Computational Genefinding

David Haussler
Computer Science Department
University of California
Santa Cruz, CA 95064
Tel: 831 459 2105
FAX: 831 459 4829
email: haussler@cse.ucsc.edu
www: http://www.cse.ucsc.edu/ haussler

Summary

We briefly review computational methods for finding genes in genomic DNA sequences. Specific programs are now available to find genes in the genomic DNA of many organisms. We discuss the approaches used by these programs, their performance, and future directions for this field.

1 Introduction

Computational methodology for finding genes and other functional sites in genomic DNA has evolved significantly over the last 20 years. Excellent recent surveys have been given by Gelfand [27], Fickett [20, 21], Guigó [31], Claverie [13], Milanesi and Rogosin [50], and Krogh [41]. Extensive bibliographies are available at `http://linkage.rockefeller.edu/wli/gene/` and `http://www-hto.usc.edu/software/procrustes/fans_ref/`. Here we give only a very brief overview.

Among the types of functional sites in genomic DNA that researchers have sought to recognize are splice sites, start and stop codons, branch points, promoters and terminators of transcription, polyadenylation sites, ribosomal binding sites, topoisomerase II binding sites, topoisomerase I cleavage sites, and various transcription factor binding sites [27]. Local sites such as these are called *signals* and methods for detecting them may be called *signal sensors*. Genomic DNA signals can be contrasted with extended and variable length regions such as exons and introns, which are recognized by different methods that may be called *content sensors* [64, 65].

2 Signal Sensors

The most basic signal sensor is a simple consensus sequence or an expression that describes a consensus sequence along with allowable variations, such as a PROSITE expression [66, 2]. More sensitive sensors can be designed using weight matrices in place of the consensus, in which each position in the pattern allows a match to any residue, but different costs are associated with matching each residue in each position [64, 67, 66, 3, 12]. The score returned

by a weight matrix sensor for a candidate site is the sum of the costs of the individual residue matches over that site. If this score exceeds a given threshold, the candidate site is predicted to be a true site. Such sensors have a natural probabilistic interpretation in which the score returned is a log likelihood ratio under a simple statistical model in which each position in the site is characterized by an independent and distinct distribution over possible residues. A mathematically equivalent interpretation of the score is that it is the discrimination energy for site recognition [3].

Weight matrices can also be viewed as a simple type of neural network, sometimes called a perceptron [67, 66]. Many investigators have also applied more complex neural networks, such as multi-layer feed-forward networks and time delay networks, to various DNA signal recognition problems [8, 19, 49, 53, 54, 46, 32]. Multi-layer nets have the ability to capture statistical dependency between the residues at different positions in a site, an ability that perceptrons (and hence weight matrices) lack. Time delay neural networks also allow insertions and deletions while evaluating a match to a prospective site, whereas weight matrices and feed-forward neural networks do not [56]. Other statistical/pattern models besides neural networks, such as nonhomogeneous Markov models (a weight matrix where the distribution at position i depends on the residue at position $i-1$, sometimes called "WAM" models), decision trees, quadratic discriminant functions, and graphical models, have also been used as biosequence signal sensors [37, 76, 63, 15, 58, 1]. In general, the penalty for these more sophisticated models is that much more training data is needed to estimate the many parameters that they contain, so they are unsuitable in cases where relatively few verified examples are known of the site to be modeled.

3 Content Sensors

The most important and most studied content sensor is the sensor that predicts coding regions. An extensive review of computational methods to detect coding regions is given by Fickett and Tung [23] (see also [20, 21]). In prokaryotes, it is still common to locate genes by simply looking for long open reading frames (ORFs); this is certainly not adequate for higher eukaryotes. To discriminate coding from non-coding regions in eukaryotes, exon content sensors often use in-frame hexamer counts or, what is nearly equivalent, a set of 3 fifth-order Markov models, one for each of the three nucleotide positions within a codon, as pioneered in the genefinder GeneMark [7]. It is also important to consider local compositional biases, as the codon preferences are quite different between genes in G+C rich regions and genes in A+T rich regions [55, 18, 7]. While many other measures of coding potential have been investigated (Fickett tested 19 different measures, which he took from the literature [21]), few others have been proven to be as effective. However, combinations of several measures can be effective, as in the popular GRAIL exon detector, in which several coding measures are combined along with base composition and signal sensor output for flanking splice sites, and fed into a neural net to predict exons [71].

