CHAPTER 8

Gene Prediction

With the advent of whole-genome sequencing projects, there is considerable use for computer programs that scan genomic DNA sequences to find genes, particularly those that encode proteins. Once a new genomic sequence has been obtained, the most likely protein-encoding regions are identified and the predicted proteins are then subjected to a database similarity search, as described in the previous chapter. The genomic DNA sequence is then annotated with information on the exon–intron structure and location of each predicted gene along with any functional information based on the database searches. This procedure is summarized in the gene prediction flowchart (p. 346).

In this chapter, I first discuss methods of predicting the genes that encode proteins and then the identification of sequences, such as promoters, that regulate the activity of protein-encoding genes. The prediction of genes that specify classes of RNA molecules is discussed in Chapter 5. The organization of genomes is discussed in Chapter 10. There are many computer programs and Web sites for gene prediction, and representative examples are shown in Table 8.1.

INTRODUCTION

The simplest method of finding DNA sequences that encode proteins is to search for open reading frames, or ORFs. An ORF is a length of DNA sequence that contains a contiguous set of codons, each of which specifies an amino acid. There are six possible reading frames in every sequence, three starting at positions 1, 2, and 3 and going in the 5′ to 3′ direction of a given sequence, and another three starting at positions 1, 2, and 3 and going in the 5′ to 3′ direction of the complementary sequence. In prokaryotic genomes, DNA sequences that encode proteins are transcribed into mRNA, and the mRNA is usually translated directly into proteins without significant modification. The longest ORFs running from the first available Met codon on the mRNA to the next stop codon in the same reading frame generally provide a good, but not assured, prediction of the protein-encoding regions (see Table 8.1 for Web sites that provide a more detailed analysis). A reading frame of a genomic sequence that does not encode a protein will have short ORFs due to the presence of many in-frame stop codons. An example of a search of the *Escherichia coli lac* operon for ORFs is shown in Figure 8.1. These predictions have to take into account the observation in *E. coli* and its phages of the presence of multiple genes on mRNA and sometimes of overlapping genes in which two different proteins may be encoded in different reading frames of the same mRNA, either on the same or complementary DNA strands. In eukaryotes, prediction of protein-encoding genes is a more difficult task.

In eukaryotic organisms, transcription of protein-encoding regions initiated at specific promoter sequences is followed by removal of noncoding sequence (introns) from pre-mRNA by a splicing mechanism, leaving the protein-encoding exons. Once the introns have been removed and certain other modifications to the mature RNA have been made, the resulting mature mRNA can be translated in the 5′ to 3′ direction, usually from the first start codon to the first stop codon. As a result of the presence of intron sequences in the genomic DNA sequences of eukaryotes, the ORF corresponding to an encoded gene will be interrupted by the presence of introns that usually generate stop codons.

Three types of posttranscriptional events influence the translation of mRNA into protein and the accuracy of gene prediction. First, the genetic code of a given genome may vary from the universal code (see Table 8.1 for reference Web sites). For the most part, the universal genetic code, shown in Table 8.2, is used.

Table 8.1. *Programs and Web pages for sequence translation and related information*

Name of translation site	Web address	Reference
Arabidopis intron splice site table	http:// www.Arabidopsis.org/splice_site.html	see Web site
Codon usage database	http://www.kazusa.or.jp/codon/	see Web site
Ecoparse for finding *E. coli* genes based on HMM model	mail server described at http://www.cbs.dtu.dk/krogh/EcoParse.info	Krogh et al. (1994)
EST-GENOME for alignment of EST/cDNA and genomic sequences	http://www.hgmp.mrc.ac.uk/Registered/Option/est_genome.html	see Web site; also see Florea et al. (1998)
Exon recognizer, including GeneScope	http://gf.genome.ad.jp/	see Web page
FGENES and related programs that use linear discriminant analysis or hidden Markov models[a]	http://genomic.sanger.ac.uk/gf/gf.shtml	Solovyev et al. (1995); see Web site
FINEX–exon intron boundary analysis	http://www.icnet.uk/LRITu/projects/finex/	Brown et al. (1995)
GeneFinder access site at Baylor College of Medicine	http://dot.imgen.bcm.tmc.edu:9331/gene-finder/gf.html	collection of methods
Genehacker for microbial genomes based on HMMs	http://www-scc.jst.go.jp/sankichi/GeneHacker/	Hirosawa et al. (1997)
GeneID-3 Web server using rule-based models, and GeneID+[b]	http://www1.imim.es/geneid.html Mail server at geneid@darwin.bu.edu	Guigó et al. (1992); Guigó (1998)
GeneMark and GeneMark.hmm[c] uses hidden Markov models	http://genemark.biology.gatech.edu/GeneMark/; http://www2.ebi.ac.uk/genemark/	Lukashin and Borodovsky (1998)
GeneMark home page (see webgenemark)	http://genemark.biology.gatech.edu/GeneMark/	Borodovsky and McIninch (1993)
GeneParser[a,b] Web page, uses combination of neural network and dynamic programming methods	http://beagle.colorado.edu/~eesnyder/GeneParser.html	Snyder and Stormo (1993, 1995)
Genescan using Fourier transform of DNA sequences to find characteristic patterns	http://202.41.10.146/GS.html	Tiwari et al. (1997)
GeneScope	http://gf.genome.ad.jp/genescope/; see Exon recognizer	Murakami and Takagi (1998)
Genetic code variations	http://www.ncbi.nlm.nih.gov/htbin-post/Taxonomy/wprintgc?mode=c	
Genie for finding human genes in 10-kb DNAs and in *Drosophila* by hidden Markov models and neural networks	http://www.cse.ucsc.edu/~dkulp/cgi-bin/genie http://www.fruitfly.org/seq_tools/genie.html http://www.tigem.it/TIGEM/HTML/Genie.html	Kulp et al. (1996); Reese et al. (1997, 2000)
GenLang using linguistic methods	http://www.cbil.upenn.edu/	Dong and Searls (1994)
GenScan based on probabilistic model of gene structure for vertebrate, *Drosophila*, and plant genes	http://genes.mit.edu/GENSCAN.html	Burge and Karlin (1998)
Genseqer for aligning genomic and EST sequences	http://gremlin1.zool.iastate.edu/cgi-bin/gs.cgi	see Web site and Splicepredictor
Glimmer uses interpolated Markov models for prokaryotic translation	http://www.tigr.org/softlab/ and http://www.cs.jhu.edu/labs/compbio/glimmer.html	Salzberg et al. (1998)

Continued.

Table 8.1. *Continued.*

Name of translation site	Web address	Reference
GrailII[a,b] prediction by neural networks based on scores of characteristic sequence patterns and composition	http://compbio.ornl.gov/	Uberbacher and Mural (1991); Uberbacher et al. (1996)
Hexon for exon prediction by linear discriminant analysis	see GeneFinder access site	Solovyev et al. (1994)
Human splice sites with decision tree analysis[d]	http://sol2.ebi.ac.uk/projects/Events/gene/genepred-thanaraj.html	Thanaraj (1999)
INFO for finding splice junctions by database similarity search	http://elcapitan.ucsd.edu/~info/	Laub and Smith (1998)
INFOGENE: a database of known gene structures and predicted genes	http://genomic.sanger.ac.uk/inf/infodb.shtml	Solovyev and Salamov (1999)
Initiation codon analysis	http://www.ncbi.nlm.nih.gov/htbin-post/Taxonomy/wprintgc?mode=c	see Web site
Microbial genome coding region identification based on Markov chains of order 5	http://igs-server.cnrs-mrs.fr/~audic/selfid.html	Audic and Claverie (1998)
Morgan for finding vertebrate genes by decision tree classification[d]	http://www.cs.jhu.edu/labs/compbio/morgan.html	see http://www.cs.jhu.edu/labs/compbio/morgan.html#refs; Salzberg (1998); Searls (1998)
MZEF uses quadratic discriminant analysis for human, mouse, *Arabidopsis*, and *S. pombe* exons	http://argon.cshl.org/genefinder/	Zhang (1997)
NetGene uses neural networks for analysis of splice sites in human, *C. elegans*, and *Arabidopsis* genes	http://www.cbs.dtu.dk/services/NetGene2/	Brunak et al. (1991); Hebsgaard et al. (1996)
NetPlantGene	http://www.cbs.dtu.dk/services/NetPGene/	see NetGene
NetStart uses neural networks for gene prediction in vertebrate and *Arabidopsis* genes	http://www.cbs.dtu.dk/services/NetStart/	Pedersen and Nielsen (1997)
Procrustes based on comparison of related genomic sequences	http://www-hto.usc.edu/software/procrustes/	Gelfand et al. (1996)
Push-button Gene Finder for gene identification using Markov and hidden Markov models	http://www.cse.ucsc.edu/research/compbio/pgf/	see Web site
Splice Predictor for plants uses trained logitlinear models	http://gremlin1.zool.iastate.edu/cgi-bin/sp.cgi	Brendel and Kleffe (1998); Brendel et al. (1998)
Splicing Sites by neural network at LBNL	http://www.fruitfly.org/seq_tools/splice.html	see Genie
Translate tool at ExPASy	http://www.expasy.ch/tools/dna.html	see Web site
Translation machine on the Web at EBI	http://www2.ebi.ac.uk/translate/	see Web site

Table 8.1. *Continued.*

Name of translation site	Web address	Reference
Translation of large genome sequences on the Web	http://alces.med.umn.edu/rawtrans.html	see Web site
Veil (Viterbi exon-intron locator) uses hidden Markov models for vertebrate DNA	http://www.cs.jhu.edu/labs/compbio/veil.html	Henderson et al. (1997)
Webgene, a set of gene prediction tools and concurrent database similarity searches	http://www.itba.mi.cnr.it/webgene/	see Web site
Webgenemark and Webgenemark.hmm[c]	http://genemark.biology.gatech.edu/GeneMark/	see GeneMark; Lukashin and Borodovsky (1998)
Yeast splice sites by M. Ares Jr. laboratory	http://www.cse.ucsc.edu/research/compbio/yeast_introns.html	Spingola et al. (1999)

Abbreviations: (LBNL) Lawrence Berkeley National Laboratory.

Lists of Web sites for gene recognition and splice site prediction with references and program availability are also available at http://linkage.rockefeller.edu/wli/gene/programs.html, http://www.bork.embl-heidelberg.de/genepredict.html, http://www.hgc.ims.u-tokyo.ac.jp/~katsu/genefinding/programs.html and http://www-hto.usc.edu/software/procrustes/links.html. A more detailed list of programs for gene recognition has been prepared (Burset and Guigó 1996).

Performance comparisons are given at http://igs-server.cnrs-mrs.fr/igs/banbury/, http://www.cs.jhu.edu/labs/compbio/veil.html#perf, http://www1.imim.es/courses/SeqAnalysis/GeneIdentification/Evaluation.html, and are also described in many of the references (see, e.g., Snyder and Stormo 1993; Zhang 1997).

[a] Programs that assemble exons into predicted genes.

[b] Prediction can be enhanced through database similarity searches. GeneParser 3 has this option.

[c] The GeneMark.hmm program is designed to use additional information at the 5′ end of bacterial sequences.

[d] A decision tree analysis has features in common with the phlyogenetic analysis described in Chapter 5 and also with the discriminant analysis described in the text. Scorable features of sequences in coding versus noncoding regions are used as a basis for optimally classifying the sequences into sets. Cutoff values for these features are then used as a basis for scoring unknown sequences as coding or noncoding. These criteria are applied in a sequential order much like starting at the root of a tree and passing through a series of nodes. At each node a further criterion is applied that is the basis for moving along one branch from that node and moving to the next node. Eventually, a terminal branch is reached that is labeled with a decision. In this case, the label is a YES if the sequence is coding, a splice site, or whatever test is being applied because it meets the criteria applied in passing through the decision nodes on the tree, or NO, the sequence is not coding, etc., and because it does not meet the applied criteria.

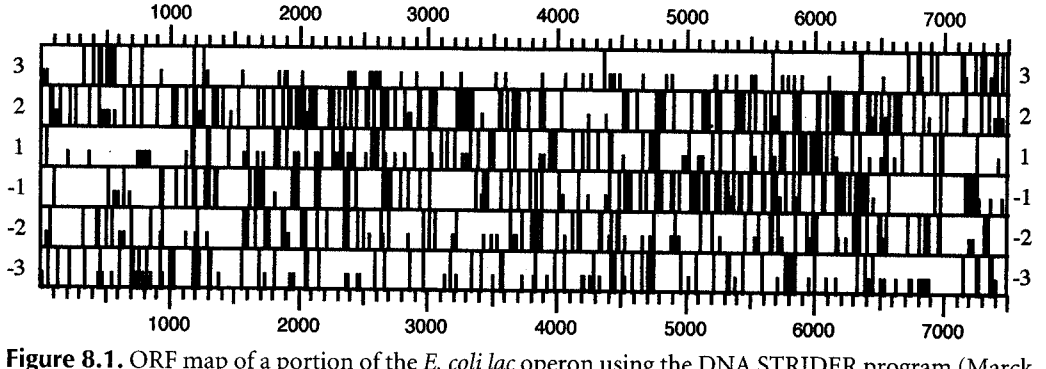


Figure 8.1. ORF map of a portion of the *E. coli lac* operon using the DNA STRIDER program (Marck 1988). Shown are AUG and termination codons as one-half and full vertical bars, respectively, in all six possible reading frames. The *lacZ* gene is visible as an ORF that runs from positions 1284 to 4355 in frame 3.

Table 8.2. *The universal or standard genetic code*

UUU-Phe	F	UCU-Ser	S	UAU-Tyr	Y	UGU-Cys	C
UUC-Phe	F	UCU-Ser	S	UAU-Tyr	Y	UGU-Cys	C
UUA-Leu	L	UCA-Ser	S	UAA-	TER	UGA-	TER
UUG-Leu	L	UCG-Ser	S	UAG-	TER	UGG--Trp	W
CUU-Leu	L	CCU-Pro	P	CAU-His	H	CGU-Arg	R
CUC-Leu	L	CCU-Pro	P	CAU-His	H	CGC-Arg	R
CUA-Leu	L	CCA-Pro	P	CAA-Gln	Q	CGA-Arg	R
CUG-Leu	L	CCG-Pro	P	CAG-Gln	Q	CGG-Arg	R
AUU-Ile	I	ACU-Thr	T	AAU-Asn	N	AGU-Ser	S
AUC-Ile	I	ACC-Thr	T	AAC-Asn	N	AGC-Ser	S
AUA-Ile	I	ACA-Thr	T	AAA-Lys	K	AGA-Arg	R
AUG-MET	M	ACG-Thr	T	AAG-Lys	K	AGG-Arg	R
GUU-Val	V	GCU-Ala	A	GAU-Asp	D	GGU-Gly	G
GUC-Val	V	GCC-Ala	A	GAC-Asp	D	GGC-Gly	G
GUA-Val	V	GCA-Ala	A	GAA-Glu	E	GGA-Gly	G
GUG-Val	V	GCG-Ala	A	GAG-Glu	E	GGG-Gly	G

Shown are each codon and the three-letter and one-letter codes for each encoded amino acid. ATG is the usual START codon and the three TER codons cause translational termination.

It is important to be aware of cellular organelles and organisms in which the genetic code varies so that the correct translation may be made.

Second, one tissue may splice a given mRNA differently from another, thus creating two similar but also partially different mRNAs encoding two related but partially different proteins (Lopez 1998). Understanding the molecular interactions between RNA and the RNA-binding proteins that perform these modifications is an area of active investigation. Availability of this information will assist in the prediction of such variations. Third, mRNAs may be edited, changing the sequence of the mRNA and, as a result, of the encoded protein (see, e.g., Gray and Covello 1993; Paul and Bass 1998; Morse and Bass 1999). Such changes also depend on interaction of RNA with RNA-binding proteins.

TESTING THE RELIABILITY OF AN ORF PREDICTION

DNA sequences that encode protein are not a random chain of available codons for an amino acid, but rather an ordered list of specific codons that reflect the evolutionary origin of the gene and constraints associated with gene expression. This nonrandom property of coding sequences can be used to advantage for finding regions in DNA sequences that encode proteins (see Fickett and Tung 1992). Each species also has a characteristic pattern of use of synonymous codons; i.e., codons that stand for the same amino acid (Table 8.3) (Wada et al. 1992). There are also different patterns of use of codons in strongly versus weakly expressed genes, as, for example, in *E. coli*. Also in *E. coli*, there is a strong preference for certain codon pairs within a coding region and for certain codons to be next to the termination codon. Some of this preference is due to constraints in amino acid sequences in proteins and some to the influence of a given codon on the translation of neighboring codons (Gutman and Hatfield 1989). There is also a strong preference for codon pairs in eukaryotic exons that has been very useful for distinguishing exons and introns in eukaryotic genomic DNAs, as described later in this chapter. Organisms with a high genome content of GC have a strong bias of G and C in the third codon position (for review, see Von Heijne 1987; Rice et al. 1991).

Table 8.3. *Codon usage table*

UUU-Phe	16.6	26.0	UCU-Ser	14.5	23.6	UAU-Tyr	12.1	18.8	UGU-Cys	9.7	8.0
UUC-Leu	20.7	18.2	UCC-Ser	17.7	14.2	UAC-Tyr	16.3	14.7	UGC-Cys	12.4	4.7
UUA-Leu	7.0	26.3	UCA-Ser	11.4	18.8	UAA-TER	0.7	1.0	UGA-TER	1.3	0.6
UUG-Leu	12.0	27.1	UCG-Ser	4.5	8.6	UAG-TER	0.5	0.5	UGG-Trp	13.0	10.3
CUU-Leu	12.4	12.2	CCU-Pro	17.2	13.6	CAU-His	10.1	13.7	CGU-Arg	4.7	6.5
CUC-Leu	19.3	5.4	CCC-Pro	20.3	6.8	CAC-His	14.9	7.8	CGC-Arg	11.0	2.6
CUA-Leu	6.8	13.4	CCA-Pro	16.5	18.2	CAA-Gln	11.8	27.5	CGA-Arg	6.2	3.0
CUG-Leu	40.0	10.4	CCG-Pro	7.1	5.3	CAG-Gln	34.4	12.2	CGG-Arg	11.6	1.7
AUU-Ile	15.7	30.2	ACU-Thr	12.7	20.2	AAU-Asn	16.8	36.0	AGU-Ser	11.7	14.2
AUC-Ile	22.3	17.1	ACC-Thr	19.9	12.6	AAC-Asn	20.2	24.9	AGC-Ser	19.3	9.7
AUA-Ile	7.0	17.8	ACA-Thr	14.7	17.7	AAA-Lys	23.6	42.1	AGA-Arg	11.2	21.3
AUG-MET	22.2	20.9	ACG-Thr	6.4	8.0	AAG-Lys	33.2	30.8	AGG-Arg	11.1	9.3
GUU-Val	10.7	22.0	GCU-Ala	18.4	21.1	GAU-Asp	22.2	37.8	GGU-Gly	10.9	23.9
GUC-Val	14.8	11.6	GCC-Ala	28.6	12.6	GAC-Asp	26.5	20.4	GGC-Gly	23.1	9.7
GUA-Val	6.8	11.7	GCA-Ala	15.6	16.2	GAA-Glu	28.6	45.9	GGA-Gly	16.4	10.9
GUG-Val	29.3	10.7	GCG-Ala	7.7	6.1	GAG-Glu	40.6	19.1	GGG-Gly	16.5	6.0

Shown are frequency of each codon per 100,000 codons obtained from http://www.kazusa.or.jp/codon/ for *Homo sapiens*; columns 2, 5, 8, and 11, and for *Saccharomyces cerevisiae*, columns 3, 6, 9, and 12.

