Nucleic Acids Research, 2002, Vol. 30 No. 19 **4103–4117**

SURVEY AND SUMMARY

Current methods of gene prediction, their strengths and weaknesses

Catherine Mathé*, Marie-France Sagot[1], Thomas Schiex[2] and Pierre Rouzé[3]

Institut de Pharmacologie et Biologie Structurale, UMR 5089, 205 route de Narbonne, F-31077 Toulouse Cedex, France, [1]INRIA Rhône-Alpes, UMR 5558 Biométrie et Biologie Évolutive, Université Claude Bernard, Lyon I, 43 Boulevard du 11 Novembre, F-69622 Villeurbanne Cedex, France, [2]INRA Toulouse, Département de Biométrie et Intelligence Artificielle, Chemin de Borde Rouge, BP 27, F-31326 Castanet-Tolosan Cedex, France and [3]Laboratoire Associé de l'INRA (France), Universiteit Gent, Ledeganckstraat 35, B-9000 Gent, Belgium

Received May 28, 2002; Revised and Accepted August 7, 2002

ABSTRACT

While the genomes of many organisms have been sequenced over the last few years, transforming such raw sequence data into knowledge remains a hard task. A great number of prediction programs have been developed that try to address one part of this problem, which consists of locating the genes along a genome. This paper reviews the existing approaches to predicting genes in eukaryotic genomes and underlines their intrinsic advantages and limitations. The main mathematical models and computational algorithms adopted are also briefly described and the resulting software classified according to both the method and the type of evidence used. Finally, the several difficulties and pitfalls encountered by the programs are detailed, showing that improvements are needed and that new directions must be considered.

INTRODUCTION

The 21st century has seen the announcement of the draft version of the human genome sequence (1). Model organisms have been sequenced in both the plant (2,3) and animal kingdoms (4) and, currently, more than 60 eukaryotic genome sequencing projects are underway (see http://igweb.integratedgenomics.com/GOLD/).

However, biological interpretation, i.e. annotation, is not keeping pace with this avalanche of raw sequence data. There is still a real need for accurate and fast tools to analyze these sequences and, especially, to find genes and determine their functions. Unfortunately, finding genes in a genomic sequence (in this paper, we will only discuss genes encoding proteins) is far from being a trivial problem (5). The widely used and recognized approach for genome annotation (6) consists of employing, first, homology methods, also called 'extrinsic methods' (7), and, second, gene prediction methods or 'intrinsic methods' (8,9). Indeed, it seems that only approximately half of the genes can be found by homology to other known genes or proteins (although this percentage is of course increasing as more genomes get sequenced). In order to determine the 50% of remaining genes, the only solution is to turn to predictive methods and to elaborate fast, accurate and reliable gene finders (10).

Many gene prediction programs are currently publicly available. Most of them are referenced in the Web site maintained by W. Li (http://linkage.rockefeller.edu/wli/gene/). Several reviews have also been written on this topic, among which the most recent are Claverie (11), Guigó (12), Haussler (13) and Burge and Karlin (14). In the last 15 years, a competitive spirit has appeared and an ever increasing number of programs are thus being created, updated and adapted from one organism to another. While such a scientific rush helps to improve the quality of existing programs, it is also confusing for the users, who wonder what makes the programs different, which one they should use in which situation and for each what the prediction confidence is. This last question was addressed, in the case of vertebrate genomes, by Burset and Guigó (15) and recently by Rogic *et al*. (16), and in the case of a plant model genome (of *Arabidopsis thaliana*) by Pavy *et al*. (17). Finally, users also wonder whether current programs can answer all their questions (11). The answer is most probably no, and will remain no as it is unrealistic to imagine that such complex biological processes as transcription and translation can be explained merely by looking at the DNA sequence.

In this review, we start by giving a general overview of the classical gene finding approaches, without going too deeply into the mathematical methods and algorithms themselves (the interested reader is invited to have a look at the original papers). For the sake of exposition, the approaches are divided into finding the evidence for a gene and combining the evidence in order to better predict the structure of a gene. We end by focusing on the many remaining problems presented by the currently available gene prediction methods.

FINDING THE EVIDENCE

We consider in this paper the problem of finding genes coding for a protein sequence in eukaryotes only. The problem of finding genes in prokaryotes presents different types of

*To whom correspondence should be addressed. Tel: +33 5 61 17 59 53; Fax: +33 5 61 17 59 94; Email: catherine.mathe@ipbs.fr

difficulties (there are no introns and the intergenic regions are small, but genes may often overlap each other and the translation starts are difficult to predict correctly). Functionally, a eukaryotic gene can be defined as being composed of a transcribed region and of regions that *cis*-regulate the gene expression, such as the promoter region which controls both the site and the extent of transcription and is mostly found in the 5′ part of the gene. The currently existing gene prediction software look only for the transcribed region of genes, which is then called 'the gene'. We will adopt this definition of a gene in this review; the region between two transcribed regions will be called intergenic. In the current practice, the promoter is seen as sitting in the intergenic region, immediately upstream of the gene and not overlapping with it. This is also a simplification of reality. A gene is further divided into exons and introns, the latter being removed during the splicing mechanism that leads to the mature mRNA. Although some exons (or parts of them) may be non-coding, most gene finding software use the term exon to denote the coding part of the exons only. In this review, however, we will refer to the correct biological definition of exons and explicitly mention when only the coding part of them is concerned. Indeed, in the mature mRNA, the untranslated terminal regions (UTRs) are the non-coding transcribed regions, which are located upstream of the translation initiation (5′-UTR) and downstream (3′-UTR) of the translation stop. They are known to play a role in the post-transcriptional regulation of gene expression, such as the regulation of translation and the control of mRNA decay (18). Inside or at the boundaries of the various genomic regions, specific functional sites (or signals) are documented to be involved in the various levels of protein encoding gene expression, e.g. transcription (transcription factor binding sites and TATA boxes), splicing (donor and acceptor sites and branch points), polyadenylation [poly(A) site], translation (initiation site, generally ATG with exceptions, and stop codons).

Essentially, two different types of information are currently used to try to locate genes in a genomic sequence. (i) Content sensors are measures that try to classify a DNA region into types, e.g. coding versus non-coding. Historically, the existence of a sufficient similarity with a biologically characterized sequence has been the main means of obtaining such a classification. Similarity-based approaches have often been called extrinsic in opposition to others that try to capture some of the intrinsic properties of the coding/non-coding sequences (compositional bias, codon usage, etc). (ii) Signal sensors are measures that try to detect the presence of the functional sites specific to a gene.

Content sensors

Extrinsic content sensors. Extrinsic content sensors simply exploit a sufficient similarity between a genomic sequence region and a protein or DNA sequence present in a database in order to determine whether the region is transcribed and/or coding. The basic tools for detecting sufficient similarity between sequences are local alignment methods ranging from the optimal Smith–Waterman algorithm to fast heuristic approaches such as FASTA (19) and BLAST (20). Besides the fact that some databases may contain information of poor quality (a topic that will be discussed later), and independently of the type of similarities that are considered, the obvious weakness of such extrinsic approaches is that nothing will be found if the database does not contain a sufficiently similar sequence. Furthermore, even when a good similarity is found, the limits of the regions of similarity, which should indicate exons, are not always very precise and do not enable an accurate identification of the structure of the gene. Small exons are also easily missed.

Overall, similarities with three different types of sequences may provide information about exon/intron locations. The first and most widely used are protein sequences that can be found in databases such as SwissProt or PIR. It is estimated that almost 50% of the genes can be identified thanks to a sufficient similarity score with a homologous protein sequence. However, even when a good hit is obtained, a complete exact identification of the gene structure can still remain difficult because homologous proteins may not share all of their domains. Furthermore, UTRs cannot be delimited in this way.

The second type of sequences are transcripts, sequenced as cDNAs (a cDNA is a DNA copy of a mRNA) either in the classical way for targeted individual genes with high coverage sequencing of the complete clone or as expressed sequence tags (ESTs), which are one shot sequences from a whole cDNA library. ESTs and 'classical' cDNAs are the most relevant information to establish the structure of a gene, especially if they come from the same source as the genome to be annotated. ESTs provide information that enable the identification of (partial) exons, either coding or non-coding, and give unbiased hints on alternative splicing. However, ESTs give only local and limited information on the gene structure as they only reflect a partial mRNA. Furthermore, the correct attribution of EST sequences to an individual member in a gene family is not a trivial task.

Finally, under the assumption that coding sequences are more conserved than non-coding ones, similarity with genomic DNA can also be a valuable source of information on exon/intron location. Two approaches are possible: intra-genomic comparisons can provide data for multigenic families, apparently representing a large percentage of the existing genes (e.g. 80% for *Arabidopsis*); inter-genomic (cross-species) comparisons can allow the identification of orthologous genes, even without any preliminary knowledge of them. Nevertheless, the similarity may not cover entire coding exons but be limited to the most conserved part of them. Alternatively, it may sometimes extend to introns and/or to the UTRs and promoter elements. This will be the case when genomes are evolutionarily close or when genome duplications are recent events. In both cases, exactly discriminating between coding and non-coding sequences is not an obvious task.

In all cases, an important strength of similarity-based approaches is that predictions rely on accumulated pre-existing biological data (with the caveat mentioned later of possible poor database quality). They should thus produce biologically relevant predictions (even if only partial). Another important point is that a single match is enough to detect the presence of a gene, even if it is not canonical. An interesting study on the human genome concluded that EST databases represent an effective general purpose probe for gene detection in the human genome (21). The authors of the study stress, however, the fact that genes expressed either

under very specific conditions or at a low level are generally not represented in EST databases.

Intrinsic content sensors. Originally, intrinsic content sensors were defined for prokaryotic genomes. In such genomes, only two types of regions are usually considered: the regions that code for a protein and will be translated, and intergenic regions. Since coding regions will be translated, they are characterized by the fact that three successive bases in the correct frame define a codon which, using the genetic code rules, will be translated into a specific amino acid in the final protein.

In prokaryotic sequences, genes define (long) uninterrupted coding regions that must not contain stop codons. Therefore, the simplest approach for finding potential coding sequences is to look for sufficiently long open reading frames (ORFs), defined as sequences not containing stops, i.e. as sequences between a start and a stop codon. In eukaryotic sequences, however, the translated regions may be very short and the absence of stop codons becomes meaningless (22).

Several other measures have therefore been defined that try to more finely characterize the fact that a sequence is 'coding' for a protein: nucleotide composition and especially (G+C) content (introns being more A/T-rich than exons, especially in plants), codon composition, hexamer frequency, base occurrence periodicity, etc. Among the large variety of coding measures that have been tested, hexamer usage (i.e. usage of 6 nt long words) was shown in 1992 to be the most discriminative variable between coding and non-coding sequences (23). This characteristic has been widely exploited by a large number of algorithms through different methods.

Thus, hexamer frequency is one of the main variables used in SORFIND (24), Genview2 (25), the quadratic discriminant analysis approach of MZEF (26) and the neural network procedure of GeneParser (27). This last program combines the use of hexamer frequency with local compositional complexity measures estimated on octanucleotide statistics. Such statistics are also efficiently used, among other variables, in the linear discriminant analysis of Solovyev's GeneFinder (28).