Other content sensors include sensors for CpG islands, which are regions that often occur near the beginnings of genes where the frequency of the dinucleotide CG is not as low as it typically is in the rest of the genome [4, 25, 47], and sensors for repetitive DNA, such as ALU sequences [36, 35, 51]. The latter sensors are often used as masks or filters that completely

remove the repetitive DNA, leaving the remaining DNA to be analyzed.

4 Integrated Gene Finding Methods

Signal and content sensors alone cannot solve the genefinding problem. The statistical signals they are trying to recognize are too weak [1], and there are dependencies between signals and contents that they cannot capture [11], such as the possible correlation between splice site strength and exon size [78]. During the last five years, a number of systems have been developed that combine signal and content sensors to try to identify complete gene structure. Such systems are capable, in principle, of handling more complex interdependencies between gene features. A linguistic metaphor is sometimes applied here, likening the process of breaking down a sequence of DNA into genes, each of which is a series of exons and introns, to the process of parsing a sentence by breaking it down into its constituent grammatical parts. Indeed this parsing metaphor can be pushed deeper. Searls[60, 16] was the first major proponent of describing gene structure in linguistic terms using a formal grammar. His genefinding program, GenLang, was one of the earliest integrated genefinders, following on the pioneering work of Gelfand [26], Gelfand and Roytberg [30], Fields and Soderlund [24], and Phil Green's GeneFinder[69], and was one of the inspirations for significant later work (e.g. [6, 5] and the HMM methods described below.)

Nearly all integrated genefinders use dynamic programming to combine candidate exons and other scored regions and sites into an complete gene prediction with maximal total score. A brief and lucid tutorial on this topic can be found in [41] and a more detailed exposition in [17]. Gelfand, *et al*, proposed a dynamic programming scheme, embodied in the genefinder GREAT[29], that calculates the set of all so-called Pareto-optimal gene structure predictions, which include the optimal predictions for a wide variety of different scoring functions. Dynamic programming methods are also used in Grail II [73], GeneParser [62], FGENEH [63], and recent versions of GeneID [31].

Dynamic programming methods find the candidate gene structure with the best overall score. The key to success in these methods is developing the right score function. A fruitful approach here has been to define a statistical model of genes that includes parameters describing codon dependencies in exons, characteristics of splice sites (e.g. the parameters of a weight matrix for splice sites), as well as "linguistic" information on what functional features are likely to follow other features (see Figure 1). In this approach the observed DNA sequences are actually modeled as if they were manifestations of a stochastic process that generates gene-containing DNA. This process includes a latent (or "hidden") variable associated with each nucleotide that represents the functional role or position of that nucleotide, e.g. a G residue might be part of a GT consensus donor splice site or it might be in the third position of a start codon. Taken together, the states of these hidden variables define a candidate gene structure. The linguistic rules for what functional features follow what other features are expressed by the parameters of a Markov process on the hidden variables. For this reason, these models are called hidden Markov models, or HMMs. Because a Markov process is just a finite state machine with probabilities on the state transitions, genefinding HMMs are merely a stochastic version of the genefinding finite state machines (regular grammars) introduced by Searls.

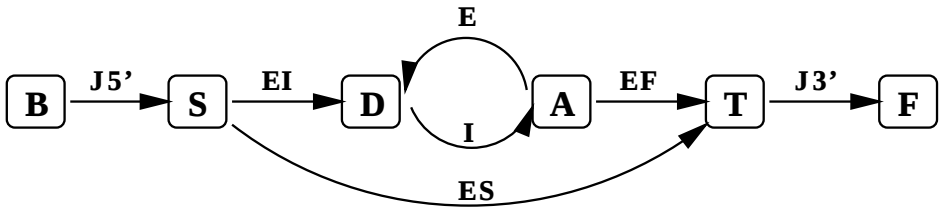

Figure 1: A simplified diagram representing the liguistic rules for what might follow what when parsing a sequence consisting of a multiple exon gene. The arcs represent contents and the nodes represent signals. The contents are J5' : 5' UTR, EI : Initial Exon, E : Exon, I : Intron, E : Internal Exon, EF: Final Exon, ES : Single Exon, and J3' : 3' UTR. The signals are B : Begin sequence, S : Start Translation, D : Donor splice site, A : Acceptor splice site, T : Stop Translation, F : End sequence. A candidate gene structure is created by tracing a path in this figure from B to F. An HMM (GHMM) is defined by attaching stochastic models to each of the arcs and nodes. Figure taken from [44].