On the basis of these characteristics of protein-encoding sequences, three tests of ORFs have been devised to verify that a predicted ORF is in fact likely to encode a protein (Staden and McLachlan 1982; Staden 1990). The first test is based on an unusual type of sequence variation that is found in ORFs; namely, that every third base tends to be the same one much more often than by chance alone (Fickett 1982). This property is due to nonrandom use of codons in ORFs and is true for any ORF, regardless of the species. No information about nucleotide or codon preference is needed for this analysis. The program TESTCODE, which is available in the Genetics Computer Group suite of programs (http://www.gcg.com), provides a plot of the nonrandomness of every third base in the sequence. An example of TESTCODE output is shown in Figure 8.2. The second test is an analysis to determine whether the codons in the ORF correspond to those used in other genes of the same organism (Staden and McLachlan1982). For this test, information on codon use for an organism is necessary, such as shown in Table 8.3 for human and yeast genes, averaged over all genes. In addition, there may be variations in codon use by different genes of an organism providing a type of gene regulation. An example of the analysis of an *E. coli* gene for the presence of more and less frequently used *E. coli* codons is shown in Figure 8.3. A parameter that reflects the frequency of codon use may also be calculated, as in the Genetics Computer Group CODONFREQUENCY program. Third, the ORF may be translated into an amino acid sequence and the resulting sequence then compared to the databases of existing sequences. If one or more sequences of significant similarity are found, there will be much more confidence in the predicted ORF (Gish and States 1993).

EUKARYOTIC GENES HAVE REPEATED SEQUENCE ELEMENTS THAT PROBABLY REFLECT NUCLEOSOME STRUCTURE

Eukaryotic DNA is wrapped around histone–protein complexes called nucleosomes. As a result, some of the base pairs in the major or minor grooves of the DNA molecule face the nucleosome surface and others face the outside of the structure. Binding sites for some

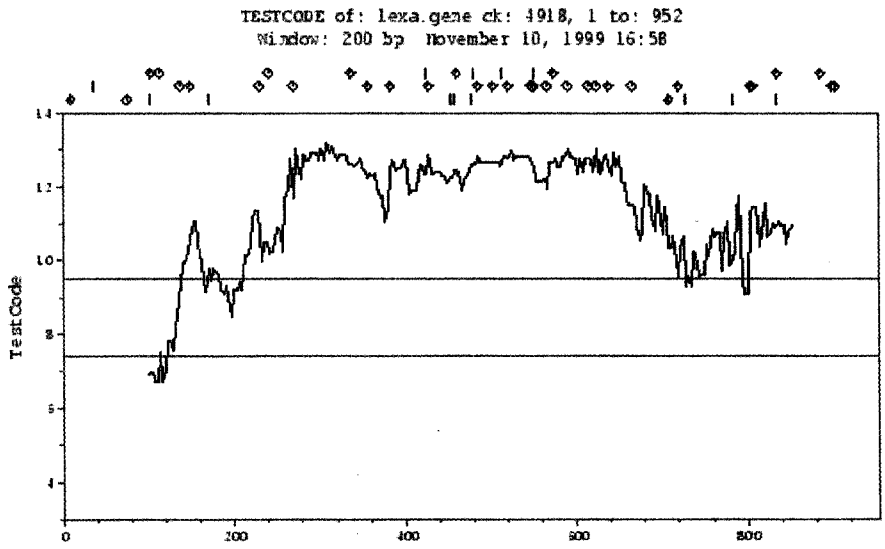


Figure 8.2. TESTCODE analysis of the *E. coli lexA* gene, which is known to extend from positions 102 to 707 in the sequence shown. The TESTCODE statistic (Fickett 1982; for comparison, see Staden 1990) was plotted for each base position in a sliding window of 200 nucleotides. The TESTCODE statistic is found in the following way: (1) The number of each base is counted at every third position starting at positions 1, 2, and 3, and going to the end of the sequence window; (2) the asymmetry statistic for each base is calculated as the ratio of the maximum count of the three possible reading frames divided by the minimum count for the same base plus 1; (3) the frequency of each base in the window is also calculated; (4) the resulting asymmetry and frequency scores are then converted to probabilities of being found in a codon region (found from an analysis of known coding and noncoding regions); and (5) the probabilities are multiplied by weighting factors that are summed. Weighting factors are chosen so that the resulting sum best discriminates coding from noncoding sequences. A value of >0.95 classifies the sequence as coding, and <0.74 classifies the sequence as noncoding. These cutoff values are indicated by red horizontal lines. TESTCODE was run and displayed using TESTCODE in the Genetics Computer Group suite of programs. Above the plot, short vertical lines indicate possible start codons, and diamonds indicate possible stop codons.

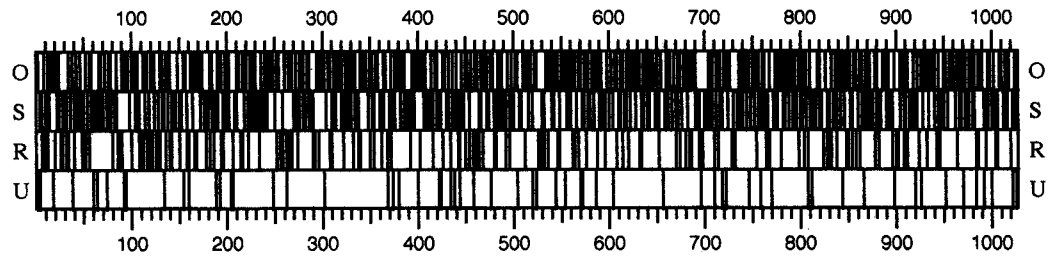


Figure 8.3. Analysis of *E. coli lacZ* gene for occurrence of frequent and infrequent codons using the codon adaptation analysis feature of DNA STRIDER. The positions of common (O for optimum), less common (S for suboptimal), rare (R), and unique (U, which includes the three stop codons, the AUG Met codon and the UGG Trp codon) codons along the sequence are shown, starting at the first nucleotide in the sequence and analyzing three at a time. These first three classes correspond, respectively, to codon adaptation values (Sharp and Li 1987) of >0.9, 0.1–0.9, and <0.1. The gene is obviously represented by commonly used codons.

proteins that regulate transcription may therefore be hidden on the inside of the structure. Nucleosomes located in the promoter region are remodeled in a manner that can influence the availability of binding sites for regulatory proteins, making them more or less available (Carey and Smale 2000).

The computational background of this model is that repeated patterns of sequence have been found in the introns and exons and near the start site of transcription of eukaryotic genes by hidden Markov model (HMM) analysis (Baldi et al. 1996; for a detailed analysis, see Baldi and Brunak 1998; see also Chapter 4, p. 185) and other types of pattern-searching methods (Ioshikhes et al. 1996). These sequences appear to be correlated with the position of nucleosomes and are not found in prokaryotic DNA (Stein and Bina 1999). An example of the HMM is shown in Figure 8.4. These patterns appear with a periodicity of 10; that is,

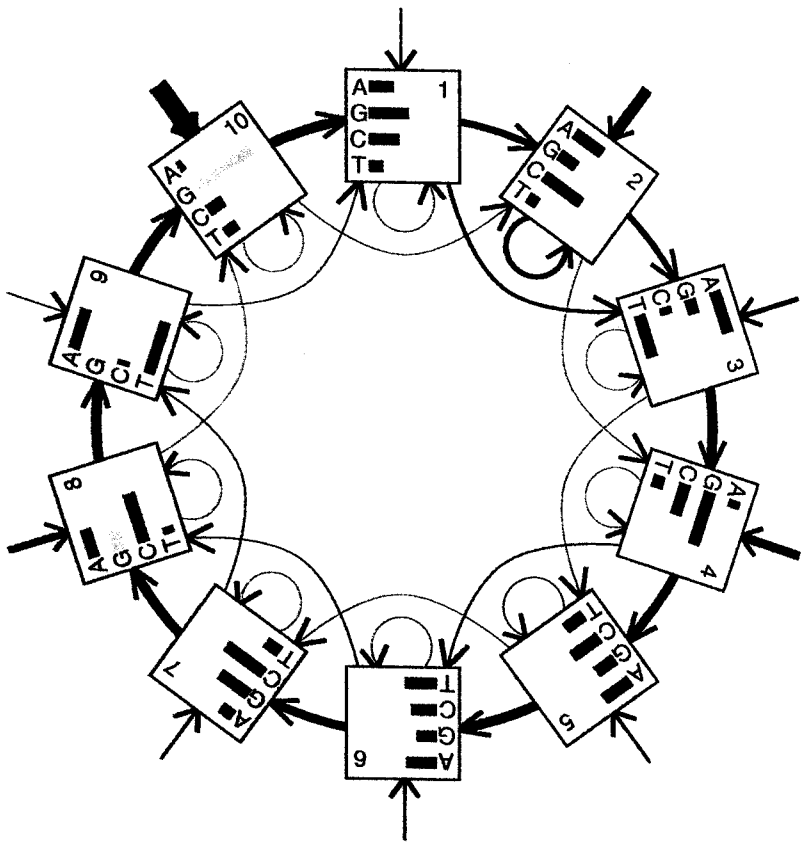


Figure 8.4. A hidden Markov model (HMM) of eukaryotic internal exons. This HMM is designed to detect a statistically significant frequency of the same base at intervals of 10 bp in sequences. Imagine feeding an exon sequence into the part of the sequence shown by the heaviest arrow at 11 o'clock on the circle and then threading the sequence clockwise around the circle, noting the base at each subsequent position in the sequence, and recording that information in the corresponding box (the state of the HMM). If there is a small repeated pattern of a few bases at every tenth position in the sequence starting at the same position from the start of the exon sequence, the distribution of bases in some of the boxes will begin to reflect that pattern. Hence, there is a repeated pattern of not-T (i.e., A, C, or G), A or T, then a G. By a slightly more sophisticated analysis similar to that discussed in Chapter 4 (p. 187), the model can be used to show that the same pattern may start at other positions with respect to the start of the sequence (other arrows feeding into the circle) and also that some sequence positions in the circle may be skipped (arrows going around some of the states) or extra sequence may be found (loop arrow returning to same state). A similar pattern is found in introns and also around the start site of transcription. This structure is modulated by histone-modifying systems as one means of gene regulation in eukaryotes. (Redrawn, with permission, from Baldi et al. 1996 [copyright Academic Press, London].)

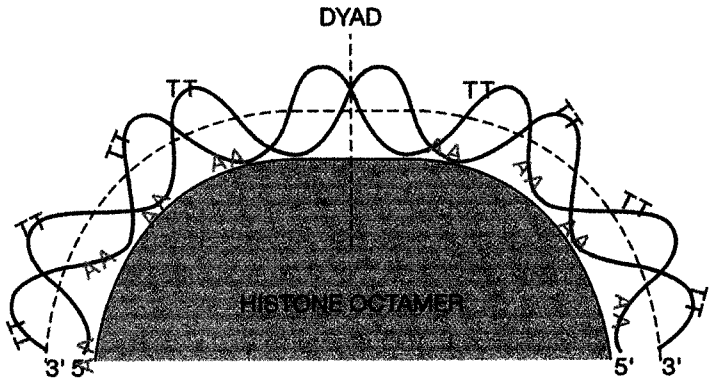
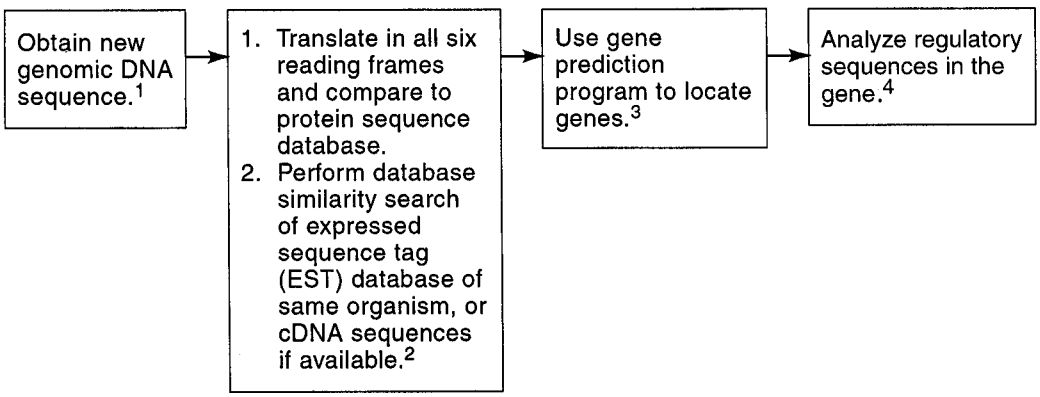


Figure 8.5. A proposed role for the repeated sequence patterns in eukaryotic genes. Shown is the portion of a DNA molecule wrapped around a nucleosome. The actual length of DNA around the nucleosome will be approximately 145 bp. The repeated patterns found in eukaryotic genes (including not-T, A or G, G) and AA/TT dinucleotides influence the bendability of the DNA strand in which they are located, and hence will facilitate the folding of DNA around a nucleosome. (Redrawn, with permission, from Ioshikhes et al. 1996 [copyright Academic Press, London].)

the number of base pairs expected in a single turn of the DNA double-stranded helix around a nucleosome. The patterns found in promoter sequences are those that bend more easily when located in the major groove of DNA and are thought to be located on the inside of the bent molecule (Ioshikhes et al. 1996; Pederson et al. 1998), as shown in Figure 8.5. Using these observations, a model has been proposed that sequence patterns located downstream from the transcription start site are suitable for positioning of nucleosomes, whereas upstream regions do not show the necessary patterns (Pederson et al. 1998).

Loops of chromatin are attached to the nuclear matrix by relatively short (100–1000 bp long) sequences called matrix attachment regions (MARS) or scaffold-associated regions (SARS). These regions are considered to be an indicator of the presence of expressed genes. Although the sequence of only a small number of such regions has been determined, several characteristic sequence patterns have been identified. The program MARS-FINDER (see Table 8.6 for Web site) searches for sequences that have a high representation of such sites in genomic DNA (Singh et al. 1997).

METHODS



1. The purpose of gene prediction is to identify regions of genomic DNA that encode proteins, although searches for RNA-encoding genes are also performed (see Chapter 5). The genomic DNA sequence may be that of an insert of genomic DNA in a bacterial artificial chromosome (BAC) or similar vector or that of an assembled chromosome or chromosomal fragment. Genome sequencing centers often search through newly acquired sequences with gene prediction programs and then annotate the sequence database entry with this information. This annotation includes gene location, gene structure (positions of predicted exons/introns and regulatory sites), and any matches of the translated exons with the protein sequence databases. The amino acid sequence of the predicted gene may also be entered in the protein sequence databases. Because the standards for identification are not uniform, and because gene predictions can be incorrect, it is a good idea to reconfirm any gene prediction of interest, perform alignments of the predicted sequence with matching database sequences to confirm statistical and biological significance (as described in Chapters 3 and 7), and confirm the predicted gene sequence by cDNA sequencing. If EST sequences are available in a sufficient coverage of the genome, these are also useful for confirmation of predicted gene sequences. For an example of the gene annotation procedure that was followed for the *Drosophila melanogaster* genome sequence, see Adams et al. (2000). The final goal of the gene annotation procedure for an organism is to produce a genome database that includes a rich supply of biological information on the function of each gene, as discussed in Chapter 10. This information will come from laboratory experimentation and manual entry of relevant published data into the genome database.
2. Database similarity searches of this type are described in the flowchart of Chapter 7. For genes of prokaryotic organisms, step 1 identifies open reading frames (ORFs, a series of amino acid-specifying codons) that encode a protein similar to one found in another organism. ORFs without a similar gene in another organism may also be found, as described in the text. Genes of eukaryotic organisms often have intron and exon sequences in the genomic DNA sequence. Step 1 provides the approximate locations of exons that encode a protein similar to one in another organism. Eukaryotic genomes may also have ORFs that do not match a database sequence, and these ORFs may or may not encode a protein. In the Genome Annotation Assessment Project (GASP) of the *Drosophila* genome, one study showed that combining gene prediction methods with homology searches generally provides a reliable annotation method (Birney and Durbin 2000). Step 2 is an additional type of database similarity search that identifies protein-encoding ORFs. Because cDNA sequences and partial cDNA sequences correspond to exons, genomic ORFs that can be aligned to these expressed gene sequences include exon sequences. This analysis can be enhanced by using databases of indexed genes in which overlapping ESTs have been identified (see flowchart, Chapter 7). EST_GENOME is a program for aligning EST and cDNA sequences to genome sequences (Table 8.1). Collections of EST sequences for an organism are often only partial collections; thus, failure to find a matching EST is not a sufficient criterion for rejecting an ORF by this test. Searching the EST collections of related organisms, e.g., another mammal or plant, may be helpful in identifying such missing EST sequences. An additional type of gene analysis is to use an already-identified ORF as a query sequence in a database search against the entire proteome (all of the predicted proteins) of an organism to find families of paralogous genes, as described in Chaper 10.
3. There are a large number of gene prediction programs available (Table 8.1). They all have in common to varying degrees the ability to differentiate between gene sequences characteristic of exons, introns, splicing sites, and other regulatory sites in expressed genes from other non-gene sequences that lack these patterns. Because these gene sequences as well as gene structure (the number and sizes of exons and introns) vary from one organism to another (see Fig. 10.3), a program trained on one organism, e.g., the bacterium *E. coli* or the worm *Caenorhabditis elegans*, is not generally useful for another organism, e.g., another bacterial species or the fruit fly *D. melanogaster*. Reliability tests of gene prediction programs have shown that the available methods for predicting known gene structure are, in general, error-prone. Referring to Web sites with this information (Table 8.1) or performing one's own reliability check is recommended. Some "reliability checks" should be eyed with suspicion because they are based on a comparison of new predictions with previous gene annotations. When gene predictions are made using gene-sized rather than large-sized, multigene sequence genomic DNA fragments, the predictions are generally more reliable (see text).
4. In prokaryotes, the predicted genes may have conserved sequence patterns such as those for promoter recognition by RNA polymerases and transcription factors, for ribosomal binding to mRNA, or for termination of transcription, as found in the model prokaryote *E. coli.* Similarly, in eukaryotes, the

region at the 5′ end of the gene may also have characteristic sequence patterns such as a high density and periodicity of putative transcription-factor-binding sites and sequence patterns characteristic of RNA polymerase II promoters. These types of analyses are enhanced by searching for similar sequence patterns in genes that are regulated by the same set of environmental conditions or that are expressed in the same tissue. Regulatory predictions are enhanced when information about conserved oligomers found in the promoters of co-regulated genes is available, as described in the text.