More generally, the *k*mer composition of coding sequences is the basis of the now ubiquitous so-called 'three-periodic Markov model' introduced in the pioneering algorithm GeneMark (29). Very briefly, a Markov model is a stochastic model which assumes that the probability of appearance of a given base (A, T, G or C) at a given position depends only on the k previous nucleotides (k is called the order of the Markov model). Such a model is defined by the conditional probabilities $P(X|k$ previous nucleotides), where X = A, T, G or C. In order to build a Markov model, a learning set of sequences on which these probabilities will be estimated is required. Given a sequence and a Markov model, one can then very simply compute the probability that this sequence has been generated according to this model, i.e. the likelihood of the sequence, given the model.

The simplest Markov models are homogeneous zero order Markov models which assume that each base occurs independently with a given frequency. Such simple models are often used for non-coding regions, although it is now frequent to use higher order models to represent introns and intergenic regions as, for instance, in GeneMark, Genscan (30) and EuGène (31). The more complex three-periodic Markov models have been introduced to characterize coding sequences. Coding regions are defined by three Markov models, one for each position inside a codon.

The larger the order of a Markov model, the finer it can characterize dependencies between adjacent nucleotides. However, a model of order k requires a very large number of coding sequences to be reliably estimated. Therefore, most existing gene prediction programs, such as GeneMark and Genscan, usually rely on a three-periodic Markov model of order five (thus exploiting hexamer composition) or less to characterize coding sequences. To cope with these limitations, interpolated Markov models (IMMs) have been introduced in the prokaryotic gene finder Glimmer (32). For each conditional probability, an IMM combines statistics from several Markov models, from order zero to a given order k (typically $k = 8$), according to the information available. These IMMs are now also used in GlimmerM (33), a version dedicated to eukaryotes, and in EuGène. The new version of Glimmer introduces yet another sophistication of Markov models called interpolated context models, which can capture dependencies among 12 adjacent nucleotides (34).

Another type of refinement is often needed in eukaryotic genomes. It consists of estimating several gene models according to the G+C content of the genomic sequence. This is done by Genscan and GeneMark.hmm (35). Indeed, it was shown that differences in gene structure and gene density along some genomes are closely related to their 'isochore' organization (36–38).

In general, most currently existing programs use two types of content sensors: one for coding sequences and one for non-coding sequences, i.e. introns, UTRs and intergenic regions. A few software refine this by using a different model for the different types of non-coding regions (e.g. one model for introns, one for intergenic regions and an optional specific 3′- and 5′-UTR model in EuGène).

Although these methods are considered as 'intrinsic', the fact that the models are built from known sequences will inherently limit the applicability of the methods to sequences that, globally, behave in the same way as the learning set.

Signal sensors

The basic and natural approach to finding a signal that may represent the presence of a functional site is to search for a match with a consensus sequence (with possible variations allowed), the consensus being determined from a multiple alignment of functionally related documented sequences. This type of method is used, for instance, for splice sites prediction in SPLICEVIEW (39) and SplicePredictor (40), the latter combining it with other kinds of information.

A more flexible representation of signals is offered by the so-called positional weight matrices (PWMs), which indicate the probability that a given base appears at each position of the signal (again computed from a multiple alignment of functionally related sequences). Equivalently, one can say that a PWM is defined by one classical zero order Markov model per position, which is called an inhomogeneous zero order Markov model. The PWM weights can also be optimized by a neural network method, as proposed by Brunak *et al.* (41) for NetPlantGene (42) and NetGene2 (43) and used in NNSplice (44).

Table 1. Splice site prediction programs

Program	Organism	Method
GeneSplicer (152)	*Arabidopsis*, human	HMM + MDD
NETPLANTGENE (42) (http://www.cbs.dtu.dk/services/NetPGene/)	*Arabidopsis*	NN
NETGENE2 (43) (http://www.cbs.dtu.dk/services/NetGene2/)	Human, *C.elegans*, *Arabidopsis*	NN + HMM
SPLICEVIEW (39) (http://l25.itba.mi.cnr.it/~webgene/wwwspliceview.html)	Eukaryotes	Score with consensus
NNSPLICE0.9 (44) (http://www.fruitfly.org/seq_tools/splice.html)	*Drosophila*, human or other	NN
SPLICEPREDICTOR (40,153) (http://bioinformatics.iastate.edu/cgi-bin/sp.cgi)	*Arabidopsis*, maize	Logitlinear models: (i) score with consensus; (ii) local composition
BCM-SPL (http://www.softberry.com/berry.phtml; http://genomic.sanger.ac.uk/gf/gf.html)	Human, *Drosophila*, *C.elegans*, yeast, plant	Linear discriminant analysis

HMM, hidden MM; MDD, maximal dependence decomposition; NN, neural networks.

In order to capture possible dependencies between adjacent positions of a signal, one may use higher order Markov models. The so-called weight array model (WAM) is essentially an inhomogeneous higher order Markov model. It was first proposed by Zhang and Marr (45) and later used by Salzberg (46), who applied it in the VEIL (47) and MORGAN (48) software. Genscan also uses a modified WAM to model acceptor splice sites and a second order WAM to represent branch point information. This is closely related to the position-dependent triplet frequency model employed by MZEF for the same signal.

These methods assume a fixed length signal. Hidden Markov models (HMMs) [see the tutorial from Rabiner (49) and, for instance, Krogh (50) for applications to biological sequences] further allow for insertions and deletions. They have been used in NetGene2 to model the branch point signal. In order to capture the most significant dependencies between adjacent as well as non-adjacent positions, Burge proposed another model for donor sites called the maximal dependence decomposition (MDD) method.

Most existing programs use such models to represent and detect splice sites. An alphabetical list of currently available splice site detection programs is presented in Table 1. These programs can already integrate the output of several signal sensors, or even signal and content sensors, as for example NetGene2, which combines splice site signal and branch point signal sensors with a global coding content sensor through a neural network. A similar approach is used in the recent program GeneSplicer, which combines MDD models for splice sites with second order Markov models that characterize coding/non-coding regions around splice sites. Recently, it was shown that combining sequence-based metrics for splice sites (WAM) with secondary structure metrics could lead to valuable improvements in splice site prediction (51).

However, when using splice site prediction programs, one ends up with a list of potential splice sites, from which various gene structures may be built. The main purpose of such programs is not to find the gene structure but to try to find the correct exon boundaries. They are thus very useful in addition to an exon or gene predictor in order to refine an existing gene structure. These programs can also provide insights into possible alternative splicing, even if, so far, this possibility has been very poorly investigated.

Finally, HMMs have also been used to represent other types of signals, such as poly(A) sites (in 3′-UTRs), promoters, etc. Although promoter detection is, by nature, closely related to gene detection, we will not discuss it in this paper, as it represents, on its own, an important area of research in computational biology (52). Recently, Pedersen *et al.* (53) reviewed the (known) biology of eukaryotic promoters and the many difficulties encountered in predicting them. As for the previous 'intrinsic' content sensors, the fact that HMMs are built from a multiple alignment of known functional sequences inherently limits the sensors to canonical signals.

Another important signal to identify when trying to predict a coding sequence is the translation initiation codon. A few programs exist specifically dedicated to this problem (54–56), but most of them have a rather limited efficiency, which is maybe related to the lack of proper learning sets for eukaryotic genomes. Experimental information on the genuine location of translation starts has indeed been scarce up to now, a situation that will likely change soon with the advent of proteome data.

COMBINING THE EVIDENCE TO PREDICT GENE STRUCTURES

Since 1990, following the example of Fields and Soderlund (57) and of Gelfand (58), programs are no longer limited to searching for independent exons, but try instead to identify the whole complex structure of a gene. Given a sequence and using signal sensors, one can accumulate evidence on the occurrence of signals: translation starts and stops and splice sites are the most important ones since they define the boundaries of coding regions. In theory, each consistent pair of detected signals defines a potential gene region (intron, exon or coding part of an exon). If one considers that all these potential gene regions can be used to build a gene model, the number of potential gene models grows exponentially with the number of predicted exons. In practice, this is slightly reduced by the fact that 'correct' gene structures must satisfy a set of properties: (i) there are no overlapping exons; (ii) coding exons must be frame compatible; (iii) merging two successive coding exons will not generate an in-frame stop at the junction. The number of candidates remains, however, exponential. In almost all existing approaches, such an

exponential number is coped with in reasonable time by using dynamic programming techniques.

Until recently, prediction methods that try to determine the whole gene structure, i.e. to assemble all the pieces, could be separated into two classes depending on whether the content of exon/intron regions was assessed using extrinsic or intrinsic content sensors. We first consider each of these approaches in turn and then see how more recent programs try to combine evidence coming from both intrinsic and extrinsic content sensors.

Extrinsic approaches

Pioneered by Gelfand *et al.* (59) with Procrustes, many software based on similarity searches have emerged during the last 5 years. They are presented in Table 2. As mentioned before, one of the main weaknesses of the pure similarity-based content sensors is that the limits of similarities are never accurately defined. The principle of most of these programs is to combine similarity information with signal information obtained by signal sensors. This information will be used to refine the region boundaries. These programs inherit all the strengths and weaknesses of the sensors used and may, for example, fail when non-canonical splice sites are present.

Very briefly, all the programs in this class may be seen as sophistications of the traditional Smith–Waterman local alignment algorithm where the existence of a signal allows for the opening (donor) or closure (acceptor) of a gap with an essentially free extension cost. They are often referred to as 'spliced alignment' programs. Existing software may be further divided according to the type of similarity exploited: genomic DNA/protein, genomic DNA/cDNA or genomic DNA/genomic DNA. Some of these methods are able to deal with more than one type and to take into account possible frameshifts in the genomic DNA or cDNA sequences.

The purpose of Procrustes is to align a genomic sequence with a protein. The selection of the target protein is left to the user, who may retrieve it from a BLASTX search for instance. Procrustes then considers all potential exons from the query DNA sequence, initially with the only constraint that they must be bordered by donor and acceptor sites [according to Gelfand *et al.* (59), it is recommended to additionally use content sensors, although this does not seem to be done in the program available on the Web]. All possible exon assemblies are explored by translating the exons and aligning them with the target protein, using the PAM120 matrix for scoring mismatches. This is done in a time proportional to the product of the lengths of the query and target sequences. As a result, it produces an assembly with the highest similarity score to the target protein. Other programs performing the same task are GeneWise (60), PredictGenes, ORFgene (61) and ALN (62), the latter introducing a new code of 23 letters for translated codons (called a tron).

Some programs, like INFO (63) and ICE (64), use a dictionary-based approach: they first create dictionaries of k long segments from a protein or an EST database and then, using a look-up procedure, find all segments in the query DNA sequence having a match in the dictionary. A related alternative is used in the mixture model of Thayer *et al.* (65), where only the highly populated protein segments, derived from multiple protein alignments, are stored and then used to develop statistical models.

