We may then alter their equation 3.4 as follows. 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: $FC(n) = \Sigma_p[\Sigma_m(EQ(\text{base } m,\text{base } m\text{-}p))]e^{\pi inp/M}$. Then define the **Fourier Measure** to be [FC(2M/2), FC(2M/3),...,FC(2M/9] (i.e. the Fourier coefficients of the autocorrelation function for periods 2 to 9).

Arques & Michel (41). When the frequency of R(i−other-bases)YR in eukaryotic coding sequences is plotted as a function of i, there is a clear separation of the points according to the remainder of i upon division by three. The first new result in this paper is that when the focus is narrowed to i congruent to 0 modulo 3, clear peaks in the frequency are observed at i = 9, 18, 27 (but not 36). The authors then specialize the above pattern to R(j−other-bases)RYR (j here corresponds to i−1 above). The second new result is that the frequency of this pattern shows a clear peak at j=8 not only in protein coding, but in tRNA genes. The periodicity of 9 is absent in rRNA genes, prokaryotic protein coding genes and introns. (Similar results obtain if Y and R are reversed.). Based on these observations we 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)].

Konopka (42) combines a measure of entropy (in a manner similar to Almagor (37); cf. the entropy measure) with a ratio of two measures of periodicity that are essentially equivalent to Fourier coefficients (cf. the Fourier measure).

Other global patterns

Erickson & Altman (43) examine both non-uniform codon usage (cf. the Codon Usage Measure) and the relationship between successive codons. For the latter, they point out that the dependency of the first base of a codon on the third base of the previous codon. Shulman, Steinberg, and Westmoreland (44) take this a step further, noting a higher dependency between nucleotides in codon positions 1 & 2, and between positions 2 & 3, than between positions 3 & 1. This effect is measured with a chi-squared test. Although there is not enough data in a window of tens of codons for a valid chi-squared test, one may still use the chi-squared function for a natural indicator of difference between two distributions. We define the **Dinucleotide Frame Measure** as follows: 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).

Shulman, Steinberg, and Westmoreland (44) also suggest two other tests. First, they point out that G is found more frequently in codon position 1 than in codon position 2 (cf. the codon prototype measure). Second, they show that the set of 'words' obtained by dividing a coding region into codons has a more biased distribution than that obtained by dividing it into words of another size, or in another frame. It appears from other results (e.g. Konopka & Smythers (45) and Arques & Michel (46)) that the most significant of these differences is between words of length 2 and words of length 3. Thus we define the **Word Measure** as follows: 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.

Blaisdell (47) finds that both coding and noncoding sequences have longer than expected runs of R and Y, and shorter than expected runs of W and S. However the tendency toward longer than expected runs of R and Y is stronger in noncoding than in coding, and the trend toward shorter than expected runs of W and S is stronger in coding than in noncoding. (This may be connected with the observation of Wada and Suyama (48) that the predicted melting temperature of coding regions is both higher and more uniform than that of noncoding regions.) In partial explanation of these results, Blaisdell notes that many codons are WWS, and that the WWS pattern is even further from random than the RNY one. The WWS pattern is dealt with by the codon prototype measure.

For the observation concerning run length, we define a **Run Measure** as follows. Let $S_1, S_2, \ldots S_{14}$ be the nontrivial 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 $[r_{ij}]$.

Blake & Early (49) find a number of notable characteristics of coding regions in *E.coli*: (a) Coding regions (both RNA and protein genes) are embedded in segments of uniform G+C content of about 53%, about 1000 bases long; noncoding sequences are embedded in segments about 500 bases long of average G+C content 46% (cf. the composition measure). (b) There is less bias in nearest neighbor frequency in coding than in noncoding regions: WW and SS nearest neighbors are about 7% more frequent than expected and WS and SW are about 7% less frequent than expected, in noncoding. A similar statement, with 7% replaced by 4%, holds for coding. To make a general measure, let f(w), for any possible word w, be the frequency of w in the sample window. Now for each dinucleotide ab let 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.

Mural, Mann, and Uberbacher (34) use both a version of the codon prototype measure (mentioned above) and a measure termed the dinucleotide usage fractal dimension. The latter is based on the relationship between the frequencies with which dinucleotides occur overall and the frequencies with which they occur together in the same trinucleotide, but details on calculating the actual value of this measure on a window are not specified.

