CHAPTER EIGHT

Gene Prediction

With the rapid accumulation of genomic sequence information, there is a pressing need to use computational approaches to accurately predict gene structure. Computational gene prediction is a prerequisite for detailed functional annotation of genes and genomes. The process includes detection of the location of open reading frames (ORFs) and delineation of the structures of introns as well as exons if the genes of interest are of eukaryotic origin. The ultimate goal is to describe all the genes computationally with near 100% accuracy. The ability to accurately predict genes can significantly reduce the amount of experimental verification work required.

However, this may still be a distant goal, particularly for eukaryotes, because many problems in computational gene prediction are still largely unsolved. Gene prediction, in fact, represents one of the most difficult problems in the field of pattern recognition. This is because coding regions normally do not have conserved motifs. Detecting coding potential of a genomic region has to rely on subtle features associated with genes that may be very difficult to detect.

Through decades of research and development, much progress has been made in prediction of prokaryotic genes. A number of gene prediction algorithms for prokaryotic genomes have been developed with varying degrees of success. Algorithms for eukarytotic gene prediction, however, are still yet to reach satisfactory results. This chapter describes a number of commonly used prediction algorithms, their theoretical basis, and limitations. Because of the significant differences in gene structures of prokaryotes and eukaryotes, gene prediction for each group of organisms is discussed separately. In addition, because of the predominance of protein coding genes in a genome (as opposed to rRNA and tRNA genes), the discussion focuses on the prediction of protein coding sequences.

CATEGORIES OF GENE PREDICTION PROGRAMS

The current gene prediction methods can be classified into two major categories, ab initio–based and homology-based approaches. The ab initio–based approach predicts genes based on the given sequence alone. It does so by relying on two major features associated with genes. The first is the existence of gene signals, which include start and stop codons, intron splice signals, transcription factor binding sites, ribosomal binding sites, and polyadenylation (poly-A) sites. In addition, the triplet codon structure limits the coding frame length to multiples of three, which can be used as a condition for gene prediction. The second feature used by ab initio algorithms is gene content, which is statistical description of coding regions. It has been observed that nucleotide composition and statistical patterns of the coding regions tend to vary significantly from those of the noncoding regions. The unique features can be detected by employing probabilistic models such as Markov models or hidden Markov models (HMMs; see Chapter 6) to help distinguish coding from noncoding regions.

The homology-based method makes predictions based on significant matches of the query sequence with sequences of known genes. For instance, if a translated DNA sequence is found to be similar to a known protein or protein family from a database search, this can be strong evidence that the region codes for a protein. Alternatively, when possible exons of a genomic DNA region match a sequenced cDNA, this also provides experimental evidence for the existence of a coding region.

Some algorithms make use of both gene-finding strategies. There are also a number of programs that actually combine prediction results from multiple individual programs to derive a consensus prediction. This type of algorithms can therefore be considered as consensus based.

GENE PREDICTION IN PROKARYOTES

Prokaryotes, which include bacteria and Archaea, have relatively small genomes with sizes ranging from 0.5 to 10 Mbp (1 Mbp = 10^6 bp). The gene density in the genomes is high, with more than 90% of a genome sequence containing coding sequence. There are very few repetitive sequences. Each prokaryotic gene is composed of a single contiguous stretch of ORF coding for a single protein or RNA with no interruptions within a gene.

More detailed knowledge of the bacterial gene structure can be very useful in gene prediction. In bacteria, the majority of genes have a start codon ATG (or AUG in mRNA; because prediction is done at the DNA level, T is used in place of U), which codes for methionine. Occasionally, GTG and TTG are used as alternative start codons, but methionine is still the actual amino acid inserted at the first position. Because there may be multiple ATG, GTG, or TGT codons in a frame, the presence of these codons at the beginning of the frame does not necessarily give a clear indication of the translation initiation site. Instead, to help identify this initiation codon, other features associated with translation are used. One such feature is the ribosomal binding site, also called the *Shine-Delgarno sequence*, which is a stretch of purine-rich sequence complementary to 16S rRNA in the ribosome (Fig. 8.1). It is located immediately downstream of the transcription initiation site and slightly upstream of the translation start codon. In many bacteria, it has a consensus motif of AGGAGGT. Identification of the ribosome binding site can help locate the start codon.

At the end of the protein coding region is a stop codon that causes translation to stop. There are three possible stop codons, identification of which is straightforward. Many prokaryotic genes are transcribed together as one operon. The end of the operon is characterized by a transcription termination signal called *ρ-independent terminator*. The terminator sequence has a distinct stem-loop secondary structure

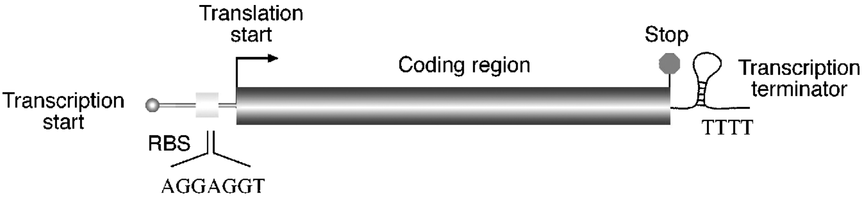

Figure 8.1: Structure of a typical prokaryotic gene structure. *Abbreviation:* RBS, ribosome binding site.

followed by a string of Ts. Identification of the terminator site, in conjunction with promoter site identification (see Chapter 9), can sometimes help in gene prediction.

