DIGIT: A Novel Gene Finding Program by Combining Gene-Finders

T. Yada, Y. Totoki, Y. Takaeda, Y. Sakaki, T. Takagi

Pacific Symposium on Biocomputing 8:375-387(2003)

DIGIT: A NOVEL GENE FINDING PROGRAM BY COMBINING GENE-FINDERS

T. YADA, T. TAKAGI

Institute of Medical Science, University of Tokyo
4-6-1, Shirokanedai, Minato-ku, Tokyo 108-8639, Japan

Y. TOTOKI, Y. SAKAKI

Genomic Sciences Center, RIKEN
1-7-22 Suehiro-cho, Tsurumi-ku, Yokohama, Kanagawa 230-0045, Japan

Y. TAKAEDA

Mitsubishi Research Institute, Inc.
2-3-6, Otemachi, Chiyoda-ku, Tokyo 100-8141, Japan

We have developed a general purpose algorithm which finds genes by combining plural existing gene-finders. The algorithm has been implemented into a novel gene-finder named DIGIT. An outline of the algorithm is as follows. First, existing gene-finders are applied to an uncharacterized genomic sequence (input sequence). Next, DIGIT produces all possible exons from the results of gene-finders, and assigns them their exon types, reading frames and exon scores. Finally, DIGIT searches a set of exons whose additive score is maximized under their reading frame constraints. Bayesian procedure and a hidden Markov model are used to infer exon scores and search the exon set, respectively. We have designed DIGIT so as to combine the results of FGENESH, GENSCAN and HMMgene, and have assessed its prediction accuracy by using recently compiled benchmark data sets. For all data sets, DIGIT successfully discarded many false-positive exons predicted by individual gene-finders and yielded remarkable improvements in sensitivity and specificity at the gene level compared with the best gene level accuracies achieved by any single gene-finder.

1 Introduction

Draft sequences corresponding to approximately 90% of the human genome have been produced [1], and interest in exhaustive gene finding in the genome has increased enormously. Over the last decade, many gene finding programs (gene-finders) for the human genome have been developed, but none of them is entirely reliable [2].

Gene-finders may be categorized into two classes: empirical gene-finders and *ab initio* gene-finders. Empirical gene-finders, which are also called 'sequence similarity-based gene-finders', detect genes by aligning known cDNA and protein sequences onto uncharacterized genomic sequences [a]. The remarkable feature of empirical gene-finders is their high specificity, that is, genes exist with high probability in genomic regions which these gene-finders detect. However, empirical gene-finders can detect only a limited number of genes (low sensitivity) because it is extremely difficult experimentally to collect mRNAs of all genes. On the other hand, *ab initio* gene-finders do not utilize sequence similarity and rely on intrinsic gene measures such as coding potential and splice signals. The remarkable feature of *ab initio* gene-finders is their high sensitivity, that is, these gene-finders are capable of detecting almost all genes. However, *ab initio* gene-finders also predict many false-positive genes (low specificity) because known gene measures are insufficient to distinguish true positives from false positives.

Several attempts to complete the human gene catalogue have been launched since the production of the draft sequences. Among them, Ensembl (`http://www.ensembl.org/`) is the most popular project. Ensembl adopts a conservative gene annotation protocol which mainly uses empirical gene-finders. Therefore, the gene catalogue compiled by Ensembl contains only a small number of false-positive genes but misses a large number of unknown genes.

Ab initio gene-finders are essential for the completion of the human gene catalogue. Hence, an improvement in their specificity has become an important problem. To address this problem, hybrid gene-finders which combine coding potential and splice signals (*ab initio* approach) with similarity to known gene sequences (empirical approach) have been recently developed [3,4]. On average, hybrid gene-finders enable us to improve the specificity of *ab initio* gene-finders. However, their specificity drops to levels comparable with that of *ab initio* gene-finders when remote homologous genes or no homologous genes are available.