Other available programs are AAT (66), GeneSeqer (67,68), SIM4 (69) and Spidey (70), all of which perform an alignment of the genomic DNA sequence against a cDNA database. This is a very reliable way of identifying exons, independently of their coding status, especially when the genomic sequence is aligned against a cDNA from the same or a close organism (71). Difficulties may, however, be encountered when trying to delineate the UTR part of the genes, and thus the correct translation initiator and stop codons. AAT and GeneSeqer also allow for the alignment against a protein database.

The SYNCOD program (72) uses a more original procedure: it compares two DNA sequences (retrieved with BLASTN) and, for each ORF contained in a high scoring pair, analyzes the ratio of the number of mismatches resulting in synonymous (silent) codons to the number of mismatches resulting in non-synonymous codons. An ORF is then assumed to be a potential coding region if this ratio significantly deviates from a random behavior, simulated with a Monte Carlo procedure.

The approach adopted is rather different for programs which try to elucidate the gene structure from EST matches, like EbEST (73), Est2genome (74) or, more recently, TAP (75) and PAGAN (http://ismb01.cbs.dtu.dk/GeneFinding.html#A308). The reason for this comes from the specific nature of an EST. A first characteristic of ESTs is that they are very redundant and a large number of them may be retrieved when performing a BLAST search against dbEST. EbEST faces this problem in its first step by clustering ESTs into non-overlapping groups (PAGAN clusters alignments of ESTs obtained from the results of similarity searches) and then by selecting the most informative ESTs within each group. A second characteristic of ESTs is that they are naturally error prone since they are generated from single-read sequences. The Smith–Waterman algorithm used in EbEST tolerates the presence of such errors. Another characteristic of ESTs is that most of them are 3′ ESTs generated from oligo(dT)-primed cDNA libraries and are therefore useful for detecting the 3′-UTRs in long sequences. This last point is an important added value for the EST-driven gene modeling approaches, as it leads to a rather confident prediction of a gene 3′-end. This importance may be somewhat weakened by the fact that ESTs represent only partial mRNA sequences and even clusters of ESTs may not lead to the complete identification of the gene structure.

Finally, several programs try to retrieve information on conservation or synteny between organisms from genomic alignments, as, for example, MUMmer (76), WABA (77), PipMaker (78) and DIALIGN (79). Recently, a few algorithms have appeared that focus more specifically on the gene recognition problem by comparison of two genomic sequences: such programs are based on the hypothesis that coding DNA sequences are more conserved than non-coding sequences (intronic and intergenic). Comparing two homologous genomic sequences (cross- or intra-species) should thus help to reveal conserved exons and allow the prediction of genes simultaneously on both sequences. Some programs like ROSETTA (80) and CEM (81) are more specifically designed for the comparison of closely related species. In particular, they make the hypothesis of conserved exon–intron structure in the two sequences. ROSETTA makes

Table 2. Homology-based gene prediction programs

Program	Organism	Databank or required input	Alignment	Gene reconstruction
AAT (66) (http://genome.cs.mtu.edu/aat.html)	Primates, rodents, other	cDNA, protein	DDS (improved BLASTX), DPS (improved BLASTN)	NAP, GAP2
ALN (62)		Protein	Tron code, PAM 250	
CEM (81)		Two genomic sequence	BLASTX output, WMM for sites	DP
EbEST (73) (http://ares.ifrc.mcw.edu/EBEST/ebest.html)	Human, other	dbEST	BLASTN, EST clustering, Smith–Waterman-based gapped alignment	3′-UTR detection, assembly of EST-tagged exons
Est2genome (74)		EST or cDNA, preferably BLASTN output	Modified Smith–Waterman Needleman–Wunsh algorithm	No
GeneSeqer (67,68) (http://bioinformatics.iastate.edu/cgi-bin/gs.cgi)	*Arabidopsis*, maize, generic plant	dbEST or EST database or proteins	Spliced alignment, splice recognition with SplicePredictor if missing EST match	Yes
GeneWise (60) (http://www.sanger.ac.uk/Software/Wise2/genewiseform.shtml)	Human	One protein or a HMM profile	Global alignment translated ORF/protein	DP (dynamite)
GENQUEST (http://compbio.ornl.gov/Grail-bin/EmptyGenquestForm)		dbEST, SwissProt, Prosite, BLOCKS, GSDB	Smith–Waterman, Blast, Fasta	
ICE (64) (http://theory.lcs.mit.edu/ice)		dbEST, OWL	Look-up	DP
INFO (63)		Nr	25mer look-up table, protein/protein alignments scored with PAM 40, PAM 120, PAM 250, BLO62	No
ORFgene2 (61) (http://l25.itba.mi.cnr.it/~webgene/wwworfgene2.html)	Human, mouse, *Drosophila*, *Aspergillus*, *Arabidopsis*, *Caenorhabditis*	SwissProt	BlastP, WAM for splice sites, identity score on frequencies of dipeptides	Compatibility graph, DP
PredictGenes (http://cbrg.inf.ethz.ch/Server/subsection3_1_8.html)	Invertebrates, vertebrates, prokaryotes, plants	SwissProt	PAM 250	DP
PROCRUSTES (59) (http://www-hto.usc.edu/software/procrustes/wwwserv.html)	Vertebrates	One homologous protein	Protein/protein alignments scored with PAM 120	DP
Pro-Gen (83) (http://www.anchorgen.com/pro_gen/pro_gen.html)		Two genomic sequences	Alignment of translated sequences scored with PAM 120	DP
ROSETTA (80) (http://crossspecies.lcs.mit.edu/)	Human, mouse	Two genomic sequences	GLASS (global alignment system), PAM 20, Genscan method for splice sites	DP
SGP-1 (82) (http://soft.ice.mpg.de/sgp-1)	Vertebrates, angiosperms	Two genomic sequences or a pairwise local alignment output	Local alignment	DP
SIM4 (69)	All eukayotes	cDNA/genomic	HSP from Blast	No
SLAM (85) (http://baboon.math.berkeley.edu/~syntenic/slam.html)	Human, mouse	Two genomic sequences	Generalized pair HMM	DP
Spidey (70) (http://www.ncbi.nlm.nih.gov/IEB/Research/Ostell/Spidey/index.html)	Vertebrates, *Drosophila*, *C.elegans*, plant	One genomic sequence/set of mRNAs	Two Blasts: high stringency and low stringency	
SYNCOD (72) (http://l25.itba.mi.cnr.it/~webgene/wwwsyncod.html)	Human, mouse, *Drosophila*, *Arabidopsis*, *Aspergillus*, *Caenorhabditis*	BLASTN output	Silent/replacement ratio, Monte Carlo simulations	No
TAP (75) (http://sapiens.wustl.edu/~zkan/TAP/)	Human, mouse, *Drosophila*	dbEST	WU-BLASTN, SIM4	Yes
Utopia (84)	All eukaryotes	Two genomic sequences	Local alignment	Yes

DP, dynamic programming; WAM, weight array matrix; WMM, weight matrix method.

the further (very strong) hypothesis that the corresponding exons in the two genes have roughly the same length. More flexibility is allowed by algorithms which do not assume that the gene structure is conserved, as in SGP-1 (82), Pro-Gen (83) and Utopia (84). All these programs, except ROSETTA, can perform the sequence comparison at the protein level. SGP-1 is also able to use a nucleotide level alignment; it further combines sequence comparison with information coming from signal sensors. It seems nevertheless to perform worse than Pro-Gen and Utopia on sequences not so closely related or for organisms for which it has not been specifically trained (P.Blayo, S.Aubourg, P.Peterlongo, P.Rouzé and M.F.Sagot, manuscript in preparation). However, SGP and a more recent version of Utopia allow the prediction of partial genes as well as multiple collinear genes in genomic sequences. Finally, an original method was recently implemented in the SLAM program, which introduces a probabilistic cross-species gene finding algorithm using generalized pair HMMs (85).

All the comparative genomic methods have, theoretically, the advantage of not being species specific. In practice, their performance will depend on the evolutionary distance between the compared sequences. Initial results show that the relationship is not straightforward. Indeed, a greater evolutionary distance allows some algorithms to more accurately discriminate between coding and non-coding sequence conservation. Such programs are often computer intensive and consequently much work remains to be done. In particular, a major challenge that could considerably improve the performance of gene finding programs would be to introduce multiple comparisons into these methods.

In order to retrieve only relevant information from homology searches against databases, the use of such programs must be coupled, if not integrated, with other specific programs to eliminate repeated sequences (SINES, LINES, etc.), which are very frequent in human genomic sequences (about one-quarter of the genome). Examples of such programs are RepeatMasker (http://ftp.genome.washington.edu/RM/RepeatMasker.html) and CENSOR (86).

Intrinsic approaches

Unlike most of the 'spliced alignment' approaches described in the previous section, which aim at producing a (single) gene structure based on similarities to known sequences, intrinsic gene finders (Table 3) aim at locating all the gene elements that occur in a genomic sequence, including possible partial gene structures at the border of the sequence.

To efficiently deal with the exponential number of possible gene structures defined by potential signals, almost all intrinsic gene finders use dynamic programming (DP) to identify the most likely gene structures according to the evidence defined by both content and signal sensors. All such gene modeling strategies can be formulated with a graph language (87). Following Guigó (88), such approaches are said to be exon based or signal based depending on whether a gene structure is considered to be defined by an assembly of segments defining the coding part of the exons (exon based) or by the presence of a succession of signals separated by 'homogeneous' regions.

In the exon-based category, the gene assembly is separated from the coding segments prediction step. The goal is to find the highest scoring genes, the gene score being a simple function (usually the sum) of the scores of the assembled segments. The strength of this two-step process comes from the fact that the score of each segment can be quite complex and may depend on some global characteristics of the coding part of exons, such as their lengths, for instance. In theory at least, the segment assembly process can be defined as the search for an optimal path in a directed acyclic graph where vertices represent exons and edges represent compatibility between exons. This is the approach adopted by the GeneId (89), GenView2, GAP3 (90), FGENE (28) and DAGGER (91) programs, where the initially adopted algorithms run in a time proportional to the square of the number of predicted segments. Recently, the DP algorithm GenAmic (88) solved the problem in a running time that grew only linearly with the number of potential segments, which itself is in the worst case quadratic with the sequence length (however, the assumption that in-frame stop codons occur following a Poisson process suffices to make the expected number of possible coding segments linear with the length of the sequence). It is currently integrated in the GeneId program as well as in FGENE, whereas the GeneGenerator (92) program has its own linear running time algorithm.