Uberbacher and Mural (50) combine a number of coding measures using a two layer neural net. The measures include the codon prototype measure, the TESTCODE algorithm output (cf. the position asymmetry measure), the dinucleotide fractal dimension (see above), and the hexamer measure, the latter calculated on three different reference sets. The first reference set for the hexamer measure is coding (on either strand and in any frame) versus noncoding. The second is coding (correct strand and phase only) versus noncoding, the third is actual versus random DNA, and the fourth is repetitive versus non-repetitive DNA. The hexamer measure, as defined above, covers the first of these, and the dicodon measure covers the second. The third is very close to the first and will not be tested separately for this study. There are problems in characterizing the repetitive elements of many genomes, including the human, both because the variation of individual repeats from the consensus is incompletely known, and because new repetitive elements are constantly being discovered. Thus for the fourth case above, we simply take all hexamers which occur, on average, more than twice every 4096 bases to be in the 'repetitive' set (we are indebted to N.Doggett for suggesting this approach (personal communication)). Using only the counts of these hexamers (324 in human, 247 in *E.coli*), in the coding and noncoding reference sets, gives the **Repeat Measure.**

MATERIALS AND METHODS

In this section we describe a benchmark by which the relative usefulness of any coding measure may be evaluated. Of course the usefulness of a measure may vary, depending on the context in which it is applied. We will show that the benchmark is reasonably general, by testing several variants of it on the set of measures defined above, and showing that while the absolute accuracies vary with incidental factors, yet our overall conclusions hold in the several variant contexts.

Some measures are most naturally used to differentiate only between coding and noncoding regions, irrespective of frame, while others are more specific. Thus we make use of two different definitions of 'coding', as follows. A (single-stranded) window of DNA sequence is 'phase-coding' if the successive trinucleotides of the window, beginning with the first base, are used as codons in some gene. A window is 'region-coding' if every nucleotide, or its complement, is in a codon (irrespective of frame) of some gene. The benchmark includes an evaluation of the measure for both of these definitions of 'coding'.

In brief, the benchmark is defined as follows. Homogeneous (fully coding or fully noncoding) windows of fixed size were taken from GenBank. The data corpus was split in half, and the first part was used as a training set. Discriminant analysis was used to define a linear function of the measure which discriminates coding from noncoding. A threshold was then set to equalize the error rates on the coding and noncoding 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 noncoding parts of the test set was taken as the overall accuracy of the measure. A more detailed description follows.

The data

All data were taken from the GenBank^{TM}/EMBL/DDBJ international collection of nucleotide sequence data, in the form of the on-line relational GenBank database (1). Genomic human sequences were extracted 30 May 1992, and *E.coli* sequences were extracted on 28 June 1992. For the primary benchmark, successive, non-overlapping windows of length 54 bases were taken from all human genomic sequences (with partial windows at sequence ends discarded). Variants of the benchmark also used 54 base windows from *E.coli* sequences and windows of length 108 and 162 from genomic human sequences. (The particular lengths chosen are not particularly significant, but were chosen to simplify some of the calculations.)

Each set of windows was split into two, with the first half to be used for training, or parametrizing the algorithm, and the second half to be used for testing, or evaluating its accuracy. It is critical to separate the training and testing sets. See (27) for examples.

Within each train and test set, only homogeneous (fully coding or fully noncoding) windows with no ambiguous bases were used. Table 1 shows the numbers of windows used in each set.

Evaluation of measures

All existing algorithms which incorporate a vector valued coding measure (as we have defined them) use (with minor variations) some linear combination of the vector elements as the basis of a coding/noncoding decision. That is, there is some coefficient vector **c** and some threshold t such that the sample window with coding measure vector **m** is thought to be coding if and only if $\mathbf{c} \cdot \mathbf{m} > t$ (or, in some cases, if and only if $k^{\mathbf{c} \cdot \mathbf{m}} > t$ for some constant k).

For this study, it was desirable to have one simple, uniform method for deriving such a coefficient vector **c**. For reasons discussed below, we used Linear Discriminant Analysis (LDA, a standard technique in multivariate analysis; see, e.g. 51, 52). Classical LDA finds a coefficient vector c such that the ratio of the between-population variation of $\mathbf{c} \cdot \mathbf{m}$ to the within-population variation of $\mathbf{c} \cdot \mathbf{m}$ is maximized. One begins by defining the total

Table 1. Numbers of windows, in thousands, used in the sixteen train and test sets

	Hum 54	Hum 108	Hum 162	Eco 54
Region Train	20.5/125.1	7.1/58.1	3.5/36.5	40.3/14.2
Region Test	22.9/122.1	8.2/57.0	4.3/35.6	38.8/15.7
Phase Train	4.2/152.0	1.5/73.3	.8/47.4	8.7/48.7
Phase Test	4.7/151.2	1.7/72.9	.9/46.9	8.3/49.1