Ab initio gene-finders are immensely powerful tools to find genes when there are no known homologues. It is for these cases where specificity should be improved. The sequencing project of human chromosome 21 addressed this problem by using plural *ab initio* gene-finders, namely, only genome regions which were simultaneously detected by more than one of them were chosen as exons [5]. Although this heuristic approach considerably improves the specificity of *ab initio* gene-finders, the following problems have newly arisen. When we assemble such exons into a gene, the frame consistencies between adjoining exons are not always ensured. Moreover, in the case when all exon scores given by plural gene-finders are low, it is questionable as to

[a] Gene-finders based on genome-genome comparisons may also be categorized into this class.

Table 1. Patterns of exons predicted by two gene-finders, where X and Y are the 5' and the 3' ends of actual exons, respectively. For example, 'Case 1' is the case where gene-finders 1 and 2 predict the same genomic region as an exon and this region corresponds exactly to an actual exon.

	Exons predicted by Gene-finder 1 ($x_1 < y_1$)		Exons predicted by Gene-finder 2 ($x_2 < y_2$)	
Case	5' end (x_1)	3'end (y_1)	5' end (x_2)	3' end (y_2)
1	$x_1 = X$	$y_1 = Y$	$x_2 = X$	$y_2 = Y$
2	$x_1 = X$	$y_1 = Y$	$x_2 = X$	$y_2 \neq Y$
3	$x_1 = X$	$y_1 = Y$	$x_2 \neq X$	$y_2 = Y$
4	$x_1 = X$	$y_1 \neq Y$	$x_2 = X$	$y_2 = Y$
5	$x_1 \neq X$	$y_1 = Y$	$x_2 = X$	$y_2 = Y$
^{*1}6	$x_1 = X$	$X < y_1 < Y$	$X < x_2 < Y$	$y_2 = Y$
^{*2}7	$X < x_1 < Y$	$y_1 = Y$	$x_2 = X$	$X < y_2 < Y$
8	$x_1 = X$	$y_1 > Y$	$x_2 < X$	$y_2 = Y$
9	$x_1 < X$	$y_1 = Y$	$x_2 = X$	$y_2 > Y$
10	$x_1 = X$	$X < y_1 < Y$	$x_2 < X$	$y_2 = Y$
	$x_1 = X$	$y_1 > Y$	$X < x_2 < Y$	$y_2 = Y$
11	$x_1 < X$	$y_1 = Y$	$x_2 = X$	$X < y_2 < Y$
	$X < x_1 < Y$	$y_1 = Y$	$x_2 = X$	$y_2 > Y$
12	$x_1 = X$	$y_1 = Y$	$x_2 \neq X$	$y_2 \neq Y$
13	$x_1 \neq X$	$y_1 \neq Y$	$x_2 = X$	$y_2 = Y$

^{*1} where $y_1 > x_2$. ^{*2} where $x_1 < y_2$.

whether we should choose it or not. Independently of the above sequencing project, Murakami & Takagi have reported that different *ab initio* gene-finders will often correctly predict different exons, suggesting that they could complement one another, yielding better predictions [6]. However, they have not proposed an algorithm which address the above problems.

We present here a general purpose algorithm which finds genes by combining plural existing gene-finders. The algorithm addresses the two problems stated above by applying the frameworks of hidden Markov model (HMM) [7] and Bayesian procedure [8], namely, an HMM is used to ensure the frame consistency between exons within a gene, and Bayesian procedure is used to take into account the exon scores given by the gene-finders. A remarkable feature of the algorithm is that it can combine most gene-finders in a systematic manner. The algorithm has been implemented into a novel gene-finder named DIGIT (Digit Integrates Gene Identification Tools). Currently, DIGIT has been designed so as to combine plural *ab initio* gene-finders. Our extensive testing has clearly shown that DIGIT can accurately identify genes with a low rate of false-positives. In this paper, we report the prediction accuracy of DIGIT as well as presenting the detailed algorithm behind DIGIT.