The advantage of HMMs is that, being probabilistic models, they define a natural score function. Let X denote the DNA sequence, Q denote a possible sequence of hidden states, one for each nucleotide in X, and θ denote the parameters of the HMM. Since Q represents a candidate gene structure for X, to find the genes in X, we want to find the Q that is most likely given the sequence X, *i.e.*, we want to find the Q that maximizes $P(Q|X,\theta)$, the probability of the gene structure Q given the DNA sequence X and the parameters θ. Equivalently, we can maximize $\log P(Q|X,\theta)$. This is the score function that is optimized in a genefinding HMM. It can be optimized using standard dynamic programming methods.

Early genfinding HMMs were EcoParse (for *E. coli* [42], also recently used in the annotation of the *M. Tuberculosis* genome [14]) and Xpound (for human) [70]. More recent programs are GeneMark-HMM (for bacterial genomes) [48] Veil [33] and HMMgene (for human) [41]. A somewhat more general class of probabilistic models, called generalized HMMs (GHMMs) or (hidden) semi-Markov models, have their roots in GeneParser [62], and were more fully developed in Genie [44, 57, 45] and then GenScan [9] (see also [72]).

The probabilistic approach has further advantages. For example, for any given feature, such as a 5' splice site, and any position in the DNA sequence X, we can calculate the probability that that feature occurs at that position. If we do this for separately for each feature of our overall predicted gene structure, then this gives us a kind of individual "confidence" value for each part of our prediction. GeneParser [62] pioneered this methodology (see further theoretical discussion in [68]), and it is used to give highly accurate confidence values for predicted exons in Genscan [9]. In addition, the probabilistic formulation provides various new ways to estimate the parameters θ of the gene-finding model. Given a large "training" DNA contig (or set of contigs) X and its correct state sequence annotation Q, we can find θ to maximize $P(X,Q|\theta)$ (the maximum likelihood approach), $P(\theta|X,Q)$ (the maximum *a posteriori* approach), or $P(Q|X,\theta)$ (the conditional maximum likelihood approach) [40]. It is even possible to estimate the parameters θ from partially annotated training sequences using the expectation-maximization method [17].

So far we have focused on genefinders that predict gene structure based only on general features of genes, rather than using explicit comparisons to other, previously known genes, or auxiliary information such as expressed sequence tag (EST) matches. One way to include information about previously known genes is to use the database of known proteins as a basis for gene prediction. Current state-of-the-art genefinding systems combine multiple statistical measures with database homology searches, obtained by translating the DNA to protein in all possible reading frames, and then searching the protein databases for similar protein sequences. Examples are Genie [45], GeneID+ [10], GeneParser3 [62], and recent versions of Grail [75]. The program AAT [34] and new versions of Grail also take into account EST information [74]. Database homology has long been used as a *post hoc* method to validate gene predictions, but these systems were among the first to integrate database homology directly into the genefinding algorithm itself. This approach has been taken to its extreme limit in a genefinding program developed by Gelfand, Mironov, and Pevzner[28]. This system, called Procrustes, requires the user to provide a close protein homolog of the gene to be predicted. Then a "spliced alignment" algorithm, similar to a Smith-Waterman[61] alignment, is used to derive a putative gene structure by aligning the DNA to the homolog. The major disadvantage to this method is the requirement of a close homolog. It is often the case that homologs are unknown or are remote, in which case this system would be inappropriate. Nevertheless, in the presence of a very close homolog, Procrustes is an extremely effective gene finding method. Recent related methods, based on HMM models, have been developed by Birney and Durbin [5] and are currently being developed by Kulp [43].

