

## Gene recognition by combination of several gene-finding programs

Katsuhiko Murakami<sup>1,2</sup> and Toshihisa Takagi<sup>1</sup>

<sup>1</sup>Human Genome Center, Institute of Medical Science, University of Tokyo, 4-6-1 Shirokane-dai Minato-ku, Tokyo 108-8639 and <sup>2</sup>Central Research Laboratory, Hitachi Ltd, 1-280 Higashi-Koigakubo, Kokubunji-shi, Tokyo 185-8601, Japan

Received on February 9, 1998; accepted on May 25, 1998

### Abstract

**Motivation:** A number of programs have been developed to predict the eukaryotic gene structures in DNA sequences. However, gene finding is still a challenging problem.

**Results:** We have explored the effectiveness when the results of several gene-finding programs were re-analyzed and combined. We studied several methods with four programs (FEXH, GeneParser3, GEN-SCAN and GRAIL2). By HIGHEST-policy combination method or BOUNDARY method, approximate correlation (AC) improved by 3–5% in comparison with the best single gene-finding program. From another viewpoint, OR-based combination of the four programs is the most reliable to know whether a candidate exon overlaps with the real exon or not, although it is less sensitive than GENSCAN for exon–intron boundaries. Our methods can easily be extended to combine other programs.

**Availability:** We have developed a server program (Shirokane System) and a client program (GeneScope) to use the methods. GeneScope is available through a WWW site (<http://gf.genome.ad.jp/>).

**Contact:** {katsu,takagi}@ims.u-tokyo.ac.jp

### Introduction

A large number of uncharacterized DNA sequences are generated in the development of the genome projects. It is essential to develop algorithms for computational gene finding. For the analysis of DNA sequence, homology searching is an effective method for gene finding if there are any similar sequences in the database, but it is estimated that ~50% of the newly discovered genes have no similar homologues in the protein sequence databases (Uberbacher *et al.*, 1996). Therefore, gene-finding programs based on the pattern recognition method have been developed. Those programs use neural network [(GRAIL (Xu *et al.*, 1994), GeneParser (Snyder and Stormo, 1993)], discriminant analysis [GeneFinder (Solovyev *et al.*, 1994, 1995), MZEF (Zhang, 1997)], hidden Markov model [VEIL (Henderson *et al.*, 1997), Genie (Kulp *et al.*, 1996), GENSCAN (Burge and Karlin, 1997) and HMMgene (Krogh, 1997)], etc. Recently, combinatorial approaches that use both homology search and pattern recogni-

tion were proposed by the developers of GeneParser (Snyder and Stormo, 1995), Genie (Kulp *et al.*, 1997), GRAIL (Xu and Uberbacher, 1997) and GeneFinder (Solovyev and Salamov, 1997).

Each program takes advantage of various measures to score partial features of genes, such as the coding region and signal patterns like splice sites. Since the programs consider the features in different ways, predictions for the same sequence by different programs are often not identical. Even if a program performs worse than the others in general cases, it sometimes predicts more correctly than the other programs for some novel sequences. This suggests that the program grasps some features of genes on DNA sequences and that the features may not be taken into account by the other programs. Thus, the accuracy of gene prediction can be improved if the outputs of several programs are combined in a proper way. Note that exons represent coding regions for convenience.

Burset and Guigó (1996) discussed that it may be beneficial to combine the outputs of several gene-finding programs. They evaluated nine gene-finding programs with 570 vertebrate DNA sequences. Among their data set, which consists of 2649 actual exons, 174 exons were predicted by all the programs; 172 of them corresponded exactly to actual exons annotated in GenBank, while the remaining two had a large overlap with actual exons. Additionally, only 33 (~1%) were missed completely by all the programs.

Although their results suggest that the combination of several gene-finding programs may be useful, the number of exons predicted by all the programs was 174, which is only 6.6% of all the 2649 coding exons. Therefore, it is not easy to detect many correct exons by combinations of several prediction programs. Development of the methods of combination is needed.

A related tool is GeneNomi, which is introduced in the paper on Genotator (Harris, 1997). Genotator automatically analyzes an input sequence by many analysis tools, such as BLAST, GRAIL, GeneFinder, Genie and so on. While Genotator is a tool to display analysis results, GeneNomi is a tool to combine several sources of information and to make

'conservative exon predictions'. The combined results of GeneNomi were tested for a standardized data set of 305 complete gene sequences. The complete gene sequence means that each piece of data has an actual initiation codon and an actual stop codon in each sequence data (Kulp *et al.*, 1996). The paper claimed that GeneNomi's prediction was slightly better than the best program among Xpound (Thomas and Skolnick, 1994), Genefinder, GRAIL and Genie. However, the details (the methods and evaluation of the combination) were not described in the paper. Besides, the result may not reflect real performance since the data are comprised of only complete gene sequences, and may contain a part of the learning data of the programs. Furthermore, it is not clear from the paper whether the effect of worse programs could spoil an outstanding prediction program, like GENSCAN; GENSCAN was developed recently and claims to be significantly more accurate than the others (Burge and Karlin, 1997).

BCM Search launcher (Smith *et al.*, 1996) also utilizes several analysis tools served on the World Wide Web, as well as gene prediction programs. However, this is not a tool that analyzes the results of gene prediction programs.

We have investigated five methods of combination of the predictions by four gene-finding programs (FEXH, GeneParser3, GENSCAN or GRAIL2). We tested the methods and compared their performance. In the experiments, the data we prepared were comprised of only new sequences whose highly homologous proteins were not found in protein databases. Using this kind of data makes it possible to conclude that the increase in accuracy by combination, if any, is not due to the compensation of learning data.

Furthermore, to test the performance in a practical way, we used DNA sequences that contain parts of genes, as well as complete genes. This point is different from the data of Burset and Guigó. It is a more natural situation where input DNA sequences contain multiple genes or parts of genes.

Here we present five combination methods. In all methods, we transformed scores of the programs into probabilistic ones to compare the predictions of different programs. Then we made final predictions by the five methods.

In some methods of combination, the performance increased especially by a method called HIGHEST-policy. Approximate correlation (AC) improved by 3–5% in comparison with the best single gene-finding program. Although we have implemented the combination of only four gene-finding programs in this paper, these methods can easily be extended to utilize other gene-finding programs. There is a tendency that combinations are more accurate as the number of programs increases. Thus, when a more precise prediction program is newly developed, we can expect to improve the accuracy by these methods of combination with the new program.