GENE PREDICTION IN MICROBIAL GENOMES

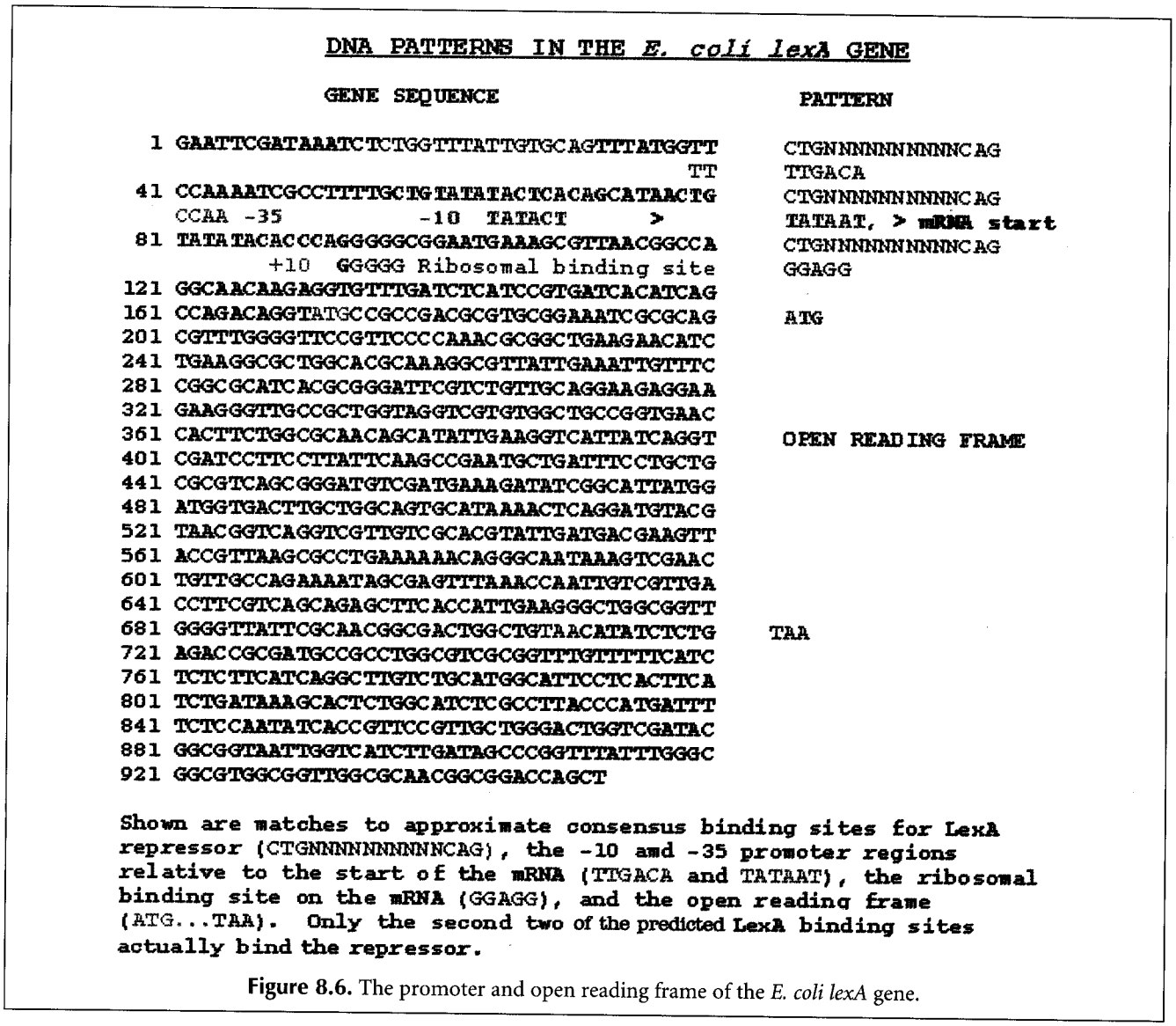
Predicting protein-encoding genes is generally easier in prokaryotic than eukaryotic organisms because prokaryotes generally lack introns and because several quite highly conserved sequence patterns are found in the promoter region and around the start sites of transcription and translation, at least in the *E. coli* model of prokaryotes. When a set of different patterns characteristic of a gene are found in the same order and with the same spacing in an unknown sequence, the prediction is more reliable than if only one pattern is found, and this type of information can be obtained in *E. coli*.

An example of the regulatory sequences for an *E. coli* gene, the *lexA* gene, is shown in Figure 8.6. Note the presence of the −10 and −35 regions (yellow) that mark the site of interaction with RNA polymerase, and the ribosomal binding site on the mRNA product (green) that is complementary to the ribosomal RNA. The ORF that encodes the LexA product is also indicated (blue). Also shown are three potential binding sites for LexA product to the promoter region, recognizable by searching for a consensus of known LexA-binding sites. Note that these sites are inverted repeats; i.e., the sequence on the forward and reverse sequence is approximately the same. This feature with minor variations is not uncommon in the binding sites of proteins that regulate transcription and is a reflection of the binding of a dimer of LexA protein to the two sites, which produces a stronger interaction than binding of a single monomer to a single site. The sites in the *lexA* promoter region represent a form of self-regulation. The two downstream sites have been shown to bind the protein and to act as a repressor that prevents further transcription. The binding at two sites may be cooperative in that two dimer molecules are more effective at preventing transcription than one, possibly because the bound proteins interact, thus making the overall binding to the promoter region stronger.

In the case of a number of other genes, binding of a regulatory protein such as LexA to a recognizable target sequence activates transcription by stimulating the binding of RNA polymerase. The consensus patterns for these various regulatory sites may be found by sequence alignment and statistical and neural network methods. These methods are discussed in Chapters 3 and 4, and also later in this chapter. Ribosomal binding sites were the first to be modeled by a neural network with no hidden layer (or perceptron), which is also discussed below (Stormo et al. 1982; Bisant and Maizel 1995).

HMMs are also used for modeling a multiple sequence alignment of many proteins and for use in identification of more members of the same family of proteins (see Chapter 3 for details.)

The highly conserved features of *E. coli* genes have made gene identification methods an attractive possibility. One such method is that of HMMs. Here a model of an *E. coli* gene is made and then expanded to include multiple genes and the sequences between the genes. The model shown in Figure 8.7 is an example of a simple HMM of a bacterial genome as a DNA molecule that is densely packed with genes with relatively short intergenic sequences and no introns. This model will read through a sequence of unknown gene composition and find the genes, i.e., a series of codons that specify amino acids flanked by start and stop codons, that are most like a set of known gene sequences and flanking regions that have been used to train or calibrate the model. Because codon usage and flanking sequence will probably vary from one genome to the next, the model trained with *E. coli* genes may not work for finding genes in other organisms. The reliability of the model depends on the



DNA PATTERNS IN THE *E. coli lexA* GENE

	GENE SEQUENCE	PATTERN
1	GAATTCGATAAATCTCTGGTTTATTGTGCAGTTTATGGTT	CTGNNNNNNNNNNCAG
	TT	TTGACA
41	CCAAAATCGCCTTTTGCTGTATATACTCACAGCATAACTG	CTGNNNNNNNNNNCAG
	CCAA -35 -10 TATACT >	TATAAT, > mRNA start
81	TATATACACCCAGGGGGCGGAATGAAAGCGTTAACGGCCA	CTGNNNNNNNNNNCAG
	+10 GGGGG Ribosomal binding site	GGAGG
121	GGCAACAAGAGGTGTTTGATCTCATCCGTGATCACATCAG	
161	CCAGACAGGTATGCCGCCGACGCGTGCGGAAATCGCGCAG	ATG
201	CGTTTGGGGTTCCGTTCCCCAAACGCGGCTGAAGAACATC	
241	TGAAGGCGCTGGCACGCAAAGGCGTTATTGAAATTGTTTC	
281	CGGCGCATCACGCGGGATTCGTCTGTTGCAGGAAGAGGAA	
321	GAAGGGTTGCCGCTGGTAGGTCGTGTGGCTGCCGGTGAAC	
361	CACTTCTGGCGCAACAGCATATTGAAGGTCATTATCAGGT	OPEN READING FRAME
401	CGATCCTTCCTTATTCAAGCCGAATGCTGATTTCCTGCTG	
441	CGCGTCAGCGGGATGTCGATGAAAGATATCGGCATTATGG	
481	ATGGTGACTTGCTGGCAGTGCATAAAACTCAGGATGTACG	
521	TAACGGTCAGGTCGTTGTCGCACGTATTGATGACGAAGTT	
561	ACCGTTAAGCGCCTGAAAAAACAGGGCAATAAAGTCGAAC	
601	TGTTGCCAGAAAATAGCGAGTTTAAACCAATTGTCGTTGA	
641	CCTTCGTCAGCAGAGCTTCACCATTGAAGGGCTGGCGGTT	
681	GGGGTTATTCGCAACGGCGACTGGCTGTAACATATCTCTG	TAA
721	AGACCGCGATGCCGCCTGGCGTCGCGGTTTGTTTTTCATC	
761	TCTCTTCATCAGGCTTGTCTGCATGGCATTCCTCACTTCA	
801	TCTGATAAAGCACTCTGGCATCTCGCCTTACCCATGATTT	
841	TCTCCAATATCACCGTTCCGTTGCTGGGACTGGTCGATAC	
881	GGCGGTAATTGGTCATCTTGATAGCCCGGTTTATTTGGGC	
921	GGCGTGGCGGTTGGCGCAACGGCGGACCAGCT	

Shown are matches to approximate consensus binding sites for LexA repressor (CTGNNNNNNNNNNCAG), the -10 and -35 promoter regions relative to the start of the mRNA (TTGACA and TATAAT), the ribosomal binding site on the mRNA (GGAGG), and the open reading frame (ATG...TAA). Only the second two of the predicted LexA binding sites actually bind the repressor.

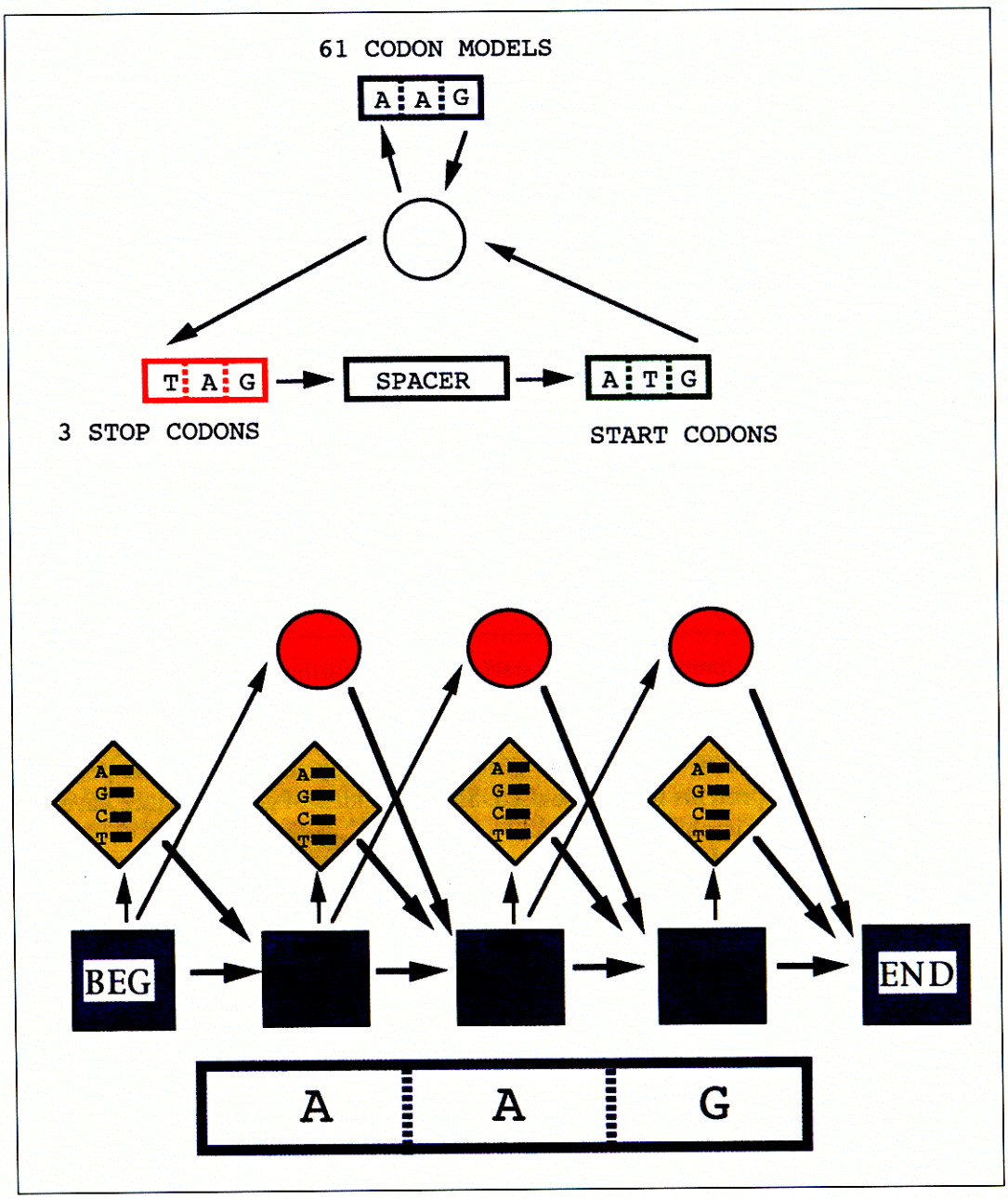
Figure 8.6. The promoter and open reading frame of the *E. coli lexA* gene.

accuracy of the gene start and stop information that is used for the training or calibration step and on the number of such genes used for training. For *E. coli,* the positions of many genes have been accurately determined. For other microbial genomes, this information is not as available, and genes predicted by alignment of the predicted proteins with *E. coli* proteins have to be used. Similar models of gene structure have been developed for other microbial genomes.

The HMM model shown in Figure 8.7 assumes that there is no relationship between each codon and later codons in the sequence; i.e., that the choice of each codon is independent of the rest. This model of genes as a Markov chain may not be fully correct because there may be long-distance correlations between some positions due to requirements for mRNA structure or translation. However, using this simplifying assumption, useful gene models can be produced. Analyses of sequential codons in genes have shown that some pairs are found at a greater frequency and others at a lesser frequency than expected by chance alone (Gutman and Hatfield 1989; Farber et al. 1992). Hence, a more appropriate choice is to design a model that uses sequence information from the previous five instead of the previous two bases to make what is called a fifth-order Markov model. In such a model, the frequency of hexamers is used to differentiate between coding and noncoding sequences. A

version of GeneMark (Borodovsky and McIninch 1993) called GeneMark.HMM uses a HMM of this type to search for *E. coli* genes (Lukashin and Borodovsky 1998).

From an information perspective, as the number of consecutive sequence positions being compared in two sequences is increased, the chance of being able to find similarities above background noise increases. For example, when using the dot matrix method for comparing sequences, a sliding window in which words of length n are compared is used to locate the most significant matches. In comparing codon and noncoding sequences, a comparison of three consecutive positions at a time can be used to find ORFs as uninterrupted runs of amino-acid-specifying codons. Extending the number of positions to a number greater than three, such as four to six, increases the chances of discovering higher-order sequence correlations in coding sequences that may be used to distinguish them from noncoding sequences.



For fifth-order Markov models to give accurate gene predictions, there must be many representatives of each hexameric sequence in genes, and if there is not, the method will be statistically limited. A new type of model, the interpolated Markov model (IMM; e.g., Glimmer; see Table 8.1), overcomes this difficulty of finding a sufficient number of patterns by searching for the longest possible patterns that are represented in the known gene sequences up to a length of eight bases. Thus, if there are not enough hexameric sequences, then pentamers or smaller may be more highly represented, and in other cases many representative patterns even longer than six bases may be found. In general, the longer the patterns, the more accurate the prediction. The IMM combines probability estimates from the different-sized patterns, giving emphasis to longer patterns and weighting more heavily the patterns that are well represented in the training sequences (Salzberg et al. 1998).

Both GeneMark.HMM and IMM find genes in microbial genomes with an apparent high degree of accuracy, assuming that gene predictions made by other methods such as sequence similarity of the translated proteins to known *E. coli* proteins are accurate. Therefore, these methods can be expected to produce reliable predictions of genes that do not match previously identified protein sequences. A further improvement of the prediction of the bacterial start codon position has been found (Hannenhalli et al. 1999). This method sorts through a set of predictions for the start codon in a set of sequences, where the actual signal is known. These predictions depend on weighting each of a set of input sequence information. The weights are adjusted so that the predictions are made more accurate by a method called mixed integer programming.

Compare this gene model with the model for protein sequence alignments shown on page 186, Figure 4.16.

Figure 8.7. HMM of an *E. coli* gene (Krogh et al. 1994). This model is designed to generate a sequence of amino-acid-encoding codons of the approximate length of an *E. coli* gene starting with an ATG codon and ending with a stop codon. A set of predicted genes are separated by intergenic spacer regions of the range of lengths actually found between *E. coli* genes. Variations in this basic model are described in the text. The model is first trained on a set of known *E. coli* gene sequences with flanking sequences. The training step is performed in very much the same manner as that described in Chapter 4 for multiple sequence alignment. The trained model may then be used to find the most probable set of genes in *E. coli* genomic sequences of unknown gene composition. The model for each codon (lower part of diagram) is represented by a set of round, diagonal, or square boxes representing match, insert, and delete states, respectively. The model shown is that for the AAG codon. Each of the 61 codons has a similar structure. If a sequence were extremely accurate, only match states would be needed in the model. The insert and delete states allow an ORF with an extra or missing base to be recognized. Similarly, the inclusion of alternative bases in each match state allows for errors in base identification. Stop codons and initiation codons are assumed to be correctly represented in each sequence and no allowance for errors is made. Hence, errors in these codons would lead to an incorrect prediction. Each match and insert state has a certain probability of producing an A, another probability for producing a G, and so on. The delete state does not produce a letter but instead acts to skip a sequence position. Directional arrows (transitions) give the probability of passing from one state to another in the model. Thus, if one state generates an A with probability of 1.0, the transition probability to the next state is 0.9, and the next state generates a G with probability 0.98, then the probability of AG is $1.0 \times 0.9 \times 0.98 = 0.88$. The model is entered at any position (upper part of diagram) and the arrows designate possible paths through the model between successive states. The central state represented by a circle does not generate a sequence position but acts as a junction between adjacent codons. For the model to generate a sequence, the probability of a codon following another codon must be quite high. Hence, the transition probability of going from the junction to a codon is much higher than for going to a stop codon. Once a stop codon has been reached, a sequence representing an intergenic spacer region is generated. Within this region is a model for sequences that are found upstream from the ATG codon for the next gene, such as the Shine-Dalgarno ribosomal binding site and other sequence information (see Hayes and Borodovsky 1998). The presence of this sequence increases the probability for a downstream gene.

GENE PREDICTION IN EUKARYOTES

A simple method for discovering protein-encoding genes within a eukaryotic genomic sequence is to perform a sequence database search by translating the sequence in all possible reading frames and comparing the sequence to a protein sequence database using the BLASTX or FASTX programs described in Chapter 7. Alternatively, if a genomic sequence is to be scanned for a gene encoding a particular protein, the protein can be compared to a nucleic acid sequence database that includes genomic sequences and is translated in all six possible reading frames by the TBLASTN or TFASTX/TFASTY programs. For proteins that are highly conserved, these methods can give a very good, albeit approximate, indication of the gene structure. If the proteins are not highly conserved, or if the exon structure of a gene is unusual, these methods may not work.

Additional information as to the locations of genes in genomic DNA sequences may be found by using cDNA sequences of expressed genes (see flowchart). An enhanced method (Pachter et al. 1999) for finding eukaryotic genes rapidly is to prepare a dictionary of sequence words (4-letter words in a protein sequence database, 11-letter words in an EST database) and to use these dictionaries to compare a genomic DNA sequence to the expressed gene and protein sequence databases.

The commonly used methods for eukaryotic gene prediction depend on training a computer program to recognize sequences that are characteristic of known exons in genomic DNA sequences. The program is then used to predict the positions of exons in unknown genomic sequences and to join these exons into a predicted gene structure. Predictions depend on analysis of a variety of sequence patterns that are characteristic of known genes in a particular organism. These include patterns characteristic of exons, intron–exon boundaries, and upstream promoter sequences. As more sequences are collected for specific organisms and the actual structures of additional genes become known, these prediction methods should become more reliable. Patterns that specify RNA splice sites are poorly conserved with only a few identical positions. Therefore, the positions of intron–exon boundaries cannot be defined precisely by simple pattern-searching methods. Neural networks (described below and in Chapter 9) provide a method of sequence analysis that has the capability of finding complex patterns and relationships among sequence positions that may not be obvious. The available methods also depend on the analysis of windows of sequence in genomic DNA to determine whether these regions are likely to be coding or noncoding. Regions that encode proteins are found to have characteristic patterns reflecting preferential codon usage and codon neighbors. These observations have led to the widely used analysis of 6-mers in DNA sequences as a basis for gene prediction.