In 1995, a number of different integrated genefinders were tested on a benchmark set of 570 vertebrate genes by Burset and Guigó [10]. They looked at not only how many bases were predicted correctly as either coding or non-coding, but how many exons were predicted exactly, with both splice sites located correctly. In the former case, accuracy was about 75-80%. In the latter it was about 40-60%. These numbers are for systems that do not employ protein database homology searches. When database homology is employed, the upper limit for the accuracy increases about 10% in both categories. Integrated eukaryotic genefinding systems based on HMM and GHMM models, starting with Genie, and followed by Veil, Genscan and HMMgene have pushed beyond these early performance numbers, with the latter two programs now obtaining upwards of 90% accuracy at the level of individual nucleotides and 80% for exact exon prediction, without the use of database homologies. A new category of completely correct gene prediction has been added to the list of performance measurements, and Genscan achieves an accuracy of about 40% on the Burset and Guigó dataset in this category [9]. Tests have also been conducted on the identification of promoters, showing that the accuracy of currently available methods is much lower on this task [22].

The currently available genefinding performance results must be approached with extreme caution. The primary reason is that they depend very strongly on the difficulty of the genes in the test set, and for some genefinders, on the homology overlap between the genes in the test set and those in the training set that is used to optimize the parameters of the models [31, 41]. The latter is a factor even when no homology is explicitly used by the genefinding method. To avoid this problem, it is best to compare genefinders by training and testing on the same genes, and to avoid homologies between genes used for training and testing. Reese has constructed benchmark sets for human and for *Drosophila* genes of this type that are ran-

domly partitioned into specified parts for use in cross-validated train-test experiments. These have been used by Genie, Genscan and HMMgene (ftp://www-hgc.lbl.gov/pub/genesets/). Reese's human dataset is a bit harder than the original Burset and Guigó dataset as well, so genefinding programs get overall lower scores on it. Furthermore, the variance in performance from one train-test partition to another is quite high, since some parts by chance ended up with more "hard-to-predict" genes (usually genes with many exons and or long introns) than others. This graphically demonstrates the unreliability of the currently available genefinding performance figures: if by chance a different set of human genes had been included in Genbank, the numbers would have been quite different, and probably lower, since Genbank is biased towards genes with fewer exons and shorter introns. We need a much larger sample of human genes before we can get stable performance numbers.

Reese's datasets, like those of Burset and Guigó, contain exactly one gene per sequence. Little is known about the accuracy of genefinders on large genomic sequences containing multiple genes. Some harder and more realistic human genomic data, consisting of large annotated contigs, is available at `http://igs-server.cnrs-mrs.fr/banbury/index.html`. Annotated *C. elegans* gene data is available at

`http://www.sanger.ac.uk/Projects/C_elegans/genefinding/`.

The latter site also proposes a standardized format (Gene Finding Format, or GFF) for both gene annotation and comparing the results of various genefinders. It would greatly aid the maturation of this field if we could agree on a simple standard data interchange format like this. Once this is established, we could then share a set of tools for the display, comparison, analysis and combination of different gene predictions, along with auxiliary sequence annotation.

5 Discussion

It is important to distinguish two different goals in genefinding research. The first goal is to provide computational methods to aid in the annotation of the large volume of genomic data that is produced by genome sequencing efforts. The second goal is to provide a computational model to help elucidate the mechanisms involved in transcription, splicing, polyadenylation and other critical processes in the pathway from genome to proteome. While there is some overlap in these goals, there is also some conflict. No one computational genefinding approach will be optimal for both goals. A "purist" system that mimics the cellular processes cannot take advantage of homologies with other proteins and matches to EST sequences when deciding where to splice. It presumably should not use codon statistics, frame consistency between exons, or lack of in-frame stop codons to predict overall gene structure, although there is some evidence that absence of early in-frame stop codons may be involved in biological start site selection [39]. One would think that these restrictions would completely cripple computational genefinding methods, however Guigó has shown that just using simple weight matrices to find the best combination of splice site signals, translation start and stop signals, along with the standard syntactic constraints on gene structure (frame consistency, no in-frame stop codons, minimum intron size), gives results on his benchmark

data set that are comparable to those obtained by most of the genefinders he and Burset tested in 1995 [31]. These results are not competitive with the older genefinders that use protein homology, nor with the newer methods that use exon coding potential but not homology, but they nevertheless indicate a surprising potential for purist genefinding models. More detailed models of the splicing process, the selection of translation start and the process of polyadenylation may significantly improve such purist models. These models may prove useful in human genome annotation for finding rapidly evolving and rarely expressed genes, especially those with unusual codon usage. However, if we simply want to produce genefinders that give the most reliable annotation in "everyday" genome center annotation efforts, it is clear that more work needs to be done to incorporate EST information along with protein homology and powerful statistical models.