## System and methods

The combination programs run on Sun Microsystems Ultra2, under SunOS 5.5.1. The core program of the combination program was written in C language, and some perl and C shell scripts invoke other gene-finding programs, sort them out and run the core program. To use the combination program and to get the results of several gene-finding programs, we have also developed a client server system. The details are described in the subsection 'Availability'.

### *Selection of gene-finding programs*

One of the objectives is to examine whether the performance becomes better if we use several gene-finding programs. In this work, we selected four gene-finding programs for combination: FEXH, GeneParser3, GENSCAN and GRAIL2 (Version 1.3). The details of the reasons why we selected the four programs are given in Results and discussion.

### *DNA sequence data*

We extracted human DNA entries with at least one 'CDS' from GenBank Release 100 (April 1997). The other constraints in extracting entries from GenBank were as follows. The term SOURCE is 'Homo sapiens'. The term 'DNA' in the first line of each entry is required. Entries with non-standard splice site conservative dinucleotides (i.e. not GT-AC) were discarded. Furthermore, entries with keywords such as pseudo, putative, ORF, alternative, predict, and fusion were discarded since it was not confirmed experimentally whether they are transcribed or not. We also discarded immunoglobulin genes due to the complexity of the gene structures. At this stage, we collected 1394 loci.

An old entry might be learned by any of the gene-finding programs. Using those data may cause confusion in understanding combined results. To study the performance against new data, the old data registered before June 1996 were discarded. The learning data of the four programs were released before that date. The remaining data consisted of 332 loci.

This data set contains partial genes. This is because it is a more natural situation where an input DNA sequence has multiple genes or partial genes rather than the situation where the sequence has only one complete gene.

### *Homology with known protein sequences*

Of the new 332 DNA sequence data, there were some sequences whose translated amino acid sequences were almost the same as some of the known protein sequences.

We studied the similarity between a protein encoded on the new DNA sequence and its most homologous protein. Given a new DNA sequence, we used BLASTX to identify a protein that is the most homologous to the protein encoded on

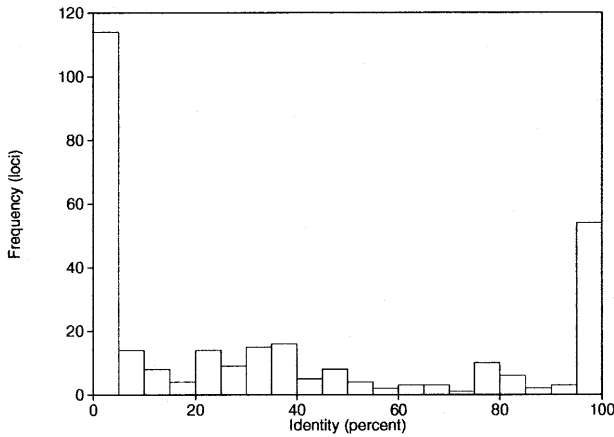


Fig. 1. The histogram of identities.

the DNA sequence. The homology was measured by the probability in the output of BLASTX. The repetitive sequences (ALU, L1, etc.) were masked beforehand using the programs BLASTN and xblast. After the most homologous protein was identified, we calculated the identity using a sequence comparison program ALIGN (Myers and Miller, 1988). Figure 1 shows the histogram of identities. Then the new DNA sequences were classified into two groups: known (identity  $\geq 80\%$ , 113 loci) and novel (219 loci). The 'known' sequences were not used in this study. The novel data were randomly divided into two data sets. The training set contains two-thirds of the novel data and the remaining data are used as the test set. The data set is available upon request.

#### Prediction using several gene-finding programs

In this section, we describe how to integrate the predictions of the four gene-finding programs (FEXH, GeneParser3, GENSCAN and GRAIL) and make a prediction using the different results of those programs.

*Transformation to probabilistic score.* First, we transformed the score of predicted exon into a probabilistic one so that we can compare the quality of the prediction of different programs. For this purpose, we examined the relationship between the score and the accuracy of prediction for each program.

We show the frequency of predicted exons and their error rates against the program's raw scores in Figure 2 for only FEXH. The error rate decreases with the score. The error rate of FEXH in the region higher than score 20 is irrelevant because of the limited number of the samples and thus should be ignored.

Using these error rate distributions, we estimated score functions  $P$  by the least-squares method on the training data. The functions represent the probabilities of predicted exons being true given the score of the program. The score of