Conventional Determination of Open Reading Frames

Without the use of specialized programs, prokaryotic gene identification can rely on manual determination of ORFs and major signals related to prokaryotic genes. Prokaryotic DNA is first subject to conceptual translation in all six possible frames, three frames forward and three frames reverse. Because a stop codon occurs in about every twenty codons by chance in a noncoding region, a frame longer than thirty codons without interruption by stop codons is suggestive of a gene coding region, although the threshold for an ORF is normally set even higher at fifty or sixty codons. The putative frame is further manually confirmed by the presence of other signals such as a start codon and Shine–Delgarno sequence. Furthermore, the putative ORF can be translated into a protein sequence, which is then used to search against a protein database. Detection of homologs from this search is probably the strongest indicator of a protein-coding frame.

In the early stages of development of gene prediction algorithms, genes were predicted by examining the nonrandomness of nucleotide distribution. One method is based on the nucleotide composition of the third position of a codon. In a coding sequence, it has been observed that this position has a preference to use G or C over A or T. By plotting the GC composition at this position, regions with values significantly above the random level can be identified, which are indicative of the presence of ORFs (Fig. 8.2). In practice, because genes can be in any of the six frames, the statistical patterns are computed for all possible frames. In addition to codon bias, there is a similar method called TESTCODE (implemented in the commercial GCG package) that exploits the fact that the third codon nucleotides in a coding region tend to repeat themselves. By plotting the repeating patterns of the nucleotides at this position, coding and noncoding regions can be differentiated (see Fig. 8.2). The results of the two methods are often consistent. The two methods are often used in conjunction to confirm the results of each other.

These statistical methods, which are based on empirical rules, examine the statistics of a single nucleotide (either G or C). They identify only typical genes and tend to miss atypical genes in which the rule of codon bias is not strictly followed. To improve the prediction accuracies, the new generation of prediction algorithms use more sophisticated statistical models.

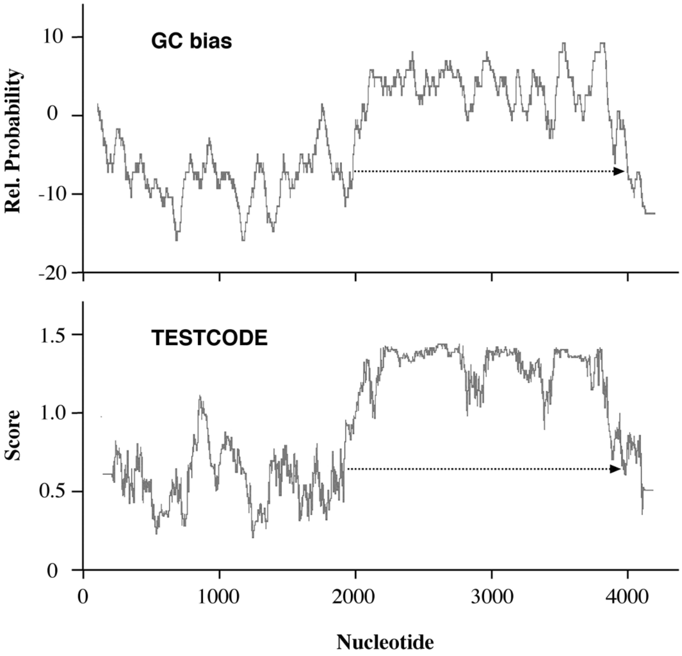

Figure 8.2: Coding frame detection of a bacterial gene using either the GC bias or the TESTCODE method. Both result in similar identification of a reading frame (*dashed arrows*).

Gene Prediction Using Markov Models and Hidden Markov Models

Markov models and HMMs can be very helpful in providing finer statistical description of a gene (see Chapter 6). A Markov model describes the probability of the distribution of nucleotides in a DNA sequence, in which the conditional probability of a particular sequence position depends on k previous positions. In this case, k is the order of a Markov model. A zero-order Markov model assumes each base occurs independently with a given probability. This is often the case for noncoding sequences. A first-order Markov model assumes that the occurrence of a base depends on the base preceding it. A second-order model looks at the preceding two bases to determine which base follows, which is more characteristic of codons in a coding sequence.

The use of Markov models in gene finding exploits the fact that oligonucleotide distributions in the coding regions are different from those for the noncoding regions. These can be represented with various orders of Markov models. Since a fixed-order Markov chain describes the probability of a particular nucleotide that depends on previous k nucleotides, the longer the oligomer unit, the more nonrandomness can be described for the coding region. Therefore, the higher the order of a Markov model, the more accurately it can predict a gene.