In the signal-based methods, the gene assembly is produced directly from the set of detected signals. In the simplest signal-based methods, there is an implicit assumption that the 'content' score of a segment is defined as the sum of the local (nucleotide-based) content scores and therefore does not depend on the global characteristics of the segment. This is, for instance, the case in a basic HMM. In such models, a given segment is considered to be generated by a Markov model associated with a state of the automaton (e.g. coding or non-coding). Since such states are unknown, they are called hidden states. In a given state, the 'content' score of a segment is defined as the log-likelihood of the region according to the corresponding Markov model, i.e. as the sum of the logarithms of the probabilities that each nucleotide appears given the k previous nucleotides in the model. Thanks to this assumption, and in theory at least, the gene parsing can be defined as the search for an optimal path in a directed acyclic graph. This search is done using the famous Viterbi algorithm (93), which produces a most likely gene structure and can be considered as a specific instance of the older Bellman shortest path algorithm (94), also used in the first versions of EuGène. Its running time grows linearly with the length of the sequence. Krogh has developed the first gene finder using a HMM, ECOPARSE (95), for *Escherichia coli*. The basic HMM model can be made more sophisticated by taking into account the length of the regions in the score. However, this leads to a quadratic running time. Additional assumptions make the algorithms usable in practice. This results in the so-called hidden semi-Markov models or generalized HMM. Such models are used in HMM-based programs like Genscan, Genie (44,96), GeneMark.hmm, FGENESH (A.Salamov and V.Solovyev, unpublished data; see http://genomic.sanger.ac.uk/) and GRPL (97). HMMgene (98) and VEIL employ a slightly different method called CHMM, for class HMM. Interestingly, the GRPL program introduces a new classification method for functional sites surrounding exons, called the regression point logistic. Other kinds of classifiers have already been integrated in some gene finders, such as a

Table 3. *Ab initio* gene prediction programs (possibly with homology integration)

Program	Organism	Gene elements	Gene model	Homology
DAGGER (91)		Site scores	Directed acyclic graphs	
EuGène (31) (http://www.inra.fr/bia/T/EuGene)	*Arabidopsis*	Three-periodic IMM for exons, one IMM for introns, one for intergenic regions, one for UTR. NetGene2/ SplicePredictor for splice sites	DP	EST/cDNA, protein
GeneId3 (89) (http://www1.imim.es/geneid.html)	Vertebrates, plants	Rule-based method; WAM, discriminant analysis.	DP	EST
GENEFINDER (28): FGENE, FEX … (http://genomic.sanger.ac.uk/gf/gf.html; http://www.softberry.com/berry.phtml)	Human, mouse, *Drosophila*, *Caenorhabditis elegans*, yeast, dicots, monocots, *Schizosaccharomyces pombe*, *Neurospora crassa*	Linear discriminant analysis	DP	Protein
GENEFINDER (Green)		Log likelihood ratio score matrix on MM	DP	
GeneGenerator (92)	Maize	Logitlinear models for splice sites, start; 3rd to 5th order MM for exons and introns	DP	
GeneMark (29) (http://opal.biology.gatech.edu/GeneMark/genemark24.cgi)	Prokaryotes, eukaryotes	5th order MM (homogeneous for introns, three-periodic for exons)	No	
GeneMark.hmm (35) (http://opal.biology.gatech.edu/GeneMark/eukhmm.cgi)	Human, mouse, *Drosophila*, *Gallus gallus*, *Arabidopsis*, rice, maize, *Chlamydomonas reinhardtii*, *C.elegans*, *Hordeum vulgare*, *Triticum aestivum*	5th order MM (homogeneous for introns, three-periodic for exons)	GHMM, DP	Under development
GeneModeler (57) (ftp://ftp.tigr.org/pub/software/gm/)	Eukaryotes	Nucleotide and dinucleotide composition, consensus for splice sites	Rule-based method	
GeneParser (27) (http://beagle.colorado.edu/~eesnyder/GeneParser.html)	Vertebrates	NN	DP	EST
Genie (44,96) (http://www.fruitfly.org/seq_tools/genie.html)	*Drosophila*, human, other	NN	GHMM, DP	Protein
GenLang (154) (http://www.cbil.upenn.edu/genlang/genlang_home.html)	Vertebrates, *Drosophila*, dicots	Grammar rules, WAM, hextuple frequencies …	Chart parsing, DP	
GenomeScan	Vertebrates	Genscan method, BLASTP or BLASTX	GHMM, DP	Protein
GENSCAN (30) (http://genes.mit.edu/GENSCAN.html)	Vertebrates, *Arabidopsis*, maize	WAM for acceptor; MDD for donor; 5th order MM (homogeneous for introns, three-periodic for exons)	GHMM, DP	Protein, GenomeScan (99)
GENVIEW2 (25) (http://l25.itba.mi.cnr.it/~webgene/wwwgene.html)	Human, mouse, diptera	Linear combination, dicodon statistic	DP	
GlimmerM (33,34) (salzberg@cs.jhu.edu)	Small eukaryotes, *Arabidopsis*, rice	Three-periodic IMM for exons (order 0–8), IMM for introns, 2nd order MM for splice sites	DP	
GRAIL/GAP3 (90,155) (http://compbio.ornl.gov/Grail-bin/EmptyGrailForm)	Human, mouse, *Arabidopsis*, *Drosophila*	NN	DP	EST, cDNA
GRPL (97)	Human, *Drosophila*, *Arabidopsis*	Reference point logistic for splice sites, 5th order MM (homogeneous for introns, three-periodic for exons)	GHMM, DP	Protein
HMMgene (98) (http://www.cbs.dtu.dk/services/HMMgene/)	Vertebrates, *C.elegans*	Three-periodic 4th order MM for exons, 3rd order MM for introns	CHMM	
MORGAN (48) (http://www.cs.jhu.edu/labs/compbio/morgan.html)	Vertebrates	Decision tree system	DP	
MZEF (26) (http://argon.cshl.org/genefinder/)	Human, mouse, *Arabidopsis*, fission yeast	Quadratic discriminant analysis	No	
SORFIND (24)		Matrix method for start and splice sites, hexamer usage (Fourier measure)	No	
Twinscan (100)	Mouse, human	Genscan method; 5th order MM for UTR and intergenic, WAM for acceptor sites	GHMM	Genomic sequence
VEIL (47) (http://www.cs.jhu.edu/labs/compbio/veil.html)	Vertebrates	HMM	DP	
Xpound (156) (http://bioweb.pasteur.fr/seqanal/interfaces/xpound-simple.html)	Human	Three-periodic 1st order MM for exons, 1st order MM for introns and intergenic	HMM	

CHMM, class HMM; GHMM, generalized HMM; IMM, interpolated MM; MM, Markov model.

decision tree system in MORGAN or discriminant analysis in FGENE and MZEF.

Integrated approaches

Aware of the added value provided by database similarities, authors are now combining both intrinsic and extrinsic approaches in recent gene predictors and, in older software, updates are made to add information from homology. A pioneer in the area was the GSA program (Gene Structure Assembly), born from the fusion between AAT and Genscan, and whose results on the Burset and Guigó dataset are better than those obtained with the two programs separately (X.Huang, personal communication). GenomeScan (99) is Burge's own extension of Genscan to incorporate similarity with a protein retrieved by BLASTX or BLASTP. Genes predicted by GenomeScan have a maximum probability conditional on such similarity information. GenomeScan is thus able to accurately predict coding regions missed by both Genscan and BLASTX used alone. Although no paper exists describing them, FGENESH+ and FGENESH_C are other extensions of an existing algorithm, FGENESH, that use similarity to a protein or a cDNA sequence, respectively, to improve gene prediction. In fact, most intrinsic prediction programs now offer (or will soon offer) the possibility of integrating similarities with expressed sequences to confirm their prediction (see Table 3). The integration of similarity between genomic sequences has so far been considered only in FGENES-H2 and Twinscan (100), which use the FGENESH and Genscan programs, respectively. Such integration represents a very promising approach that will undoubtedly be much further developed in the future.

A highly integrative approach is used in the EuGène program: it combines NetGene2 and SplicePredictor for splice site prediction, NetStart (54) for translation initiation prediction, IMM-based content sensors and similarity information from protein, EST and cDNA matches. All these sensors are weighted, with the weights being optimized in order to maximize the number of successes (as is done in the GeneParser and Dagger programs). This approach produces a reliable software, as attested by the results obtained on the *A.thaliana* genes that were presented at the Georgia Tech *In silico* Biology International Conference in November 1999. Its good performance probably relies to a large part on the maximum of success criterion and on the fact that the program is using a specific model for intergenic regions. It is also worth mentioning GAZE (http://ismb01.cbs.dtu.dk/GeneFinding.html#A304), which should allow the integration of arbitrary prediction information from multiple sources supplied by the user.

In the same order of ideas, it is possible to directly combine the predictions of several programs in order to obtain a sort of consensus. Such an approach was tested by several authors and involved different software (17,101,102). DIGIT (http://ismb01.cbs.dtu.dk/GeneFinding.html#A303) is such a software. It integrates FGENESH, Genscan and HMMgene. It is quite obvious that only keeping exons shared by two or more predictions has the advantage of significantly decreasing the number of over-predictions but may lead to a poor sensibility and possible inconsistencies at the gene level. However, promising results were recently obtained with a new flexible method for combining any number of gene prediction systems, as a combination of experts, inside a Bayesian framework (103).

Finally, it must be mentioned that there exist annotation platforms that allow the running of several selected gene prediction programs and/or which graphically display their results. Among the more widely used are Genotator (104), MagPie (105) and Ensembl (106). If such platforms are not prediction programs themselves, they gather evidence obtained from *ab initio* or homology-based prediction programs and thus are complementary and useful tools that facilitate either human driven or automated annotations.

QUALITY OF THE PREDICTIONS AND PRINCIPAL REMAINING PROBLEMS

Prediction accuracy

For numerical data on the quality of the predictions obtained by most *ab initio* programs, the reader is invited to see Burset and Guigó (15), Rogic *et al.* (16) and Pavy *et al*. (17) or to refer to the homepage of each software, which sometimes provides such data.

Although the statistical properties of coding regions allow for a good discrimination between large coding and non-coding regions, the exact identification of the limits of the coding parts of the exons or of gene boundaries remains difficult. In the first case, predicted coding region limits are often incorrect. In the second case, the predicted structure frequently splits a single true gene into several or, alternatively, merges several genes into one. To address these problems, several teams are currently working on the terminal parts of genes. Already, a program especially dedicated to predicting 3′-terminal exons (107) and another for identifying 5′-terminal exons (108) have been made available from M. Zhang's laboratory. Such problems are, however, very complex, as intergenic and intronic sequences do not differ much. Furthermore, specific signals in the 5′- or 3′-ends of genes (e.g. the TATA box and the polyadenylation signal), which could be expected to be useful for predicting gene boundaries, are often too variable. Once more, improvements can be achieved by looking for a combination of several signals together with compositional biases (109). Sometimes, however, such signals are not even present (53,110).

It is clear, and has recently been underlined (111), that gene prediction accuracy drops significantly with large DNA sequences. This is due to a decrease in gene density and to the presence of larger introns.

No exhaustive evaluation of the homology-based prediction programs has been published so far, but some general comments can be made that are based on both experience and the results discussed in the papers describing each algorithm. As expected, the programs that use expression data, such as protein or mRNA sequences, have the advantage of generating fewer false predictions, i.e. they tend to have good specificity. The obvious counterpart is of course that when no gene is predicted, this does not imply that no gene is present. Moreover, determination of the complete gene structure by such methods is also difficult or impossible when either the target expressed sequence is partial or when the evolutionary distance between the compared sequences is too great. However, even partial hits can give useful clues to

the presence of a gene and, therefore, the homology-based methods are important in all annotation processes.