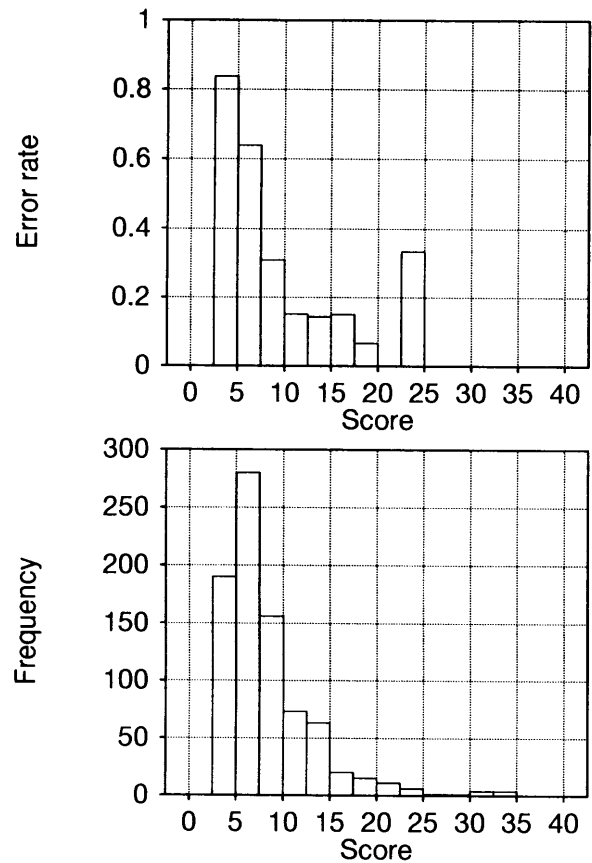


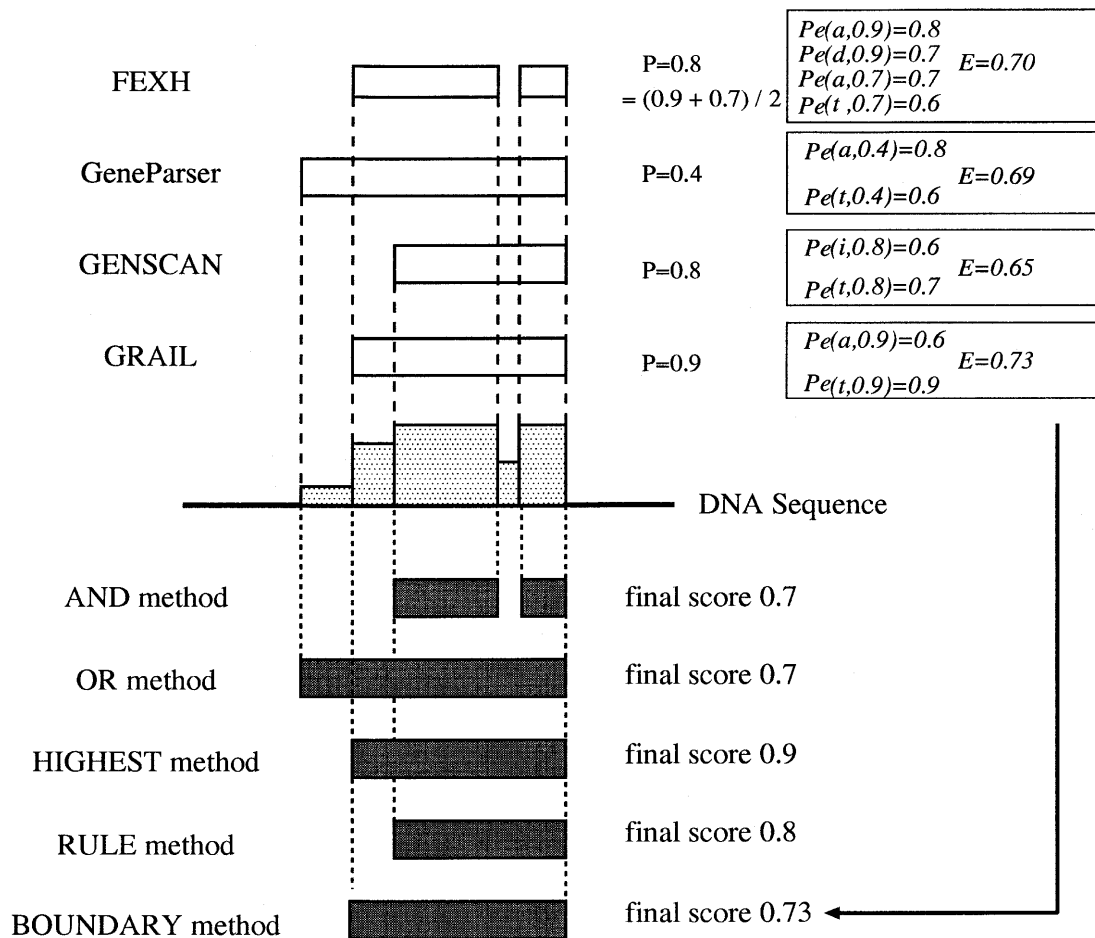
Fig. 2. The score dependency of error rate and frequency (FEXH).

GRAIL and GP3 should be equal to  $P$  because they employ multilayer perceptron whose output will approximate the posterior probability, given a training data set (Gish, 1990). The score function  $P$  for GENSCAN is the score itself, as the score is intended to be the probability. However, outputs of programs should be normalized, because training sets used in each program are different from each other. Moreover, the error rate distributions which we plotted differed from the output scores slightly. This is partly because the size of our data set is not large. Another reason could be that the perceptron is not trained enough; the data size of the training set of the programs might be too small to make a precise posterior probability function.

If we use linear functions, it may occur that  $P$  exceeds the region of  $0 \leq P \leq 1$ . To prevent such an event, we used logistic functions for estimation of  $P(\text{score})$ , defined as follows:

$$P_i(\text{score}) = \frac{1}{1 + e^{A_i + B_i \times \text{score}}} \quad (1)$$

where  $\text{score}$  is an output of a gene-finding program, and  $A_i$  and  $B_i$  are constants determined by the error rate distribution.  $i$  corresponds to one of the programs ( $i = \text{GP3, GENSCAN,}$



**Fig. 3.** An artificial example of a cluster of predicted exons. The four programs predicted exons in the cluster with different coding–non-coding boundaries. Each open rectangle represents a predicted exon by each program with their  $P$  written on the right. In this case, FEXH predicted two exons at this region with the two  $P$ s (0.9 and 0.7). The  $P$ s are modified to their average. At the middle, the X-axis represents the position of a DNA sequence. The Y-axis represents the number of programs that predicted those positions as positive (coding region). The closed rectangles at the bottom represent new predicted regions by these methods with their final scores on the right.

GRAIL). For FEXH,  $P(score)$  was also defined by a logistic function, because the program uses discriminant analysis. All the scores of the programs are transformed by these score functions.

*Determination of coding region.* We describe how to determine coding regions from the results of different gene-finding programs after the score transformation. We considered five different methods to combine the results of prediction.

We explain the methods with an example where the same region is predicted differently by the several programs with different coding–non-coding boundaries, as shown in Figure 3. From the output of the programs, predicted exons in the same region are clustered. A cluster is defined as a region where any base is predicted positively by at least one program. If a program predicted more than one exon in the

cluster, we take the average of the predicted exons and assign the average as new  $P$ s of the exons so that we can compare the tools in the cluster (like the results of FEXH in the example of Figure 3).

For the cluster of predicted exons, we determine exon candidates by the following five methods.

1. AND-based method: exon candidates are the regions predicted by all the programs. This method is supposed to result in the lowest rate of wrong exons.
2. OR-based method: exon candidates are the regions predicted by at least one of the programs, which is the same region of the cluster in Figure 3. This method helps us to grasp the maximum sensitivity (including overlapped match) at the exon level.