Because a protein-encoding gene is composed of nucleotides in triplets as codons, more effective Markov models are built in sets of three nucleotides, describing nonrandom distributions of trimers or hexamers, and so on. The parameters of a Markov model have to be trained using a set of sequences with known gene locations. Once the parameters of the model are established, it can be used to compute the nonrandom

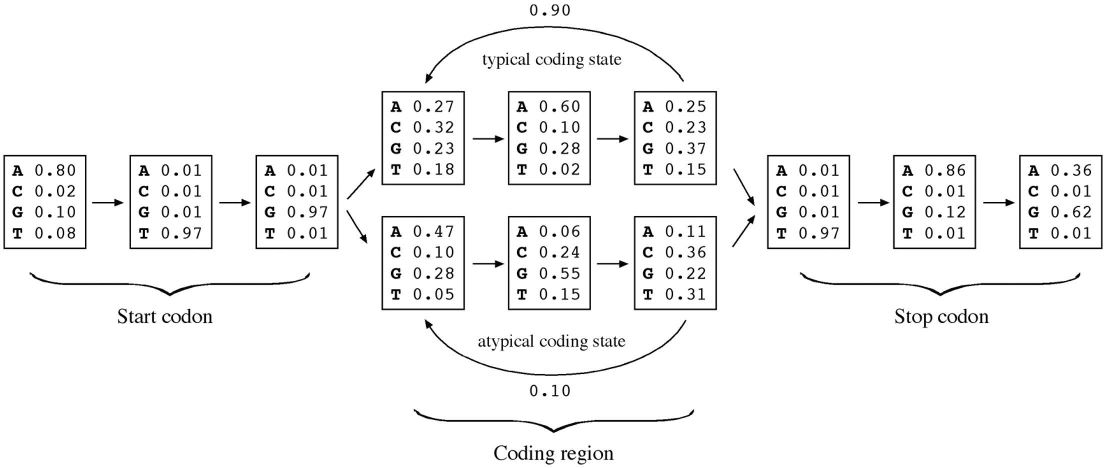

Figure 8.3: A simplified second-order HMM for prokaryotic gene prediction that includes a statistical model for start codons, stop codons, and the rest of the codons in a gene sequence represented by a typical model and an atypical model.

distributions of trimers or hexamers in a new sequence to find regions that are compatible with the statistical profiles in the learning set.

Statistical analyses have shown that pairs of codons (or amino acids at the protein level) tend to correlate. The frequency of six unique nucleotides appearing together in a coding region is much higher than by random chance. Therefore, a fifth-order Markov model, which calculates the probability of hexamer bases, can detect nucleotide correlations found in coding regions more accurately and is in fact most often used.

A potential problem of using a fifth-order Markov chain is that if there are not enough hexamers, which happens in short gene sequences, the method's efficacy may be limited. To cope with this limitation, a variable-length Markov model, called an *interpolated Markov model* (IMM), has been developed. The IMM method samples the largest number of sequence patterns with k ranging from 1 to 8 (dimers to ninemers) and uses a weighting scheme, placing less weight on rare k-mers and more weight on more frequent k-mers. The probability of the final model is the sum of probabilities of all weighted k-mers. In other words, this method has more flexibility in using Markov models depending on the amount of data available. Higher-order models are used when there is a sufficient amount of data and lower-order models are used when the amount of data is smaller.

It has been shown that the gene content and length distribution of prokaryotic genes can be either typical or atypical. Typical genes are in the range of 100 to 500 amino acids with a nucleotide distribution typical of the organism. Atypical genes are shorter or longer with different nucleotide statistics. These genes tend to escape detection using the typical gene model. This means that, to make the algorithm capable of fully describing all genes in a genome, more than one Markov model is needed. To combine different Markov models that represent typical and atypical nucleotide distributions creates an HMM prediction algorithm. A simplified HMM for gene finding is shown in Fig. 8.3.

The following describes a number of HMM/IMM-based gene finding programs for prokaryotic organisms.

GeneMark (http://opal.biology.gatech.edu/GeneMark/) is a suite of gene prediction programs based on the fifth-order HMMs. The main program – GeneMark.hmm – is trained on a number of complete microbial genomes. If the sequence to be predicted is from a nonlisted organism, the most closely related organism can be chosen as the basis for computation. Another option for predicting genes from a new organism is to use a self-trained program GeneMarkS as long as the user can provide at least 100 kbp of sequence on which to train the model. If the query sequence is shorter than 100 kbp, a GeneMark heuristic program can be used with some loss of accuracy. In addition to predicting prokaryotic genes, GeneMark also has a variant for eukaryotic gene prediction using HMM.

Glimmer (Gene Locator and Interpolated Markov Modeler, www.tigr.org/softlab/glimmer/glimmer.html) is a UNIX program from TIGR that uses the IMM algorithm to predict potential coding regions. The computation consists of two steps, namely model building and gene prediction. The model building involves training by the input sequence, which optimizes the parameters of the model. In an actual gene prediction, the overlapping frames are "flagged" to alert the user for further inspection. Glimmer also has a variant, GlimmerM, for eukaryotic gene prediction.

FGENESB (www.softberry.com/berry.phtml?topic=gfindb) is a web-based program that is also based on fifth-order HMMs for detecting coding regions. The program is specifically trained for bacterial sequences. It uses the Vertibi algorithm (see Chapter 6) to find an optimal match for the query sequence with the intrinsic model. A linear discriminant analysis (LDA) is used to further distinguish coding signals from noncoding signals.

These programs have been shown to be reasonably successful in finding genes in a genome. The common problem is imprecise prediction of translation initiation sites because of inefficient identification of ribosomal binding sites. This problem can be remedied by identifying the ribosomal binding site associated with a start codon. A number of algorithms have been developed solely for this purpose. RBSfinder is one such algorithm.

RBSfinder (ftp://ftp.tigr.org/pub/software/RBSfinder/) is a UNIX program that uses the prediction output from Glimmer and searches for the Shine–Delgarno sequences in the vicinity of predicted start sites. If a high-scoring site is found by the intrinsic probabilistic model, a start codon is confirmed; otherwise the program moves to other putative translation start sites and repeats the process.