There are other key issues that will effect future research in both of the above computational genefinding paradigms. One is the issue of alternative splicing. No currently available genefinders handle alternative splicing in an effective manner. Intimately tied with this issue is that of gene regulation. The abundant regulatory signals flanking genes, and appearing in introns (and sometimes in exons [52]), combined with regulatory proteins specific to the cell type and cell state, determine the expression of the gene. Gene annotation is not complete until these signals are identified, and the cellular conditions that give rise to differing expression levels for different transcripts are worked out. This implies, among other things, that future genefinders will need to explicitly take into account experimental data relating to differential expression, along with the other types of data we have discussed (see e.g. [38]). It may be anticipated that this task will occupy genefinding researchers for some years to come.

Acknowledgments

The author gratefully acknowledges the support of DOE grant DE-FG03-95ER6211, and thanks R. Guigó, D. Kulp, M. Reese and the editor for helpful suggestions.

URLs

Computational genefinding bibliographies:
`http://linkage.rockefeller.edu/wli/gene/`
`http://www-hto.usc.edu/software/procrustes/fans_ref/`

Genfinding Datasets:
Single genes: ftp://www-hgc.lbl.gov/pub/genesets/
Annotated contigs: `http://igs-server.cnrs-mrs.fr/banbury/index.html`
`http://www.sanger.ac.uk/Projects/C_elegans/genefinding/`

Some HMM-based genefinders genes:
Genie [44, 57, 45]: `http://www.cse.ucsc.edu/~dkulp/cgi-bin/genie`
GenScan [9]: http://CCR-081.mit.edu/GENSCAN.html

HMMgene [41]: http://www.cbs.dtu.dk/services/HMMgene/
GeneMark-HMM [48]: http://genemark.biology.gatech.edu/GeneMark/hmmchoice.html Veil [33]: http://www.cs.jhu.edu/labs/compbio/veil.html

Some further genefinders:
AAT [34]: http://genome.cs.mtu.edu/aat.html
FGENEH [63]: `http://dot.imgen.bcm.tmc.edu:9331/gene-finder/gf.html`
GENEID [31]: `http://www.imim.es/GeneIdentification/Geneid/geneid_inp ut.html`
Genlang [60]: `http://cbil.humgen.upenn.edu/~sdong/genlang_home.html`
GeneParser [62]: `http://beagle.colorado.edu/~eesnyder/GeneParser.html`
Glimmer [59]: `http://www.cs.jhu.edu/labs/compbio/glimmer.html`
Grail [73]: http://compbio.ornl.gov/
MZEF [77]: http://www.cshl.org/genefinder
Procrustes [28]: `http://www-hto.usc.edu/software/procrustes/`

Full version of this review: `http://www.cse.ucsc.edu/~haussler/pubs.html`

References

[1] P. Agarwal and V. Bafna. The ribosome scanning model for translation initiation for gene prediction and full-length cDNA detection. In *Proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology*, pages 2–7, 1998.

[2] A. Bairoch. Prosite: a dictionary of sites and patterns in proteins. *Nucleic Acids Research*, 20:2013–2018, 1992.

[3] O. Berg and P. von Hippel. Selection of DNA binding sites by regulatory proteins. *J. Mol. Biol.*, 193:723–750, 1987.

[4] A. P. Bird. CpG islands as gene markers in the vertebrate nucleus. *Trends Genet*, 3:342–347, 1987.

[5] E. Birney and R. Durbin. Dynamite: a flexible code generating language for dynamic programming methods used in sequence comparison. In *ismb97*, pages 56–64, 1997.

[6] E. Birney, J. Thompson, and T. Gibson. PairWise and SearchWise: finding the optimal alignment is a simultaneous comparison of a protein profile against all DNA translation frames. *NAR*, 24:2730–2739, 1996.

[7] M. Borodovsky and J. McIninch. Genmark: Parallel gene recognition for both DNA strands. *Computers and Chemistry*, 17(2):123–133, 1993.

[8] S. Brunak, J. Engelbrecht, and S. Knudsen. Prediction of human mRNA donor and acceptor sites from the DNA sequence. *JMB*, 220:49–65, 1991.