Whatever the approach, gene prediction depends, to a large extent, on the current biological knowledge, especially knowledge at the molecular level of gene expression which keeps evolving and accumulating. Besides mainstream mechanisms, some unexpected non-canonical cases are regularly reported in the literature, which again increase the complexity of the problem.

Pitfalls and issues to be addressed

Several issues make the problem of eukaryotic gene finding extremely difficult. (i) Very long genes: for example, the largest human gene, the dystrophin gene, is composed of 79 exons spanning nearly 2.3 Mb (112). (ii) Very long introns: again, in the human dystrophin gene, some introns are >100 kb long and >99% of the gene is composed of introns. (iii) Very conserved introns or 3′-UTRs (113), either between different species or within gene families: this is particularly a problem when gene prediction is addressed through similarity searches. A comparative analysis of 77 orthologous mouse and human gene pairs concluded that 56% of 3′-UTRs are covered by alignable blocks (114). (iv) Very short exons: some exons are only 3 bp long in *Arabidopsis* genes and probably even 1 bp for the coding part of exons at either end of the coding sequence, meaning that start or stop codons can be interrupted by an intron (S.Aubourg, K.Vandepoele, P.Déhais, C.Mathé and P.Rouzé, personal communication). Such small exons are easily missed by all content sensors, especially if bordered by large introns. The more difficult cases are those where the length of a coding exon is a multiple of three (typically 3, 6 or 9 bp long), because missing such exons will not cause a problem in the exon assembly as they do not introduce any change in the frame.

Other issues could be addressed by making the gene model currently adopted in most gene finders more sophisticated.

(i) Overlapping genes: though very rare in eukaryotic genomes, there are some documented cases in animals as well as in plants (115). The number of such cases will probably increase as soon as we consider them as probable. Overlapping generally involves the 3′-UTR part of genes, but it can also happen that there is a gene within an intron of another gene [the first example was reported by Henikoff *et al.* (116)].

(ii) Polycistronic gene arrangement: albeit this situation was initially thought to occur only in prokaryotes, polycistronic genes have been found in eukaryotes. Many such cases are known for snoRNA genes in plants (117), as well as for protein encoding genes in nematodes, and so far isolated examples have been observed in mammals (118).

(iii) Frameshifts: some sequences stored in databases may contain errors (either sequencing errors or simply errors made when editing the sequence) resulting in the introduction of artificial frameshifts (deletion or insertion of one base). Such frameshifts greatly increase the difficulty of the computational gene finding problem by producing erroneous statistics and masking true solutions. Some programs are said to be able to detect frameshifts, like GeneMark and AAT. Some homology-based programs, such as Utopia and Pro-Frame (119), can handle such cases. Some programs specifically dedicated to this problem exist, mainly for prokaryotic genomes, but also for coding eukaryotic sequences such as FSED (120). More recent programs such as FrameD (121) and ESTScan (122) are able to deal with noisy sequences and are typically aimed at detecting and correcting frameshifts in cDNA and more specifically in EST sequences.

(iv) Introns in non-coding regions: there are genes for which the genomic region corresponding to the 5′- and/or 3′-UTR in the mature mRNA is interrupted by one or more intron(s). The extremity of the gene is then composed of non-coding exons and intron(s). It was shown that such non-coding exons might also support alternative splicing (123). Some of the programs that exploit similarities with ESTs or cDNAs, such as SplicePredictor, are actually able to predict such introns. Without such evidence, and since the base composition of UTRs is more intron-like than coding exon-like (42), most programs simply ignore this problem.

(v) Non-canonical splice sites (reviewed in 124): they are not really handled by any program yet. As long as examples of non-canonical introns are provided in the training set, programs such as GeneParser or NetPlantGene/NetGene2 (and therefore EuGène) should be able to identify some splice sites that do not use a GT-AG rule. The new FGENESH program (A.Salamov and V.Solovyev, unpublished) specifically allows for the existence of a GC donor site instead of a GT. Recently, mammalian annotated genes were controlled with EST matching sequences, producing new sets of EST-supported non-canonical splice sites (125). It would be interesting to do the same work on other genomes, as it provides very useful data that could be integrated into gene prediction programs.

(vi) Cases of an alternative biological processing. (a) Alternative transcription start: e.g. three alternative promoters regulate the transcription of the 14 kb full-length dystrophin mRNAs and four ‘intragenic’ promoters control that of smaller isoforms (112). Another problem related to promoters concerns the fact that there is not one single class of core promoter. Instead, many combinations of small elements are possible (53). (b) Alternative splicing: EST-based methods are widely used to identify more systematically such events and to understand their roles (126,127). A recent review on this topic (128) reports an estimated 35–59% of human genes showing evidence for at least one alternative splice site form. Some gene prediction programs try to handle this through the identification of sub-optimal exons (Genscan and MZEF) and sub-optimal gene structures [GeneParser, HMMgene, GeneGenerator and FGENES-M (V.Solovyev, unpublished data)]. Nevertheless, a more relevant approach would consist of improving the identification of the intronic and/or exonic signals that dictate the choice of alternative sites (129). (c) Alternative polyadenylation: here again, EST data are very informative and lead to an estimated 20% of human transcripts showing evidence of alternative polyadenylation (130). (d) Alternative initiation of translation: finding the right AUG initiator is still a major concern for gene prediction methods. The main reason is that experimental data on native proteins remain scarce. The biological process itself is far from simple and is therefore not yet fully elucidated [for a recent review on the initiation of translation, see Kozak (131)]. Briefly, the rule stating that the first AUG in the mRNA is the initiator codon can be escaped through three mechanisms: context-dependent leaky scanning, re-initiation and direct internal initiation. Furthermore, it has been observed that a

non-AUG triplet can sometimes act as the functional codon for translation initiation, as ACG in *Arabidopsis* (132) or CUG in human sequences (133). However, no current program considers such alternative forms.

Some of these cases remain rare enough so that taking them into account in the gene prediction algorithms would lower the overall accuracy of the programs while greatly increasing the level of difficulty. Current programs thus aim at optimizing performance on the majority of data. This is laudable, but one must be aware that this is at the expense of obtaining a good performance on 'outliers' of the population. This may be a problem, particularly if one considers that such outlier cases appear marginal only because nobody (including 'wet' biologists) is looking for them and that they may in fact be more frequent than suspected. As mentioned, this may be the case for overlapping genes.

Problems with databases and/or training sets

A major concern, already stressed by Claverie (11), is that existing sensors are very conservative since they nearly always rely on already known sequences, either in the form of training sets or of databases against which homology searches are performed. Attempts to tackle this problem have been proposed (134,135), which were dedicated to prokaryotic genomes and were anyway not fully satisfying. However, it seems to be a natural and reasonable procedure to first look for what is already known and then to try to extrapolate from our current knowledge. This approach should probably be considered only as a first step. A first parsing of the data could be done using one of these 'conservative' methods and, either in a second run or when unusual situations occur, the possibility of non-canonical cases should be examined, allowing a reconsideration of the first prediction proposed.

Another remark about training sets is that several datasets may be better than a single big one. Indeed, an important fact considered by only a few programs is that, from a biological point of view, there is not a unique type of gene. Besides the fact that genes code for various types of proteins, they also exhibit differences in their level of expression, condition and cellular location of expression. There is therefore no reason to assume that only a unique gene model exists (136,137). Taking this into account has already proved to be relevant in the case of *E.coli* (138). As a consequence, an interesting problem would be to create such sub-data sets, with enough data to derive statistical models in a biologically relevant way. This problem was addressed in the case of prokaryotic genomes by GeneMark-Genesis, which automatically clusters ORFs on the basis of their codon usage and derives Markov models for each cluster obtained (139), and more recently by GeneMarkS (140). In eukaryotes, the problem is again more complex since, within a gene, some exons may better fit one model and others another model. There are nevertheless some clues indicating that the same kind of procedure would be relevant (141). This remark also applies to signal sensors, e.g. for splice sites. Several consensus sequences could thus be searched for.

Last but not least, whatever method is adopted (intrinsic or extrinsic), there is a great need for 'clean' sequence databases, i.e. for databases that are not redundant, contain reliable and relevant annotations and provide all necessary links to further data (142,143). Without such conditions, much false information can be (and currently is) propagated by the predictions made. If, in the case of extrinsic gene prediction, the usage of erroneous data has consequences only at the level of the analyzed sequence itself, in the case of intrinsic prediction, using corrupted training sets can dramatically affect the whole performance of the program. In both situations, it is therefore very important to filter the data retrieved from the databanks in order to use only experimentally documented sequences as references and not data coming from predictions, such as those generated by automatic genome annotations.

CONCLUSIONS

The prediction of protein encoding genes is obviously still in need of improvement, as discussed in the previous section, especially for larger genomes. It has also to better tackle the problem of alternative gene models and to take into account non-canonical, but nevertheless biologically significant, cases. Since gene prediction leads to a structural annotation of the genomes which is then used for experimentation, it would be wise to weight the predictions by giving a confidence value for each predicted gene, from high for a gene whose full structure has been obtained in a non-ambiguous way using cognate cDNA data to low for a gene whose prediction totally depends on intrinsic approaches.

Moreover, algorithms and software descriptions provided by the authors are too often somewhat superficial; in particular, most papers do not give enough details on precise parameter tuning, for instance. Until recently, there was also no common vocabulary adopted among developers, with no common definitions used between different software for the phase or frames, etc. It is worth observing an interesting effort to design a general feature format (GFF) (http://www.sanger.ac.uk/Software/formats/GFF/). This was the result of an agreement between many developers. The format aims at standardizing gene predictor outputs and vocabulary. A standard output for all gene predictors allows for the development of common tools that can be used for downstream analysis: evaluation, graphical representation and combination of predictions. A few software are already producing GFF output, such as HMMgene, SGP-1 and the exon assembly program GenAmic, which is using GFF files for both input and output. At a larger scale, the Gene Ontology consortium (http://www.geneontology.org/) aims at providing a structured vocabulary that would allow the description of gene products in any organism (144). Model organism groups have started joining this consortium, largely contributing to the unification of biological information, whose importance was recently emphasized (145).

With the huge amount of EST and cDNA sequences now available, programs based on the existence of homology with such expressed sequences are playing an ever increasingly crucial role in current genome annotations, at least for genes for which expression can be shown. Even in the case of genes that are scarcely or tissue specifically expressed, complementary information can be provided by similarity between genomic sequences. Consequently, a great deal of effort is now expended on trying to gather information from genome comparisons. This is particularly true in the case of the human genome annotation process, where the availability of other

complete vertebrate genomes, such as those of mouse and fish, is a great advantage. With the many genome sequencing projects currently under way, and although there are still problems to be solved (146), the comparative genome approach seems to be a very promising approach not only in the field of gene prediction but also for the identification of regulatory sequences and the deciphering of the so-called junk DNA. The latter has been largely ignored until now, yet much may be expected to be learned from its analysis (147,148). Even if in this review we have just discussed programs to detect protein coding genes, there is also an undetermined (but probably high) number of genes producing functional non-coding RNAs (reviewed in 149), which may be identified by genomic comparison. More interest is now devoted to such non-coding RNAs (150,151), and this probably stands among the main future directions in computational approaches for genome analysis.