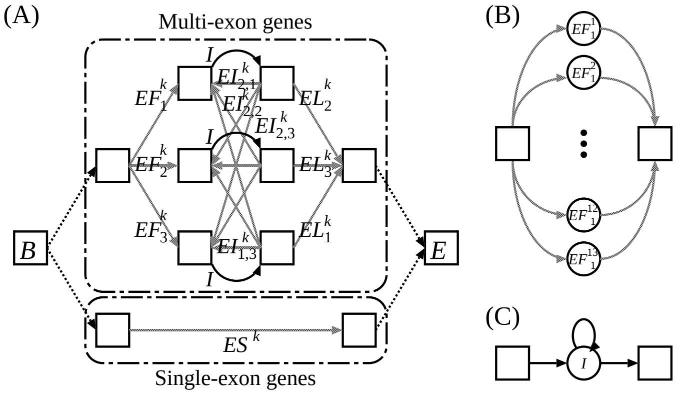

Figure 1. (A) the basic architecture of the HMM which DIGIT employs. EF_i^k indicates a first coding exon which ends at the i-th position of a codon. $EI_{i,j}^k$ indicates an internal coding exon which starts at at the i-th position of a codon and ends at the j-th position of a codon. EL_i^k indicates a last coding exon which starts the i-th position of a codon. ES^k and I indicate single exon genes and introns, respectively. Squares are null states. B and E are the begin and the end states, respectively. This topology includes frame constraints within genes. For example, a downstream exon which adjoins with the first exon ending at the first base of codon (EF_1^k) through an intron always starts at the second base of codon ($EI_{2,j}^k$ or EL_2^k). (B) a detailed architecture corresponding to EF_1^k in the left figure, where k indicates a pattern of exons predicted by gene-finders. There are thirteen possible patterns in the case of combining two gene-finders. Each of them are listed in Table 1. Note that these states are able to emit patterns of the predicted exons. (C) a detailed architecture corresponding to I in the left figure. This state is able to emit four bases i.e. A, C, G and T.

2 Methods

We describe below the computer algorithm which is employed by DIGIT. Currently, DIGIT combines two kinds of gene-finders. However, we can easily extend the algorithm so as to combine more than two kinds of gene-finders. In such case, although the number of model parameters which we should estimate increases exponentially, we can apply the same technique described in 'Parameter estimation' in order to reduce the number.

2.1 HMM architecture

Since DIGIT employs an HMM whose architecture includes frame constraints in gene structure (Figure 1 A), the parsing of genome sequences by DIGIT

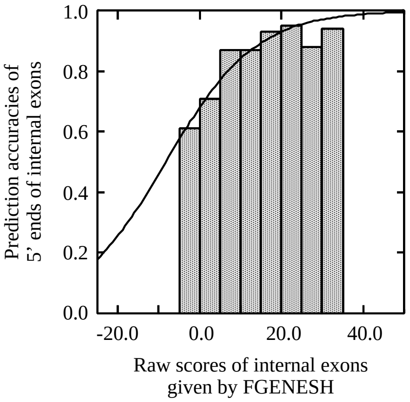

Figure 2. The logistic function which transforms raw exon scores given by gene-finders into probabilities. The solid curve is a logistic function which enables us to transform raw exon scores given by FGENESH into the prediction accuracies of 5' ends of internal coding exons. This function is adjusted to the correlation, shown here by the bar graph, between the exon scores and the prediction accuracies observed in the training data.

necessarily results in finding of frame consistent genes. This feature seems to be very similar to that of Genie [9]; however, DIGIT employs a more complicated architecture. The state transitions corresponding to each exon type in Figure 1 A are divided into thirteen sets of state transitions and states in fact (Figure 1 B), each of which corresponds to a pattern of exons predicted by the gene-finders. In other words, these states emit patterns of the predicted exons. According to the positional relationship between both ends of the predicted exons and the actual exons, the patterns can be classified into thirteen cases (Table 1). Note that the number of cases exponentially increases as the number of gene-finders increases. That is, the number of model parameters which we should estimate increases exponentially in such cases. As for introns, states which emit four bases and possess in/out- and self-transitions are prepared (Figure 1 C).

In addition, several state transitions, i.e. from the begin state to $EI^k_{i,j}$ and EL^k_i, and from those to the end state, are defined in order to find 'partial' genes. For example, transitions from the begin state to $EI^k_{i,j}$ enables us to find partial genes starting in internal coding exons. These transitions are not shown in Figure 1 for reason of simplicity.