3. HIGHEST-method: exon candidates are the regions which have the highest  $P$  among the programs.
4. RULE-method: this method is based on the performance test of Burset and Guigó (1996) and that of Burge and Karlin (1997). GENSCAN has the best accuracy for the coding–non-coding boundary among the four programs in terms of sensitivity and specificity at the exon level according to the Burge’s result. Without GENSCAN, the program FGENEH (a derivation of FEXH) marked the best accuracy of coding–non-coding boundary among the remaining three programs, and GeneParser3 follows. Thus, in the fourth method, the coding region is determined by one of the programs selected in the priority order (GENSCAN, FEXH, GeneParser and GRAIL). For each cluster of predicted exons, we select a program with the highest priority which predicted the region as a coding region. Then an exon candidate is the region predicted by the selected program.
5. BOUNDARY-method: here we define the probability  $Pb(bt, P)$  ( $bt = \{i, d, a, t\}$ ), which is the probability of the coding–non-coding boundary being correct, given the  $P$  and a boundary type  $bt$ . The boundary type is either the initiation codon (i), donor site (d), acceptor site (a) or termination codon (t).  $Pb$  was estimated for each program on the training data. If we have an exon with two boundaries whose types are  $l$  and  $r$  ( $l = \{i, a\}$ ,  $r = \{d, t\}$ ) with a  $P$ , we calculate a new score  $E$  defined by:

$$E(l, r, P) = \sqrt{Pb(l, P) \times Pb(r, P)} \quad (2)$$

If the program predicted more than one exon in the cluster, we calculate the  $N$ th root of multiplied  $Pbs$  as in Figure 3. In this method, we selected the exon(s) of the program with the best  $E$ .

The exon candidates considered by the five methods are shown in the example of Figure 3 as closed rectangles.

At this stage, each exon is just a candidate. Each exon will be given a final score, as described in the next subsection. Then, the final prediction was produced in accordance with both the final score and the threshold determined for each method and combination.

*Final score.* The final scores for the three methods, AND, OR and HIGHEST, are defined as the average of all the scores returned by the programs. If any programs predict the region as a non-coding region, the  $P$  of the program is regarded as zero in the calculation. Therefore, negative predictions (prediction as non-coding) were taken into account for the final scores. In the fourth method (RULE), the final score is identical to the  $P$  of the selected program in the cluster. In the fifth method (BOUNDARY), the final score is identical to the  $E$ .

*Threshold.* We set thresholds for all methods to cut exon candidates that have low scores. Therefore, the methods did not answer all the candidate exons automatically. A threshold (TH) was determined using Missing Exons (ME) and Wrong Exons (WE) for each method and for each combination of the programs. ME is the proportion of actual exons without overlap to predicted exons and WE is the proportion of predicted exons without overlap to actual exons. TH was determined as a value at which

$$\frac{1}{2}(ME + WE) \quad (3)$$

was the lowest on the training set, under the condition that  $|ME - WE| \leq 0.2$ . Without the condition, ME (or WE) can be too high, although WE (or ME) is very close to zero. Since negative data (bases of non-coding region) are huge compared to positive data (bases of coding region), maximizing the correlation coefficient (CC) produces too many false positives (Salzberg, 1997). Therefore, we used equation (3). This gives reasonable predictions in which sensitivity and specificity are balanced. The same TH values of the corresponding combination and method were used in the test set.

## Results and discussion

### Evaluation numbers

We studied four combinatorial methods. To evaluate the general accuracy for these methods, we calculated the AC:

$$AC = \frac{1}{2} \left[ \frac{TP}{TP + FN} + \frac{TP}{TP + FP} + \frac{TN}{TN + FP} + \frac{TN}{TN + FN} \right] - 1 \quad (4)$$

for each sequence. TP is ‘true positive’, which is the number of coding nucleotides predicted as coding. FN is ‘false negative’, which is the number of coding nucleotides predicted as non-coding. TN is ‘true negative’, which is the number of non-coding nucleotides predicted as non-coding. FP is ‘false positive’, which is the number of non-coding nucleotides predicted as coding. AC were averaged over the test sets.

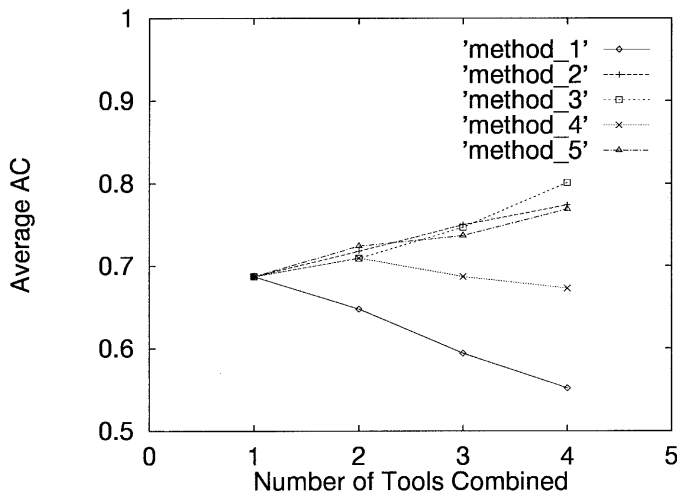
With regard to the performance for exact match (both boundaries are correct) at the exon level, sensitivity (SN) and specificity (SP) are defined by:

$$SN = \frac{TP}{TP + FN} \quad SP = \frac{TP}{TP + FP} \quad (5)$$

where TP, TN, FP and FN are similar to the previous abbreviations used in the definition of AC at the nucleotide level, except that they are the total number of exons in the data set.

### Results of combination

We investigated all the combinations of the gene-finding programs by all the methods. As a result, we observed that the combination of several programs generally improved in



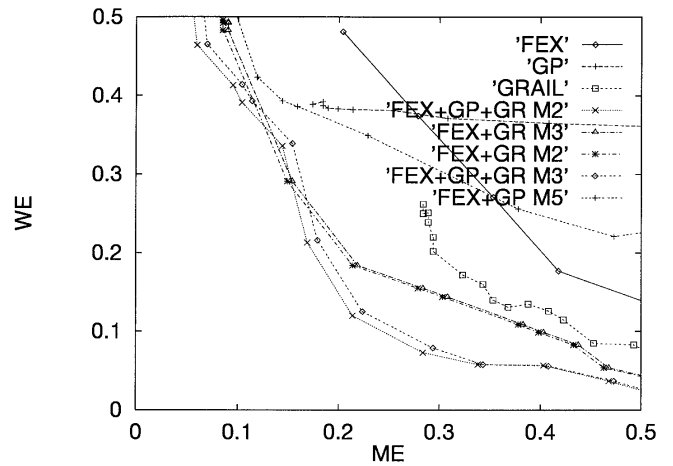
**Fig. 4.** Average AC against the number of programs combined. Each line represents each method. The average AC of all combinations by methods 2 (OR), 3 (HIGHEST) and 5 (BOUNDARY) increased as more programs were combined.

terms of AC. Figure 4 shows the relationship between AC which are averaged over the possible combination and the number of programs combined. The number of possible combinations is  $4C_n$ , when we pick up  $n$  program(s) out of the four programs. For methods 2, 3 and 5, the average AC increased as many programs were combined. For method 1 (AND) and method 4 (RULE), the AC decreased as the number of combined programs increased. Therefore, if we use methods 2, 3 and 5, the combinations improve accuracy on average.