Finally, we wish to again warn the users of gene prediction software that the results produced should be taken with caution: although such results are becoming increasingly more reliable, they do only remain predictions. These are very useful for speeding up gene discovery and knowledge mining thereof, but biological expertise remains necessary in order to confirm the existence of a virtual protein and to find or prove its biological function and its condition of expression in the organism.

ACKNOWLEDGEMENTS

We wish to thank the anonymous referees for their very interesting and constructive comments on the manuscript. We are also very grateful to Jim Middleton and to Alain Vignal for revising the English text.

REFERENCES

1. The International Human Genome Sequencing Consortium (2001) Initial sequencing and analysis of the human genome. *Nature*, **409**, 860–921.
2. The Arabidopsis Genome Initiative (2000) Analysis of the genome sequence of the flowering plant *Arabidopsis thaliana*. *Nature*, **408**, 796–815.
3. Goff,S.A., Ricke,D., Lan,T.H., Presting,G., Wang,R., Dunn,M., Glazebrook,J., Sessions,A., Oeller,P., Varma,H. *et al.* (2002) A draft sequence of the rice genome (*Oryza sativa* L. ssp. *japonica*). *Science*, **296**, 92–100.
4. Myers,E., Sutton,G., Delcher,A., Dew,I., Fasulo,D., Flanigan,M., Kravitz,S., Mobarry,C., Reinert,K., Remington,K. *et al.* (2000) A whole-genome assembly of *Drosophila*. *Science*, **287**, 2196–2204.
5. Claverie,J.M., Poirot,O. and Lopez,F. (1997) The difficulty of identifying genes in anonymous vertebrate sequences. *Comput. Chem.*, **21**, 203–214.
6. Cho,Y. and Walbot,V. (2001) Computational methods for gene annotation: the *Arabidopsis* genome. *Curr. Opin. Biotechnol.*, **12**, 126–130.
7. Borodovsky,M., Rudd,K.E. and Koonin,E.V. (1994) Intrinsic and extrinsic approaches for detecting genes in a bacterial genome. *Nucleic Acids Res.*, **22**, 4756–4767.
8. Fickett,J.W. (1996) The gene identification problem: an overview for developer. *Comput. Chem.*, **20**, 103–118.
9. Rouzé,P., Pavy,N. and Rombauts,S. (1999) Genome annotation: which tools do we have for it? *Curr. Opin. Plant Biol.*, **2**, 90–95.
10. Fickett,J.W. (1996) Finding genes by computer: the state of the art. *Trends Genet.*, **12**, 316–320.
11. Claverie,J.M. (1997) Computational methods for the identification of genes in vertebrate genomic sequences. *Hum. Mol. Genet.*, **6**, 1735–1744.
12. Guigó,R. (1997) Computational gene identification: an open problem. *Comput. Chem.*, **21**, 215–222.
13. Haussler,D. (1998) Computational genefinding. *Trends Biochem. Sci.*, 12–15.
14. Burge,C. and Karlin,S. (1998) Finding the genes in genomic DNA. *Curr. Opin. Struct. Biol.*, **8**, 346–354.
15. Burset,M. and Guigó,R. (1996) Evaluation of gene structure prediction programs. *Genomics*, **34**, 353–367.
16. Rogic,S., Mackworth,A. and Ouellette,F. (2001) Evaluation of gene-finding programs on mammalian sequences. *Genome Res.*, **11**, 817–832.
17. Pavy,N., Rombauts,S., Déhais,P., Mathé,C., Ramana,D.V.V., Leroy,P. and Rouzé,P. (1999) Evaluation of gene prediction software using a genomic data set: application to *Arabidopsis thaliana* sequences. *Bioinformatics*, **15**, 887–899.
18. Mignone,F., Gissi,C., Liuni,S. and Pesole,G. (2002) Untranslated regions of mRNAs. *Genome Biol.*, **3**, reviews0004.1-0004.10.
19. Pearson,W.R. and Lipman,D.J. (1988) Improved tools for biological sequence comparison. *Proc. Natl Acad. Sci. USA*, **85**, 2444–2448.
20. Altschul,S.F., Gish,W., Miller,W., Myers,E.W. and Lipman,D.J. (1990) Basic local alignment search tool. *J. Mol. Biol.*, **215**, 403–410.
21. Bailey,L.C., Searls,D.B. and Overton,G.C. (1998) Analysis of EST-driven gene annotation in human genomic sequence. *Genome Res.*, **8**, 362–376.
22. Fickett,J.W. (1995) ORFs and genes: how strong a connection? *J. Comput. Biol.*, **2**, 117–123.
23. Fickett,J.W. and Tung,C.S. (1992) Assessment of protein coding measures. *Nucleic Acids Res.*, **20**, 6441–6450.
24. Hutchinson,G.B. and Hayden,M.R. (1992) The prediction of exons through an analysis of spliceable open reading frames. *Nucleic Acids Res.*, **20**, 3453–3462.
25. Milanesi,L., Kolchanov,N.A., Rogozin,I.B., Ischenko,I.V., Kel,A.E., Orlov,Y.L., Ponomarenko,M.P. and Vezzoni,P. (1993) GenView: a computing tool for protein-coding regions prediction in nucleotide sequences. In Lim,H.A., Fickett,J.W., Cantor,C.R. and Robbins,R.J. (eds), *Proceedings of the Second International Conference on Bioinformatics, Supercomputing and Complex Genome Analysis*. World Scientific Publishing, Singapore, pp. 573–588.
26. Zhang,M.Q. (1997) Identification of protein coding regions in the human genome by quadratic discriminant analysis. *Proc. Natl Acad. Sci. USA*, **94**, 565–568. [Erratum (1997) *Proc. Natl Acad. Sci. USA*, **94**, 5495]
27. Snyder,E.E. and Stormo,G.D. (1995) Identification of protein coding regions in genomic DNA. *J. Mol. Biol.*, **248**, 1–18.
28. Solovyev,V. and Salamov,A. (1997) The Gene-Finder computer tools for analysis of human and model organisms genome sequences. In Gaasterland,T., Karp,P., Karplus,K., Ouzounis,C., Sander,C. and Valencia,A. (eds), *The Fifth International Conference on Intelligent Systems for Molecular Biology*. AAAI Press, Menlo Park, CA, pp. 294–302.
29. Borodovsky,M. and McIninch,J. (1993) GENMARK: parallel gene recognition for both DNA strands. *Comput. Chem.*, **17**, 123–133.
30. Burge,C. and Karlin,S. (1997) Prediction of complete gene structures in human genomic DNA. *J. Mol. Biol.*, **268**, 78–94.
31. Schiex,T., Moisan,A. and Rouzé,P. (2001) EuGène: an eukaryotic gene finder that combines several sources of evidence. In Gascuel,O. and Sagot,M.-F. (eds), *Lecture Notes in Computer Science*, Vol. 2006, *First International Conference on Biology, Informatics, and Mathematics, JOBIM 2000*. Springer-Verlag, Germany, pp. 111–125.
32. Salzberg,S., Delcher,A., Kasif,S. and White,O. (1998) Microbial gene identification using interpolated Markov models. *Nucleic Acids Res.*, **26**, 544–548.
33. Salzberg,S.L., Pertea,M., Delcher,A.L., Gardner,M.J. and Tettelin,H. (1999) Interpolated Markov models for eukaryotic gene finding. *Genomics*, **59**, 24–31.
34. Delcher,A.L., Harmon,D., Kasif,S., White,O. and Salzberg,S.L. (1999) Improved microbial gene identification with GLIMMER. *Nucleic Acids Res.*, **27**, 4636–4641.
35. Lukashin,A.V. and Borodovsky,M. (1998) GeneMark.hmm: new solutions for gene finding. *Nucleic Acids Res.*, **26**, 1107–1115.
36. Bernardi,G. (1989) The isochore organization of the human genome. *Annu. Rev. Genet.*, **23**, 637–661.