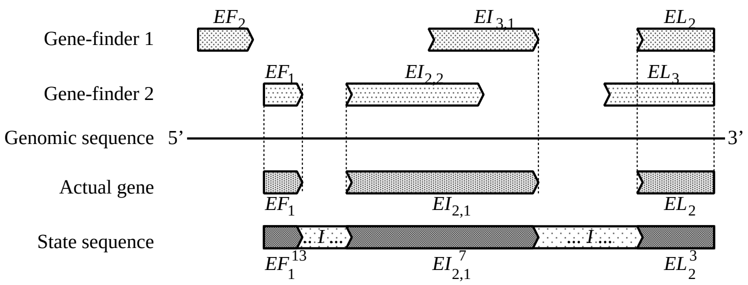

Figure 3. Identification of state sequences in the HMM shown in Figure 1. When we apply gene-finders to genomic sequences of the training data, we obtain sets of predicted exons with their types and frames. Since the training data contains information concerning the types and frames of actual exons, we can easily identify state sequences in the HMM by refering to Figure 1 and Table 1. For example, the actual exon EF_1 in this figure corresponds to 'Case 13' in Table 1 because gene-finder 2 exactly predicted this exon but gene-finder 1 missed.

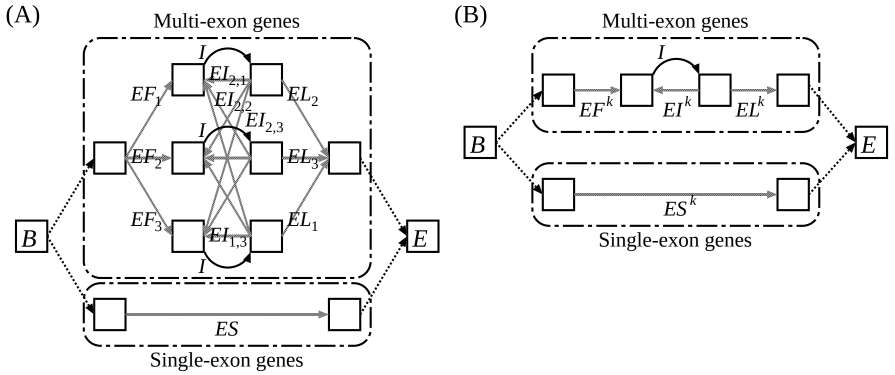

Figure 4. Two types of HMMs introduced in order to reduce the number of parameters which need to be estimated. (A) the HMM where information for the pattern of predicted exons is removed from the HMM shown in Figure 1. (B) the HMM where frame constraints are removed from the HMM shown in Figure 1.

2.2 Parameter estimation

Emission probabilities of exon states must reflect probabilities that a genomic region $[X \dots Y]$ is an actual exon for each case in Table 1. We assume that such probabilities can be calculated by $\alpha\beta$, where α and β are probabilities that X and Y are actual 5' and 3' ends of exons, respectively. Although α and β should be calculated from exon scores given by the gene-finders, these scores are usually given as log-odds measure. Thus, we have to begin with the transformation of raw exon scores into probabilities. First, we plot graphs representing correlations between the exon scores given by each gene-finder and the prediction accuracies of 5' and 3' ends of exons based on training data (Figure 2). Next, we adjust logistic functions to the plots by using a non-linear least square method, namely, the Marquardt method [10]. These logistic functions enable us to transform raw exon scores given by gene-finders into prediction accuracies of 5' and 3' ends of exons. Finally, we calculate probabilities that a genomic region $[X \dots Y]$ is an actual exon for each case in Table 1. Bayesian procedure tells us that these probabilities can be calculated in the following manner. For example, the probability for 'Case 1' is given by

$$\frac{\alpha_1\alpha_2}{\alpha_1\alpha_2 + (1-\alpha_1)(1-\alpha_2)} \times \frac{\beta_1\beta_2}{\beta_1\beta_2 + (1-\beta_1)(1-\beta_2)}$$

where α_1 and β_1 are the prediction accuracies of 5' and 3' ends of exons by 'Gene-Finder 1', respectively. α_2 and β_2 are the prediction accuracies of 5' and 3' ends of exons by 'Gene-Finder 2', respectively. The first term corresponds to α, and the second term corresponds to β. $\alpha_1\alpha_2$ indicates the probability that X is the 5' end of an actual exon. $\beta_1\beta_2$ indicates the probability that Y is the 3' end of an actual exon. $(1-\alpha_1)(1-\alpha_2)$ indicates the probability that X is not the 5' end of an actual exon. $(1-\beta_1)(1-\beta_2)$ indicates the probability that Y is not the 3' end of an actual exon. We assume that the background probability is given by setting $\alpha_1 = \alpha_2 = \beta_1 = \beta_2 = 0.5$ in the equation above. We can calculate the probabilities for other cases in a similar way and assign them to the emission probabilities of exon states.