[9] C. Burge and S. Karlin. Predictions of complete gene structures in human genomic DNA. *JMB*, 268:78–94, 1997.

[10] M. Burset and R. Guigo. Evaluation of gene structure prediction programs. *Genomics*, 34(3):353–367, 1996. Data set and evaluation results can be found at http://www.imim.es/GeneIdentification/Evaluation/Index.html.

[11] J.-M. Claverie. Sequence "signals": Artifact or reality? *Computers and Chemistry*, 16(2):89–91, 1992.

[12] J.-M. Claverie. Some useful statistical properties of position-weight matrices. *Computers and Chemistry*, 18(3):287–294, 1994.

[13] J.-M. Claverie. Computational methods for the identification of genes in vertebrate genomic sequences. *Human Molecular Genetics*, 6(10):1735–1744, 1997.

[14] S. Cole et al. Deciphering the biology of mycobecterium tuberculosis from the complete genome sequence. *Nature*, 393(6685):537–544, 1998.

[15] M. Craven and J. Shavlik. Learning to predict reading frames in *e. coli* DNA sequences. In *Proceedings of the Hawaii International Conference on System Sciences*, pages 773–782, Los Alamitos, CA, 1993. IEEE Computer Society Press.

[16] S. Dong and D. B. Searls. Gene structure prediction by linguistic methods. *Genomics*, 162:705–708, 1994.

[17] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998.

[18] L. Duret, D. Mouchiroud, and C. Gautier. Statistical analysis of vertebrate sequences reveals that long genes are scarce in CG-rich isochores. *Journal of Molecular Evolution*, 40:308–317, 1995.

[19] R. Farber, A. Lapedes, and K. Sirotkin. Determination of eukaryotic protein coding regions using neural networks and information theory. *JMB*, 226:471–479, 1992.

[20] J. Fickett. Finding genes by computer - the state of the art. *Trends in Genetics*, 12(8):316–320, 1996.

[21] J. Fickett. The gene identification problem — an overview for developers. *Computers and Chemistry*, 20(1):103–118, 1996.

[22] J. Fickett and A. G. Hatzigeorgiou. Eukaryotic promoter recognition. *Genome Research*, 7(9):861–878, 1997.

[23] J. W. Fickett and C.-S. Tung. Assessment of protein coding measures. *Nucl. Acids Res.*, 20:6441–6450, 1992.

[24] C. Fields and C. Soderlund. A practical tool for automating DNA sequence analysis. *Comp. Appl. Biosci.*, 6:263–270, 1990.

[25] M. Gardiner-Garden and M. Frommer. CpG islands in vertebrate genomes. *JMB*, 196:261–282, 1987.

[26] M. S. Gelfand. Computer prediction of exon-intron structure of mammalian pre-mrnas. *NAR*, 18:5865–5869, 1990.

[27] M. S. Gelfand. Prediction of function in DNA sequence analysis. *Jour. Comp. Biol.*, 2(1):87–115, 1995.

[28] M. S. Gelfand, A. A. Mironov, and P. A. Pevzner. Gene recognition via spliced sequence alignment. *PNAS*, 93(17):9061–9066, 1996.

[29] M. S. Gelfand, L. I. Podolsky, T. V. Astakhova, and M. A. Roytberg. Recognition of genes in human DNA sequences. *Jour. Comp. Biol.*, 3(2):223–234, 1996.

[30] M. S. Gelfand and M. A. Roytberg. Prediction of the exon-intron structure by a dynamic programming approach. *BioSystems*, 30:173–182, 1993.

[31] R. Guigo. Computational gene identification: an open problem. *Computers and Chemistry*, 21(4):215–222, 1997.

[32] A. Hatzigeorgiou and M. Reczko. Recognition of coding regions and reading frames in DNA. In *Gene-Finding and Gene Structure Prediction Workshop*, 1995.

[33] J. Henderson, S. Salzberg, and K. Fasman. Finding genes in human DNA with a hidden Markov model. *Journal of Computational Biology*, 4(2):119–126, 1997.

[34] X. Huang, M. Adams, H. Zhou, and A. Kerlavage. A tool for analyzing and annotating genomic sequences. *Genomics*, 46:37–45, 1997.