Since GENSCAN predicts significantly better than the others (Burge and Karlin, 1997), and there are some differences between combinations with and without GENSCAN, we show the results of combination with and without GENSCAN from here.

**Table 1.** The results of the top five combinations without GENSCAN and results of individual programs. The abbreviation ‘FEX + GR’ means the combination of FEX and GRAIL. ‘GP’ is GeneParser3. The last column, ‘TH’, is the threshold of the final score used in the combination in the method. Bold numbers indicate that the combination became better in terms of the column than any individual programs

Programs	Method	Nucleotide level			Exon level				TH
		sn	sp	AC	SN	SP	ME	WE	
FEX + GP + GR	M2	<b>0.798</b>	0.778	<b>0.739</b>	0.299	0.343	0.214	<b>0.120</b>	0.35
FEX + GR	M3	0.706	<b>0.848</b>	<b>0.722</b>	0.408	<b>0.488</b>	0.284	<b>0.155</b>	0.25
FEX + GR	M2	0.716	0.821	<b>0.715</b>	0.318	0.381	0.279	<b>0.155</b>	0.25
FEX + GP + GR	M3	0.659	<b>0.891</b>	<b>0.709</b>	0.388	<b>0.513</b>	0.294	<b>0.079</b>	0.40
FEX + GP	M5	<b>0.754</b>	0.792	<b>0.701</b>	<b>0.468</b>	0.339	<b>0.159</b>	0.386	0.15
GRAIL	–	0.667	0.832	0.671	0.418	0.431	0.284	0.262	0.00
FEX	–	0.735	0.743	0.671	0.338	0.221	0.204	0.481	0.00
GP	–	0.711	0.795	0.640	0.388	0.284	0.174	0.389	0.00



**Fig. 5.** The relationship between WE and ME, for FEX, GP and GRAIL, and for the top five combinations of FEX, GP and GRAIL. The points correspond to various thresholds.

Without GENSCAN. Table 1 shows the results of the top five combinations without GENSCAN and that of individual programs. This table clearly shows the effectiveness of the combinations. The bold figures in Table 1 indicate that the combination became better in terms of the column than any individual programs. All the AC in Table 1 were superior to the best single program (GRAIL). As for the exact exon prediction, although SN were slightly decreased, SP of the prediction by method 3 (HIGHEST) were significantly higher than the best single program. With respect to the overlapping matches (including exact matches), ME and WE of the top five combinations gave better results, as shown in Table 1 and Figure 5 for various thresholds. Method 2 gave low WE/ME, as shown in Figure 5. This also illustrates the effectiveness of the combinations.

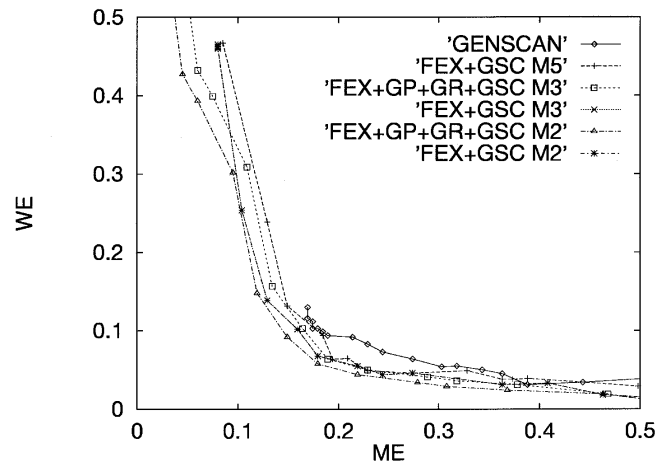
From Figure 5, we can see the contribution of GP. Although the AC value of 'FEX + GR M2/M3' was better than 'FEX + GP + GR M3' (combination of all programs) in Table 1, ME of the 'FEX + GR M2/M3' was worse than 'FEX + GP + GR M2/M3' in Figure 5 when WE is <0.25. The result indicates that GP contributed to the discrimination of the regions of candidates, but not to the discrimination of exon-intron boundaries. A reason might be that GP was more sensitive than the others at the exon level (Table 1) and it provided actual exons. Therefore, if users are first interested in identifying coding regions rather than the boundaries, GP should be taken into account.

*With GENSCAN.* The improvement by combination was less than the previous combinations without GENSCAN. There were seven combinations which were better in AC than the best single program, GENSCAN. The best combination (FEX + GSC M5) was better than GENSCAN in most of the numbers in Table 2. Some other combinations by methods 2 and 3 also improved accuracy, as they did in the previous combinations.

The combination of all programs 'FEX + GP + GR + GSC M2/M3' gave lower ME and WE than the combination 'EX + GP + GR' without GENSCAN. Other combinations in Table 2 were better than GENSCAN with regard to the nucleotide level (sn, sp and AC), although the power of exact exon detection became weak. ME and WE of the combination was plotted in Figure 6 for various thresholds.

#### Features of individual gene-finding programs

To discuss the combination effect more precisely, we summarize the features of each gene-finding program. These programs are common in that they integrate various features of DNA sequence, which are mainly coding regions and exon-intron boundaries (such as splice sites). However, the details of the methods are different from each other. In brief, FEX characterizes the coding region and splice sites by frequencies of different oligomers for regions with different relative distance to the exon-intron boundary. GeneParser



**Fig. 6.** The relationship between WE and ME of GENSCAN and the combination of all four programs (FEX, GP, GENSCAN and GRAIL) for methods 2 and 3. The points correspond to various thresholds.

uses hexamer frequency and complexity to characterize coding potential, and uses weight matrices to characterize the exon-intron boundary. GRAIL uses three coding potentials based on hexamer frequency, and uses weight matrices and neural networks for the exon-intron boundary. GENSCAN uses hexamer-based coding potential and uses maximal dependence decomposition (MDD) to identify donor site, and uses the windowed weight array model (WWAM) to identify acceptor site (and branch point). Although techniques for splice site identification in GENSCAN have been shown to be better than weight matrices, which are used in GeneParser, it is unknown whether this technique is still better than the techniques used in FEXH or GRAIL. GENSCAN deals with more features than the other programs, such as TATA-box, cap-site and poly(A) signal by weight matrix. All the programs except FEXH take account of the GC content of DNA sequence. However, FEXH did not appear to have weakness for sequences with low GC content (Bursat Guigó, 1996).