Each entry of the table shows first the number of coding windows, then the number of noncoding.

covariance matrix **T** and the within-population covariance matrix **W**, as follows: let x_{wfm} be the value of the m^th^ scalar component of the measure on the w^th^ window of function f (f=0 for noncoding, f=1 for coding). Let $\bar{x}_{fm}$ be the mean of x_{wfm} over w, and $\bar{x}_m$ the mean of x_{wfm} over both f and w. Then the element in the r^th^ row and c^th^ column of **T** is $t_{rc} = \Sigma_f\Sigma_w(x_{wfr}-\bar{x}_r)(x_{wfc}-\bar{x}_c)$. And the element in the r^th^ row and c^th^ column of **W** is $w_{rc} = \Sigma_f\Sigma_w(x_{wfr}-\bar{x}_{fr})(x_{wfc}-\bar{x}_{fc})$. **T** and **W** can be calculated with a single pass through the data, using the formulae $S_{rc} = \Sigma_f\Sigma_w(x_{wfr}*x_{wfc})$, $t_{rc} = S_{rc}-(n_0 + n_1) *\bar{x}_r *\bar{x}_c$, and $w_{rc} = S_{rc}-\bar{x}_{0r} * \bar{x}_{0c} * n_0-\bar{x}_{1r} * \bar{x}_{1c} * n_1$, where n_0 is the number of non-coding windows, and n_1 is the number of coding windows. Next we define the between-population covariance matrix as $\mathbf{B} = \mathbf{T}-\mathbf{W}$, and diagonalize $\mathbf{W}^{-1}\mathbf{B}$ to find its eigenvalues. The eigenvector corresponding to the largest eigenvalue of $\mathbf{W}^{-1}\mathbf{B}$ is the desired coefficient vector **c**.

Classical LDA requires the inversion of the within-sample covariance matrix **W**. For many of the measures defined above, especially those with hundreds or thousands of elements, this presents a problem. High redundancy of information in a measure leads to a covariance matrix that is very nearly singular, making standard inversion algorithms highly unstable (small perturbations in the data result in large perturbations in the discriminant vector). Thus for the primary benchmark we used a form of LDA where one ignores all off-diagonal elements of the covariance matrix. In this case each scalar component of the discriminant vector is calculated from the means and variances of the corresponding component variable on the noncoding and coding sets, by the formula $(\mu_{cds}-\mu_{ncd}) /(V_{cds} + V_{ncd})$. Classical LDA was used as one of the variants of the benchmark, in those cases where it could be applied.

The form of LDA which ignores off-diagonal elements of the covariance matrix is equivalent to classification on the basis of Penrose distance (53), and we term the resulting coefficient vector the Penrose Discriminant vector. Geometrically, Penrose discrimination amounts to rescaling each axis of the space of observations so that all variables have the same average within-set variance (i.e. the average of the variance within the coding and within the noncoding sets), and then projecting all points onto a line between the centroid of the two sets. Classical LDA is similar, except that the rescaling may also move the axes of the observation space relative to each other.

Given a scalar measure, or a vector-valued measures and a discriminant vector, a pass through the training set was made to choose an appropriate threshold by which to make the coding/noncoding decision. In the case of scalar measures m, the threshold was simply applied to m; in the case of vector measures **m**, the threshold was applied to **c**·**m**. In each case the threshold was chosen so that the fraction of errors on the coding windows (i.e. the false negative rate), was equal to the fraction of errors on the noncoding windows (the false positive rate).

Note that both the mean and variance of many of the following indicators depend on the window length, so the above procedure must be carried out separately for each window length.

Finally, the resulting real-valued function on sequence windows, either m or **c**·**m**, with the appropriate threshold for obtaining a coding/noncoding decision, was applied to the testing set. The resulting accuracy was taken to be the average of the correct prediction rate on the true coding and true noncoding subsets, i.e. the accuracy is the average of the sensitivity and the specificity.

RESULTS

We applied the primary benchmark and four variants to each of the measures defined above. The primary benchmark is made by applying a Penrose discriminant function to 54 base windows of human genomic sequences. In the four variants, one condition of the benchmark is varied at a time: window length, discriminant function, or organism. In the first two variants, the window length is varied to 108 and 162 bases. In the third variant just the organism is changed, to *E.coli*. In the fourth variant the discriminant method is changed to classical linear discriminant analysis.