Performance Evaluation

The accuracy of a prediction program can be evaluated using parameters such as sensitivity and specificity. To describe the concept of sensitivity and specificity accurately, four features are used: true positive (TP), which is a correctly predicted feature; false positive (FP), which is an incorrectly predicted feature; false negative (FN), which is a missed feature; and true negative (TN), which is the correctly predicted absence of

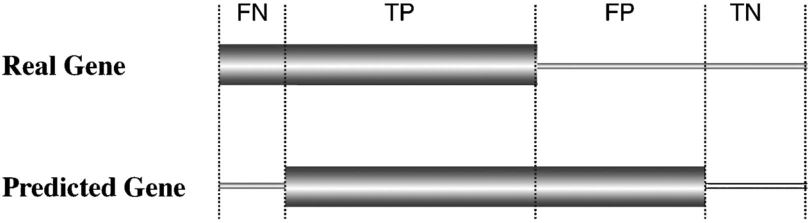

Figure 8.4: Definition of four basic measures of gene prediction accuracy at the nucleotide level. *Abbreviations:* FN, false negative; TP, true positive; FP, false positive; TN, true negative.

a feature (Fig. 8.4). Using these four terms, sensitivity (Sn) and specificity (Sp) can be described by the following formulas:

$$Sn = TP/(TP + FN) \tag{Eq. 8.1}$$

$$Sp = TP/(TP + FP) \tag{Eq. 8.2}$$

According to these formulas, *sensitivity* is the proportion of true signals predicted among all possible true signals. It can be considered as the ability to include correct predictions. In contrast, *specificity* is the proportion of true signals among all signals that are predicted. It represents the ability to exclude incorrect predictions. A program is considered accurate if both sensitivity and specificity are simultaneously high and approach a value of 1. In a case in which sensitivity is high but specificity is low, the program is said to have a tendency to overpredict. On the other hand, if the sensitivity is low but specificity high, the program is too conservative and lacks predictive power.

Because neither sensitivity nor specificity alone can fully describe accuracy, it is desirable to use a single value to summarize both of them. In the field of gene finding, a single parameter known as the correlation coefficient (CC) is often used, which is defined by the following formula:

$$CC = \frac{TP \bullet TN - FP \bullet FN}{\sqrt{(TP + FP)(TN + FN)(FP + TN)}} \tag{Eq. 8.3}$$

The value of the CC provides an overall measure of accuracy, which ranges from −1 to +1, with +1 meaning always correct prediction and −1 meaning always incorrect prediction. Table 8.1 shows a performance analysis using the Glimmer program as an example.

GENE PREDICTION IN EUKARYOTES

Eukaryotic nuclear genomes are much larger than prokaryotic ones, with sizes ranging from 10 Mbp to 670 Gbp (1 Gbp = 10^9 bp). They tend to have a very low gene density. In humans, for instance, only 3% of the genome codes for genes, with about 1 gene per 100 kbp on average. The space between genes is often very large and rich in repetitive sequences and transposable elements.

Most importantly, eukaryotic genomes are characterized by a mosaic organization in which a gene is split into pieces (called *exons*) by intervening noncoding sequences

TABLE 8.1. Performance Analysis of the Glimmer Program for Gene Prediction of Three Genomes

Species	GC (%)	FN	FP	Sensitivity	Specificity
Campylobacter jejuni	30.5	10	19	99.3	98.7
Haemophilus influenzae	38.2	3	54	99.8	96.1
Helicobacter pylori	38.9	6	39	99.5	97.2

Note: The data sets were from three bacterial genomes (Aggarwal and Ramaswamy, 2002).
Abbreviations: FN, false negative; FP, false positive.

(called *introns*) (Fig. 8.5). The nascent transcript from a eukaryotic gene is modified in three different ways before becoming a mature mRNA for protein translation. The first is capping at the 5′ end of the transcript, which involves methylation at the initial residue of the RNA. The second event is splicing, which is the process of removing introns and joining exons. The molecular basis of splicing is still not completely understood. What is known currently is that the splicing process involves a large RNA-protein complex called spliceosome. The reaction requires intermolecular interactions between a pair of nucleotides at each end of an intron and the RNA component of the spliceosome. To make the matter even more complex, some eukaryotic genes can have their transcripts spliced and joined in different ways to generate more than one transcript per gene. This is the phenomenon of alternative splicing. As to be discussed in more detail in Chapter 16, alternative splicing is a major mechanism for generating functional diversity in eukaryotic cells. The third modification is polyadenylation, which is the addition of a stretch of As (~250) at the 3′ end of the RNA.