For RNA PolII genes, gene prediction programs give possible locations of exons that can then be joined to predict the sequence of the mRNA of the gene. This sequence will include an upstream 5′ region (5′ untranslated region, UTR) extending from the start site of transcription to the initiation codon, the ORF for the protein ending in a translational termination codon, and the downstream region (3′UTR) extending to the termination of transcription in the region where the signal for polyadenylation of the mRNA may be found. The initiation site for translation in eukaryotic mRNAs is usually the AUG codon nearest the 5′ end of the mRNA, but sometimes downstream AUG codons still close to the 5′ end of the mRNA may also be used (Kozak 1999).

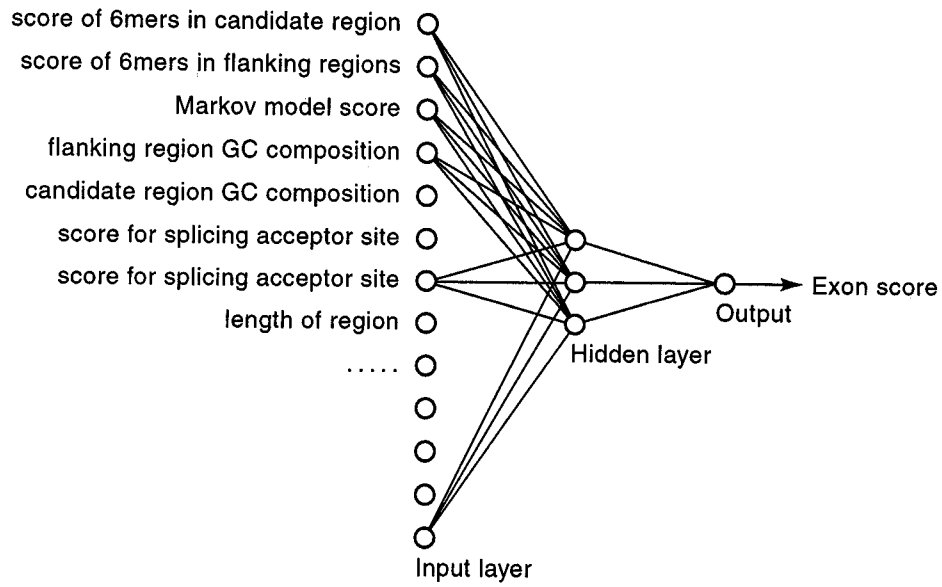
As examples of the types of analyses that are available, two types of gene prediction methods, neural networks and pattern discrimination methods, are described below. Other methods and Web sites for finding genes in eukaryotic DNA are described in Table 8.1.

Neural Networks

Grail II

Grail II provides analyses of protein-coding regions, poly(A) sites, and promoters; constructs gene models; predicts encoded protein sequences; and provides database searching capabilities. A list of most likely exon candidates is first established, and these are evaluated further using the neural network described in Figure 8.8. The algorithm makes its final prediction by picking the best candidates. A dynamic programming approach is then used to define the most probable gene models (Uberbacher et al. 1996).

Input for Grail II includes several indicators of sequence patterns. These inputs include several from different types of analyses, including a Markov model for gene recognition that, in principle, resembles the one shown in Figure 8.7, and inputs from two additional neural networks that evaluate the region for potential splice sites. One important indicator is the in-frame 6-mer preference score. Recall that the occurrence of codon pairs in coding regions is not random, whereas in noncoding regions their occurrence is more random.



Compare the use of neural networks for gene prediction with that for protein secondary structure prediction shown in Figure 9.29 (p. 453).

Figure 8.8. The Grail II system for finding exons in eukaryotic genes (Uberbacher and Mural 1991; Uberbacher et al. 1996). The method uses a neural network to identify patterns characteristic of coding sequences. The method has similarities to and differences from that used for predicting secondary structure of proteins and described in Chapter 9. Similarities include the use of three layers, an input layer for the data with the data coming from a candidate exon sequence, and a hidden layer for discerning relationships among the input data. An output layer comprising one neuron indicates whether or not the region is likely to be an exon. Each neuron receives information from a set in the layer above, some with a positive value and others with a negative value, sums these values; and then converts them to an output of approximately 0 or 1. The system is trained using a set of known coding sequences, and as each sequence is utilized, the strengths and types of connections (positive or negative) between the neurons are adjusted, decreasing or increasing the signal to the next neuron in a manner that produces the correct output. The major difference between neural networks for exon and secondary structure prediction is that the exon prediction uses sequence pattern information as input whereas secondary structure prediction uses a window of amino acid sequence in the protein. In Grail II, a candidate sequence is evaluated by calculating pattern frequencies in the sequence and applying these values to the neural network. If the output is close to a value of 1, then the region is predicted to be an exon.

Consequently, higher frequencies of 6-mers in genomic DNA that are more commonly found in coding regions can be an indicator of the presence of an exon. For various organisms, tables have been constructed giving the frequency of each 6-mer (base 1 of first codon to base 3 of second, base 2 of first codon to base 1 of the third codon, and so on) of known cDNAs divided by the frequency of the 6-mer in noncoding DNA. The logarithm of this ratio gives what is called an in-frame preference value for the 6-mer. These 6-mer preference scores increase as GC composition rises, thus increasing the preference scores of a 6-mer with GC richness. Grail II automatically corrects for this increase to put predictions from GC-rich regions on an even footing with other regions.

The log ratios for a potential ORF starting at base 1 in the test sequence, another for an ORF starting at base 2, and a third starting at base 3 are calculated by adding the logarithms of these individual 6-mers. These sums provide a log likelihood score for an exon starting at the first, second, or third positions in the given genomic sequence. These likelihoods are further modified by including conditional information on the likelihood of the next 5 bases on coding and noncoding regions, given the current 6-mer. The probability of an exon starting at base 1 is then given by a Bayesian formulation

$$P = a_1 / a + C n_1 \tag{1}$$

where a_1 is the score for an exon starting at base 1; a is the sum of scores for base 1, base 2, and base 3; n_1 is the score for a noncoding region starting at base 1; and C is the ratio of coding to noncoding bases in the organism. This value is used as the score of 6-mers in the candidate region (Uberbacher et al. 1996). A similar score is calculated for the regions 60 bases on each side of the candidate region. If these regions also appear to be encoding exons, the examined region will be enlarged and the prediction repeated. In this manner, a given exon candidate sequence will be enlarged until the coding signals from flanking sequences are no longer to be found.

GeneParser

This program predicts the most likely combination of exons and introns in a genomic sequence by a dynamic programming approach. Dynamic programming was introduced in Chapter 3 as a way for aligning sequences to obtain a most likely alignment for a given scoring system with scores for matches, mismatches, and gaps. The alignment up to a given set of sequence positions is stored in a scoring matrix, and the dynamic programming algorithm provides a method for finding the best score at that position. GeneParser uses a likelihood score for each sequence position being in an intron or exon. The intron and exon positions are then aligned with the constraint that they must alternate within a gene structure. In this manner, a combination of the most likely intron and exon regions that comprise a gene structure is found. GeneParser includes one other novel feature, a scheme for adjusting the weights used for several types of sequence patterns that make up the intron and exon scores.

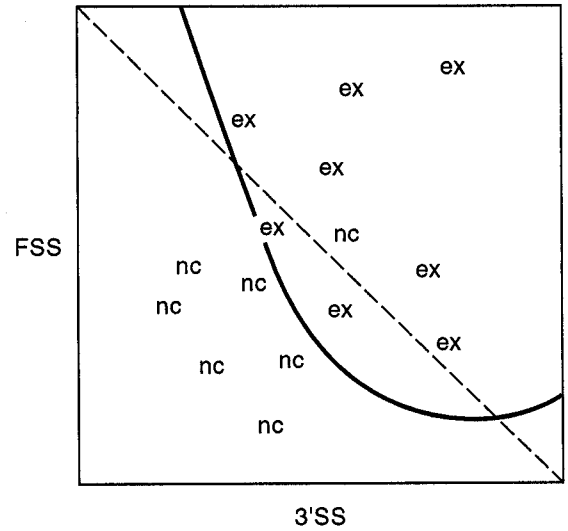
A neural network is used to adjust the weights given to the sequence indicators of known exon and intron regions, including codon usage, information content (see Chapter 4, p. 195), length distribution, hexamer frequencies, and scoring matrices (see Chapter 4, p. 192) for splicing signals. The integration of the dynamic programming and neural network methods works as follows:

1. The characteristics described above of a set of intron sequences and a second set of exon sequences are determined. For example, a table of hexamer frequencies is prepared.

2. For a training gene sequence, a series of indicator matrices is prepared. The sequence is listed both down the side of the matrix and across the top. Each position in one of the matrices representing positions a and b in the sequence gives the likelihood for an exon or intron that starts at position a and ends at position b. One such matrix would be the likelihood of an exon based on hexamer frequency in the a–b interval. Another matrix (or the other half of the same matrix, since only one half is needed for exon values) gives the likelihood of an intron based on the same criterion. Other sets of matrices for the sequence based on compositional complexity, length distribution or exons, or splice signals on weight matrices are also prepared.
3. The a,b values in the above indicator matrices for exons are each transformed by a weight and bias, and the sum of the weighted values is obtained. An initial arbitrary set of weights is chosen for each type of sequence information. These weights are later adjusted until they provide the correct gene structure of the training sequence. This sum (s) is then further transformed to a number (L) that is either close to 0 or 1 by using the neural network gating function $L = 1/[1 - e^{-s}]$. The transformed a,b values are then placed in another matrix L_E that gives the weighted score for exons going from position a to position b in the sequence. A similar set of transformed values for an intron at position a,b, but not necessarily weighted the same way, is placed in another matrix L_I at position a,b (which can be the other half of L_E since only half of the L_E matrix is needed). The reason for this transformation is to use the information at a later stage as input to a neural network, in the same manner as used in neural networks for prediction of protein secondary structure and discussed in Chapter 9.
4. Dynamic programming is used to predict by the most compatible number and lengths of introns in the training gene up to any position j in the sequence.
5. Steps 2–4 are repeated for each training sequence.
6. The accuracy of the predictions is then determined.
7. If a certain required level of accuracy is not achieved, a neural network similar to that described above for Grail II is used to adjust the weights used for the input exon and intron features.
8. If the required level of accuracy is reached, the method is ready to be used for determining the structure of an unknown genomic DNA sequence.

Pattern Discrimination Methods

Discrimination methods applied to DNA sequences are statistical methods used for classifying the sequence based on one or more observed sequence patterns. For gene prediction, features of patterns found in genomic sequences are examined statistically to determine whether they are like those found in coding sequences. One such feature that is characteristic of coding sequences is the 6-mer exon preference score (EPS) described above. Another is a score for a 3′-flanking splice site (3′SS) calculated in a similar manner. In effect, the distribution of these two scores and a number of others is obtained for a large set of known exons and also for a set of noncoding sequences. Using the EPS and 3′SS as examples, the pair of scores for each sequence is plotted on a graph and each point is labeled as coding or noncoding, as illustrated in Figure 8.9. A line is then positioned between the two groups of sequences. A sequence of unknown coding capability is similarly analyzed to determine whether the features of the sequence place it on one. HEXON and FGENEH (combines exon prediction into a gene structure) use linear discriminant



Similar types of discriminant analyses are used to classify microarray data (Fig. 10.11, p. 522).

Figure 8.9. Analysis of candidate sequences for exon status by a discriminant function. Up to nine different pattern features of sequences are analyzed in coding and noncoding sequence. Shown is a plot of two of these features for several exon (ex) and noncoding (nc) sequences. The object of the discriminant analysis is to define a boundary between these two groups of sequences such that they are maximally separated, or that the sum of distances from a boundary line to each point is a minimum. A linear discriminant analysis (Solovyev et al. 1994) assumes that the covariations among the data are the same for the exon and noncoding sequences and provides a straight line boundary (*dotted straight line*) between the two sets of data. Such a boundary may miss some of the data points. A quadratic discriminant analysis (Zhang 1997) is more flexible, does not assume a similar covariation in the exon and noncoding sequences, and provides a curved boundary formed by a quadratic equation that can, in principle, provide a better separation of the groups (*solid line*). Once these boundary lines have been calculated, the EPS and 3′SS values of a query sequence will indicate whether the sequence belongs to the exon group or noncoding group of sequences. For an actual analysis, multiple analyses are performed on a candidate sequence leading to a more complex, multidimensional type of analysis.

analysis (Solovyev et al. 1994) and MZEF uses quadratic discriminant analysis (see Table 8.1) (Zhang 1997).

EVALUATION OF GENE PREDICTION METHODS

A comparison of the above methods for accuracy and reliability must take into account the type of analysis, whether neural network, linear discriminant, or other; the number and types of sequences used for training and evaluation; and the method used for evaluation. In addition, choice of program variables by the user will affect the predictions that are made. As more gene sequences become known, more are becoming available for training and evaluation. The ideal method for evaluation uses a known set of gene structures for training the method and a second set that is not used in the training or similar to those used in the training for evaluation (Burset and Guigó 1996). The evaluation is usually more stringent if the evaluation set includes a gene and neighboring sequence rather than just the sequence between the first and last exons. A current evaluation of most methods is available at the Web sites for these methods listed in the footnotes to Table 8.1. These evaluations are most useful when different prediction methods are used in combination.

The method for evaluation is similar to that used for testing the reliability of protein secondary structure prediction as described in Chapter 9 (Mathews 1975; Burset and Guigó 1996). The program, which is trained on a set of sequences from a given organism, is used to predict the exons, or set of exons, comprising a gene of a set of genomic evaluation sequences from the same organism. An evaluation is then made of the number of true positives (TP) where the length and end sequence positions are correctly predicted, the number of over-predicted positive predictions or false positives (FP), true negative (TN), and number of underpredicted residues as misses or false negative (FN) predictions. The following calculations are made: (1) Number of actual positives is AP = TP + FN; (2) the number of actual negatives is AN = FP + TN; (3) the predicted number of positives is PP = TP + FP; and (4) the predicted number of negatives is PN = TN + FN. The sensitivity of a method SN is given by SN true positives/actual positives = TP / (TP + FN), the specificity by SP = true positives/predicted positives = TP / (TP + FP), and a correlation coefficient CC by

$$CC = [(TP)(TN) - (FP)(FN)] / \sqrt{[(AN)(PP)(AP)(PN)]}$$

By this coefficient, a method given all correct gene predictions would score 1, and the worst possible prediction would be −1. In tests of this kind on three sets of human sequences, GeneParser, GenID, and Grail gave (1) sensitivities of 0.68–0.75, 0.65–0.67, and 0.48–0.65; (2) specificities of 0.68–0.78, 0.74–0.78, and 0.86–0.87; and (3) correlation coefficients of 0.66–0.69, 0.66–0.67, and 0.61–0.72, respectively, for the accuracy of finding the correct nucleotide ends of exons. GeneParser was also shown to be more reliable for genes with short exons and least reliable for genes with long exons (Snyder and Stormo 1993).

A detailed evaluation of the available gene prediction programs has been performed, and the correlation coefficent was found to lie between 0.6 and 0.7, and the fraction of correctly found exons was generally less than 50%. The performance decreased when longer test sequences were used and when a 1% level of artificial frameshift mutations was introduced. Programs including protein sequence database searches (GeneID+ and GeneParser3) showed substantially greater accuracy (Burset and Guigó 1996). These studies therefore indicate that gene prediction programs reliably locate genomic regions that encode genes, but they provide an only approximate indication of the gene structure. In a later similar study using the same data set as the above study, and comparing Grail II, FGENEH, and MZEF, these numbers were: (1) sensitivities 0.79, 0.93, 0.95; (2) specificities 0.92, 0.93, 0.95; and (3) 0.83, 0.85, 0.89, respectively (Zhang 1997).

To illustrate the results obtained by the gene prediction programs, an *Arabidopsis* genomic sequence was submitted to several Web servers, as shown in Table 8.4. Because the cDNA sequence was also available, the accuracy of the programs could be determined. There is a computer program designed for aligning the cDNA and genomic DNA sequences of a gene (Florea et al. 1998; and see Table 8.1). As shown, the results of the analyses vary considerably and the program variables must sometimes be optimized to find the correct translation. In this case, GeneMark gave a fully accurate translation of the sequence. Other programs, such as NetPlantGene, gave a large number of possible exon–intron boundaries including some of the actual ones.

PROMOTER PREDICTION IN *E. COLI*

The method that has most often been used to analyze *E. coli* promoters is to align a set of promoter sequences by the position that marks the known transcription start site (TSS)

Table 8.4. *Example of exons predicted in an* Arabidopsis *genomic sequence by gene prediction programs*

cDNA	Netgene[b]	GeneMark[c]	FgeneP[d]	GeneScan	Mzeff[e]
345[a]–1210	x 1210	345–1210	345–1210	530–1210	276–1210
1290–1513	1290 1513	1290–1513	x 1513	1242–1513	1290–1513
1611–1696	1611* 1696	1611–1696	x x	1611–1696	1611–1696
1880–2029	1880* 2034	1880–2029	x x	1880–2029	1880–2029
2143–2880	2143 2880	2143–2880	x 2880	2143–2880	x 2880
3143–3253	x 3253	3143–3253	x x	x x	3143–3253
3339–3599	3339* 3599	3339–3599	3339-3599	3339–3599	3339–3599
3698–3921	3698 3921	3698–3921	3698-3921	3698–3921	3698–3921
4010–4217	4010 x	4010–4220[f]	x x	4010–4220[f]	x x

This test is given as an example and should not be taken as a measure of the reliability of these programs. The Web sites were provided with the genomic sequences of the *Arabidopis UVH1* gene with approximately 250 bp upstream from the first exon and 200 bp downstream beyond the last exon. As indicated in the text, these programs are more reliable when they are presented with short genomic sequences, as was done in this example. The consensus splice sites for *Arabidopsis* may be found at http://genome-www.stanford.edu/Arabidopsis/splice_site.html. A more detailed assessment of the reliability of gene prediction programs on Arabidopsis genomic sequences has been published (Pavy et al. 1999).

[a] Predicted.

[b] NetPlantGene was used. This program predicts intron–exon and exon–intron junctions and not most probable combinations of the two. In this case many false-positive intron–exon junctions were predicted with low probability. The highest scoring junctions are marked by *. x are actual sites not predicted. The intron–exon junctions are predicted much more reliably, and three false positives were reported.

[c] GeneMark shows a remarkably good frequency of prediction for these exons and usually joins the exons in the correct reading frame, but not always. Therefore, some parts of the predicted protein sequence are not correct.

[d] x are actual sites not predicted. Exon start sites of 1370–1513 and 2779–2880 were found illustrating a difficulty with finding exon start sites.

[e] The prior probability was set at 0.6–0.8 to obtain these results. The higher this value, the lower the level of discrimination used, the more sensitive the test, and the greater the number of exons that is predicted. x was not predicted; instead a start site of 2709 was predicted. This program predicts internal exons only.

[f] The 4220 end includes the termination codon.

and then to search for conserved regions in the sequences. Following such an alignment, *E. coli* promoters are found to contain three conserved sequence features: a region approximately 6 bp long with consensus TATAAT at position −10 (the Pribnow box), a second region approximately 6 bp long with consensus TTGACA at position −35, and a distance between these regions of approximately 17 bp that is relatively constant (see Fig. 8.6 for an example). A weaker region exists around +1, the designation given to the start of transcription, and an AT-rich region is found before the −35 region (Hawley and McClure1983; Mulligan and McClure 1986). The sequences changed to some extent as the number of sequences and the types of promoters analyzed were varied. For example, promoters that are activated by transcription factors have more variable sequences (Hertz and Stormo 1996). The RegulonDB (http://www.cifn.unam. mx/Computational_Biology/regulondb/; Salgado et al. 1999), Dpinteract (http://arep.med.harvard.edu/dpinteract-database; Robison et al. 1998), and regulatory site database (Thieffry et al. 1998; http://www.cifn.unam.mx/Computational_Biology/E.coli-predictions) have been developed with information on the *E. coli* genome. With the availability of a large number of prokaryotic genomes (see Chapter 10 and http://www.tigr.org/tdb/mdb/mdb.html), a similar analysis of the genes and regulatory sites in these other genomes has become possible.