For classical linear discriminant analysis the variables making up the measure must not be linearly dependent. Thus for the fourth variant we used a subset of the variables in some of the measures, as follows: for amino acid usage, we removed the 'stop' count; for codon usage, the TTT count; for composition, the three T counts; and for run, we used only the counts for R, Y, W and S.

Results of the tests are shown in Table 2 (for the 'region' definition of coding) and Table 3 (for the 'phase' definition of coding). Both tables are ordered according to results on the primary benchmark.

Many of the measures are independent of coding phase, and it might have been thought that these would perform much better on the region coding test. But in fact all such measures performed either better on the phase coding test, or only slightly worse. This leads to the pleasing conclusion that it will probably not be necessary to apply certain measures in a region coding test, others in a phase coding test, and to combine the results in a post-processing step. Rather, all measures can simply be used to discriminate coding regions in phase.

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. composition can be derived from codon usage counts). Figure 1 shows which measures can be derived from others.

The tree in the right half of Figure 1 contains most of the measures currently used. It is remarkable that, without exception,

Table 2. Percentage accuracy (average of specificity and sensitivity) of the coding measures in predicting region coding

Measure	Human 54 Penrose	Human 108 Penrose	Human 162 Penrose	*E.coli* 54 Penrose	Human 54 Classical
Hexamer	70.5	73.1	74.2	67.5	–
Position Asymmetry	70.2	76.6	80.6	61.6	70.3
Dicodon Usage	70.2	72.9	73.9	67.5	–
Fourier	69.9	76.5	80.8	61.3	69.9
Hexamer-1	69.9	72.6	73.8	66.8	–
Hexamer-2	69.9	72.6	73.8	66.7	–
Run	66.6	70.3	71.3	63.6	67.9
Codon Usage	65.2	68.0	69.5	64.1	66.
Repeat	65.1	69.9	73.1	62.4	–
Autocorrelation	64.9	71.1	77.0	58.2	64.9
Dinucleotide Bias	62.9	55.5	50.7	55.9	62.7
Diamino Acid Usage	62.8	66.3	67.8	61.3	–
Composition	61.7	64.1	65.9	61.7	61.3
Amino Acid Usage	60.6	63.4	64.7	59.7	61.3
Word	59.5	66.4	71.4	56.6	61.0
Entropy	58.4	63.1	66.2	55.0	58.4
Dinucleotide Frame	58.0	62.9	66.6	54.6	58.0
Open Reading Frame	57.8	59.2	60.7	57.4	57.8
Stability Hydrophobicity	55.5	57.5	58.7	55.5	55.5
Codon Prototype	54.7	56.1	56.4	54.7	54.7
Period 9	52.5	53.0	52.8	51.8	52.4

Data from five benchmark situations are shown, with varying data set (Human or *E.coli*), window length (54, 108, or 162) and decision method (Penrose discriminant or Classical linear discriminant).

Table 3. Percentage accuracy (average of specificity and sensitivity) of the coding measures in predicting phase-specific coding

Measure	Human 54 Penrose	Human 108 Penrose	Human 162 Penrose	*E.coli* 54 Penrose	Human 54 Classical
Dicodon Usage	80.7	84.3	85.4	88.7	–
Hexamer-2	79.5	82.8	84.2	87.2	–
Hexamer-1	78.6	82.0	83.3	87.1	-
Codon Usage	78.0	81.0	82.1	86.9	81.7
Diamino Acid Usage	77.2	84.9	87.7	84.2	–
Amino Acid Usage	75.3	81.1	83.6	83.3	76.2
Codon Prototype	74.3	78.2	80.5	78.8	74.3
Open Reading Frame	72.9	83.3	88.0	75.6	72.9
Composition	72.2	74.7	75.9	78.8	75.0
Hexamer	71.7	74.3	75.4	70.5	–
Position Asymmetry	68.1	74.7	77.5	59.7	68.3
Fourier	67.8	74.8	77.6	54.7	67.5
Run	66.1	69.6	71.1	62.5	67.0
Repeat	65.5	70.4	73.8	63.0	–
Autocorrelation	64.5	71.4	76.3	58.6	64.6
Dinucleotide Bias	61.9	56.4	55.5	50.3	61.4
Entropy	61.1	64.7	69.2	56.2	61.2
Stability Hydrophobicity	59.8	62.5	63.8	60.4	59.8
Word	58.4	65.6	72.9	57.6	60.7
Dinucleotide Frame	58.4	62.6	65.7	52.1	56.5
Period 9	55.0	58.4	58.9	53.9	55.0

Data from five benchmark situations are shown, with varying data set (Human or *E.coli*), window length (54, 108, or 162) and decision method (Penrose discriminant or Classical linear discriminant).