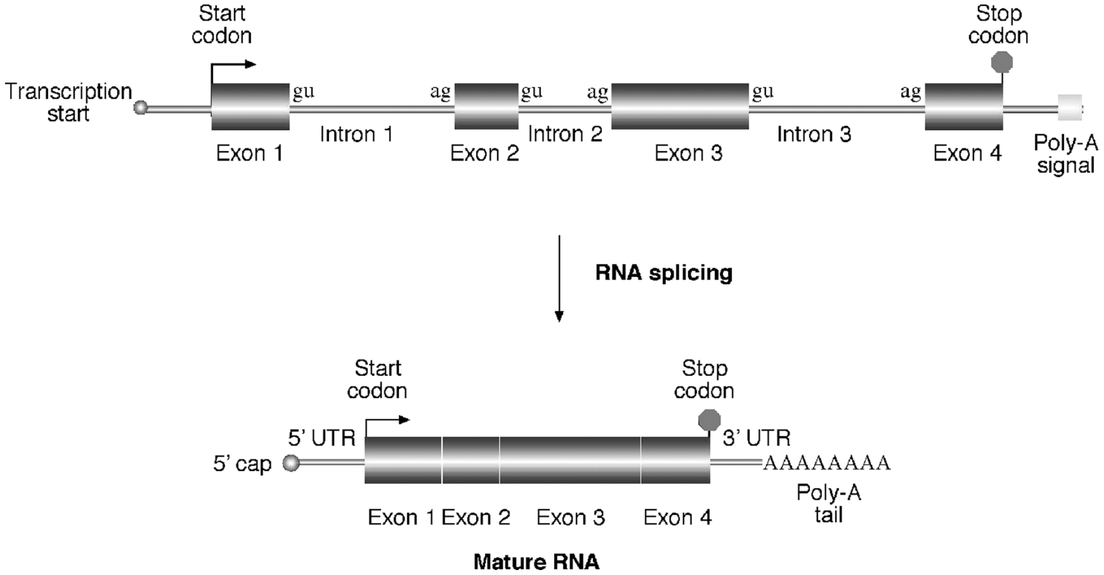

Figure 8.5: Structure of a typical eukaryotic RNA as primary transcript from genomic DNA and as mature RNA after posttranscriptional processing. *Abbreviations:* UTR, untranslated region; poly-A, polyadenylation.

This process is controlled by a poly-A signal, a conserved motif slightly downstream of a coding region with a consensus CAATAAA(T/C).

The main issue in prediction of eukaryotic genes is the identification of exons, introns, and splicing sites. From a computational point of view, it is a very complex and challenging problem. Because of the presence of split gene structures, alternative splicing, and very low gene densities, the difficulty of finding genes in such an environment is likened to finding a needle in a haystack. The needle to be found actually is broken into pieces and scattered in many different places. The job is to gather the pieces in the haystack and reproduce the needle in the correct order.

The good news is that there are still some conserved sequence features in eukaryotic genes that allow computational prediction. For example, the splice junctions of introns and exons follow the GT–AG rule in which an intron at the 5′ splice junction has a consensus motif of GTAAGT; and at the 3′ splice junction is a consensus motif of $(Py)_{12}$NCAG (see Fig. 8.5). Some statistical patterns useful for prokaryotic gene finding can be applied to eukaryotic systems as well. For example, nucleotide compositions and codon bias in coding regions of eukaryotes are different from those of the noncoding regions. Hexamer frequencies in coding regions are also higher than in the noncoding regions. Most vertebrate genes use ATG as the translation start codon and have a uniquely conserved flanking sequence call a *Kozak sequence* (CCGCCATGG). In addition, most of these genes have a high density of CG dinucleotides near the transcription start site. This region is referred to as a CpG island (*p* refers to the phosphodiester bond connecting the two nucleotides), which helps to identify the transcription initiation site of a eukaryotic gene. The poly-A signal can also help locate the final coding sequence.

Gene Prediction Programs

To date, numerous computer programs have been developed for identifying eukaryotic genes. They fall into all three categories of algorithms: ab initio based, homology based, and consensus based. Most of these programs are organism specific because training data sets for obtaining statistical parameters have to be derived from individual organisms. Some of the algorithms are able to predict the most probable exons as well as suboptimal exons providing information for possible alternative spliced transcription products.

Ab Initio–Based Programs

The goal of the ab initio gene prediction programs is to discriminate exons from noncoding sequences and subsequently join the exons together in the correct order. The main difficulty is correct identification of exons. To predict exons, the algorithms rely on two features, gene signals and gene content. Signals include gene start and stop sites and putative splice sites, recognizable consensus sequences such as poly-A sites. *Gene content* refers to coding statistics, which includes nonrandom nucleotide distribution, amino acid distribution, synonymous codon usage, and hexamer frequencies. Among these features, the hexamer frequencies appear to be most discriminative for

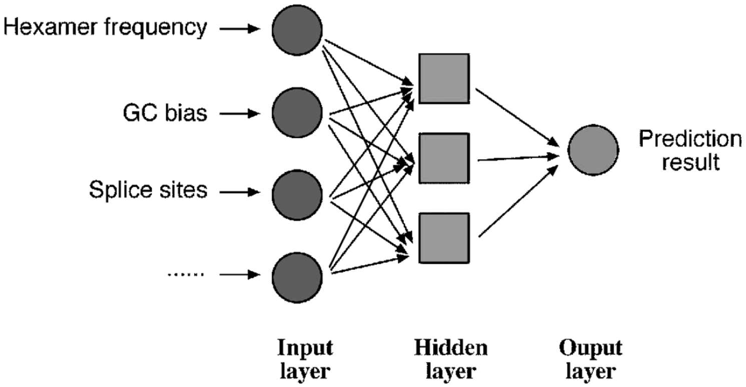

Figure 8.6: Architecture of a neural network for eukaryotic gene prediction.

coding potentials. To derive an assessment for this feature, HMMs can be used, which require proper training. In addition to HMMs, neural network-based algorithms are also common in the gene prediction field. This begs the question of what is a neural network algorithm. A brief introduction is given next.