Emission and transition probabilities related to introns (Figure 1 C) are set so as not to contribute to the Viterbi score [7]. That is, emission probabilities of all bases are 0.25, and the corresponding background probabilities are also 0.25. All transition probabilities are 1.0.

Transition probabilities to exon states must reflect the frequency for each case in Table 1 as observed in the training data. A straightforward way for the estimation of transition probabilities is to count the number of degrees for each case observed in the training data. However, since the HMM shown in Figure 1 contains a large number of transition parameters to exon states, an overfitting problem might occur. Thus, we adopted the following strategy in order to reduce the number of parameters which need to be estimated. First, we apply the gene-finders to genomic sequences of the training data and identify state sequences of the HMM for the genomic sequences (Figure 3). Next,

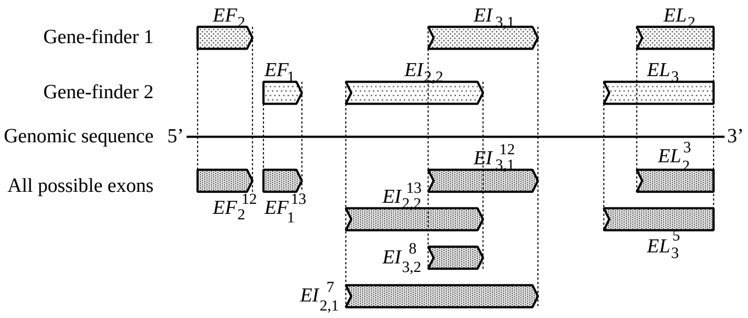

Figure 5. All possible exons produced from the prediction results of gene-finders and the identification of their state labels. This figure shows the case when exons which overlap each other have the same reading frame, i.e. $EI_{3,1}$ and $EI_{2,2}$ have the same frame. If $EI_{3,1}$ and $EI_{2,2}$ have different reading frames, we do not generate possible exons $EI^8_{3,2}$ and $EI^7_{2,1}$ because their frames can not be determined. When two overlapping exons have the same reading frame but their exon types are different, we do not generate exons for their intersection and union regions.

we prepare two types of HMMs (Figure 4); one is an HMM where information for the pattern of predicted exons is removed from the HMM shown in Figure 1, and the other is an HMM where frame constraints are removed from the HMM shown in Figure 1. For each HMM, we count the number of degrees corresponding to the transition probabilities based on the state sequences of the training data, add simple pseudocounts [7] to these numbers, and then calculate the frequencies from these numbers. Finally, we obtain the transition probabilities for the exon states in Figure 1 by assuming that these probabilities can be calculated by multiplying the corresponding parameters of the two HMMs. For example, a transition probability for EF_i^k can be calculated by multiplying the transition probability for EF_i by that of EF^k. This procedure enables us to reduce the number of transition probabilities which need to be estimated. For example, although the HMM contains 39 (3×13) transition probabilities for EF_i^k, we only have to estimate 16 ($3 + 13$) probabilities by applying the above procedure.