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. This is very clearly the case for most of the measures. One case which is less clear is that of the diamino (or amino) acid usage measure, which with the Penrose discriminant on longer human windows scores higher than the dicodon (respectively, codon) usage measure. There are several reasons, however, for preferring the in-phase hexamer (including dicodon usage) measure. First, the Penrose discriminant can be improved upon significantly for measures with high redundancy of information. We may note that while the score of the amino acid usage measure improves by only 0.9 percentage points when the classical linear discriminant is used in place of Penrose, the accuracy of the codon usage measure improves by 3.7 percentage points. So when more sophisticated techniques are used to take advantage of the information in the measure, we think that the in-phase hexamer

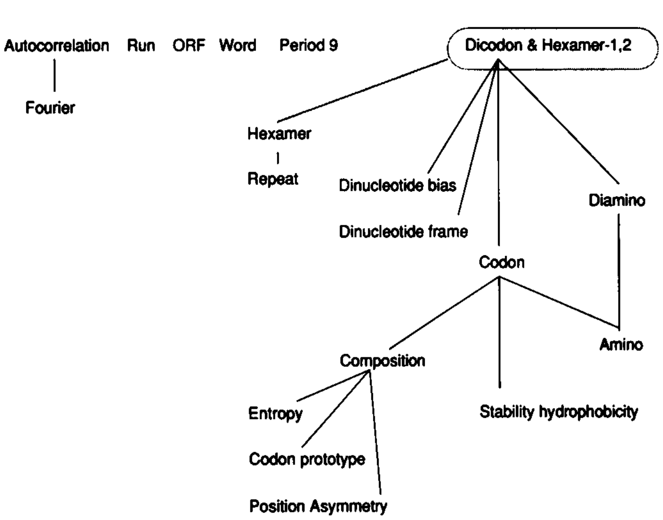

Figure 1. Derivability of measures. Whenever two measures are connected by a line, the lower one is a function of the higher one. The hexamer measure is derivable from the dicodon, hexamer-1 and hexamer-2 measures together. The Fourier measure is derivable from the autocorrelation function (as we have defined it), together with the base composition of the window.

measures will outperform the diamino acid measure even for longer windows. Second, the in-phase hexamer measures very clearly outperform the diamino acid measure on 54 base windows, in both species tested. There are already algorithms that work well on longer windows; we think the direction of research should be to develop algorithms for shorter windows.

Of the measures not in the main tree at the right of Figure 1, the period 9 measure and the word measure yield rather poor results, and the autocorrelation measure is essentially equivalent to the Fourier measure. Taking all the results together thus suggests that of the measures tested here, future algorithms should be based on Fourier, run, ORF, and the in-phase hexamer measures.

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 E.Uberbacher kindly applied the Coding Recognition Module of GRAIL (50) 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).

All discriminant vectors, as well as values for thresholds, sensitivity, and specificity may be obtained by sending an e-mail message containing only the text 'cds__discriminant' to bioserve@genome.lanl.gov.

DISCUSSION

Computer-assisted recognition of genes in DNA sequences is widely recognized as a central problem in computational molecular biology. Many techniques applicable to this problem have been developed in the past decade or so, and much current effort is being directed at combining component techniques into integrated algorithms. This has been, and will continue to be, a very fruitful line of work, yet it has suffered seriously from one weakness. Namely, it is often the case that the component techniques have not been carefully assessed for their effectiveness. We propose that one of the best ways to improve the accuracy of gene recognition algorithms is to reverse this situation.

A natural place to begin this process is in the systematic evaluation of what we have termed coding measures—scalar or vector-valued functions on windows of sequence, intended to measure resemblance to typical exonic DNA. Our aim has been to assess the value of all coding measures from the literature, apart, as much as possible, from both their algorithmic and their application context, so that in building the next generation of integrated algorithms we may with some confidence know what are the most informative coding measures. (One method for discovering genes, namely similarity searches on sequence databases, is not amenable to the same sort of accuracy assessment that we have carried out here, because a negative result means little. Cf., however, 54 and 55.)