**Table 2.** The results of the top five combinations and the best single program (GENSCAN). The abbreviation 'FEX + GP + GR + GSC' means the combination of FEX and GeneParser and GRAIL and GENSCAN. Bold numbers indicate that the combination became better in terms of the column than GENSCAN

Programs	Method	Nucleotide level			Exon level				TH
		sn	sp	AC	SN	SP	ME	WE	
FEX + GSC	M5	<b>0.785</b>	0.903	<b>0.803</b>	<b>0.697</b>	<b>0.711</b>	<b>0.149</b>	0.132	0.10
FEX + GP + GR + GSC	M3	<b>0.781</b>	<b>0.912</b>	<b>0.801</b>	0.572	0.622	<b>0.164</b>	<b>0.103</b>	0.30
FEX + GSC	M3	0.742	<b>0.918</b>	<b>0.778</b>	0.657	<b>0.750</b>	0.179	<b>0.068</b>	0.30
FEX + GP + GR + GSC	M2	<b>0.827</b>	0.804	<b>0.774</b>	0.343	0.375	<b>0.149</b>	<b>0.092</b>	0.30
FEX + GSC	M2	<b>0.760</b>	0.879	<b>0.770</b>	0.453	0.517	0.179	<b>0.068</b>	0.30
GENSCAN	–	0.745	0.904	0.767	0.672	0.703	0.169	0.130	0.00

GENSCAN characterizes many features than the others, and it has the excellent characterizations of other features which other program has already dealt with. Therefore, GENSCAN provides better prediction than the others in most cases, and it is difficult to improve performance by combination of GENSCAN and others. However, some combinations of GENSCAN and other programs succeeded in increasing accuracy. We have studied some combinations which improved accuracy, as follows.

### *Analysis of combination effects*

To analyze why the combinations performed well, we have calculated how many exons are commonly predicted in a test set and their probability of being accurate depending on the commonly predicted program set. It was generally observed that commonly predicted exons are more likely to be actual exons. There were 146 exons predicted by all four programs; the predicted exons included 143 actual exons (only ~2% of the exons were wrong; this is much lower than the 13% of GENSCAN). On the other hand, when exons are predicted by only one program, the accuracy of such exons is very low (6/109, 2/68, 1/24 and 3/9 for FEXH, GeneParser, GRAIL and GENSCAN). This may be due to the deviation intrinsic to the statistical features, such as coding potentials based on oligomer frequency.

The number of programs which predicted a common exon is indeed informative, and a procedure of averaging *Pscore* worked well by this fact. However, combination by method 1 failed to improve accuracy because it takes only commonly predicted regions so that the prediction gives very low sensitivity, as one would expect.

### *Features of combination methods 2 and 3*

There was a common tendency for combinations with/without GENSCAN. Method 3 is better than method 2 in AC, SN and SP (Table 1 and 2). On the other hand, method 3 is worse than method 2 in ME and WE (Figures 5 and 6). We examined several examples which result in different WE or ME between methods 2 and 3. For all the cases, the differences in WE and ME were due to the longer predicted exons in method 2. All the exons which make differences were predicted by both method 2 and method 3. Method 2 is OR based (bases were included to exon if a program called the base coding) so that predicted exons tend to be longer than corresponding exons by method 3. The longer predicted exons overlapped with a true exon or two. On the other hand, in method 3, corresponding predicted exons did not overlap with any true exons. For some cases, predicted exons still

overlapped with other true exons (the predicted exon in method 2 overlapped with more than one true exon).

Boundary predictions in method 2 were worse than in method 3. This is reasonable because longer exons tend to be produced without a quality check of boundaries in method 2.

Therefore, method 3 is more suitable for prediction of boundaries, and method 2 is more suitable for the prediction of broad coding regions.

### *Violation of rule in method 4*

Method 4, which is based on priority-rule, succeeded to some extent only in the case of combination without GENSCAN. AC for 'FEX + GR M4' and 'FEX + GP3 M4' were 0.698 and 0.697; they were sixth and seventh best of the combinations without GENSCAN. They were better than any single programs. In contrast, the combination with GENSCAN did not perform so well. This is probably because the rule is not always true. In comparison with method 4, methods 2, 3 and 5 were able to make probably correct predictions in each case.

### *Analysis of method 5 with GENSCAN*

The top combination was established by the combination of FEXH and GENSCAN using method 5. We have found that this method has utilized good features of GENSCAN and FEXH. First, boundary prediction of GENSCAN was more reliable than the others. Of the 143 commonly predicted exons, the number of exactly predicted exons (this means that the boundary was predicted accurately) was 82, 70, 90 and 121 for FEXH, GeneParser, GRAIL and GENSCAN, respectively. There was the same tendency for the other combination and for the training set. Second, the accuracy of exon-intron boundaries depends on either the prediction program or boundary type. The boundary prediction of FEXH was more erroneous than the others as a whole (see SN and SP in Table 1), but was relatively accurate when its *Pscore* is >0.8 (Table 3). When GENSCAN have low *Pe* and FEXH have high *Pe*, the combination method 5 tends to adopt correct predictions of FEXH. For example, in the locus HSU41284, FEXH predicted an actual exon and GENSCAN did not predict it. Method 3 discards this exon, while method 5 predicts it. If only FEXH gave false exon predictions, the *Pe* would be so low that those false positives would be discarded in method 5. FEXH presumably complemented GENSCAN's prediction by such a mechanism.

From these observations, if we have another program and if it can, with high accuracy, detect exons which GENSCAN does not, we can expect that the combination of the new one with GENSCAN (or further with others) will give much better prediction.