Prediction Using Neural Networks. A *neural network* (or *artificial neural network*) is a statistical model with a special architecture for pattern recognition and classification. It is composed of a network of mathematical variables that resemble the biological nervous system, with variables or nodes connected by weighted functions that are analogous to synapses (Fig. 8.6). Another aspect of the model that makes it look like a biological neural network is its ability to “learn” and then make predictions after being trained. The network is able to process information and modify parameters of the weight functions between variables during the training stage. Once it is trained, it is able to make automatic predictions about the unknown.

In gene prediction, a neural network is constructed with multiple layers; the input, output, and hidden layers. The input is the gene sequence with intron and exon signals. The output is the probability of an exon structure. Between input and output, there may be one or several hidden layers where the machine learning takes place. The machine learning process starts by feeding the model with a sequence of known gene structure. The gene structure information is separated into several classes of features such as hexamer frequencies, splice sites, and GC composition during training. The weight functions in the hidden layers are adjusted during this process to recognize the nucleotide patterns and their relationship with known structures. When the algorithm predicts an unknown sequence after training, it applies the same rules learned in training to look for patterns associated with the gene structures.

The frequently used ab initio programs make use of neural networks, HMMs, and discriminant analysis, which are described next.

GRAIL (Gene Recognition and Assembly Internet Link; http://compbio.ornl.gov/public/tools/) is a web-based program that is based on a neural network algorithm. The program is trained on several statistical features such as splice junctions, start

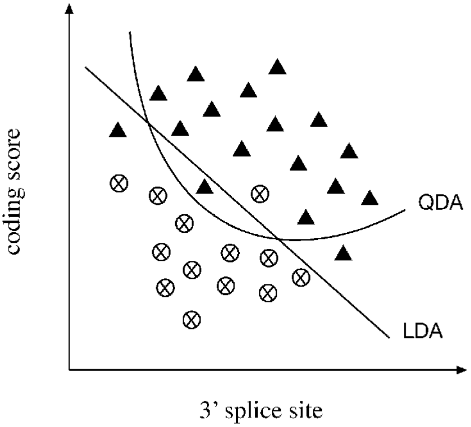

Figure 8.7: Comparison of two discriminant analysis, LDA and QDA. ▲ coding features; ⊗ noncoding features.

and stop codons, poly-A sites, promoters, and CpG islands. The program scans the query sequence with windows of variable lengths and scores for coding potentials and finally produces an output that is the result of exon candidates. The program is currently trained for human, mouse, *Arabidopsis*, *Drosophila,* and *Escherichia coli* sequences.

Prediction Using Discriminant Analysis. Some gene prediction algorithms rely on discriminant analysis, either LDA or quadratic discriminant analysis (QDA), to improve accuracy. LDA works by plotting a two-dimensional graph of coding signals versus all potential 3′ splice site positions and drawing a diagonal line that best separates coding signals from noncoding signals based on knowledge learned from training data sets of known gene structures (Fig. 8.7). QDA draws a curved line based on a quadratic function instead of drawing a straight line to separate coding and noncoding features. This strategy is designed to be more flexible and provide a more optimal separation between the data points.

FGENES (Find Genes; www.softberry.com/) is a web-based program that uses LDA to determine whether a signal is an exon. In addition to FGENES, there are many variants of the program. Some programs, such as FGENESH, make use of HMMs. There are others, such as FGENESH_C, that are similarity based. Some programs, such as FGENESH+, combine both ab initio and similarity-based approaches.

MZEF (Michael Zhang's Exon Finder; http://argon.cshl.org/genefinder/) is a web-based program that uses QDA for exon prediction. Despite the more complex mathematical functions, the expected increase in performance has not been obvious in actual gene prediction.

Prediction Using HMMs. GENSCAN (http://genes.mit.edu/GENSCAN.html) is a web-based program that makes predictions based on fifth-order HMMs. It combines hexamer frequencies with coding signals (initiation codons, TATA box, cap site, poly-A, etc.) in prediction. Putative exons are assigned a probability score (P) of being a true exon. Only predictions with $P > 0.5$ are deemed reliable. This program is trained for sequences from vertebrates, *Arabidopsis,* and maize. It has been used extensively in annotating the human genome (see Chapter 17).

HMMgene (www.cbs.dtu.dk/services/HMMgene) is also an HMM-based web program. The unique feature of the program is that it uses a criterion called the *conditional maximum likelihood* to discriminate coding from noncoding features. If a sequence already has a subregion identified as coding region, which may be based on similarity with cDNAs or proteins in a database, these regions are locked as coding regions. An HMM prediction is subsequently made with a bias toward the locked region and is extended from the locked region to predict the rest of the gene coding regions and even neighboring genes. The program is in a way a hybrid algorithm that uses both ab initio-based and homology-based criteria.

Homology-Based Programs

Homology-based programs are based on the fact that exon structures and exon sequences of related species are highly conserved. When potential coding frames in a query sequence are translated and used to align with closest protein homologs found in databases, near perfectly matched regions can be used to reveal the exon boundaries in the query. This approach assumes that the database sequences are correct. It is a reasonable assumption in light of the fact that many homologous sequences to be compared with are derived from cDNA or expressed sequence tags (ESTs) of the same species. With the support of experimental evidence, this method becomes rather efficient in finding genes in an unknown genomic DNA.