37. Montero,L.M., Salinas,J., Matassi,G. and Bernardi,G. (1990) Gene distribution and isochore organization in the nuclear genome of plants. *Nucleic Acids Res.*, **18**, 1859–1867.
38. Duret,L., Mouchiroud,D. and Gautier,C. (1995) Statistical analysis of vertebrate sequences reveals that long genes are scarce in GC-rich isochores. *J. Mol. Evol.*, **40**, 308–317.
39. Rogozin,I.B. and Milanesi,L. (1997) Analysis of donor splice signals in different organisms. *J. Mol. Evol.*, **45**, 50–59.
40. Kleffe,J., Hermann,K., Vahrson,W., Wittig,B. and Brendel,V. (1996) Logitlinear models for the prediction of splice sites in plant pre-mRNA sequences. *Nucleic Acids Res.*, **24**, 4709–4718.
41. Brunak,S., Engelbrecht,J. and Knudsen,S. (1991) Prediction of human mRNA donor and acceptor sites from the DNA sequence. *J. Mol. Biol.*, **220**, 49–65.
42. Hebsgaard,S.M., Korning,P.G., Tolstrup,N., Engelbrecht,J., Rouzé,P. and Brunak,S. (1996) Splice site prediction in *Arabidopsis thaliana* pre mRNA by combining local and global sequence information. *Nucleic Acids Res.*, **24**, 3439–3452.
43. Tolstrup,N., Rouzé,P. and Brunak,S. (1997) A branch point consensus from *Arabidopsis* found by non-circular analysis allows for better prediction of acceptor sites. *Nucleic Acids Res.*, **25**, 3159–3163.
44. Reese,M.G., Eeckman,F.H., Kulp,D. and Haussler,D. (1997) Improved splice site detection in Genie. In Istrail,S., Pevzner,P. and Waterman,M. (eds), *First Annual International Conference on Computational Molecular Biology (RECOMB)*. ACM Press, New York, NY, pp. 232–240.
45. Zhang,M.Q. and Marr,T.G. (1993) A weight array method for splicing signal analysis. *Comput. Appl. Biosci.*, **9**, 499–509.
46. Salzberg,S.L. (1997) A method for identifying splice sites and translational start sites in eukaryotic mRNA. *Comput. Appl. Biosci.*, **13**, 365–376.
47. Henderson,J., Salzberg,S. and Fasman,K. (1997) Finding genes in human DNA with a hidden Markov model. *J. Comput. Biol.*, **4**, 127–141.
48. Salzberg,S., Delcher,A., Fasman,K. and Henderson,J. (1998) A decision tree system for finding genes in DNA. *J. Comput. Biol.*, **5**, 667–680.
49. Rabiner,L.R. (1989) A tutorial on hidden Markov models and selected applications for speech recognition. *Proc. IEEE*, **77**, 257–285.
50. Krogh,A. (1998) An introduction to hidden Markov models for biological sequences. In Salzberg,S.L., Searls,D.B. and Kasif,S. (eds), *Computational Methods in Molecular Biology*. Elsevier, Amsterdam, The Netherlands, pp. 46–63.
51. Patterson,D.J., Yasuhara,K. and Ruzzo,W.L. (2002) Pre-mRNA secondary structure prediction aids splice site prediction. In Altman,R.B., Dunker,A.K., Hunter,L., Lauderdale,K. and Klein,T.E. (eds), *Pacific Symposium on Biocomputing*, Vol. 7, World Scientific, Singapore, pp. 223–234.
52. Ohler,U. and Niemann,H. (2001) Identification and analysis of eukaryotic promoters: recent computational approaches. *Trends Genet.*, **17**, 56–60.
53. Pedersen,A.G., Baldi,P., Chauvin,Y. and Brunak,S. (1999) The biology of eukaryotic promoter prediction—a review. *Comput. Chem.*, **23**, 191–207.
54. Pedersen,A.G. and Nielsen,H. (1997) Neural network prediction of translation initiation sites in eukaryotes: perspectives for EST and genome analysis. In Gaasterland,T., Karp,P., Karplus,K., Ouzounis,C., Sander,C. and Valencia,A. (eds), *The Fifth International Conference on Intelligent Systems for Molecular Biology*. AAAI Press, Menlo Park, CA, pp. 226–233.
55. Zien,A., Ratsch,G., Mika,S., Scholkopf,B., Lengauer,T. and Muller,K. (2000) Engineering support vector machine kernels that recognize translation initiation sites. *Bioinformatics*, **16**, 799–807.
56. Nishikawa,T., Ota,T. and Isogai,T. (2000) Prediction whether a human cDNA sequence contains initiation codon by combining statistical information and similarity with protein sequences. *Bioinformatics*, **16**, 960–967.
57. Fields,C.A. and Soderlund,C.A. (1990) gm: a practical tool for automating DNA sequence analysis. *Comput. Appl. Biosci.*, **6**, 263–270.
58. Gelfand,M.S. (1990) Computer prediction of the exon–intron structure of mammalian pre-mRNAs. *Nucleic Acids Res.*, **18**, 5865–5869.
59. Gelfand,M.S., Mironov,A.A. and Pevzner,P.A. (1996) Gene recognition via spliced sequence alignment. *Proc. Natl Acad. Sci. USA*, **93**, 9061–9066.
60. Birney,E. and Durbin,R. (1997) Dynamite: a flexible code generating language for dynamic programming methods used in sequence comparison. *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, **5**, 56–64.
61. Rogozin,I.B., Milanesi,L. and Kolchanov,N.A. (1996) Gene structure prediction using information on homologous protein sequence. *Comput. Appl. Biosci.*, **12**, 161–170.
62. Gotoh,O. (2000) Homology-based gene structure prediction: simplified matching algorithm using a translated codon (tron) and improved accuracy by allowing for long gaps. *Bioinformatics*, **16**, 190–202.
63. Laub,M.T. and Smith,D.W. (1998) Finding intron/exon splice junctions using INFO, INterruption Finder and Organizer. *J. Comput. Biol.*, **5**, 307–321.
64. Pachter,L., Batzoglou,S., Spitkovsky,V.I., Banks,E., Lander,E.S., Kleitman,D.J. and Berger,B. (1999) A dictionary-based approach for gene annotation. *J. Comput. Biol.*, **6**, 419–430.
65. Thayer,E., Bystroff,C. and Baker,D. (2000) Detection of protein coding sequences using a mixture model for local protein amino acid sequence. *J. Comput. Biol.*, **7**, 317–327.
66. Huang,X., Adams,M.D., Zhou,H. and Kerlavage,A.R. (1997) A tool for analyzing and annotating genomic sequences. *Genomics*, **46**, 37–45.
67. Usuka,J. and Brendel,V. (2000) Gene structure prediction by spliced alignment of genomic DNA with protein sequences: increased accuracy by differential splice site scoring. *J. Mol. Biol.*, **297**, 1075–1085.
68. Usuka,J., Zhu,W. and Brendel,V. (2000) Optimal spliced alignment of homologous cDNA to a genomic DNA template. *Bioinformatics*, **16**, 203–211.
69. Florea,L., Hartzell,G., Zhang,Z., Rubin,G.M. and Miller,W. (1998) A computer program for aligning a cDNA sequence with a genomic DNA sequence. *Genome Res.*, **8**, 967–974.
70. Wheelan,S.J., Church,D.M. and Ostell,J.M. (2001) Spidey: a tool for mRNA-to-genomic alignments. *Genome Res.*, **11**, 1952–1957.
71. Fukunishi,Y., Suzuki,H., Yoshino,M., Konno,H. and Hayashizaki,Y. (1999) Prediction of human cDNA from its homologous mouse full-length cDNA and human shotgun database. *FEBS Lett.*, **464**, 129–132.
72. Rogozin,I.B., D'Angelo,D. and Milanesi,L. (1999) Protein-coding regions prediction combining similarity searches and conservative evolutionary properties of protein-coding sequences. *Gene*, **226**, 129–137.
73. Jiang,J. and Jacob,H.J. (1998) EbEST: an automated tool using expressed sequence tags to delineate gene structure. *Genome Res.*, **8**, 268–275.
74. Mott,R. (1997) EST_GENOME: a program to align spliced DNA sequences to unspliced genomic DNA. *Comput. Appl. Biosci.*, **13**, 477–478.
75. Kan,Z., Rouchka,E.C., Gish,W.R. and States,D.J. (2001) Gene structure prediction and alternative splicing analysis using genomically aligned ESTs. *Genome Res.*, **11**, 889–900.
76. Delcher,A.L., Kasif,S., Fleischmann,R.D., Peterson,J., White,O. and Salzberg,S.L. (1999) Alignment of whole genomes. *Nucleic Acids Res.*, **27**, 2369–2376.
77. Kent,W.J. and Zahler,A.M. (2000) Conservation, regulation, synteny, and introns in a large-scale *C. briggsae–C. elegans* genomic alignment. *Genome Res.*, **10**, 1115–1125.
78. Schwartz,S., Zhang,Z., Frazer,K., Smit,A., Riemer,C., Bouck,J., Gibbs,R., Hardison,R. and Miller,W. (2000) PipMaker—a web server for aligning two genomic DNA sequences. *Genome Res.*, **10**, 577–586.
79. Morgenstern,B. (2000) A space-efficient algorithm for aligning large genomic sequences. *Bioinformatics*, **16**, 948–949.
80. Batzoglou,S., Pachter,L., Mesirov,J., Berger,B. and Lander,E.S. (2000) Human and mouse gene structure: comparative analysis and application to exon prediction. *Genome Res.*, **10**, 950–958.
81. Bafna,V. and Huson,D. (2000) The conserved exon method for gene finding. In Bourne,P., Gribskov,M., Altman,R., Jensen,N., Hope,D., Lengauer,T., Mitchell,J., Scheeff,E., Smith,C., Strande,S. and Weissig,H. (eds), *Eighth International Conference on Intelligent Systems for Molecular Biology*. AAAI Press, Menlo Park, CA, pp. 3–12.
82. Wiehe,T., Gebauer-Jung,S., Mitchell-Olds,T. and Guigo,R. (2001) SGP-1: prediction and validation of homologous genes based on sequence alignments. *Genome Res.*, **11**, 1574–1583.
83. Novichkov,P.S., Gelfand,M.S. and Mironov,A.A. (2001) Gene recognition in eukaryotic DNA by comparison of genomic sequences. *Bioinformatics*, **17**, 1011–1018.
84. Blayo,P., Rouzé,P. and Sagot,M.-F. (2002) Orphan gene finding—an exon assembly approach. *Theor. Comput. Sci.*, in press.