The conceptual model by which we divide an algorithm into a coding measure and a decision method is a natural one. Many general decision methods fit well in this context: neural nets, discriminant analysis, and many probability calculations take a vector of measurements and produce a single number by which a decision may be made. Of course many algorithms process the data in several stages, so that there is some ambiguity in dividing the algorithm into a coding measure and a decision method. Our criteria in defining the coding measure used by a published algorithm have been (a) that any biological insights of the authors be represented in the coding measure we extract, and (b) that the final statistic by which the authors make the coding decision be a linear function of our coding measure or a slight variant thereof.

One may note that many of the measures are linear combinations of other measures. Thus for example choosing a set of coefficients for the amino acid usage measure is just one way of choosing a set of coefficients for the codon usage measure. Nevertheless, since no one way of choosing the coefficient vector is guaranteed to maximize the accuracy, we felt it important to test all the measures.

Our benchmarking procedure was defined not to obtain the utmost accuracy from each measure, but to provide a straightforward assessment of how well each measure already accomplishes the task of summarizing important information about codingness. Linear discriminant analysis is simple, relatively uncontroversial in its mode of application, and easily reproduced. In addition, it is reasonably effective, as seen by the fact that the linear discriminant of each measure was more accurate than the linear discriminant of other measures derived from the given one. Classical linear discriminant analysis corrects the greatest weakness of the Penrose discriminant in that the former takes into account inter-variable correlations. But the average increase in accuracy made by using the classical discriminant was only 0.6%.

All this is only to say that the Penrose discriminant is well suited to comparative benchmarking. Naturally, much more sophisticated decision methods will be used to build state-of-the-art gene recognition algorithms.

Human sequence is a natural choice for the primary benchmark both because of its intrinsic interest and because it is a significant challenge: most algorithms don't work as well on human as on prokaryotic or simpler eukaryotic sequence (data not shown). The choice of window size was similarly based: good algorithms already exist for 100 base windows; windows of half that size are an important and significant challenge. In other words, we believe that discrimination of human sequence windows of about 50 bases is a significant and difficult, but achievable, goal. Results using the variants of the primary benchmark show that for the purposes of this study these choices are not critical.

Most known human DNA sequence is in the vicinity of highly expressed genes, and it is possible that some of our conclusions may need refinement as more sequence is determined from random genomic locations. We do not expect major changes in the conclusions reached here, for two reasons. First, preliminary results on genomic sequencing of yeast and the nematode indicate that there is far less intergenic DNA than was once supposed, even in eukaryotes (56, 57). So known noncoding DNA is likely to be a reasonable approximation to genomic noncoding DNA. Second, in a separate study (Fickett and Guigo, manuscript in preparation) we have compared the distribution of the codon usage measure on the recently determined yeast chromosome III sequence as against previously known yeast sequences, and found that the two distributions are very similar. So while weakly expressed genes will give a somewhat lower signal with most coding measures, the difference in performance will likely not be great.

There is no standard definition for the accuracy of a recognition algorithm. The most important feature of the measure we chose is that it gives equal weight to the coding and noncoding sets (many definitions of accuracy give equal weight to each window; in this study that would have given too much emphasis to noncoding windows). The other main choice we made was to use the number of true positives rather than the number of predicted positives in the denominator of the sensitivity fraction, and similarly for specificity.

Source code, instructions for anonymous ftp retrieval of data, and instructions for carrying out the primary benchmark described here may be obtained by sending e-mail containing only the text 'cds__benchmark' to bioserve@genome.lanl.gov.

Our most important conclusion is that a measure which seems to embody little biological understanding—counts of in-phase hexanucleotides—is in fact the most effective one. (One might of course distinguish between the goals of biological insight about coding regions and accuracy in discriminating coding regions; in this paper we are concerned with the latter.) In-phase word count measures have a long history. The first use we know of the codon usage measure in a published algorithm is by Staden and McLachlan (12). Separate word counts of different lengths for each phase were considered by Borodovsky et al. (17–21). These papers considered words of length 1, 2 and 3. More recently the same author (personal communication) has extended his work to include words of length 6. Claverie, Sauvaget, and Bougeleret (24) were the first (as far as we know) to use the in-phase hexamer count measures.

Our second main conclusion is that it is probably most useful to discriminate the coding region and its phase in a single step, rather than doing the two tasks separately and combining results.

The third main conclusion of this study is that for accurate exon discrimination most of the measures have been superceded. Either they have low accuracy, or they are measuring just one facet of what a more general, and more accurate measure, is sensing. We think that, of the measures surveyed here, future development can be limited to application of the in-phase word count measures, the ORF measure, the Fourier measure, and the run measure.