2.3 Prediction algorithm

Since gene-finders tend to perform better if provided with a genomic region containing only a single gene and its immediate neighborhood [2], the prediction algorithm should begin with the extraction of such genomic regions. First, we apply an *ab initio* gene-finder FGENESH [11] to an uncharacterized genomic sequence and extract genomic regions as candidate gene regions, each of which includes a single predicted gene and the surrounding regions (up to 1,000 bps on each side). Our experiments preliminary to this study have shown that FGENESH is capable of detecting gene boundaries with high reliability (data not shown). Second, we apply gene-finders to the candidate gene regions. This second analysis often leads to different results from the first analysis [2]. Third, we generate all possible exons and identify their state labels in the HMM shown in Figure 1 (Figure 5). Fourth, we calculate emission probabilities for all possible exons based on the Bayesian procedure explained in 'Parameter estimation'. The state labels are used to calculate these probabilities. Last, we parse the candidate gene regions by using the HMM shown in Figure 1. For the parsing algorithm, semi-global search algorithm [7], which aligns an entire HMM to partial genomic sequences, is used. The local search algorithm enables us to find plural genes within a candidate gene region. The search threshold is set so as to maximize the average value of sensitivity and specificity of gene level in the training data (see below).

3 Data

In order to evaluate the prediction accuracy of DIGIT, we used three various data sets each of which has different characteristics. The first one is the data set, HMR195, compiled by Rogic *et al.* [12]. They attempted to create a data set which did not have many overlaps with the training sets used for the gene-finders. Then, they selected only sequences entered in GenBank after Aug., 1997. Since all sequences in HMR195 are relatively short and each of them contains exactly one complete gene, it can be said that HMR195 is a typical benchmark set for gene-finders. HMR195 consists of 195 human/murine DNA sequences, 43 of them contain single-exon genes, and 152 of them contain multi-exon genes. The average sequence length is 7,096 bps, the proportion of coding sequence is 14%, of intronic sequence is 46% and of intergenic sequence is 40%. Note that these sequences are highly gene dense. The second data set, Gen178, was compiled by Guigò *et al.* [2]. They attempted to overcome the lack of well-annotated large genomic sequences, by preparing a set of well-annotated DNA sequences each of which contains exactly one complete gene and embedding them in simulated intergenic DNA. Gen178 is a more realistic benchmark set for gene-finders, that is, all sequences are fully long, some of them contain several genes including partial ones, and some of them do not contain any genes. Gen178 consists of 43 semiartificial genomic sequences and contains 178 human genes, 40 are single-exon genes, and 138 are multi-exon

Table 2. Exon- and gene-level prediction accuracies of DIGIT and three *ab initio* gene-finders on three data sets. Statistics on annotation are also summarized, i.e. numbers of actual exons and actual genes in each data set. Exon level sensitivity (Sn) is the percent of annotated exons predicted correctly. Exon level specificity (Sp) is the percent of predicted exons which are correct. Gene level Sn is the percent of annotated genes predicted correctly. Gene level Sp is the percent of predicted genes which are correct.

Data	Program	Exon level			Gene level		
		#	Sn	Sp	#	Sn	Sp
HMR195	Annotated	948			195		
	DIGIT	899	0.795	0.838	188	0.507	0.526
	FGENESH	1006	0.819	0.772	222	0.476	0.418
	GENSCAN	1011	0.773	0.725	221	0.364	0.321
	HMMgene	1181	0.754	0.605	299	0.446	0.290
Gen178	Annotated	900			178		
	DIGIT	911	0.786	0.777	166	0.449	0.481
	FGENESH	1055	0.771	0.657	187	0.376	0.358
	GENSCAN	1332	0.665	0.449	222	0.185	0.148
	HMMgene	1711	0.696	0.366	362	0.258	0.127
Chr22	Annotated	3660			522		
	DIGIT	4513	0.695	0.563	654	0.123	0.098
	FGENESH	5734	0.708	0.452	866	0.115	0.069
	GENSCAN	6588	0.707	0.393	803	0.067	0.044
	HMMgene	6840	0.629	0.336	1508	0.090	0.031

genes. The average sequence length is 177,160 bps, the proportion of coding sequence is 2.3%, of intronic sequence is 6.3% and of intergenic sequence is 91.4%. Note that these statistics are highly similar to ones observed in real genomic sequences. The third data set is the finished human chromosome 22 (Chr 22) sequences (May 19, 2000, version) [13] with the annotation provided by the Sanger Institute (March 6, 2001, version). It is one of the few well-annotated large genomic sequence sets for human. Chr22 consists of 12 contiguous segments covering 33.4 million bps. It consists of 504 multi-exon genes and 18 single-exon genes. The proportion of coding sequence is 1.7%, of intronic sequence is 32.3% and of intergenic sequence is 66.0%. Note that some uncharacterized genes may still remain in these sequences.