The aligned promoter regions provide a consensus sequence that may be used to search for matching regions as potential promoters in *E. coli* sequences. Each column in the alignment gives the variation found in that position of the promoter. Programs such as the

Genetics Computer Group program FINDPATTERNS and PatScan (http://www-unix.mcs.anl.gov/compbio/PatScan/HTML/patscan.html) may be used to search for matches to the consensus sequence or the variation found in each column in a target DNA sequence. The difficulty with using the consensus sequence to search for new promoters is that most sequence positions in the aligned regions vary to some extent, and some regions are much less variable than others; e.g., the first, second, and sixth positions in the −10 region.

An alternative is to use the search features of FINDPATTERNS and PatScan that allow alternative symbols at one sequence position, repeats of a symbol, inverted repeats, gaps, and so on. For example, providing the pattern GAT (TG, T, G) {1,4} to FINDPATTERNS means to search for GAT followed by a TG, or a T, or a G repeated up to four times. These types of pattern expressions are similar to regular expressions that are used to specify PROSITE patterns in protein sequences and to inititate PHI-BLAST searches of protein sequence databases (see Chapter 7, p. 331). Although these expressions are extremely useful for locating complex regulatory patterns in DNA sequence, they do not take into account the frequency of each residue at each pattern position. What is needed is a more quantitative way to use these known sequence variations to search a target sequence. The scoring matrix method provides such an analysis.

The Scoring Matrix Method Used with Aligned Promoter Sequences

A more complex type of promoter analysis used for both prokaryotic and eukaryotic sequences is a scoring or weight matrix. This kind of matrix was previously described in Chapter 4 (p. 192) as a method for representing the variation in a set of sequence patterns in a multiple sequence alignment, and in Chapter 7 (p. 320) as a tool for finding additional sequences with the same pattern in a database search. The scoring matrix has also been used to analyze promoters, ribosomal binding sites, and eukaryotic splice junctions (Staden 1984).

An example using a scoring matrix for representing the −10 region of *E. coli* promoters is illustrated in Table 8.5. In this example, *N* sequences have been aligned by their −10 regions and a count of each base in each column of the alignment has been made. These counts are converted to frequencies. For example, if 79 of 100 sequences have a T in column 1, the frequency of T in column 1 of the matrix is 0.79. Similarly, a T occurs in column 2 with a frequency of 0.94. These frequencies are converted into log odds scores, as described in Table 8.5. An example of using the scoring matrix in Table 8.5 to locate the most likely −10 sites in a query sequence is shown in Figure 8.10. The matrix is moved along the query sequence one position at a time. At each position, the base in the sequence is noted and the corresponding score of that base in the matrix is then used. This procedure is repeated for the remaining positions. The log odds scores are then added to obtain a combined log odds score for the particular position in the sequence that is a −10 region in a promoter. The sum of the log odds scores in bits may be converted to odds scores by the formula odds score $= 2^{(\text{log odds score})}$ or if the log odds score is in nats, by the formula odds score $= e^{(\text{log odds score})}$. These numbers vary from small fractions to large numbers reflecting variations in the likelihood of a −10 region at each sequence position.

The odds scores at every possible matching location along the sequence may be used to find the probability of each sequence location. The odds scores are first summed to give sum *S*. The odds score at a particular location of six bases in the sequence divided by *S* then provides a probability that the location is a −10 region. To give a simple example, of the three matches in Figure 8.10, the probability of the match at the third location shown is $391/[(1/786)+(1/630)+(391)] = 1.000$.

Table 8.5. *A scoring matrix representing the frequency of DNA bases found in the −10 position in* E. coli *promoters*

A. Fraction of each base at each column of the aligned promoters in the −10 region

Position	*A*	*C*	*G*	*T*
1	0.02	0.09	0.10	0.79
2	0.94	0.02	0.01	0.03
3..6	....	....	....	....

B. Log odds score

Position	*A*	*C*	*G*	*T*
1	−3.80	−1.49	−1.34	1.67
2	1.92	−3.81	−4.81	−3.22
3	−0.06	−0.81	−0.66	0.81
4	1.24	−1.00	−0.72	−0.89
5	1.02	−0.35	−1.00	−0.56
6	−4.81	−3.22	−4.81	1.95

(*A*) Frequency of each base found, showing two positions as examples. (*B*) Conversion of frequencies to log odds scores. The first step is to convert the frequency of each base at each sequence position into an odds score. The odds score is simply the frequency observed in the column divided by the frequency expected, or the background frequency of the base, usually averaged over the genome. Thus, if the position frequency is 0.79 and the background 0.25, the odds score is 0.79/0.25 = 3.16. This number means that if a sequence is being examined for the presence of a promoter, and a T is present in the sequence at predicted position 1, the odds of the sequence representing a promoter (a win) to the sequence not representing a promoter (a loss) is 3.16/1. Finally, the odds score is converted to a log odds score by taking the logarithm of the odds score, usually to the base 2 (units of bits) and sometimes to the natural logarithm (units of nats). As described in Chapter 4, bit units have a special meaning in information theory. They represent the number of questions that must be asked to decide whether or not the base in the column of the scoring matrix is a match to the aligned sequence position. This number is called the information content of the matrix position. On the one hand, if all four bases are equally represented in the matrix position, the number of questions that must be asked is two. The first question might be is the sequence position one of A or T, or one of G and C. The second question will then find the correct base. On the other hand, if only one base is found in the matrix position, then no question need be asked of the sequence position. The fewer questions that have to be asked, the more information in the matrix, and the more discriminatory it is for distinguishing real matches from random matches (Schneider et al. 1986). A set of log odds scores for the major six positions in the −10 region of *E. coli* promoters is shown (Hertz and Stormo 1996). In the actual matrices that are used, an additional 6–12 base positions that flank these major positions are also used. There is a zero occurrence of one particular base in the matrix, thus creating a problem because the logarithm of zero is infinity. In this case, a single count is substituted for the zeros and the resulting small fraction will calculate to a large negative log odds score. Alternatively, a large negative log odds score may be used at such positions in a scoring matrix.

Another formula for calculating the scoring matrix value of base i in column j, $w_{i,j}$, is given by

$$w_{i,j} = \log\left[(n_{i,j} + P_i)/\{(N+1)P_i\}\right] \approx \ln\left(f_{i,j} / P_i\right)$$

where $n_{i,j}$ is the count of base i in column j, P_i is the background frequency of base i, N is the total number of sequences, and $f_{i,j} = n_{i,j}/N$ (Hertz and Stormo 1999). Bucher (1990) uses the formula

$$w_{i,j} = \log\left[(n_{i,j}/P_i) + (s/100)\right] + C_j$$

where s is a smoothing percentage for the column values and C_j is a column-specific constant. Bucher sometimes also uses dinucleotide composition for calculating the background base frequency to accommodate local sequence complexity (Bucher 1990). These formulas both accommodate zero occurrences of a base by adding a small value in a scoring matrix to zero positions. Another method is to add pseudocounts to these positions, as described in Chapter 4 (p. 193).

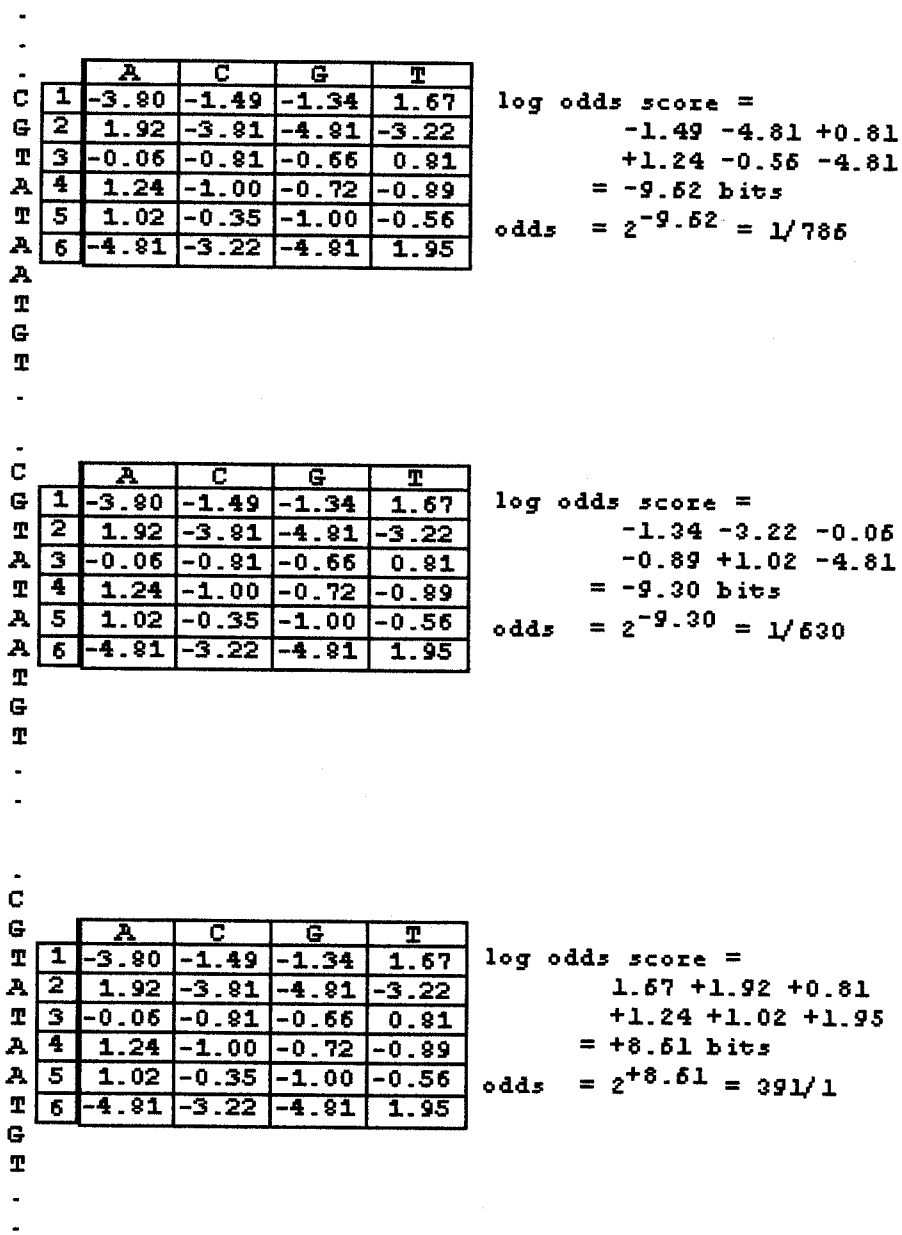


Figure 8.10. Locating −10 promoter sites in *E. coli* sequences using a −10 scoring matrix. The matrix is moved along a query sequence one position at a time, and at each location, the sequence window is scored for a match to the matrix by summing the log odds scores. A cutoff score may be defined that permits recognition of known promoter sequences while minimizing the prediction of false-positive sites.

For scoring *E. coli* sequences for the presence of promoters, scoring matrices for a 35-bp region encompassing the −35 region, a 19-bp region encompassing the −10 region, and a 12-bp region encompassing the +1 region are each applied to both strands of a query DNA sequence. Each matrix will provide a distribution of odds scores that predict possible locations for matches to itself in the query sequence. These matches are then examined for spacings that are characteristic of the known promoter sequences. The region between the −10 and −35 regions varies from 15 to 21 but is usually 17, and the region between −10 and +1 is 4–8 bp. When a suitably oriented combination of high-scoring matches is found,

the log odds scores of each sequence region are added. From this sum, a penalty may be subtracted if the distance between the −10 and −35 regions is not an optimal 17 in length or if the distance between the −10 and +1 regions is not optimal (Hertz and Stormo 1996). The resulting log odds score represents an overall likelihood that a test sequence includes regions characteristic of *E. coli* promoters in the correct spacing. A similar application of weight matrices for identifying the start of prokaryotic genes (program ORPHEUS) has been described previously (Frishman et al. 1998).

Reliability of the Matrix Method

The reliability of a combination of scoring matrices for promoter prediction can be assessed by comparing the range of scores found in a set of known promoters versus scores in a set of random sequences. A threshold score that is achieved by most of the known promoters, but only by a small number of false positives in random sequences, may then be chosen (Bucher 1990; Hertz and Stormo 1996). For example, 0.048 or 0.27 of the positions in random sequences may achieve such a score, when compared to promoters that are not activated by additional transcription factors (are more alike) versus all promoters (are more variable), respectively. When a lower threshold is chosen that gives a lower false prediction rate of 0.0005, 0.26 of all promoters and 0.60 of activated promoters achieve such a score. To try to improve the predictive values, the lengths of the scoring matrices and the gap penalty values have been varied, but the predictive value of the matrices is not much improved above these values.

There are several reasons that matrix methods do not achieve a better prediction of *E. coli* promoters. The first is that the matrix method adds the scores for each sequence position, whereas in reality, one position in the −10 region, for example, may play a role in one stage of transcription such as promoter recognition by RNA polymerase, whereas another may play a role in a subsequent stage of transcription, such as initiation of transcription or elongation of the mRNA. Matching positions with these types of functional separations are not expected to be additive, as assumed by the matrix method. A second difficulty that the matrix method shares with most other methods of promoter prediction is that all promoters are treated as being in the same class, whereas different forms of RNA polymerase that are complexed with a set of transcriptional activators (σ factors) may have preference for different sequence positions in the promoter region. With the whole genome of *E. coli* now available for analysis (see http://www.genetics.wisc.edu), such additional classification may become a possibility (Hertz and Stormo 1996). A third difficulty is that the promoter sequence is treated as a Markov chain, meaning that each sequence position acts independently of the others so that a match at each position may be individually scored without reference to the other positions. According to a statistical mechanical theory discussed below, the most conserved positions are thought to act independently. However, as evidenced by the fact that some weight matrices are not efficient in locating matching sites, there may be correlations between the sequence positions so that covariation of the bases at these positions occurs at frequencies greater than expected by chance. Such correlations are not easily found in a small number of training sequences. Methods include using decision trees and locating specific oligonucleotides, discussed later in the chapter. A number of ways to improve matrix methods, including corrections for base composition, utilizing a different number of matrix positions, have been tried, but none of these is significantly better than the basic scoring matrix described above. In addition to the matrix methods, a number of additional methods for predicting *E. coli* promoters and other regulatory sites have been developed, but without much improvement over the scoring matrix method (Hertz and Stormo 1996).

A second method that has been used for promoter prediction is the use of neural networks, which are described in Chapter 9, page 450. In this case, a neural network is trained to distinguish *E. coli* sequences from nonpromoter sequences (Horton and Kanehisa 1992; Pedersen et al. 1996). The network is like that used for prediction of protein secondary structure and is trained by similar methods. Horton and Kanehisa used a network lacking a hidden layer, called a perceptron (see Fig. 8.11). This type of network scans the sequence to be analyzed using a sliding window and at each location reads each of the sequence positions within the window. Some positions within the window may not be counted corresponding to the spaces between the conserved regions. The sequence characters are given a simple identification scheme to avoid any bias (e.g., A is 1000, G 0100, etc.) and the sum of these sequence values after weighting is used as input for a single output neuron, which produces a number close to 1 if the region is within a promoter or 0 if the region is not in

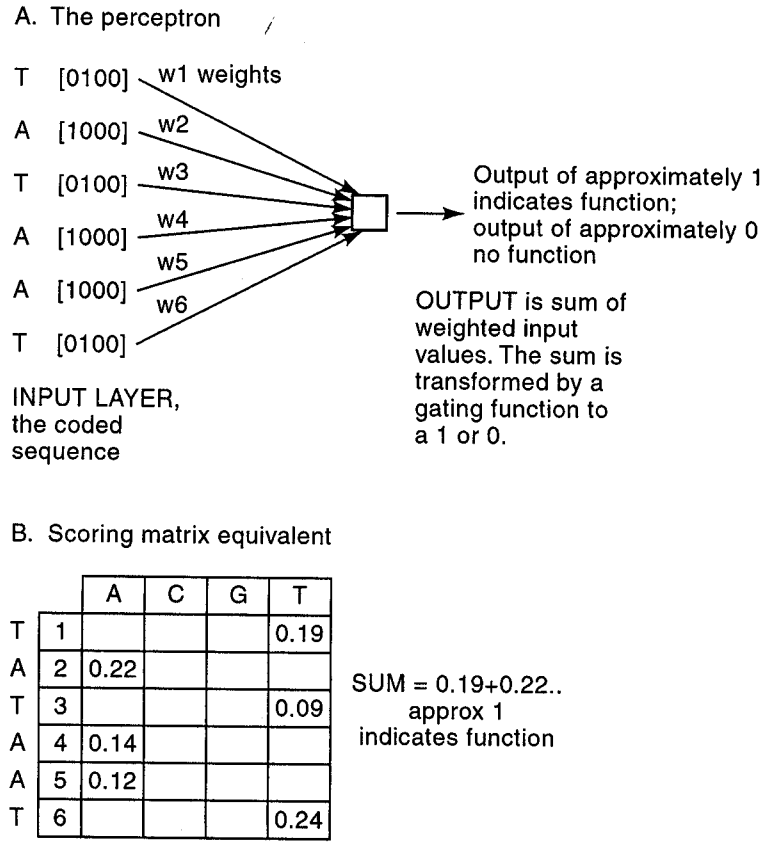


Figure 8.11. The model of a perceptron used for locating *E. coli* promoters. A portion of the sequence in the −10 region that is to be scanned is shown for illustrative purposes. (*A*) A known promoter sequence is encoded and used as input into a single output neuron. The input signals are weighted, the weighted values summed, and the sum transformed into a number that is approximately 0 or 1. The network is trained by starting with an initial set of weights, then adjusting each weight by a small amount until the correct output is found for as many of the training sequences as possible. (*B*) The trained perceptron used for scanning unknown sequences for promoter-like patterns is a special type of scoring matrix. In this case, the matrix is aligned with the sequence, and the matrix values that match the sequence are added. If the sum of numbers is approximately 1, then a promoter is predicted, and if zero, then a promoter is not predicted. The difference between this matrix and the scoring matrix described above is that in this matrix each position within the sequence window is given a different weight.

a promoter. The network is trained on known promoter sequences by adjusting the weights of the input sequence positions so that the output produces the correct response. However, the perceptron method was not found to be any more effective than scoring matrices for finding *E. coli* promoters.

Finding Less-conserved Binding Sites for Regulatory Proteins in Sequences That Do Not Readily Align

In the above example of finding consensus binding sites for RNA polymerase in *E. coli* promoters, the sequences could be quite readily aligned by the transcriptional start site and the −10 and −35 regions. The binding sites for other regulatory proteins, such as the LexA protein described above, are also quite readily found because the sequence of the binding sites is conserved. However, in many other cases, particularly those for eukaryotic transcription factor binding sites described later in this chapter, the sites vary considerably and the surrounding regions are also variable so that it is impossible to find conserved positions in the binding site by aligning the sequences. Thus, methods are needed to find a common but degenerate pattern in sequence fragments that are expected to carry a binding site but that cannot be aligned.