Other useful measures may of course be discovered. New measures, as well as new algorithms, will be most valuable if they are carefully benchmarked. While the benchmark we have introduced should suffice for many purposes, different benchmarks may of course be appropriate in different situations. When a different benchmark is desired it would be very useful if developers would benchmark their own measure, decision method, or full algorithm, together with its main competitors, in a uniform way. It is unfortunate, and difficult for users, that the accuracy figures for different algorithms are calculated on different sets and by different means, so that a meaningful comparison is rarely possible.

For further development of coding region recognition methods it will also be very valuable to systematically compare the methods by which a decision is deduced from the values of one or more measures. Now that it is fairly clear which are the best measures, a systematic evaluation of methods for making use of those measures is a natural next step.

ACKNOWLEDGEMENTS

This work was performed under the auspices of the U.S. Department of Energy and the Los Alamos Center for Human Genome Studies, with funding by DOE/OHER grant ERFWPF116, NIH grant GM-37812 to C.Burks, and LANL grant X16K. We are grateful for helpful comments from C.Burks, M.J.Cinkosky, A.S.Lapedes, M.S.Waterman, K.M.Sirotkin, and T.F.Smith. A stimulating environment for part of this work was provided by the Aspen Center for Physics.

REFERENCES

1. Cinkosky, M.J., Fickett, J.W., Gilna, P., and Burks, C. (1991) Science., 252, 1273–1277.
2. Higgins,D.G., Fuchs,R., Stoehr,P.J., Cameron,G.N. (1992) Nucleic Acids Res., 20 supplement, 2071–2074.
3. Miyazawa, S. (1990)in Bell, G.I. and Marr, T.G. (eds.) Computers and DNA: Proceedings of the Interface Between Computational Science and Nucleic Acid Sequencing Workshop, Addison-Wesley, Redwood City, CA, pp 47–62.
4. Stormo, G.D. (1987) in Bishop, M.J. and Rawlings, C.J. (eds) Nucleic Acid and Protein Sequence Analysis: A Practical Approach, IRL Press, Oxford.
5. Stormo, G.D. (1988) Annu. Rev. Biophys. Chem., 17, 241–63.
6. von Heijne, G. (1988) Sequence Analysis in Molecular Biology: Treasure Trove or Trivial Pursuit. Academic Press, San Diego, CA.
7. Staden, R. (1990) Meth. Enzymol., 183, 163–180.
8. Rice, P., Elliston, K., and Gribskov, M. (1991) in Gribskov, M. and Devereux, J. (eds) Sequence Analysis Primer, Stockton Press, New York, NY.
9. Gelfand, M.S. (1990) Biotechnology Software 7, 3–9.
10. Fields, C.A. and Soderlund, C.A. (1990) Comp. Appl. Biosci., 6, 263–270.
11. Guigo, R., Knudsen, S., Drake, N., and Smith, T. (1992) J. Mol. Biol. 226, 141–157.
12. Staden, R. and McLachlan, A.D. (1982) Nucleic Acids Res., 10, 141–156.
13. Staden, R. (1984) Nucleic Acids Res., 12, 551–567.
14. McCaldon,P. and Argos, P. (1988) Proteins: Structure, Function, and Genetics, 4, 99–122.
15. Gribskov, M., Devereux, J., and Burgess, R.R. (1984) Nucleic Acids Res., 12, 539–549
16. Sharp, P.M., and Li, W.-H. (1987) Nucleic Acids Res., 15, 1281–1295.
17. Hinds, P.W. and Blake, R.D. (1985) J. Biomolec. Struct. Dynam., 3, 543–549
18. Kolaskar, A.S. and Reddy, B.V.B. (1985) Nucleic Acids Res., 13, 185–194
19. Borodovsky, M.Y., Sprizhitskii, Y.A., Golovanov, E.I., and Aleksandrov, A.A. (1986) Molekulyarnaya Biologiya, 20, 1014–1023.