[35] J. Jurka, P. Klonowski, V. Dagman, and P. Pelton. Censor - a program for identification and elimination of repetitive elements from DNA sequences. *Computers and Chemistry*, 20(1):119–122, 1996.

[36] J. Jurka, J. Walichiewicz, and A. J. Milosavljevic. Prototypic sequences for human repetitive DNA. *J. Mol. Evol.*, 35:286–291, 1992.

[37] T. Klinger and D. Brutlag. Detection of correlations in tRNA sequences with structural implications. In L. Hunter, D. Searls, and J. Shavlik, editors, *ISMB-93*, Menlo Park, 1993. AAAI Press.

[38] N. Kolchanov et al. GenExpress: a computer system for description, analysis, and recognition of regulatory sequences in eukaryotic genome. In *Proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology*, pages 95–104, 1998.

[39] M. Kozak. Interpreting cDNA sequences: some insights from studies on translation. *Mammalian Genome*, 7:563–574, 1996.

[40] A. Krogh. Two methods for improving performace of a HMM and their application for gene finding. In *Proceedings, 5th International Conference on Intelligent Systems for Molecular Biology*, pages 179–186, 1997.

[41] A. Krogh. Gene finding: putting the parts together. In M. J. Bishop, editor, *Guide to Human Genome Computing*, chapter 11, pages 261–274. Academic Press, 2nd edition, 1998.

[42] A. Krogh, I. S. Mian, and D. Haussler. A Hidden Markov Model that finds genes in *E. coli* DNA. *NAR*, 22:4768–4778, 1994.

[43] D. Kulp and D. Haussler. Embedding HMMs: A Method for Recognizing Protein Homologs in DNA, 1997. http://www.ornl.gov/hgmis/publicat/97santa/infortoc.html.

[44] D. Kulp, D. Haussler, M. Reese, and F. Eeckman. A generalized hidden Markov model for the recognition of human genes in DNA. In *ISMB-96*, pages 134–142, St. Louis, June 1996. AAAI Press. http://www.cse.ucsc.edu/~dkulp/cgi-bin/genie.

[45] D. Kulp, D. Haussler, M. G. Reese, and F. H. Eeckman. Integrating database homology in a probabilistic gene structure model. In R. B. Altman, A. K. Dunker, L. Hunter, and T. E. Klein, editors, *Proceedings of the Pacific Symposium on Biocomputing*, pages 232–244. World Scientific, New York, 1997.

[46] A. Lapedes, C. Barnes, C. Burks, R. Farber, and K. Sirotkin. Application of neural networks and other machine learning algorithms to DNA sequence analysis. In G. Bell and T. Marr, editors, *Computers and DNA, SFI Studies in the Sciences of Complexity*, volume VII, pages 157–182. Addison-Wesley, 1989.

[47] F. Larsen, R. Gundersen, R. Lopez, and H. Prydz. CpG islands as gene markers in the human genome. *Genomics*, 13:1095–1107, 1992.

[48] A. V. Lukashin and M. Borodovsky. Genemark.hmm: new solutions for gene finding. *Nucleic Acids Research*, 26(4):1107–1115, 1998.

[49] S. Matis, Y. Xu, M. B. Shah, D. Buley, X. Guan, J. R. Einstein, R. J. Mural, and E. C. Uberbacher. Detection of RNA Polymerase II Promoters and Polyadenylation Sites in Human DNA Sequence. *Computers and Chemistry*, 20:135–140, 1995.

[50] L. Milanesi and I. Rogozin. Prediction of human gene structure. In M. J. Bishop, editor, *Guide to Human Genome Computing*. Academic Press, 2nd edition, 1998.

[51] A. Milosavljević and J. Jurka. Discovering simple DNA sequences by the algorithmic similarity method. *CABIOS*, 9(4):407–411, 1993.

[52] R. Nagel, A. Lancaster, and A. Zahler. Specific binding of an exonic splicing enhancer by the pre-mrna splicing factor srp55. *RNA*, 4:11–23, 1998.

[53] C. M. O'Neill. Training back-propagation neural networks to define and detect DNA-binding sites. *Nucl. Acids Res.*, 19:313–318, 1991.