The problem is similar to that described in Chapter 4 for finding patterns that are common to a set of related protein sequences that cannot be readily aligned. However, there is one important difference. In proteins, there are a possible 20 amino acids in each matching position of the sequence pattern, but in DNA-binding sites there are only four possible bases in the pattern—the alphabet is much smaller in DNA sequences. Hence, it is more difficult to detect DNA sequence patterns above background noise. Some of the statistical methods used for finding protein patterns, e.g., expectation maximization and hidden Markov models, are also used for identifying DNA patterns in unaligned DNA sequences.

The expectation maximization method is described in Chapter 4. Briefly, an initial scoring matrix of estimated length is made by a guessed alignment of the known promoter sequences (the expectation step). The scoring matrix is then used to scan each sequence in turn, and the probability of a match to each position in each sequence is calculated as discussed above. The scoring matrix is then updated by the sequence pattern found at each scanned position times the probability of a match to that position (the maximization step). The two steps are repeated until there is no improvement. The method has been adapted to find multiple patterns separated by a variable spacer region, to take into account the −10 and −35 regions of *E. coli* promoters (Cardon and Stormo 1992). These studies have provided useful information as to which positions in the promoter sequences provide information that enhances specificity. Hidden Markov models such as those described in Chapter 4 (p. 185) and earlier in this chapter have also been used for prokaryotic promoter prediction (Pedersen et al. 1996). In principle, because HMM methods are based on the expectation maximization method, they should be comparable in effectiveness to the EM method.

Another statistical method of finding patterns in unaligned sequences has also been used for DNA sequences. In one case, this method was used with a dinucleotide analysis to reduce background noise (Ioshikhes et al. 1999). A Gibbs sampling method that takes into account additional features of DNA sequences such as inverted repeats has been described (Zhang 1999b). Align Ace is a program designed for promoter analysis that uses a Gibbs sampling strategy (see Table 10.1E). The inverted repeat feature is designed to identify binding sites of regulatory proteins that are inverted repeats, like LexA-binding sites in Figure 8.6.

A different method has been developed for searching through a set of unaligned sequences for a common but degenerate sequence pattern (Stormo and Hartzell 1989;

Hertz et al. 1990). The program developed for this purpose, consensus, was used to produce a set of scoring matrices for eukaryotic transcription-factor-binding sites (Chen et al. 1995). Recently, a theory was developed that allows a statistical evaluation of the results (Hertz and Stormo 1999). In its simplest form, illustrated in Figure 8.11, a sliding window of sequence in each of the sequences is matched against similar windows in the remaining sequences, searching for the best scoring matrix, as judged by the information content of the matrix (p. 195). There is no allowance made for gaps, and the choice of a base at each matrix position is assumed to be independent of the other positions, although the development of methods for including such features has been described previously (Hertz and Stormo 1995). In consensus, parameters such as window width, whether or not each sequence can contribute at most one word, whether or not there are additional words after an initial one, whether or not words overlap, whether or not the complementary sequence is used, and the maximum number of alignments to be saved are set by the user. In a related program, wconsensus, the optimum window size is not set by the user. Instead, biases are used and subtracted from the information content of each column in the scoring matrix to make the amount of information a smaller number, called the crude information content. The object is to reduce the average alignment score to a negative value so that an interesting alignment appears as a positive score, much like the procedure used in the Smith-Waterman algorithm for sequence alignment by dynamic programming. wconsensus finds the scoring matrix that maximizes this crude information content. At the same time, wconsensus also saves the flanking sequence regions from each sequence included in the matrix. As more sequences are added, these regions may also become incorporated into the alignment and help to locate additional matching regions.

The time required for computing these patterns is extensive and increases as a linear function of the number of sequences and as the square of the sequence lengths. The programs accept user input to reduce the computational time. These programs are not guaranteed to provide the best possible matrix, but by trying out several reasonable values for user-provided variables, there is a strong possibility of finding the best matrix. Associated with these programs is a statistical evaluation of each matrix. If I is the information content of the matrix calculated and N the number of sequences used to create the matrix, the probability of obtaining a greater product $I \times N$ from random sequences of the same length and base composition is determined. This procedure is similar in principle to the methods used to evaluate scores found in sequence alignments and database searches, except that the statistical models are quite complex (Hertz and Stormo 1999). Similar numerical methods for calculating the significance of scoring matrices and matches to scoring matrices have also been developed (Staden 1989). Thus, different matrices found by using different matrix widths, base compositions, and other variables may be evaluated for significance, and the best ones chosen. The consensus programs run under the UNIX operating system and are available by anonymous FTP from beagle.colorado.edu in the directory /pub/consensus.

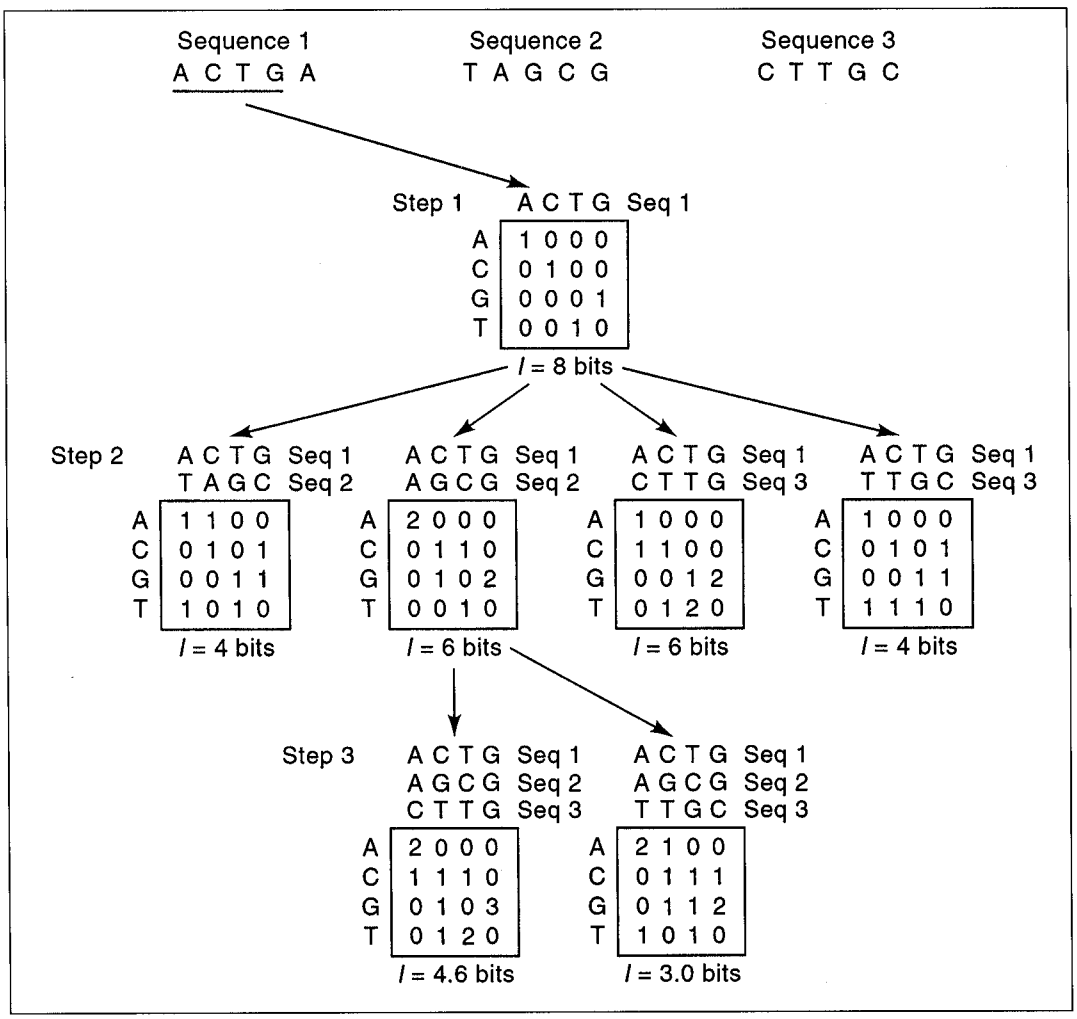
Binding sites for repressors and activators of *E. coli* and other bacteria have been analyzed for conserved patterns by the above methods. An example is the set of bacterial and bacteriophage genes that is repressed by the *E. coli lexA* gene product (Lewis et al. 1994). As illustrated in Figure 8.6, these genes carry the binding site for LexA repressor, which is located in the vicinity of the promoter and transcription start site and has the consensus sequence CTGTNNNNNNNNNCAG. The more conserved positions in the binding site contribute the most to the binding of the LexA protein to these sites and, in general, the closer the binding site to consensus, the more tightly bound the protein to that site. Similar observations of several transcriptional regulators and promoters of *E. coli* have led to a statistical mechanical theory that the most conserved positions each independently contribute the most binding energy to the interaction (Berg and von Hippel 1987; Fields et al. 1997; Stormo and Fields 1998).

PROMOTER PREDICTION IN EUKARYOTES

Transcriptional Regulation in Eukaryotes

The regulation of transcription of protein-encoding genes by RNA polymerase II (RNA PolII) involves the interaction of a large number of protein complexes, called transcription factors (TFs), with each other and with DNA-binding sites in the promoter region. The regions upstream from the start point of transcription, but also just downstream, influence the regulation and degree of expression of the gene. The region immediately upstream, the core promoter, has DNA-binding sites to which a preinitiation complex comprising RNA PolII and TFIIA, B, D, E, F, and H binds (Tjian 1996).

The position of binding sites is given with reference to the start site of transcription (TSS). A box defined as TATA is present in about 75% of vertebrate RNA PolII promoters. A TATA box HMM trained on vertebrate promoter sequences has the consensus sequence TATAWDR (W = A/T, D is not C, R is G or A) starting at approximately −17 bp from TSS (Bucher 1990; http://www.epd.isb-sib.ch/promoter_elements/). This sequence is thought to position the initiation complex around TSS. A component of TFIID, TATA-binding protein (TBP), recognizes and binds to this sequence. INR is a loosely defined sequence around TSS that also influences the start position of transcription and may be recognized by other protein subunits of TFIID (Chalkley and Verrijzer 1999).



Another conserved sequence lying upstream of TATA and present in about one-half of vertebrate promoters is the CCAAT box, which is thought to be the site of binding of additional proteins that influence preinitiation and later stages of transcription. Another conserved regulatory site is the GC box. These boxes lie at variable distances from TSS and function in either orientation. Weight matrices that describe them have been produced (Bucher 1990).

The region upstream of the core promoter and other enhancer sites in the neighborhood of a gene also influences gene expression. A variety of transcription factors, some affected by environmental influences such as hormone levels, bind to DNA-binding sites in these regions. These factors can also form large multiprotein complexes that interact with a preinitiation complex to induce or repress transcription. These interactions can cause remodeling of the local nucleosome structure by histone acetylation or deacetylation, conformational changes in the transcription complex, and possibly phosphorylation of

Figure 8.12. The Hertz, Stormo, and Hartzell method for locating common DNA-binding sites for regulatory proteins in unaligned sequences (Hertz and Stormo 1999). This example illustrates how the algorithm compares a fixed window of sequence (length 4 in this example) in a set of sequences assumed to carry one site for a DNA-binding protein that cannot be readily found by aligning the sequences. The object is to find the 4-mer in each sequence that constitutes as nearly identical a pattern as can be found in all of the sequences. The user specifies the number of matrices that can be saved by the program for further analysis. Redundant matrices are eliminated. *Step 1.* The sequence of the first four bases from sequence 1 is first chosen. An analysis of only this one window is shown in this example. Normally the program would start with all possible 4-long words in each of the sequences, thus producing a total number of 6 possible step 1 matrices in this example. *Step 2.* The sequence window chosen in step 1 is moved across sequence 2, then sequence 3, and so on until all possible windows in all sequences have been selected. If a sufficient number of saved matrices is specified, this procedure would be repeated for all of the six saved matrices in step 1. Only one matrix is shown for illustration purposes. At each selected position, the number of matches with sequence 1 is recorded in a scoring matrix. The amount of sequence conservation in each column is calculated as the information content (*Ic*) of the column, and the *Ic* values for each column are then added to give *I* of the matrix. The best-scoring matrix is chosen. Calculation of the information content of a scoring matrix is discussed in detail in Chapter 4 (p. 195). Given a position in a test sequence that is being examined for a match to a matrix column, the maximum uncertainty of a matrix column is the number of questions that must be asked to find a match to the position in a test sequence. Uncertainty is zero if only one base is represented and 2 if all four bases are represented equally. Information content of a column is 2 minus the uncertainty of the column. For example, as each column in the first matrix in step 2 requires a single question to identify a match to a sequence (for column 1, one question must be asked: "Does the matching sequence position have an A or a T?"), then *I* of the matrix is $1+1+1+1=4$. The first column of the second matrix in step 2 has two As, and no other base is represented. Because no question need be asked, *I* is 2. A general method for calculating the amount of information in a column *c* is given by $Ic = \Sigma_i \{f_{ic} \log (f_{ic}/p_i)\}$ where f_{ic} is the fraction of each base in the column and p_i is the background frequency of base i in the sequences. If logarithms to the base 2 are used, then *I* units are in bits, and if natural logarithms are used, *I* units are in nats. *Step 3.* The sequence windows found in the highest-scoring matrix in step 2 are now compared to all other possible windows in the remaining sequences. In this case, only one sequence remains and the next high-scoring matrix is identified. Only one matrix is shown as an example; the maximum number that can be used for further analysis will be determined by the specified number of matrices that can be saved by the program. Additional steps (not shown) are then used to compare this best matrix with any remaining sequences until all have been included. The final matrices provide a consensus sequence by using the base in each column that has the highest score. The algorithm is greedy because the development of the highest-scoring matrix depends on matches found in ancestor matrices based on a smaller number of alignments. On the basis of this limitation and constraints provided by the user such as window size or matrix bias (see text), and on the number of matrices saved, the algorithm is not guaranteed to provide the optimal matrix for a large number of sequences.

RNA PolII. The independent binding of proteins to separate DNA sites in the initiation and upstream control regions is cooperative in that the binding of one protein to one site enhances the binding of another protein molecule to a second site. In this manner, a series of weak interactions between individual components is amplified by protein–protein interactions to give an overall strong binding of the complex to the promoter.

An example of a mammalian gene with multiple regulatory elements that have been defined by experiment is shown in Figure 8.13. The gene is the rat *pepCK* gene, which encodes phosphoenol pyruvate kinase, a major enzyme for metabolism of glucose in mammals. This gene is regulated by four different hormones—glucocorticoids, glucagon, retinoic acid, and insulin—through a system of binding sites for particular transcription factors in the promoter region. The response of the cell to these agents involves binding of the hormone to a specific receptor protein and the subsequent binding of the hormone–receptor complex to specific sequences called response elements (REs) in the promoter region of responsive genes. The *pepCK* gene also responds to the level of cyclic AMP (cAMP) through a similar interaction. In addition, the gene has other characteristic and essential sequence features for RNA PolII recognition, such as the TATA box, the initiation region (INR) that includes the transcription start site (TSS) at +1. The REs are flanked by binding sites for other transcription factors that influence the effect of the bound receptor through protein–protein interactions.

Thus, many different transcription factors may be involved in the regulation of a particular eukaryotic gene. The sequence of the DNA-binding site recognized by many of these TFs is not known, or only a few sites are known, thus limiting the ability to predict promoter-binding sites for these TFs. In some cases, enough DNA-binding sites are known to produce a weight matrix, described earlier in this chapter (Table 8.5). However, such scoring matrices tend to be much more variable than prokaryotic matrices, so that the matrix is less useful for discriminating true binding sites from random sequence variation.

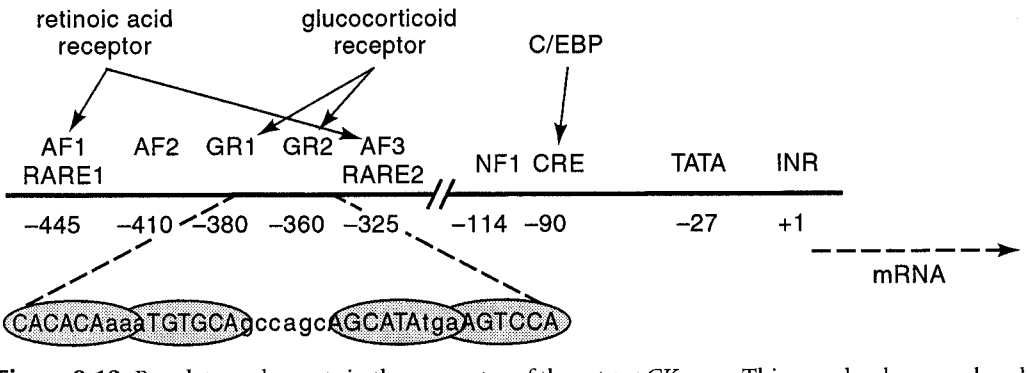


Figure 8.13. Regulatory elements in the promoter of the rat *pepCK* gene. This gene has been analyzed experimentally for the presence of transcription-factor-binding sites. The relative positions of these elements in a fusion of the *pepCK* promoter to a reporter gene are illustrated (Yamada et al. 1999). The glucocorticoid response unit (GRU) includes three accessory factor binding sites (AF1, AF2, and AF3), two glucocorticoid response elements (GR1 and GR2), and a cAMP response element (CRE). A dimer of glucocorticoid receptor bound to each GR element is depicted. The retinoic response unit (RAU) includes two retinoic acid response elements (RARE1 and RARE2) that coincide with the AF1 and AF3, respectively (Sugiyama et al. 1998). The sequences of the two GR sites and the binding of the receptor to these sites are shown. These sites deviate from the consensus sites and depend on their activity on accessory proteins bound to other sites in the GRU. This dependence on accessory proteins is reduced if a more consensus-like (canonical) GR element comprising the sequence TGTTCT is present. The CRE that binds factor C/EBP is also shown.

Such a matrix can be used to predict putative binding sites for a TF in a particular promoter. Because TF binding sites may be detectable on either forward and complementary strands or present on both forward and complementary strands in a repeated configuration, both strands of the test sequence are generally searched for binding sites. Interpolated HMMs described previously for identifying prokaryotic genes have also been used for eukaryotic promoter identification (Ohler et al. 1999). This method identifies the most informative lengths of sequence in promoters and uses them for promoter prediction in test sequences.

As shown in Figure 8.13, binding sites for TFs cluster in the promoter region. This clustering is the basis of one method for promoter prediction discussed below. A search for binding sites in the EPD database (Table 8.6) and a human first exon database showed that tandem binding sites for the same TF that are approximately 10 bp apart and expressing with a periodicity of 145 bp can be detected. Such studies confirm that searching for multiple TF binding sites can provide a more reasonable prediction of promoter function (Ioshikhes et al. 1999).

Complexes of TFs bound to DNA can either activate or repress transcription through their interaction with RNA PolII. Some quite remarkable variations of this theme can occur (Yamamoto et al. 1998). First, changes in the RE or in the binding of nearby accessory proteins can determine whether the binding of glucocorticoid response elements (GR) activates or represses transcription. Second, the GR can influence transcription simply by forming a complex with other factors and without binding to DNA itself. Thus, predicting the regulatory behavior solely on the basis of finding REs in a promoter region is probably not feasible without additional consideration of interactions among the regulatory elements and proteins themselves (Bucher et al. 1996).

RNA PolII Promoter Classification

Eukaryotic promoter sequences show variation not only between species, but also among genes within a species. A gene that is regulated by a certain set of signals during development will have a significantly different promoter than a second gene that responds to a different set of signals. For this reason, a set of promoters in an organism that share a regulatory response have been analyzed, as these promoters are expected to share common regulatory elements. Such an analysis has been performed on the genes expressed in skeletal muscle. Binding sites for TFs in skeletal muscle promoters are used to make scoring matrices, which are then used to find other muscle-regulated genes in genomic sequences. The ability of individual scoring matrices to locate signals in known muscle promoters, while at the same time not finding signals in control promoters, is determined. The alignment scores for each matrix are then weighted in favor of the most informative matrices. The sum of these weighted scores gives a value between 0 (no promoter) and 1 (has promoter function), called the logit value of the promoter. Similar promoters from closely related species are also used to enhance the ability of the method to discriminate muscle promoters from other promoters in a method described as phylogenetic footprinting (Wasserman and Fickett 1998).