85. Pachter,L., Alexandersson,M. and Cawley,S. (2002) Applications of generalized pair hidden Markov models to alignment and gene finding problems. *J. Comput. Biol.*, **9**, 389–399.
86. Jurka,J., Klonowski,P., Dagman,V. and Pelton,P. (1996) CENSOR—a program for identification and elimination of repetitive elements from DNA sequences. *Comput. Chem.*, **20**, 119–112.
87. Roytberg,M.A., Astakhova,T.V. and Gelfand,M.S. (1997) Combinatorial approaches to gene recognition. *Comput. Chem.*, **21**, 229–235.
88. Guigó,R. (1998) Assembling genes from predicted exons in linear time with dynamic programming. *J. Comput. Biol.*, **5**, 681–702.
89. Guigó,R., Knudsen,S., Drake,N. and Smith,T. (1992) Prediction of gene structure. *J. Mol. Biol.*, **226**, 141–157.
90. Xu,Y., Mural,R.J. and Uberbaker,E.C. (1994) Constructing gene models from accurately predicted exons: an application of dynamic programming. *Comput. Appl. Biosci.*, **10**, 613–623.
91. Chuang,J.S. and Roth,D. (2001) Gene recognition based on DAG shortest paths. *Bioinformatics*, **1**, 1–9.
92. Kleffe,J., Hermann,K., Vahrson,W., Wittig,B. and Brendel,V. (1998) GeneGenerator–a flexible algorithm for gene prediction and its application to maize sequences. *Bioinformatics*, **14**, 232–243.
93. Viterbi,A. (1967) Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Trans. Informat. Theor.*, **IT-13**, 260–269.
94. Bellman,R.E. (1957) *Dynamic Programming*. Princeton University Press, Princeton, NJ.
95. Krogh,A., Mian,I.S. and Haussler,D. (1994) A hidden Markov model that finds genes in *E. coli* DNA. *Nucleic Acids Res.*, **22**, 4768–4778.
96. Kulp,D., Haussler,D., Reese,M.G. and Eeckman,F.H. (1996) A generalized Hidden Markov Model for the recognition of human genes in DNA. In States,D.J., Agarwal,P., Gaasterland,T., Hunter,L. and Smith,R.F. (eds), *Proceedings of the Fourth International Conference on Intelligent Systems for Molecular Biology*. AAAI Press, Menlo Park, CA, pp. 134–142.
97. Hooper,P., Zhang,H. and Wishart,D. (2000) Prediction of genetic structure in eukaryotic DNA using reference point logistic regression and sequence alignment. *Bioinformatics*, **16**, 425–438.
98. Krogh,A. (1997) Two methods for improving performace of a HMM and their application for gene finding. In Gaasterland,T., Karp,P., Karplus,K., Ouzounis,C., Sander,C. and Valencia,A. (eds), *The Fifth International Conference on Intelligent Systems for Molecular Biology*. AAAI Press, Menlo Park, CA, pp. 179–186.
99. Yeh,R.-F., Lim,L.P. and Burge,C.B. (2001) Computational inference of homologous gene structures in the human genome. *Genome Res.*, **11**, 803–816.
100. Korf,I., Flicek,P., Duan,D. and Brent,M.R. (2001) Integrating genomic homology into gene structure prediction. *Bioinformatics*, **17**, S140–S148.
101. Murakami,K. and Tagaki,T. (1998) Gene recognition by combination of several gene-finding programs. *Bioinformatics*, **14**, 665–675.
102. Solovyev,V.V. and Salamov,A.A. (1999) INFOGENE: a database of known gene structures and predicted genes and proteins in sequences of genome sequencing projects. *Nucleic Acids Res.*, **27**, 248–250.
103. Pavlovic,V., Garg,A. and Kasif,S. (2002) A Bayesian framework for combining gene predictions. *Bioinformatics*, **18**, 19–27.
104. Harris,N.L. (1997) Genotator: a workbench for sequence annotation. *Genome Res.*, **7**, 754–762.
105. Gaasterland,T. and Sensen,C.W. (1996) Fully automated genome analysis that reflects user needs and preferences. A detailed introduction to the MAGPIE system architecture. *Biochimie*, **78**, 302–310.
106. Hubbard,T., Barker,D., Birney,E., Cameron,G., Chen,Y., Clark,L., Cox,T., Cuff,J., Curwen,V., Down,T. *et al.* (2002) The Ensembl genome database project. *Nucleic Acids Res.*, **30**, 38–41.
107. Tabaska,J., Davuluri,R. and Zhang,M. (2001) Identifying the 3′-terminal exon in human DNA. *Bioinformatics*, **17**, 602–607.
108. Davuluri,R.V., Grosse,I. and Zhang,M.Q. (2001) Computational identification of promoters and first exons in the human genome. *Nature Genet.*, **29**, 412–417.
109. Down,T.A. and Hubbard,T.J. (2002) Computational detection and location of transcription start sites in mammalian genomic DNA. *Genome Res.*, **12**, 458–461.
110. Graber,J.H., Cantor,C.R., Mohr,S.C. and Smith,T.F. (1999) *In silico* detection of control signals: mRNA 3′-end-processing sequences in diverse species. *Proc. Natl Acad. Sci. USA*, **96**, 14055–14060.
111. Guigó,R., Agarwal,P., Abril,J., Burset,M. and Fickett,J. (2000) An assessment of gene prediction accuracy in large DNA sequences. *Genome Res.*, **10**, 1631–1642.
112. Nobile,C., Marchi,J., Nigro,V., Roberts,R.G. and Danieli,G.A. (1997) Exon-intron organization of the human dystrophin gene. *Genomics*, **45**, 421–424.
113. Duret,L., Dorkeld,F. and Gautier,C. (1993) Strong conservation of vertebrate non-coding sequences during vertebrate evolution: potential involvement in post-transcriptional regulation of gene expression. *Nucleic Acids Res.*, **21**, 2315–2322.
114. Jareborg,N., Birney,E. and Durbin,R. (1999) Comparative analysis of noncoding regions of 77 orthologous mouse and human gene pairs. *Genome Res.*, **9**, 815–824.
115. Quesada,V., Ponce,M.R. and Micol,J.L. (1999) OTC and AUL1, two convergent and overlapping genes in the nuclear genome of *Arabidopsis thaliana*. *FEBS Lett.*, **461**, 101–106.
116. Henikoff,S., Keene,M.A., Fechtel,K. and Fristrom,J.W. (1986) Gene within a gene: nested *Drosophila* genes encode unrelated proteins on opposite DNA strands. *Cell*, **44**, 33–42.
117. Leader,D.J., Clark,G.P., Watters,J., Beven,A.F., Shaw,P.J. and Brown,J.W. (1997) Clusters of multiple different small nucleolar RNA genes in plants are expressed as and processed from polycistronic pre-snoRNAs. *EMBO J.*, **16**, 5742–5751.
118. Blumenthal,T. (1998) Gene clusters and polycistronic transcription in eukaryotes. *Bioessays*, **20**, 480–487.
119. Mironov,A.A., Novichkov,P.S. and Gelfand,M.S. (2001) Pro-Frame: similarity-based gene recognition in eukaryotic DNA sequences with errors. *Bioinformatics*, **17**, 13–15.
120. Fichant,G.A. and Quentin,Y. (1995) A frameshift error detection algorithm for DNA sequencing projects. *Nucleic Acids Res.*, **23**, 2900–2908.
121. Salanoubat,M., Genin,S., Artiguenave,F., Gouzy,J., Mangenot,S., Arlat,M., Billault,A., Brottier,P., Camus,J.C., Cattolico,L. *et al.* (2002) Genome sequence of the plant pathogen *Ralstonia solanacearum*. *Nature*, **415**, 497–502.
122. Iseli,C., Jongeneel,C.V. and Bucher,P. (1999) ESTScan: a program for detecting, evaluating, and reconstructing potential coding regions in EST sequences. *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, 138–148.
123. Klein,M., Pieri,I., Uhlmann,F., Pfizenmaier,K. and Eisel,U. (1998) Cloning and characterization of promoter and 5′-UTR of the NMDA receptor subunit epsilon 2: evidence for alternative splicing of 5′-non-coding exon. *Gene*, **208**, 259–269.
124. Sharp,P.A. and Burge,C.B. (1997) Classification of introns: U2-type or U12-type. *Cell*, **91**, 875–879.
125. Burset,M., Seledtsov,I. and Solovyev,V. (2000) Analysis of canonical and non-canonical splice sites in mammalian genomes. *Nucleic Acids Res.*, **28**, 4364–4375.
126. Croft,L., Schandorff,S., Clark,F., Burrage,K., Arctander,P. and Mattick,J. (2000) ISIS, the intron information system, reveals the high frequency of alternative splicing in the human genome. *Nature Genet.*, **24**, 340–341.
127. Modrek,B., Resch,A., Grasso,C. and Lee,C. (2001) Genome-wide detection of alternative splicing in expressed sequences of human genes. *Nucleic Acids Res.*, **29**, 2850–2859.
128. Modrek,B. and Lee,C. (2002) A genomic view of alternative splicing. *Nature Genet.*, **30**, 13–19.
129. Hastings,M.L. and Krainer,A.R. (2001) Pre-mRNA splicing in the new millennium. *Curr. Opin. Cell Biol.*, **13**, 302–309.
130. Gautheret,D., Poirot,O., Lopez,F., Audic,S. and Claverie Jean,M. (1998) Alternate polyadenylation in human mRNAs: a large-scale analysis by EST clustering. *Genome Res.*, **8**, 524–530.
131. Kozak,M. (1999) Initiation of translation in prokaryotes and eukaryotes. *Gene*, **234**, 187–208.
132. Riechmann,J.L., Ito,T. and Meyerowitz,E. (1999) Non-AUG initiation of AGAMOUS mRNA translation in *Arabidpsis thaliana*. *Mol. Cell. Biol.*, **19**, 8505–8512.
133. Vagner,S., Touriol,C., Galy,B., Audigier,S., Gensac,M., Amalric,F., Bayard,F., Prats,H. and Prats,A. (1996) Translation of CUG- but not AUG-initiated forms of human fibroblast growth factor 2 is activated in transformed and stressed cells. *J. Cell Biol.*, **135**, 1391–1402.
134. Audic,S. and Claverie,J.-M. (1998) Self-identification of protein-coding regions in microbial genomes. *Proc. Natl Acad. Sci. USA*, **95**, 10026–10031.

135. Besemer,J. and Borodovsky,M. (1999) Heuristic approach to deriving models for gene finding. *Nucleic Acids Res.*, **27**, 3911–3920.
136. Médigue,C., Rouxel,T., Vigier,P., Hénaut,A. and Danchin,A. (1991) Evidence for horizontal gene transfer in *Escherichia coli* speciation. *J. Mol. Biol.*, **222**, 851–856.
137. Mathé,C., Peresetsky,A., Déhais,P., Van Montagu,M. and Rouzé,P. (1999) Classification of *Arabidopsis thaliana* gene sequences: clustering of coding sequences into two groups according to codon usage improves gene prediction. *J. Mol. Biol.*, **285**, 1977–1991.
138. Borodovsky,M., McIninch,J.D., Koonin,E.V., Rudd,K.E., Médigue,C. and Danchin,A. (1995) Detection of new genes in a bacterial genome using Markov models for three gene classes. *Nucleic Acids Res.*, **23**, 3554–3562.
139. Hayes,W.S. and Borodovsky,M. (1998) How to interpret an anonymous bacterial genome: machine learning approach to gene identification. *Genome Res.*, **8**, 1154–1171.
140. Besemer,J., Lomsadze,A. and Borodovsky,M. (2001) GeneMarkS: a self-training method for prediction of gene starts in microbial genomes. Implications for finding sequence motifs in regulatory regions. *Nucleic Acids Res.*, **29**, 2607–2618.
141. Mathé,C., Déhais,P., Pavy,N., Rombauts,S., Van Montagu,M. and Rouzé,P. (2000) Gene prediction and gene classes in *Arabidopsis thaliana. J. Biotechnol.*, **78**, 293–299.
142. Pennisi,E. (1999) Keeping genome databases clean and up to date. *Science*, **286**, 447–450.
143. Smith,T.F. (1998) Functional genomics–bioinformatics is ready for the challenge. *Trends Genet.*, **14**, 291–293.
144. The Gene Ontology Consortium (2001) Creating the gene ontology resource: design and implementation. *Genome Res.*, **11**, 1425–1433.
145. Brazma,A. (2001) On the importance of standardisation in life sciences. *Bioinformatics*, **17**, 113–114.
146. Miller,W. (2001) Comparison of genomic DNA sequences: solved and unsolved problems. *Bioinformatics*, **17**, 391–397.
147. Makalowski,W. (2000) Genomic scrap yard: how genomes utilize all that junk. *Gene*, **259**, 61–67.
148. Bergman,C. and Kreitman,M. (2001) Analysis of conserved noncoding DNA in *Drosophila* reveals similar constraints in intergenic and intronic sequences. *Genome Res.*, **11**, 1335–1345.
149. Eddy,S.R. (1999) Noncoding RNA genes. *Curr. Opin. Genet. Dev.*, **9**, 695–699.
150. Erdmann,V., Szymanski,M., Hochberg,A., Groot,N. and Barciszewski,J. (2000) Non-coding, mRNA-like RNAs database Y2K. *Nucleic Acids Res.*, **28**, 197–200.
151. Rivas,E. and Eddy,S. (2000) Secondary structure alone is generally not statistically significant for the detection of noncoding RNAs. *Bioinformatics*, **16**, 583–605.
152. Pertea,M., Lin,X. and Salzberg,S. (2001) GeneSplicer: a new computational method for splice site prediction. *Nucleic Acids Res.*, **29**, 1185–1190.
153. Brendel,V., Kleffe,J., Carle Urioste,J.C. and Walbot,V. (1998) Prediction of splice sites in plant pre-mRNA from sequence properties. *J. Mol. Biol.*, **276**, 85–104.
154. Dong,S. and Searls,D.B. (1994) Gene structure prediction by linguistic methods. *Genomics*, **23**, 540–551.
155. Xu,Y.X. and Uberbacher,E.C. (1997) Automated gene identification in large-scale genomic sequences. *J. Comput. Biol.*, **4**, 325–338.
156. 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.