The drawback of this approach is its reliance on the presence of homologs in databases. If the homologs are not available in the database, the method cannot be used. Novel genes in a new species cannot be discovered without matches in the database. A number of publicly available programs that use this approach are discussed next.

GenomeScan (http://genes.mit.edu/genomescan.html) is a web-based server that combines GENSCAN prediction results with BLASTX similarity searches. The user provides genomic DNA and protein sequences from related species. The genomic DNA is translated in all six frames to cover all possible exons. The translated exons are then used to compare with the user-supplied protein sequences. Translated genomic regions having high similarity at the protein level receive higher scores. The same sequence is also predicted with a GENSCAN algorithm, which gives exons probability scores. Final exons are assigned based on combined score information from both analyses.

EST2Genome (http://bioweb.pasteur.fr/seqanal/interfaces/est2genome.html) is a web-based program purely based on the sequence alignment approach to define intron–exon boundaries. The program compares an EST (or cDNA) sequence with a genomic DNA sequence containing the corresponding gene. The alignment is done using a dynamic programming–based algorithm. One advantage of the approach is the ability to find very small exons and alternatively spliced exons that are very difficult to predict by any ab initio–type algorithms. Another advantage is that there is no need for model training, which provides much more flexibility for gene prediction. The limitation is that EST or cDNA sequences often contain errors or even introns if the transcripts are not completely spliced before reverse transcription.

SGP-1 (Syntenic Gene Prediction; http://195.37.47.237/sgp-1/) is a similarity-based web program that aligns two genomic DNA sequences from closely related organisms. The program translates all potential exons in each sequence and does pairwise alignment for the translated protein sequences using a dynamic programming approach. The near-perfect matches at the protein level define coding regions. Similar to EST2Genome, there is no training needed. The limitation is the need for two homologous sequences having similar genes with similar exon structures; if this condition is not met, a gene escapes detection from one sequence when there is no counterpart in another sequence.

TwinScan (http://genes.cs.wustl.edu/) is also a similarity-based gene-finding server. It is similar to GenomeScan in that it uses GenScan to predict all possible exons from the genomic sequence. The putative exons are used for BLAST searching to find closest homologs. The putative exons and homologs from BLAST searching are aligned to identify the best match. Only the closest match from a genome database is used as a template for refining the previous exon selection and exon boundaries.

Consensus-Based Programs

Because different prediction programs have different levels of sensitivity and specificity, it makes sense to combine results of multiple programs based on consensus. This idea has prompted development of consensus-based algorithms. These programs work by retaining common predictions agreed by most programs and removing inconsistent predictions. Such an integrated approach may improve the specificity by correcting the false positives and the problem of overprediction. However, since this procedure punishes novel predictions, it may lead to lowered sensitivity and missed predictions. Two examples of consensus-based programs are given next.

GeneComber (www.bioinformatics.ubc.ca/genecomber/index.php) is a web server that combines HMMgene and GenScan prediction results. The consistency of both prediction methods is calculated. If the two predictions match, the exon score is reinforced. If not, exons are proposed based on separate threshold scores.

DIGIT (http://digit.gsc.riken.go.jp/cgi-bin/index.cgi) is another consensus-based web server. It uses prediction from three ab initio programs – FGENESH, GENSCAN, and HMMgene. It first compiles all putative exons from the three gene-finders and assigns ORFs with associated scores. It then searches a set of exons with the highest additive score under the reading frame constraints. During this process, a Bayesian procedure and HMMs are used to infer scores and search the optimal exon set which gives the final designation of gene structure.

Performance Evaluation

Because of extra layers of complexity for eukaryotic gene prediction, the sensitivity and specificity have to be defined on the levels of nucleotides, exons, and entire genes.

TABLE 8.2. Accuracy Comparisons for a Number of Ab Initio Gene Prediction Programs at Nucleotide and Exon Levels

	Nucleotide level			Exon level				
	Sn	Sp	CC	Sn	Sp	(Sn + Sp)/2	ME	WE
FGENES	0.86	0.88	0.83	0.67	0.67	0.67	0.12	0.09
GeneMark	0.87	0.89	0.83	0.53	0.54	0.54	0.13	0.11
Genie	0.91	0.90	0.88	0.71	0.70	0.71	0.19	0.11
GenScan	0.95	0.90	0.91	0.70	0.70	0.70	0.08	0.09
HMMgene	0.93	0.93	0.91	0.76	0.77	0.76	0.12	0.07
Morgan	0.75	0.74	0.74	0.46	0.41	0.43	0.20	0.28
MZEF	0.70	0.73	0.66	0.58	0.59	0.59	0.32	0.23

Note: The data sets used were single mammalian gene sequences (performed by Sanja Rogic, from www.cs.ubc.ca/∼rogic/evaluation/tablesgen.html.

Abbreviations: Sn, sensitivity; Sp, specificity; CC, correlation coefficient; ME, missed exons; WE, wrongly predicted exons.

The sensitivity at the exon and gene level is the proportion of correctly predicted exons or genes among actual exons or genes. The specificity at the two levels is the proportion of correctly predicted exons or genes among all predictions made. For exons, instead of using CC, an average of sensitivity and specificity at the exon level is used instead. In addition, the proportion of missed exons and missed genes as well as wrongly predicted exons and wrong genes, which have no overlaps with true exons or genes, often have to be indicated.