Because the usefulness of different scoring matrices for TF binding sites is variable, other methods have been devised for weighting the scores obtained for an individual weight matrix on test sequences. An additional development includes a new algorithm for determining the cutoff value using the background rate estimated on non-promoters (see TFBIND in Table 8.6). Scores of matches of weight matrices to test sequences follow the extreme value distribution (p. 326), and have also been used to evaluate matches (Claverie 1994; Claverie and Audic 1996). The application of neural networks for devising a

Table 8.6. *Promoter prediction programs, Web pages, and related information*

Name	Web address	Reference
BDNA video analysis of transcription factor binding sites using conformational and physicochemical DNA features	see GeneExpress	Ponomarenko et al. (1999)
ConsInspector–see Transfac database[a]	http://www.gsf.de/biodv/consinspector.html	
Core-Promoter–for finding RNAPII promoters of human genes by quadratic discriminant analysis	http://argon.cshl.org/genefinder/CPROMOTER/index.htm	Zhang (1998a, b)
EPD Eukaryotic promoter database	http://www.epd.isb-sib.ch/; http://www.epd.isb-sib.ch/promoter_elements/	Bucher (1990); Périer et al. (1999, 2000)
EpoDB genes expressed during vertebrate erythropoiesis	http://www.cbil.upenn.edu/	Stoeckert et al. (1999)
FastM for transcription factor binding sites	http://genomatix.gsf.de/cgi-bin/fastm2/fastm.pl	Klingenhoff et al. (1999)
GeneExpress analysis of transcriptional regulations with TRRD database	http://wwwmgs.bionet.nsc.ru/systems/GeneExpress/	Kolchanov et al. (1999a, b)
Genome inspector for combined analysis of multiple signals in genomes	http://www.gsf.de/biodv/genomeinspector.html	Quandt et al. (1997)
GrailII[b] prediction of TSS by neural networks based on scores of characteristic sequence patterns and composition	http://compbio.ornl.gov/ see also book Web site	Uberbacher and Mural (1991); Uberbacher et al. (1996)
MAR-FINDER for finding matrix attachment regions	http://www.ncgr.org/MarFinder/	Kramer et al. (1997); Singh et al. (1997)
MatInd—see Transfac database		
MatInspector[a]–see Transfac database	http://www.gsf.de/biodv/matinspector.html (for downloading) http://www.gsf.de/cgi-bin/matsearch.pl (for interactive web page)	
Nuclear (including glucocorticoid) receptor resource[c]	http://nrr.georgetown.edu/GRR/GRR.html	Martinez et al. (1997)
Mirage (Molecular Informatics Resource for the Analysis of Gene Expression)[d]	http://www.ifti.org/	see Web page
NNPP Promoter Prediction by Neural Network for prokaryotes or eukaryotes	http://www.fruitfly.org/seq_tools/promoter.html	Reese et al. (1996)
NSITE–search for TF binding sites or other consensus regulatory sequences	http://genomic.sanger.ac.uk/gf/gf.shtml	see Web site
OOTFD Object-Oriented Transcription Factor Database	http://www.ifti.org/cgi-bin/ifti/ootfd.pl	Ghosh (1998)
PLACE plant *cis*-acting regulatory elements	http://www.dna.affrc.go.jp/htdocs/PLACE/	Higo et al. (1999)
PlantCARE plants *cis*-acting regulatory elements	http://sphinx.rug.ac.be:8080/PlantCARE/index.htm	Rombauts et al. (1999)
Pol3scan for RNAP III/tRNA promoter sequences using pattern scoring matrices	http://irisbioc.bio.unipr.it/genomics.html	Pavesi et al. (1994)
Polyadq for locating polyadenylation sites	http://argon.cshl.org/tabaska/polyadq_form.html	Tabaska and Zhang (1999)

Continued.

Table 8.6. *Continued.*

Name	Web address	Reference
Promoter element weight matrices and HMMs	http://www.epd.isb-sib.ch/promoter_elements/	Bucher (1990)
Promoter II for recognition of PolII sequences by neural networks	http://www.cbs.dtu.dk/services/promoter/	Knudsen (1999)
PromoterScan[e]	http://cbs.umn.edu/software/proscan/promoterscan.htm	Prestridge (1995) and see Web site
RegScan for promoter classification	http://wwwmgs.bionet.nsc.ru/mgs/programs/classprom/	Babenko et al. (1999)
Sequence walkers for graphical viewing of the interaction of regulatory protein with DNA binding site	http://www-lecb.ncifcrf.gov/~toms/walker/narcoverlogowalker.html	Schneider (1997)
Signal scan for transcriptional elements	http://bimas.dcrt.nih.gov:80/molbio/signal/	Prestridge (1991, 1996)
TargetFinder for promoter searching in selected annotated sequences	http://hercules.tigem.it/TargetFinder.html	Lavorgna et al. (1999)
TESS for searching for transcription factor binding sites	http://www.cbil.upenn.edu/tess/	Schug and Overton (1997a, b)
Tfbind for transcription factor binding sites	http://tfbind.ims.u-tokyo.ac.jp	Tsunoda and Takagi (1999)
Thyroid receptor resource[c]	http://xanadu.mgh.harvard.edu/receptor/trrfront.html	see Web page
Transfac programs providing search for TF binding sites. MatInd for making scoring matrices and MatInspector for searching for matches to matrices	http://www.gsf.de/cgi-bin/matsearch.pl	see http://www.gsf.de/biodv/staff_pub.html; Knüppel et al. (1994); Quandt et al. (1995); Heinemeyer et al. (1999); Klingenhoff et al. (1999)
TRRD transcriptional regulatory region database; see GeneExpress		Kolchanov et al. (1999a)
TSSG, like TSSW but based on sequences from a different promoter database	http://genomic.sanger.ac.uk/gf/gf.shtml; http://dot.imgen.bcm.tmc.edu:9331/seq-search/gene-search.html	see Web site
TSSW; recognition of human PolII promoter region and start of transcription by linear discriminant function analysis	http://genomic.sanger.ac.uk/gf/gf.shtml; http://dot.imgen.bcm.tmc.edu:9331/seq-search/gene-search.html	see Web site
Yeast cell cycle gene retrieval and promoter analysis	http://www.ncbi.nlm.nih.gov/CBBresearch/Landsman/Cell_cycle_data/upstream_seq.html; http://www.ncbi.nlm.nih.gov/CBBresearch/Landsman/Cell_cycle_data/	Wolfsberg et al. (1999)
Yeast cell cycle analysis project	http://genome-www.stanford.edu/cellcycle/info	Spellman et al. (1998)

Multiple methods of analysis are offered at sites http://dot.imgen.bcm.tmc.edu:9331/seq-search/gene-search.html and on http://genomic.sanger.ac.uk/gf/gf.shtml. Lists of Web sites are given at: http://linkage.rockefeller.edu/wli/gene/programs.html. A comparison of many of the promoter prediction programs included in this table and several additional ones on a small number of promoter-containing sequences not used in program training is available (Fickett and Hatzigeorgiou 1997).

[a] MatInspector DOS, Windows 95 and NT, and Mac versions and ConsInspector DOS and Mac versions available by FTP from ariane.gsf.de/pub/.

[b] GrailII must be given both gene and promoter sequences.

[c] Includes links to other receptor databases.

[d] The transcriptional informatics site MIRAGE includes links to regulatory data sites and programs.

[e] Accepts one person at a time; DOS version also available.

weighting scheme as used in the gene prediction program GeneParser would be another method for weighting a group of scoring matrices to give maximum discrimination between promoter and non-promoter sequences.

Gene microarrays discussed in Chapter 10 (p. 519) can assist with discovering which genes are regulated in the same manner and therefore should have binding sites for the same TFs (Cho et al. 1998; Eisen et al. 1998; Claverie 1999; Golub et al. 1999; Zhang 1999a,b). The promoter regions of these genes can be compared. The 5-mer content promoter sequences of yeast genes that are co-regulated during the cell cycle have been analyzed by the program wconsensus (described above, p. 365) and a Gibbs DNA sampler (similar to the Gibbs motif sampler described in Chapter 4 but adapted for DNA sequences) (Spellman et al. 1998; Zhang 1999b) and the results are available on a Web site (see Table 8.6).

In another study, pentamers and hexamers that are overrepresented among the upstream regions of cell-cycle-regulated genes were identified using a simple statistical sampling procedure. The sequences are divided into two sets; one set comprises cell cycle genes and the second set is the rest of the genome. A hexamer is then counted in both sets. The background number in the control set is used to identify overrepresented oligonucleotides in the cell cycle set. The actual number counted in the cell cycle genes is then compared to this expected value using a Chi-square test. For example, the hexamer ACGCGT is found with a variable location and orientation in the promoters of many cell cycle genes that are expressed during the late phase of the G_1 phase of the yeast cell cycle, whereas the pentamer CCCTT is located at positions -104 to -202 in one orientation in early G_1 (Wolfsberg et al. 1999). These types of analyses, which are available on Web sites (Table 8.6), demonstrate that computational analysis of the promoters of co-related genes reveals the presence of highly representative sequence patterns. Although some of these patterns correspond to the binding sites of transcription factors, others play a role that has yet to be determined. A similar method of oligomer counting has been used to identify overrepresented oligonucleotides with intron-containing genes in yeast and also to identify signals for localization of RNAs to mitochondria (Jacobs Anderson and Parker 2000). Hence, the oligonucleotide scoring method shows considerable promise for the identification of regulatory sites in co-regulated genes.

Prediction Methods for RNA PolII Promoters

A number of methods for predicting the location of RNA PolII promoters in genomic DNA have been derived. Several Web sites that offer an analysis are listed in Table 8.6. Also shown in this table are a number of Web sites that provide databases and information on TFs and their DNA-binding sites and other information related to transcriptional regulation in eukaryotes. A test analysis of these and several additional programs not listed in the table on a small number of new promoter sequences has been described previously (Fickett and Hatzigeorgiou 1997). The programs predicted 13–54% of the TSSs correctly, but each program also predicted a number of false-positive TSSs.

Samples of methods of analysis and programs included in Table 8.6 are listed below (for additional information on program availability, see Fickett and Hatzigeorgiou 1997; Frech et al 1997).

1. Use of a neural network trained on the TATA and Inr sites, allowing for a variable spacing between the sites (NNPP) or a neural network–genetic algorithm approach to identify conserved patterns in RNA PolII promoters and conserved spacing among the patterns (PROMOTER2.0).
2. Recognition of a TATA box using a weight matrix and an analysis of the density of TF sites. The density of TF sites at least 50 bp apart in known promoter sequences of the

eukaryotic promoter database (EPD) and on a set of non-promoter primate sequences from GenBank is compared and used to produce a promoter recognition profile (PromoterScan).

3. Use of a linear discriminant function as described above for gene prediction, but in this case, used for distinguishing features of promoter sequences from non-promoter sequences. The function is based on a TATA box score, triplet base-pair preferences around TSS, hexamer frequencies in consecutive 100-bp upstream regions, and TF binding-site prediction (TSSD and TSSW).
4. A quadratic discriminant analysis similar to that described above for gene prediction, but in this case, applied to variable lengths of sequence in the promoter region. The frequency of pentamers in a contiguous set of thirteen 30-bp windows and also in a second set of five 45-bp windows in the same 240-bp region was compared. This double-overlapping window appeared to reduce the background noise and to enhance the transcriptional signal from the promoter region (CorePromoter).
5. Searches of weight matrices for different organisms against a test sequence (TFSearch/TESS). Use of user-provided limits on type of weight matrix, key set of matches (core similarity) to individual matrices, and range of match scores (matrix similarity), and also generation of new matrices (MatInspector and ConsInspector).
6. Evaluation of test sequences for the presence of clustered groups or modules of TF binding sites that are characteristic of a given pattern of gene regulation (FastM).

REFERENCES

Adams M.D., Celniker E., Holt R.A., Evans C.A., Gocayne J.D., Amanatides P.G., Scherer S.E., Li P.W., Hoskins R.A., Galle R.F., George R.A., et al. 2000. The genome sequence of *Drosophila melanogaster*. *Science* **287:** 2185–2196.

Audic S. and Claverie J.M. 1998. Self-identification of protein-coding regions in microbial genomes. *Proc. Natl. Acad. Sci.* **95:** 10026–10031.

Babenko V.N., Kosarev P.S., Vishnevsky O.V., Levitsky V.G., Basin V.V., and Frolov A.S. 1999. Investigating extended regulatory regions of genomic DNA sequences. *Bioinformatics* **15:** 644–653.

Baldi P. and Brunak S. 1998. *Bioinformatics: The machine learning approach.* MIT Press, Cambridge, Massachusetts.

Baldi P., Brunak S., Chauvin Y., and Krogh A. 1996. Naturally occurring nucleosome positioning signals in human exons and introns. *J. Mol. Biol.* **263:** 503–510.

Berg O.G. and von Hippel P.H. 1987. Selection of DNA binding sites by regulatory proteins. Statistical-mechanical theory and application to operators and promoters. *J. Mol. Biol.* **193:** 723–750.

Birney E. and Durbin R. 2000. Using GeneWise in the *Drosophila* annotation experiment. *Genome Res.* **10:** 547–548.

Bisant D. and Maizel J. 1995. Identification of ribosome binding sites in *Escherichia coli* using neural network models. *Nucleic Acids Res.* **23:** 1632–1639.

Borodovsky M. and McIninch J. 1993. GeneMark: Parallel gene recognition for both DNA strands. *Comput. Chem.* **17:** 123–133.

Brendel V. and Kleffe J. 1998. Prediction of locally optimal splice sites in plant pre-mRNA with applications to gene identification in *Arabidopsis thaliana* genomic DNA. *Nucleic Acids Res.* **26:** 4748–4757.

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.

Brown N.P., Whittaker A.J., Newell W.R., Rawlings C.J., and Beck S. 1995. Identification and analysis of multigene families by comparison of exon fingerprints. *J. Mol. Biol.* **249:** 342–359.

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.

Bucher P. 1990. Weight matrix descriptions of four eukaryotic RNA polymerase II promoter elements derived from 502 unrelated promoter sequences. *J. Mol. Biol.* **212:** 563–578.

Bucher P., Fickett J.W., and Hatzigeorgiou A. 1996. Computational analysis of transcriptional regulatory elements: A field in flux. *Comput. Appl. Biosci.* **12:** 361–362.

Burge C.B. and Karlin S. 1998. Finding the genes in genomic DNA. *Curr. Opin. Struct. Biol.* **8:** 346–354.

Burset M. and Guigó R. 1996. Evaluation of gene structure prediction programs. *Genomics* **34:** 353–367.

Cardon L.R. and Stormo G.D. 1992. Expectation maximization algorithm for identifying protein-binding sites with variable lengths from unaligned DNA fragments. *J. Mol. Biol.* **223:** 159–170.

Carey M. and Smale S.T. 2000. *Transcriptional regulation in eukaryotes: Concepts, strategies, and techniques.* Cold Spring Harbor Laboratory Press, Cold Spring Harbor, New York.

Chalkley G.E. and Verrijzer C.P. 1999. DNA binding site selection by RNA polymerase II TAFs: A TAF(II)250-TAF(II)150 complex recognizes the initiator. *EMBO J.* **18:** 4835–4845.

Chen Q.K., Hertz G.Z., and Stormo G.D. 1995. MATRIX SEARCH 1.0: A computer program that scans DNA sequences for transcriptional elements using a database of weight matrices. *Comput. Appl. Biosci.* **11:** 563–566.

Cho R.J., Campbell M.J., Winzeler E.A., Steinmetz L., Conway A., Wodicka L., Wolfsberg T.G., Gabrielian A.E., Landsman D., Lockhart D.J., and Davis R.W. 1998. A genome-wide transcriptional analysis of the mitotic cell cycle. *Mol. Cell* **2:** 65–73.

Claverie J.-M. 1994. Some useful statistical properties of position-weight matrices. *Comput. Chem.* **18:** 287–294.

———. 1999. Computational methods for the identification of differential and coordinated gene expression. *Hum. Mol. Genet.* **8:** 1821–1832.

Claverie J.-M. and Audic S. 1996. The statistical significance of nucleotide position-weight matrix matches. *Comput. Appl. Biosci.* **12:** 431–439.

Dong S. and Searls D.B. 1994. Gene structure prediction by linguistic methods. *Genomics* **23:** 540–551.

Eisen M.B., Spellman P.T., Brown P.O., and Botstein D. 1998. Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci.* **95:** 14863–14868.

Farber R., Lapedes A., and Sirotkin K. 1992. Determination of eukaryotic protein coding regions using neural networks and information theory. *J. Mol. Biol.* **226:** 471–479.

Fickett J.W. 1982. Recognition of protein coding regions in DNA sequences. *Nucleic Acids Res.* **10:** 5303–5318.

Fickett J.W. and Hatzigeorgiou A.G. 1997. Eukaryotic promoter recognition. *Genome Res.* **7:** 861–878.

Fickett J.W. and Tung C.S. 1992. Assessment of protein coding measures. *Nucleic Acids Res.* **20:** 6441–6450.

Fields D.S., He Y., Al-Uzri A.Y., and Stormo G.D. 1997. Quantitative specificity of the Mnt repressor. *J. Mol. Biol.* **271:** 178–194.

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.

Frech K., Quandt K., and Werner T. 1997. Finding protein-binding sites in DNA sequences: The next generation. *Trends Biochem. Sci.* **22:** 103–104.

Frishman D., Mironov A., Mewes H.W., and Gelfand M. 1998. Combining diverse evidence for gene recognition in completely sequenced bacterial genomes. *Nucleic Acids Res.* **26:** 2941–2947.

Gelfand M.S., Mironov A.A., and Pevzner P.A. 1996. Gene recognition via spliced sequence alignment. *Proc. Natl. Acad. Sci.* **93:** 9061–9066.

Ghosh D. 1998. OOTFD (object-oriented transcription factors database): An object-oriented successor to TFD. *Nucleic Acids Res.* **26:** 360–362.

Gish W. and States D.J. 1993. Identification of protein coding regions by database similarity search. *Nat. Genet.* **3:** 266–272.

Golub T.R., Slonim D.K., Tamayo P., Huard C., Gaasenbeek M., Mesirov J.P., Coller H., Loh M.L., Downing J.R., Caligiuri M.A., Bloomfield C.D., and Lander E.S. 1999. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science* **286:** 531–537.

Gray M.W. and Covello P.S. 1993. RNA editing in plant mitochondria and chloroplasts. *FASEB J.* **7:** 64–71.

Guigó R. 1998. Assembling genes from predicted exons in linear time with dynamic programming. *J. Comput. Biol.* **5:** 681–702.

Guigó R., Knudsen S., Drake N., and Smith T. 1992. Prediction of gene structure. *J. Mol. Biol.* **226:** 141–157.

Gutman G.A. and Hatfield G.W. 1989. Nonrandom utilization of codon pairs in *Escherichia coli*. *Proc. Natl. Acad. Sci.* **86:** 3699–3703.

Hannenhalli S.S., Hayes W.S., Hatzigeorgiou A.G., and Fickett J.W. 1999. Bacterial start site prediction. *Nucleic Acids Res.* **27:** 3577–3582.

Hawley D.K. and McClure W.R. 1983. Compilation and analysis of *Escherichia coli* promoter DNA sequences. *Nucleic Acids Res.* **11:** 2237–2255.