20. Borodovsky, M.Y., Sprizhitskii, Y.A., Golovanov, E.I., and Aleksandrov, A.A. (1986) Molekulyarnaya Biologiya, 20, 1024–1033.
21. Borodovsky, M.Y., Sprizhitskii, Y.A., Golovanov, E.I., and Aleksandrov, A.A. (1986) Molekulyarnaya Biologiya, 20, 1390–1398.
22. Claverie, J.-M. and Bougueleret, L. (1986) Nucleic Acids Res., 14, 179–196.
23. Volinia, S., Gambari, R., Bernardi, F., and Barrai, I. (1989) Comp. Appl. Biosci., 5, 33–40.
24. Claverie, J.-M., Sauvaget, I., and Bougueleret, L. (1990) Meth. Enzymol., 183, 237–252.
25. Fichant, G. and Gautier, C. (1987) Comp. Appl. Biosci., 3, 287–295
26. Hill, M.O. (1974) Appl. Statist., 23, 340–354.
27. Lapedes, A.S., Barnes,C., Burks, C., Farber, R.M. and Sirotkin, K.M. (1990) in Bell, G.I. and Marr, T.G. (eds.) Computers and DNA: Proceedings of the Interface Between Computational Science and Nucleic Acid Sequencing Workshop, Addison-Wesley, Redwood City, CA, pp 157–182.
28. Lippmann, R.P. (1987) IEEE ASSP Mag., April, 4–22.
29. Minsky, M.L. and Papert, S.A. (1988) Perceptrons: Expanded Edition. MIT Press, Cambridge, MA.
30. Farber, R.M., Lapedes, A.S., Sirotkin, K.M. (1992) J. Mol. Biol., in press
31. Tramontano, A. and Macchiato, M.F. (1986) Nucleic Acids Res., 14, 127–135
32. Moody, M.E. and Fristensky, B. (1987) DNA, 6, 493–495
33. Shepherd, J.C.W. (1981) Proc. Nat. Acad. Sci. USA, 78, 1596–1600.
34. Mural, R.J., Mann, R.C., and Uberbacher, E.C. (1991) in, Cantor, C.C. and Lim, H.A., (eds) Proceedings of the First International Conference on Electrophoresis, Supercomputing and the Human Genome, World Scientific Co., Singapore, pp 164–172.
35. Fickett, J.W. (1982) Nucleic Acids Res., 10, 5303–5318
36. Bibb, M.J., Findlay, P.R., and Johnson, M.W. (1984) Gene 30, 157–166.
37. Almagor, H. (1985) J. Theor. Biol., 117, 127–136.
38. Trifonov, E.N. (1987) J. Mol. Biol., 194, 643–652.
39. Michel, C.J. (1986) J. Theor. Biol., 120, 223–236.
40. Silverman, B.D. and Linsker, R. (1986) J. Theor. Biol., 118, 295–300
41. Arques, D.G. and Michel, C.J. (1987) Math. Biosci., 86, 1–14.
42. Konopka, A.K. (1990) in Sarma, R.H. and Sarma, M.H. (eds) Structure and Methods: V1. Human Genome Initiative & DNA Recombination, Adenine Press, Guilderland, NY, pp 113–125.
43. Erickson, J.W. and Altman, G.G. (1979) J. Math. Biol., 7, 219–230.
44. Shulman, M.J., Steinberg, C.M. and Westmoreland, N. (1981) J. Theor. Biol., 88, 409–420.
45. Konopka, A.K. and Smythers, G.W. (1987) Comp. Appl. Biosci., 3, 193–201.
46. Arques, D.G. and Michel, C.J. (1987) Nucleic Acids Res., 15, 7581–7592.
47. Blaisdell, B.E. (1983) J. Molec. Evol., 19, 122–133.
48. Wada, A. and Suyama, A (1985) in Molecular Basis of Cancer, Part A. Alan R. Liss
49. Blake, R.D. and Early, S. (1986) J. Biomolec. Struct. Dynam., 4, 291–307.
50. Uberbacher, E.C. and Mural, R.J. (1992) Proc. Nat. Acad. Sci. USA, 88, 11261–11265.
51. Manly, B.F.J. (1986) Multivariate Statistical Methods: A Primer. Chapman and Hall, Bristol, England.
52. Lachenbruch, P.A. and Goldstein, M. (1979) Biometrics, 35, 69–85.
53. Penrose, L.W. (1953) Annals of Eugenics, 18, 337–343.
54. Seely, O.Jr., Feng, D.-F., Smith, D.W., Sulzbach, D., and Doolittle, R.F. (1990) Genomics 8, 71–82.
55. Claverie, J.-M. (1992) Genomics 12, 838–841.
56. Sulston, J., Du, Z., Thomas, K., Wilson, R., Hillier, L., Staden, R., Halloran, N., Green, P., Thierry-Mieg, J., Qiu, L., Dear, S., Coulson, A., Craxton, M., Durbin, R., Berks, M., Metzstein, M., Hawkins, T., Ainscough, R., and Waterston, R. (1992) Nature 356, 37–41.
57. Oliver, S.G., et al. (1992) Nature 357, 38–46.