**Table 3.** Accuracy of the boundary prediction ( $P_e$ ) where  $P_{score}$  values are  $>0.8$  for various boundary types. The left column lists the program names in order of accuracy of boundary prediction (SN + SP) for the training set. The second to fifth columns represent boundary type (sites around initiation codon, acceptor site, donor site, termination codon). The sixth column is unknown type; GRAIL prediction gives no direct information about site type. Despite FEXH being more erroneous than the others as a whole, it was relatively accurate when  $P_{score}$  is high

Program	Initiation	Acceptor	Donor	Termination	All
GENSCAN	0.83	0.80	0.86	0.86	0.83
GRAIL	–	–	–	–	0.64
GP	0.48	0.43	0.56	0.11	0.42
FEXH	0.70	0.71	0.88	0.71	0.72

### Limitation

There is an obvious limitation dependent on the discriminative powers of the prediction programs for combination of predicted results. If no program detects a real exon, it will inevitably be missed. In the test set, ~3% of the coding exons were not detected by any of the programs. Therefore, the lower limit of ME can be estimated. On the contrary, the limit of WE or specificity at the exon level is difficult to discuss, because specificity or WE can be refined by improving the combination method. In other words, if we can cut wrong (but commonly predicted) exons in some way, WE will be reduced, keeping the same level of sensitivity or ME.

### Program selection

The main objective is to examine whether the performance becomes better if we use several gene-finding programs. We must select the gene-finding programs carefully. If we use all the available programs, it takes computational time and human cost.

We considered that it would be better to avoid combining similar programs, such as FGENEH and FEXH. The reason is that combining programs which employ similar methods would show a low increase in total performance.

Further, it is desirable that the selected programs perform well even when they run alone. This is because it is unlikely that a program with poor performance contributes to the combination of better programs. The selection of programs here is based on the performance test by Burset and Guigó (1996) and another test by Burge and Karlin (1997).

We did not select the programs that have strong constraints in prediction. For instance, FGENEH and GeneID have the constraint that the final predicted gene structure starts with a start codon and ends with a stop codon.

Consequently, we selected four gene-finding programs: FEXH, GeneParser3, GENSCAN and GRAIL2 (Version 1.3).

### Meaning of novelty of the data

In this study, we used new DNA sequence data, each of which is not very similar to any of the known protein sequences. Despite this, the combinations of the programs were improved in accuracy. This indicates that the increase in accuracy is not due to the compensation of individually learned data themselves, but the compensation of feature extraction abilities of different programs.

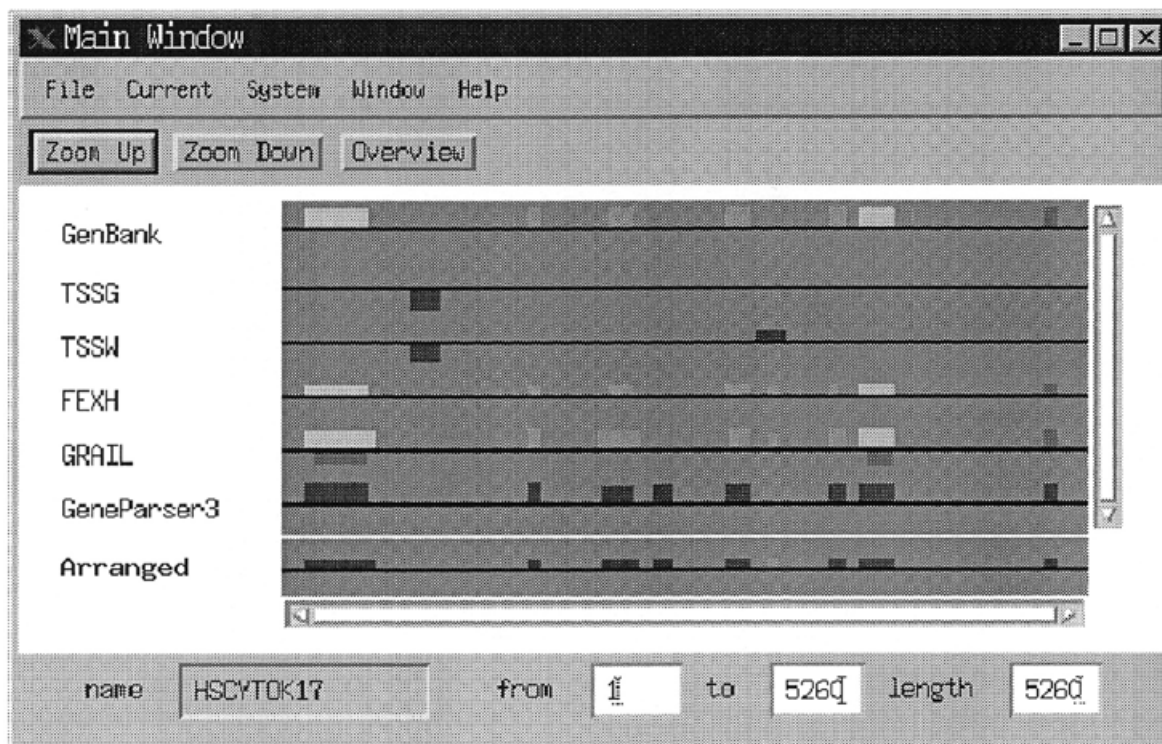
The compensation effect suggests that none of the four programs grasp the (known/unknown) features of a gene in the best way. Even if a program has the best detection technique for a certain feature of a gene, it does not necessarily have the best detection techniques for different features. Hence, there is still room for improvement by just making a new combination of existing techniques employed in those programs.

### Availability

We have developed a client program, called GeneScope, to use the methods described here. GeneScope was implemented in Java language, and run on a user's local machine on any platforms of UNIX, Macintosh and Windows95. The GeneScope sends the server (a httpd program) a query DNA sequence. Then the server (called Shirokane System) receives the query by Common Gateway Interface (CGI) and makes a queue file in the server machine and sends a short E-mail to notify the user of the receipt of the query. After that, a daemon program running in the server machine sends the query to several gene-finding programs in the server machine or in the WWW server of the gene-finding programs. The CGI programs and daemon programs were written in C shell and perl language. After all the programs have analyzed the query sequence, the server returns a short E-mail to the user to notify them that analysis is complete. In fact, the user can get some partial results of the requested analysis by gene-finding programs at any time, as long as the program has completed the analysis at the time. After the user gets the E-mail of completion, they are supposed to open a query file by using GeneScope menu again. The server returns all the results of individual programs and the combined results (only one combination by one method is set in the server). Users can view those results with GUI using GeneScope, as shown in Figure 7. Compiled Class files of GeneScope are available through the URL <http://gf.genome.ad.jp/>.