Hayes W.S. and Borodovsky M. 1998. Deriving ribosome binding site (RBS) statistical models from unannotated DNA sequences and the use of the RBS model for N-terminal prediction. *Pac. Symp. Biocomput.*, 1998: 279–290.

Hebsgaard S.M., Korning P.G., Tolstrup N., Engelbrecht J., Rouze P., and Brunak S. 1996. Splice site prediction in *Arabidopsis thaliana* DNA by combining local and global sequence information. *Nucleic Acids Res.* **24:** 3439–3452.

Heinemeyer T., Chen X., Karas H., Kel A.E., Kel O.V., Liebich I., Meinhardt T., Reuter I., Schacherer F., and Wingender E. 1999. Expanding the TRANSFAC database towards an expert system of regulatory molecular mechanisms. *Nucleic Acids Res.* **27:** 318–322.

Henderson J., Salzberg S., and Fasman K.H. 1997. Finding genes in DNA with a hidden Markov model. *J. Comput. Biol.* **4:** 127–141.

Hertz G.Z. and Stormo G.D. 1995. Identification of consensus patterns in unaligned DNA and protein sequences: A large deviation statistical basis for penalizing gaps. In *Proceedings of the 3rd International Conference on Bioinformatics and Genome Research* (ed. H.A. Lim and C.R. Cantor), pp. 201–216. World Scientific, Singapore.

———. 1996. *Escherichia coli* promoter sequences: Analysis and prediction. *Methods Enzymol.* **273:** 30–42.

———. 1999. Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics* **15:** 563–577.

Hertz G.Z., Hartzell G.W., III, and Stormo G.D. 1990. Identification of consensus patterns in unaligned DNA sequences known to be functionally related. *Comput. Appl. Biosci.* **6:** 81–92.

Higo K., Ugawa Y., Iwamoto M., and Korenaga T. 1999. Plant *cis*-acting regulatory DNA elements (PLACE) database: 1999. *Nucleic Acids Res.* **27:** 297–300.

Hirosawa M., Sazuka T., and Yada T. 1997. Prediction of translation initiation sites on the genome of *Synechocystis* sp. strain PCC6803 by hidden Markov model. *DNA Res.* **4:** 179–184.

Horton P.B. and Kanehisa M. 1992. An assessment of neural network and statistical approaches for prediction of *E. coli* promoter sites. *Nucleic Acids Res.* **20:** 4331–4338.

Ioshikhes I., Trifonov E.N., and Zhang M.Q. 1999. Periodical distribution of transcription factor sites in promoter regions and connection with chromatin structure. *Proc. Natl. Acad. Sci.* **96:** 2891-2895.

Ioshikhes I., Bolshoy A., Derenshteyn K., Borodovsky M., and Trifonov E.N. 1996. Nucleosome DNA sequence pattern revealed by multiple alignment of experimentally mapped sequences. *J. Mol. Biol.* **262:** 129–139.

Jacobs Anderson J.S. and Parker R. 2000. Computational identification of *cis*-acting elements affecting post-transcriptional control of gene expression in *Saccharomyces cerevisiae*. *Nucleic Acids Res.* **28:** 1604–1617.

Klingenhoff A., Frech K., Quandt K., and Werner T. 1999. Functional promoter modules can be detected by formal models independent of overall nucleotide sequence similarity. *Bioinformatics* **15:** 180–186.

Knudsen S. 1999. Promoter2.0: For the recognition of PolII promoter sequences. *Bioinformatics* **15:** 356–361.

Knüppel R., Dietze P., Lehnberg W., Frech K., and Wingender E. 1994. TRANSFAC retrieval program: A network model database of eukaryotic transcription regulating sequences and proteins. *J. Comput. Biol.* **1:** 191–198.

Kolchanov N.A., Ananko E.A., Podkolodnaya O.A., Ignatieva E.V., Stepanenko I.L., Kel-Margoulis O.V., Kel A.E., Merkulova T.I., Goryachkovskaya T.N., Busygina T.V., Kolpakov F.A., Podkolodny N.L., Naumochkin A.N., and Romashchenko A.G. 1999a. Transcription regulatory regions database (TRRD): Its status in 1999. *Nucleic Acids Res.* **27:** 303–306.

Kolchanov N.A., Ponomarenko M.P., Frolov A.S., Ananko E.A., Kolpakov F.A., Ignatieva E.V., Podkolodnaya O.A., Goryachkovskaya T.N., Stepanenko I.L., Merkulova T.I., Babenko V.V., Ponomarenko Y.V., Kochetov A.V., Podkolodny N.L., Vorobiev D.V., Lavryushev S.V., Grigorovich D.A., Kondrakhin Y.V., Milanesi L., Wingender E., Solovyev V., and Overton G.C. 1999b. Integrated databases and computer systems for studying eukaryotic gene expression. *Bioinformatics* **15:** 669–686.

Kozak M. 1999. Initiation of translation in prokaryotes and eukaryotes. *Gene* **234:** 187–208.

Kramer J.A., Singh G.B., and Krawetz S.A. 1997. Computer assisted search for sites of nuclear matrix attachment. *Genomics* **33:** 302–308.

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.

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. *Ismb* **4:** 134–142.

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.

Lavorgna G., Guffanti A., Borsani G., Ballabio A., and Boncinelli E. 1999. TargetFinder: Searching annotated sequence databases for target genes of transcription factors. *Bioinformatics* **5:** 172–173.

Lewis L.K., Harlow G.R., Gregg-Jolly L.A., and Mount D.W. 1994. Identification of high affinity binding sites for LexA which define new DNA damage-inducible genes in *Escherichia coli*. *J. Mol. Biol.* **241:** 507–523.

Lopez A.J. 1998. Alternative splicing of pre-mRNA: Developmental consequences and mechanisms of regulation. *Annu. Rev. Genet.* **32:** 279–305.

Lukashin A.V. and Borodovsky M. 1998. GeneMark.hmm: New solutions for gene finding. *Nucleic Acids Res.* **26:** 1107–1115.

Marck C. 1988. DNA Strider: A 'C' program for the fast analysis of DNA and protein sequences on the Apple Macintosh family of computers. *Nucleic Acids Res.* **16:** 1829–1836.

Martinez E., Moore D.D., Keller E., Pearce D., Robinson V., MacDonald P.N., Simons S.S., Jr., Sanchez E., and Danielsen M. 1997. The nuclear receptor resource project. *Nucleic Acids Res.* **25:** 163–165.

Mathews B. 1975. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochim. Biophys. Acta* **405:** 442–451.

Morse D.P. and Bass B.L. 1999. Long RNA hairpins that contain inosine are present in *Caenorhabditis elegans* poly(A)$^+$ RNA. *Proc. Natl. Acad. Sci.* **96:** 6048– 6053.

Mulligan M.E. and McClure W.R. 1986. Analysis of the occurrence of promoter-sites in DNA. *Nucleic Acids Res.* **14:** 109–126.

Murakami K. and Takagi T. 1998. Gene recognition by combination of several gene-finding programs. *Bioinformatics* **14:** 665–675.

Ohler U., Harbeck S., Niemann H., Noth E., and Reese M.G. 1999. Interpolated Markov chains for eukaryotic promoter recognition. *Bioinformatics* **15:** 362–369.

Pachter L.K., Batzoglou S., Spitkovsky B.I., Banks E., Lander E.S., Leitman D.J., and Berger B. 1999. A dictionary-based approach for gene annotation. *J. Comput. Biol.* **6:** 419–430.

Paul M.S. and Bass B.L. 1998. Inosine exists in mRNA at tissue-specific levels and is most abundant in brain mRNA. *EMBO J.* **17:** 1120–1127.

Pavesi A., Conterio F., Bolchi A., Dieci G., and Ottonello S. 1994. Identification of new eukaryotic tRNA genes in genomic databases by a multistep weight matrix analysis of transcriptional control regions. *Nucleic Acids Res.* **22:** 1247–1256.

Pavy N., Rombauts S., Dehais P., Mathe C., Ramana D.V., Leroy P., and Rouze P. 1999. Evaluation of gene prediction software using a genomic data set: Application to *Arabidopsis thaliana* sequences. *Bioinformatics* **15:** 887–899.

Pedersen A.G. and Nielsen H. 1997. Neural network prediction of translation initiation sites in eukaryotes: Perspectives for EST and genome analysis. *Ismb* **5:** 226–233.

Pedersen A.G., Baldi P., Brunak S., and Chauvin Y. 1996. Characterization of prokaryotic and eukaryotic promoters using hidden Markov models. *Ismb* **4:** 182–191.

Pedersen A.G., Baldi P., Chauvin Y., and Brunak S. 1998. DNA structure in human RNA polymerase II promoters. *J. Mol. Biol.* **281:** 663–673.

Périer R.C., Junier T., Bonnard C., and Bucher P. 1999. The eukaryotic promoter database (EPD): Recent developments. *Nucleic Acids Res.* **27:** 307–309.

Périer R.C., Praz V., Junier T., Bonnard C., and Bucher P. 2000. The eukaryotic promoter database (EPD). *Nucleic Acids Res.* **28:** 302–303.

Ponomarenko J.V., Ponomarenko M.P., Frolov A.S., Vorobyev D.G., Overton G.C., and Kolchanov N.A. 1999. Conformational and physicochemical DNA features specific for transcription factor binding sites. *Bioinformatics* **15:** 654–668.

Prestridge D.S. 1991. SIGNAL SCAN: A computer program that scans DNA sequences for eukaryotic transcriptional elements. *Comput. Appl. Biosci.* **7:** 203–206.

———. 1995. Prediction of Pol II promoter sequences using transcription factor binding sites. *J. Mol. Biol.* **249:** 923–932.

———. 1996. SIGNAL SCAN 4.0: Additional databases and sequence formats. *Comput. Appl. Biosci.* **12:** 157–160.

Quandt K., Grote K., and Werner T. 1997. GenomeInspector: A new approach to detect correlation patterns of elements on genomic sequences. *Comput. Appl. Biosci.* **12:** 405–413.

Quandt K., Frech K., Karas H., Wingender E., and Werner T. 1995. MatInd and MatInspector: New, fast, and versatile tools for detection of consensus matches in nucleotide sequence data. *Nucleic Acids Res.* **23:** 4878–4884.

Reese M.G., Harris N.L., and Eeckman F.H. 1996. Large scale sequencing specific neural networks for promoter and splice site recognition. In *Biocomputing: Proceedings of the 1996 Pacific Symposium* (ed. L. Hunter and T.E. Klein). World Scientific, Singapore.

Reese M.G., Eeckman F.H., Kulp D., and Haussler D. 1997. Improved splice site detection in Genie. *J. Comput. Biol.* **4:** 311–323.

Reese M.G., Kulp D., Tammana H., and Haussler D. 2000. Genie — Gene finding in *Drosophila melanogaster*. *Genome Res.* **10:** 529–538.

Rice P.M., Elliston K.E., and Gribskov M. 1991. DNA. In *Sequence analysis primer* (ed. M. Gribskov and J. Devereux), pp. 43–49. Stockton Press, New York.

Robison K., McGuire A.M., and Church G.M. 1998. A comprehensive library of DNA-binding site matrices for 55 proteins applied to the complete *Escherichia coli* K-12 genome. *J. Mol. Biol.* **284:** 241–254.

Rombauts S., Dehais P., Van Montagu M., and Rouze P. 1999. PlantCARE, a plant *cis*-acting regulatory element database. *Nucleic Acids Res.* **27:** 295–296.

Salgado H., Santos A., Garza-Ramos U., van Helden J., Diaz E., and Collado-Vides J. 1999. RegulonDB (version 2.0): A database on transcriptional regulation in *Escherichia coli*. *Nucleic Acids Res.* **27:** 59–60.

Salzberg S. 1998. Decision trees and Markov chains for gene finding. In *Computational methods in molecular biology* (ed. S.L. Salzberg et al.), chap. 10, pp. 187–203. Elsevier, Amsterdam, The Netherlands.

Salzberg S., Delcher A., Kasif S., and White O. 1998. Microbial gene identification using interpolated Markov models. *Nucleic Acids Res.* **26:** 544–548.

Schneider T.D., Stormo G.D., Gold L., and Ehrenfeucht A. 1986. Information content of binding sites on nucleotide sequences. *J. Mol. Biol.* **188:** 415–431.

Schug J. and Overton G.C. 1997a. Modeling transcription factor binding sites with Gibbs sampling and minimum description length encoding. *Ismb* **5:** 268–271.

———. 1997b. TESS: Transcription element search software on the WWW. Computational Biology and Informatics Laboratory, University of Pennsylvania School of Medicine (Technical Report CBIL-TR-1997-1001-v0.0).

Schneider T.D. 1997. Sequence walkers: A graphical method to display how binding proteins interact with DNA or RNA sequences (erratum appears in *Nucleic Acids Res.* [1998] **26:** following 1134). *Nucleic Acids Res.* **25:** 4408–4415.

Searls D. 1998. Grand challenges in computational biology. In *Computational methods in molecular biology* (ed. S.L. Salzberg et al.), chap. 1, pp. 1–10. Elsevier, Amsterdam, The Netherlands.

Sharp P.M. and Li W.H. 1987. The codon adaptation index — A measure of directional synonymous codon usage bias, and its potential applications. *Nucleic Acids Res.* **15:** 1281–1295.

Singh G.B., Kramer J.A., and Krawetz S.A. 1997. Mathematical model to predict regions of chromatin attachment to the nuclear matrix. *Nucleic Acids Res.* **25:** 1419–1425.

Snyder E.E. and Stormo G.D. 1993. Identification of coding regions in genomic DNA sequences: An application of dynamic programming and neural networks. *Nucleic Acids Res.* **21:** 607–613.

———. 1995. Identification of protein coding regions in genomic DNA. *J. Mol. Biol.* **248:** 1–18.

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.

Solovyev V.V., Salamov A.A., and Lawrence C.B. 1994. Predicting internal exons by oligonucleotide composition and discriminant analysis of spliceable open reading frames. *Nucleic Acids Res.* **22:** 5156–5163.

———. 1995. Identification of human gene structure using linear discriminant functions and dynamic programming. *Ismb* **3:** 367–375.

Spellman P.T., Sherlock G., Zhang M.Q., Iyer V.R., Anders K., Eisen M.B., Brown P.O., Botstein D., and Futcher B. 1998. Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. *Mol. Biol. Cell* **9:** 3273–3297.

Spingola M., Grate L., Haussler D., and Ares M., Jr. 1999. Genome-wide bioinformatic and molecular analysis of introns in *Saccharomyces cerevisiae*. *RNA* **5:** 221–234.

Staden R. 1984. Computer methods to locate signals in nucleic acid sequences. *Nucleic Acids Res.* **12:** 505–519.

———. 1989. Methods for calculating the probabilities of finding patterns in sequences. *Comput. Appl. Biosci.* **5:** 89–96.

———. 1990. Finding protein coding regions in genomic sequences. *Methods Enzymol.* **183:** 163–180.

Staden R. and McLachlan A.D. 1982. Codon preference and its use in identifying protein coding regions in long DNA sequences. *Nucleic Acids Res.* **10:** 141–156.

Stein A. and Bina M. 1999. A signal encoded in vertebrate DNA that influences nucleosome positioning and alignment. *Nucleic Acids Res.* **27:** 848–853.

Stoeckert C.J., Jr., Salas F., Brunk B., and Overton G.C. 1999. EpoDB: A prototype database for the analysis of genes expressed during vertebrate erythropoiesis. *Nucleic Acids Res.* **27:** 200–203.

Stormo G.D. and Fields D.S. 1998. Specificity, free energy and information content in protein-DNA interactions. *Trends Biochem. Sci.* **23:** 109–113.

Stormo G.D. and Hartzell G.W., III. 1989. Identifying protein-binding sites from unaligned DNA fragments. *Proc. Natl. Acad. Sci.* **86:** 1183–1187.

Stormo G.D., Schneider T.D., Gold L., and Ehrenfeucht A. 1982. Use of the 'Perceptron' algorithm to distinguish translational initiation sites in *E. coli*. *Nucleic Acids Res.* **10:** 2997–3011.

Sugiyama T., Scott D.K., Wang J.C., and Granner D.K. 1998. Structural requirements of the glucocorticoid and retinoic acid response units in the phosphoenolpyruvate carboxykinase gene promoter. *Mol. Endocrinol.* **12:** 1487–1498.

Tabaska J.E. and Zhang M.Q. 1999. Detection of polyadenylation signals in human DNA sequences. *Gene* **231:** 77–86.

Thanaraj T.A. 1999. A clean data set of EST-confirmed splice sites from *Homo sapiens* and standards for clean-up procedures. *Nucleic Acids Res.* **27:** 2627–2637.

Thieffry D., Salgado H., Huerta A.M., and Collado-Vides J. 1998. Prediction of transcriptional regulatory sites in the complete genome sequence of *Escherichia coli* K-12. *Bioinformatics* **14:** 391–400.

Tiwari S., Ramachandran S., Bhattacharya A., Bhattacharya S., and Ramaswamy R. 1997. Prediction of probable genes by Fourier analysis of genomic sequences. *Comput. Appl. Biosci.* **13:** 263–270.

Tjian R. 1996. The biochemistry of transcription in eukaryotes: A paradigm for multisubunit regulatory complexes. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* **351:** 491–499.

Tsunoda T. and Takagi T. 1999. Estimating transcription factor bindability on DNA. *Bioinformatics* **15:** 622–630.

Uberbacher E.C. and Mural R.J. 1991. Locating protein-coding regions in human DNA sequences by a multiple sensor-neural network approach. *Proc. Natl. Acad. Sci.* **88:** 11261–11265.

Uberbacher E.C., Xu Y., and Mural R.J. 1996. Discovering and understanding genes in human DNA sequence using GRAIL. *Methods Enzymol.* **266:** 259–281.

von Heijne G. 1987. *Sequence analysis in molecular biology — Treasure trove or trivial pursuit*, pp. 50–54. Academic Press, San Diego, California.

Wada K., Wada Y., Ishibashi F., Gojobori T., and Ikemura T. 1992. Codon usage tabulated from the GenBank genetic sequence data. *Nucleic Acids Res.* (suppl.) **20:** 2111–2118.

Wasserman W.W. and Fickett J.W. 1998. Identification of regulatory regions which confer muscle-specific gene expression. *J. Mol. Biol.* **278:** 167–181.

Wolfsberg T.G., Gabrielian A.E., Campbell M.J., Cho R.J., Spouge J.L., and Landsman D. 1999. Candidate regulatory sequence elements for cell cycle- dependent transcription in *Saccharomyces cerevisiae*. *Genome Res.* **9:** 775–792.

Yamada K., Duong D.T., Scott D.K., Wang J.C., and Granner D.K. 1999. CCAAT/enhancer-binding protein beta is an accessory factor for the glucocorticoid response from the cAMP response element in the rat phosphoenolpyruvate carboxykinase gene promoter. *J. Biol. Chem.* **274:** 5880–5887.

Yamamoto K.R., Darimont B.D., Wagner R.L., and Iniguez-Lluhi J.A. 1998. Building transcriptional regulatory complexes: Signals and surfaces. *Cold Spring Harbor Symp. Quant. Biol.* **63:** 587–598.

Zhang M.Q. 1997. Identification of protein coding regions in the human genome based on quadratic discriminant analysis (erratum appears in *Proc. Natl. Acad. Sci.* [1997] **94:** 5495). *Proc. Natl. Acad. Sci.* **94:** 565–568.

———. 1998a. Identification of human gene core promoters in silico. *Genome Res.* **8:** 319–326.

———. 1998b. A discrimination study of human core-promoters. *Pac. Symp. Biocomput.*, 1998: 240–251.

———. 1999a. Large-scale gene expression data analysis: A new challenge to computational biologists. *Genome Res.* **9:** 681–688.

———. 1999b. Promoter analysis of co-regulated genes in the yeast genome. *Comput. Chem.* **23:** 233–250.