By introducing these measures, the criteria for prediction accuracy evaluation become more stringent (Table 8.2). For example, a correct exon requires all nucleotides belonging to the exon to be predicted correctly. For a correctly predicted gene, all nucleotides and all exons have to be predicted correctly. One single error at the nucleotide level can negate the entire gene prediction. Consequently, the accuracy values reported on the levels of exons and genes are much lower than those for nucleotides.

When a new gene prediction program is published, the accuracy level is usually reported. However, the reported performance should be treated with caution because the accuracy is usually estimated based on particular datasets, which may have been optimized for the program. The datasets used are also mainly composed of short genomic sequences with simple gene structures. When the programs are used in gene prediction for truly unknown eukaryotic genomic sequences, the accuracy can become much lower. Because of the lack of unbiased and realistic datasets and objective comparison for eukaryotic gene prediction, it is difficult to know the true accuracy of the current prediction tools.

At present, no single software program is able to produce consistent superior results. Some programs may perform well on certain types of exons (e.g., internal or single exons) but not others (e.g., initial and terminal exons). Some are sensitive to the G-C content of the input sequences or to the lengths of introns and exons. Most programs make overpredictions when genes contain long introns. In sum, they all suffer from the problem of generating a high number of false positives and false negatives. This is especially true for ab initio–based algorithms. For complex genomes such as the human genome, most popular programs can predict no more than 40% of the genes exactly right. Drawing consensus from results by multiple prediction programs may enhance performance to some extent.

SUMMARY

Computational prediction of genes is one of the most important steps of genome sequence analysis. For prokaryotic genomes, which are characterized by high gene density and noninterrupted genes, prediction of genes is easier than for eukaryotic genomes. Current prokaryotic gene prediction algorithms, which are based on HMMs, have achieved reasonably good accuracy. Many difficulties still persist for eukaryotic gene prediction. The difficulty mainly results from the low gene density and split gene structure of eukaryotic genomes. Current algorithms are either ab initio based, homology based, or a combination of both. For ab initio–based eukaryotic gene prediction, the HMM type of algorithm has overall better performance in differentiating intron–exon boundaries. The major limitation is the dependency on training of the statistical models, which renders the method to be organism specific. The homology-based algorithms in combination with HMMs may yield improved accuracy. The method is limited by the availability of identifiable sequence homologs in databases. The combined approach that integrates statistical and homology information may generate further improved performance by detecting more genes and more exons correctly. With rapid advances in computational techniques and understanding of the splicing mechanism, it is hoped that reliable eukaryotic gene prediction can become more feasible in the near future.

FURTHER READING

Aggarwal, G., and Ramaswamy, R. 2002. Ab initio gene identification: Prokaryote genome annotation with GeneScan and GLIMMER. *J. Biosci.* 27:7–14.

Ashurst, J. L., and Collins, J. E. 2003. Gene annotation: Prediction and testing. *Annu. Rev. Genomics Hum. Genet.* 4:69–88.

Azad, R. K., and Borodovsky, M. 2004. Probabilistic methods of identifying genes in prokaryotic genomes: Connections to the HMM theory. *Brief. Bioinform.* 5:118–30.

Cruveiller, S., Jabbari, K., Clay, O., and Bemardi, G. 2003. Compositional features of eukaryotic genomes for checking predicted genes. *Brief. Bioinform.* 4:43–52.

Davuluri, R. V., and Zhang, M. Q. 2003. "Computer software to find genes in plant genomic DNA." In *Plant Functional Genomics*, edited by E. Grotewold, 87–108. Totowa, NJ: Human Press.

Guigo, R., and Wiehe, T. 2003. "Gene prediction accuracy in large DNA sequences." In *Frontiers in Computational Genomics,* edited by M. Y. Galperin and E. V. Koonin, 1–33. Norfolk, UK: Caister Academic Press.

Guigo, R., Dermitzakis, E. T., Agarwal, P., Ponting, C. P., Parra, G., Reymond, A., Abril, J. F., et al R. 2003. Comparison of mouse and human genomes followed by experimental verification yields an estimated 1,019 additional genes. *Proc. Natl. Acad. Sci. USA* 100:1140–5.

Li, W., and Godzik, A. 2002. Discovering new genes with advanced homology detection. *Trends Biotechnol*. 20:315–16.

Makarov, V. 2002. Computer programs for eukaryotic gene prediction. *Brief. Bioinform*. 3:195–9.

Mathe, C., Sagot, M. F., Schiex, T., and Rouze, P. 2002. Current methods of gene prediction, their strengths and weaknesses. *Nucleic Acids Res*. 30:4103–17.

Parra, G., Agarwal, P., Abril, J. F., Wiehe, T., Fickett, J. W., and Guigo, R. 2003. Comparative gene prediction in human and mouse. *Genome Res*. 13:108–17.

Wang, J., Li, S., Zhang, Y., Zheng, H., Xu, Z., Ye, J., Yu, J., and Wong, G. K. 2003. Vertebrate gene predictions and the problem of large genes. *Nat. Rev. Genet*. 4:741–9.

Wang, Z., Chen, Y., and Li, Y. 2004. A brief review of computational gene prediction methods. *Geno. Prot. Bioinfo*. 4:216–21.

Zhang, M. Q. 2002. Computational prediction of eukaryotic protein coding genes. *Nat. Rev. Genetics*. 3:698–709.