### Future development

A more sophisticated rule-based approach is possible. One idea is to take account of the advantages of each program. For instance, FGENEH performed well for especially low %G + C sequences (Burset and Guigó, 1996), while other programs like GRAIL have low accuracy for low %G + C sequences (Lopez *et al.*, 1994; Snyder and Stormo, 1995; Burset and



**Fig. 7.** An example view of GeneScope. The 'Arranged' at the bottom is the result of the OR-method. The Shirokane System server provides the results of other analysis programs, such as promoter prediction tools (TSSG and TSSW) and homology search tool (BLAST).

Guigó, 1996). So if the %G + C of input sequence is low, heavier weight would be given to FGENEH in the combination.

Considering the consistency of the reading frame of continuous exons is another approach. In this case, it is necessary to develop a method to deal with the programs like GeneParser which do not predict no translation frame.

Another possible direction of study is to construct a dynamic method that changes the rule of prediction depending on the features of DNA sequences.

Using promoter prediction programs and homology search program is another way for improved re-analysis. Another approach is to make a new gene-finding system that has original detectors to extract signals, such as splice sites and promoter region, as well as prediction by some programs.

## Conclusions

We demonstrated the effectiveness of the re-analysis of different predictions by multiple gene-finding programs (FEXH, GeneParser3, GENSCAN and GRAIL2). This is a practical and easy way to make more correct annotations from uncharacterized DNA sequences. We observed that even a worse program contributed to the combination with more accurate programs. By the combination method called

HIGHEST or BOUNDARY, ACs showed improvements ranging from 3 to 5% in comparison with the best single gene-finding program. From another viewpoint, the OR-based combination of the four programs (FEXH, GeneParser, GENSCAN and GRAIL) is the most reliable to know whether a candidate exon overlaps with the real exon or not. Our methods can modify the prediction of current programs and can be easily extended to combine other programs. We have developed a client program, called GeneScope, which allows users to use the combination methods and to view the results of individual programs.

## Acknowledgement

This work is partially supported by a Grant-in-Aid for Scientific Research on Priority Areas, 'Genome informatics' from the Ministry of Education, Science, Sports and Culture, Japan.

## References

- Burge, C. and Karlin, S. (1997) Prediction of complete gene structures in human genomic DNA. *J. Mol. Biol.*, **268**, 78–94.
- Burset, M. and Guigó, R. (1996) Evaluation of gene structure prediction programs. *Genomics*, **34**, 353–367.

- Gish, H. (1990) A probabilistic approach to the understanding and training of neural network classifiers. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Albuquerque, New Mexico.
- Harris, N.L. (1997) Genotator: a workbench for sequence annotation. *Genome Res.*, **7**, 754–762.
- Henderson, J., Salzberg, S. and Fasman, K. (1997) Finding genes in DNA with a hidden Markov model. *J. Comp. Biol.*, **4**, 127–141.
- Krogh, A. (1997) Two methods for improving performance of an HMM and their application for gene finding. In *Proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology*. AAAI Press, Menlo Park, CA, pp. 179–186.
- Kulp, D., Haussler, D., Reese, M. and Eeckman, F. (1996) A generalized hidden Markov model for the recognition of human genes in DNA. In *Proceedings of the Fourth International Conference on Intelligent Systems for Molecular Biology*. AAAI Press, Menlo Park, CA, pp. 134–142.
- Kulp, D., Haussler, D., Reese, M. and Eeckman, F. (1997) Integrating database homology in a probabilistic gene structure model. In Altman, R.B., Dunker, A.K., Hunter, L. and Klein, T.E. (eds), *Pacific Symposium on Biocomputing '97*. Hawaii, World Scientific.
- Lopez, R., Larsen, F. and Prydz, H. (1994) Evaluation of the exon predictions of the GRAIL software. *Genomics*, **24**, 133–136.
- Myers, E. and Miller, W. (1988) Optimal alignments in linear space. *Comput. Applic. Biosci.*, **4**, 11–17.
- Salzberg, S.L. (1997) A method for identifying splice sites and translational start sites in eukaryotic mRNA. *Comput. Applic. Biosci.*, **13**, 365–376.
- Smith, R.F., Wiese, B.A., Wojzynski, M.K., Davison, D.B. and Worley, K.C. (1996) BCM Search Launcher—an integrated interface to molecular biology data base search and analysis services available on the world wide web. *Genome Res.*, **6**, 454–462.
- Snyder, E. and Stormo, G. (1993) Identification of coding regions in genomic DNA sequences: an application of dynamic programming and neural networks. *Nucleic Acids Res.*, **21**, 607–613.
- Snyder, E. and Stormo, G. (1995) Identification of protein coding regions in genomic DNA. *J. Mol. Biol.*, **248**, 1–18.
- Solovyev, V.A. and Salamov, A.A. (1997) The Gene-Finder computer tools for analysis of human and model organisms genome sequences. In *Proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology*. AAAI Press, Menlo Park, CA, pp. 294–302.
- Solovyev, V.V., Salamov, A.A. and Lawrence, C.B. (1994) The prediction of human exons by oligonucleotide composition and discriminant analysis of spliceable open reading frames. In *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology*. AAAI Press, Menlo Park, CA, pp. 354–362.
- Solovyev, V.V., Salamov, A.A. and Lawrence, C.B. (1994) Predicting internal exons by oligonucleotide composition and discriminant analysis of spliceable open reading frames. *Nucleic Acids Res.*, **22**, 5156–5163.
- Solovyev, V.V., Salamov, A.A. and Lawrence, C.B. (1995) Identification of human gene structure using linear discriminant functions and dynamic programming. In *Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology*. AAAI Press, Menlo Park, CA, pp. 367–375.
- Thomas, A. and Skolnick, M.H. (1994) A probabilistic model for detecting coding regions in DNA sequences. *IMA J. Math. Appl. Med. Biol.*, **11**, 149–160.
- Uberbacher, E., Xu, Y. and Mural, R. (1996) Discovering and understanding genes in human DNA sequence using GRAIL. *Methods Enzymol.*, **266**, 259–281.
- Xu, Y. and Uberbacher, E.C. (1997) Reference-based gene model prediction on DNA contigs. *J. Comp. Biol.*, **4**, 325–338.
- Xu, Y., Einstein, J., Mural, R., Shah, M. and Uberbacher, E. (1994) An improved system for exon recognition and gene modeling in human DNA sequences. In *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology*. AAAI Press, Menlo Park, CA, pp. 376–384.
- Zhang, M. (1997) Identification of protein coding regions in the human genome based on quadratic discriminant analysis. *Proc. Natl Acad. Sci. USA*, **94**, 565–568.