4 Results

In this study, we have constructed DIGIT so as to combine plural *ab initio* gene-finders. More explicitly, DIGIT has been designed so as to combine the first coding exons predicted by FGENESH and HMMgene [14], the internal and last coding exons predicted by FGENESH and GENSCAN [15], and single exon genes predicted by FGENESH and HMMgene. Among *ab initio* gene-finders, these programs are highly reliable and widely used. The selection and the combination of gene-finders was determined on the basis of our experiments preliminary to this study. We have evaluated the prediction accuracy of nearly twenty *ab initio* gene-fingers by using various benchmark data. Our experiments showed that (1) FGENESH had the best prediction accuracy on average, (2) the second best was GENSCAN, and (3) HMMgene was good at predicting coding regions including start codons compared with other gene-finders.

We performed three tests to evaluate the prediction accuracy of DIGIT. The first test was a 10-fold cross validation test using the HMR195 data set. The second and third tests were open tests using the Gen178 and Chr22 data sets, respectively. In the open tests, the model parameters of DIGIT were optimized by using the HMR195 data set. In all tests, RepeatMasker was used to mask the interspersed repeats (replace interspersed repeats with Ns) in the sequence data in order to reduce the number of false positive exons detected by *ab initio* gene-finders. All the *ab initio* gene-finders including DIGIT were applied to the masked sequences.

The prediction accuracy of DIGIT with three *ab initio* gene-finders on the three data sets is summarized in Table 2. For all data sets, although the exon-level sensitivity of DIGIT was roughly comparable to that of the other gene-finders, the exon-level specificity of DIGIT was remarkably higher than that of the other gene-finders. Remarkably, DIGIT outperformed the other gene-finders for sensitivity and specificity of gene level in all data sets. Note that the numbers of exons and genes detected by DIGIT decreased drastically in comparison with the other gene-finders. However, these numbers are closer to the actual numbers of the annotated exons and genes.

5 Discussion

Table 2 clearly shows that DIGIT significantly improves exon-level specificity in comparison with the other *ab initio* gene-finders. This indicates that DIGIT successfully discards many false positive exons predicted by the gene-finders. In such cases, although actual exons are also frequently discarded, DIGIT prevents the sensitivity from dropping by generating all possible exons and applying Bayesian procedure to the inference of exon scores. That is, in addition to candidate exons detected by the other gene-finders, DIGIT produces intersection and union regions of overlapping exons as novel candidate exons, and then, Bayesian procedure tells us the likelihood of all possible exons on the basis of positional relationships between these exons and their scores given

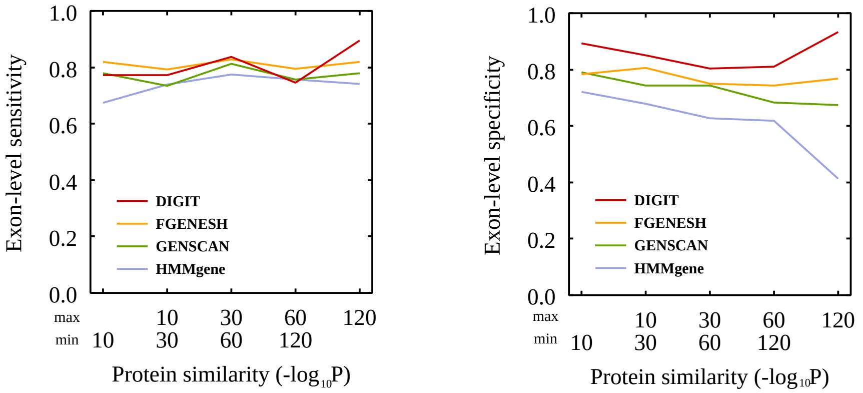

Figure 6. Exon-level accuracies of DIGIT and three *ab initio* gene-finders on the HMR195 data set as a function of protein similarity. Left, exon-level sensitivity. Right, exon-level specificity. These measures were calculated for subsets of the HMR195 data set and grouped according to the level of BLASTX similarity between HMR195 entries and the Swiss-Prot database `http://www.expasy.ch/sprot/sprot-top.html`. The definitions of the subsets and number of genes per subset were as follows: $P > 10^{-10}$ (29); $10^{-10} > P > 10^{-30}$ (48); $10^{-30} > P > 10^{-60}$ (45); $10^{-60} > P > 10^{-120}$ (36); and $10^{-120} > P$ (37).

by the other gene-finders. This significant improvement of exon-level specificity naturally leads DIGIT to remarkable improvements in sensitivity and specificity at the gene level as compared with the best gene-level accuracies achieved by any single *ab initio* gene-finder.