[54] C. M. O'Neill. Escherichia coli promoters: neural networks develop distinct descriptions in learning to search for promoters of different spacing classes. *Nucl. Acids Res.*, 20:3471–3477, 1992.

[55] G. R. and J. Fickett. Distinctive sequence features in protein coding genic non-coding, and intergenic human DNA. *JMB*, 253(1):51–60, October 13, 1995.

[56] M. Reese and F. Eeckman. Novel neural network prediction systems for human promoters and splice sites. In *Gene-Finding and Gene Structure Prediction Workshop*, 1995.

[57] M. G. Reese, F. H. Eeckman, D. Kulp, and D. Haussler. Improved splice site detection in genie. *Jour. Comp. Biol.*, 4:311–323, 1997.

[58] S. L. Salzberg. Locating protein coding regions in human DNA using a decision tree algorithm. *Jour. Comp. Biol.*, 2:473–485, 1995.

[59] S. L. Salzberg, A. L. Delcher, , S. Kasif, and O. White. Microbial gene identification using interpolated Markov models. *Nucleic Acids Research*, 26(2):544–548, 1998.

[60] D. B. Searls. The computational linguistics of biological sequences. In L. Hunter, editor, *Artificial Intelligence and Molecular Biology*, chapter 2, pages 47–120. AAAI Press, 1993.

[61] T. F. Smith and M. S. Waterman. Comparison of bio-sequences. *Adv. Appl. Math*, 2:482–489, 1981.

[62] E. Snyder and G. Stormo. Indentification of protein coding regions in genomic DNA. *JMB*, 248:1–18, 1995.

[63] V. Solovyev, S. A., and C. Lawrence. Predicting internal exons by oligonucleotide composition and discriminant analysis of splicable open reading frames. *Nucl. Acids Res.*, 22:5156–5163, 1994.

[64] R. Staden. Computer methods to locate signals in nucleic acid sequences. *NAR*, 12:505–519, 1984.

[65] R. Staden. Finding protein coding regions in genomic sequences. *Methods in Enzymology*, 183:163–180, 1990.

[66] G. Stormo. Consensus patterns in DNA. *Methods in Enzymology*, 183:211–220, 1990.

[67] G. D. Stormo. Computer methods for analyzing sequence recognition of nucleic acids. *Annu. Rev. Biophys. Biophys. Chem.*, 17:241–263, 1988.

[68] G. D. Stormo and D. Haussler. Optimally parsing a sequence into different classes based on multiple types of information. In *ISMB-94*, Menlo Park, CA, Aug. 1994. AAAI/MIT Press.

[69] J. Sulston et al. The C. elegans genome sequencing project: A beginning. *Nature*, 356:37–41, 1992.

[70] A. Thomas and M. Skolnick. A probabilistic model for detecting coding regions in DNA sequences. *IMA Journal of Mathematics Applied in Medicine and Biology*, 11:149–160, 1994.

[71] E. C. Uberbacher and R. J. Mural. Locating protein-coding regions in human DNA sequences by a multiple sensor-neural network approach. *PNAS*, 88:11261–11265, 1991.

[72] T. Wu. A segment-based dynamic programing algorithm for predicting genes. *Jour. Comp. Biol.*, 3:375–394, 1996.

[73] Y. Xu, J. R. Einstein, M. Shah, and E. C. Uberbacher. An improved system for exon recognition and gene modeling in human DNA sequences. In *ISMB-94*, pages 376–383, Menlo Park, CA, 1994. AAAI/MIT Press.

[74] Y. Xu, R. Mural, and E. Uberbacher. Inferring gene structures in genomic sequences using pattern recognition and expressed sequence tags. In *Proceedings, 5th International Conference on Intelligent Systems for Molecular Biology*, pages 344–353, 1997.

[75] Y. Xu and E. C. Uberbacher. Automated gene identification in large-scale genomic sequences. *Journal of Computational Biology*, 4(3):325–338, 1997.

[76] M. Zhang and T. Marr. A weighted array method for splicing and signal analysis. *CABIOS*, 9:499–509, 1993.

[77] M. Q. Zhang. Identification of protein coding regions in the human genome based on quadratic discriminantanalysis. *PNAS*, 94:559–564, 1998.

[78] M. Q. Zhang. Statistical features of human exons and their flanking regions. *Human Molecular Genetics*, 7(5):919–932, 1998.