One remarkable feature of the algorithm which DIGIT employs is that it can combine most gene-finders in a systematic manner. Although we introduced here the algorithm as it combines two gene-finders, we can easily extend it so as to combine more than two gene-finders (see 'Methods'). Moreover, the algorithm can combine any gene-finder which gives predicted exons their types, reading frames and scores. Therefore, it enables us to combine not only *ab initio* gene-finders but also empirical gene-finders. When we combined an *ab initio* gene-finder and an empirical gene-finder by using the algorithm, the exon-level sensitivity was significantly improved instead of the exon-level specificity (data not shown). This is due to differences in genes which *ab initio* and empirical gene-finders can detect. On the other hand, the main limitation of the algorithm is that it cannot predict exons whose boundaries and regions are not detected by gene-finders. We can however reduce this limitation by extending DIGIT so as to combine more than two gene-finders.

We strongly believe that DIGIT is an essential program for exhaustive gene finding in the human genome. Figure 6 shows the exon-level accuracy of DIGIT on the HMR195 data set as a function of protein similarity. The prediction accuracy of DIGIT remains high (the sensitivity is comparable to that of other gene-finders, and the specificity is higher than that of other gene-finders) even if only remote homologous genes are available ($P > 10^{-10}$). For $P > 10^{-10}$, it is known that the prediction accuracy of empirical gene-finders drops remarkably [2], and that the prediction accuracy of hybrid gene-finders drops to levels comparable with that of *ab initio* gene-finders [3]. This strongly suggests that DIGIT is needed to help complete the human gene catalogue, since DIGIT can find genes with high accuracy which empirical and hybrid gene-finders almost never find. DIGIT is made available upon request to the authors.

Acknowledgments

We thank Kiyoshi Asai and Kyosuke Nagasaka for their helpful discussions and valuable advice; Todd Taylor for critically reading the manuscript. This work was partly supported by a Grant-in-Aid for Scientific Research on Priority Areas (C) "*Genome Information!Science*" from The Ministry of Education, Culture, Sports, Science and Technology of Japan.

References

1. International Human Genome Sequencing Consortium, *Nature* **409**, 860 (2001).
2. R. Guigò *et al.*, *Genome Res.* **10**, 1631 (2000).
3. R.-F. Yeh *et al.*, *Genome Res.* **11**, 803, (2001).
4. I. Korf *et al.*, *Bioinformatics* **17**, S140, (2001).
5. The Chromosome 21 Mapping and Sequencing Consortium, *Nature* **405**, 311, (2000).
6. K. Murakami and T. Takagi. *Bioinformatics* **14**, 665, (1998).
7. R. Durbin *et al.*, *Biological sequence analysis: Probabilistic models of proteins and nucleic acids*, (Cambridge University Press, Cambridge, 1998).
8. A. Gelman *et al.*, *Bayesian Data Analysis*, (Champman & Hall/CRC, Boca Raton, FL, 1995).
9. D. Kulp *et al.*, *Proc. of the ISMB* **4**, 134, (1996).
10. W.H. Press *et al.*, *Numerical Recipes in C: The art of scientific computing*, (William H. Press, Cambridge, MA, 1993).
11. A.A. Salamov and V.V. Solovyev, *Genome Res.* **10**10, 516, (2000).
12. S. Rogic *et al.*, *Genome Res.* **11**, 817, (2001).
13. I. Dunham *et al.*, *Nature* **402** 489, (1999).
14. A. Krogh, *Proc. of the ISMB* **5**, 179, (1997).
15. C. Burge and S. Karlin, *J. Mol. Biol.* **268**, 78, (1997).