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 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 Guigó [10], Claverie [3], Krogh [14] and others.

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 [8]. 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* [26].

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. 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 [27]. 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. More sophisticated types of signal sensors, such as neural nets, are extensively used, and are reviewed in [8].

3 Content Sensors

The most important and most studied content sensor is the sensor that predicts coding regions, reviewed in [7]. 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 use statistical models of the nucleotide frequencies and dependencies present in codon structure. The most commonly used statistical models are known as *Markov models*, popularized for genefinding in GeneMark [1]. Neural nets are used to combine several coding measures along with signal sensors for the flanking splice sites in Grail's exon detector [29]. 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, and sensors for repetitive DNA, such as human ALU sequences. 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, and there are dependencies between signals and contents that they cannot capture [4], such as the possible correlation between splice site strength and exon size [32]. 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 was the first major proponent of describing gene structure in linguistic terms using a formal grammar [22], and his GenLang genefinding program, based on this idea, was one of the earliest integrated genefinders. This program, like nearly all integrated genefinders to date, used 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 [14] and a more detailed exposition in [6].

The key to success in dynamic programming methods is developing the right score function to optimize. 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). This model 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. 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. Genefinding HMMs can be viewed as stochastic versions of the gene structure grammars used by Searls.

Early genefinding HMMs were EcoParse (for *E. coli* [15], also recently used in the annotation of the *M. Tuberculosis* genome [5]) and Xpound (for human) [28]. More recent programs are GeneMark-HMM (for bacterial genomes) [18] Veil [11] and HMMgene (for human) [14]. A somewhat more general class of probabilistic models, called generalized HMMs (GHMMs) or (hidden) semi-Markov models, have their roots in GeneParser [24], and were more fully developed in Genie [17, 20] and then GenScan [2].

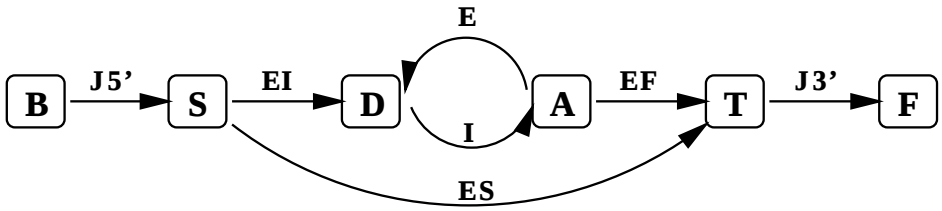

Figure 1: A simplified diagram representing the liguistic rules for what might follow what when parsing a sequence consisting of a multiple exon gene. The arcs represent contents and the nodes represent signals. The contents are J5' : 5' UTR, EI : Initial Exon, E : Exon, I : Intron, E : Internal Exon, 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 [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 previously known genes and their corresponding proteins, or auxiliary information such as expressed sequence tag (EST) matches. Protein database homology and EST matches have long been used as *post hoc* methods to validate gene predictions, but newer methods integrate this information directly into the genefinding algorithm itself. Some 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 [16, 30, 10].

The homology approach has been taken to its extreme limit in a genefinding program developed by Gelfand, Mironov, and Pevzner[9]. 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[23] 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 genefinding method.

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. 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. Presumably it 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 [13]. 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 [10]. These results are not competitive with the older genefinders that use protein homology, nor with the newer HMM-based 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 [19]), 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. It may be anticipated that this task will occupy genefinding researchers for some years to come.

Glossary

content: an extended or variable length region of genomic DNA with a particular function, such as an exon

dynamic programming: method to evaluate all possible candidate gene structures (or hidden state sequences) in an efficient manner.

expressed sequence tag: a cDNA (complementary DNA) sequence made from an mRNA transcript.

hidden Markov model: extension of a Markov model that employs a hidden state sequence, used for identifying complex patterns (such as genes) within sequences.

Markov model: a statistical model for sequences in which the probability of each letter depends on what letters precede it.

neural net: a statistical pattern recognition method; type of nonlinear regression

parsing: finding the best candidate gene structure; metaphor from linguistics

protein database homology: relationship between the protein produced by the gene you are analyzing and a similar protein taken from a database of protein sequences.

score: a function used to evaluate different candidate gene predictions; dynamic programming finds the candidate gene structure with the best score

signal: a local functional site in genomic DNA, such as a splice site.

weight matrix: a statistical model in which each position in a sequence is modeled with a separate, independent probability distribution.

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 [17, 20]: `http://www.cse.ucsc.edu/~dkulp/cgi-bin/genie`
GenScan [2]: http://CCR-081.mit.edu/GENSCAN.html
HMMgene [14]: http://www.cbs.dtu.dk/services/HMMgene/
GeneMark-HMM [18]: http://genemark.biology.gatech.edu/GeneMark/hmmchoice.html

Veil [11]: http://www.cs.jhu.edu/labs/compbio/veil.html

Some further genefinders:
AAT [12]: http://genome.cs.mtu.edu/aat.html
FGENEH [25]: `http://dot.imgen.bcm.tmc.edu:9331/gene-finder/gf.html`
GENEID [10]: `http://www.imim.es/GeneIdentification/Geneid/geneid_input.html`
Genlang [22]: `http://cbil.humgen.upenn.edu/~sdong/genlang_home.html`
GeneParser [24]: `http://beagle.colorado.edu/~eesnyder/GeneParser.html`
Glimmer [21]: `http://www.cs.jhu.edu/labs/compbio/glimmer.html`
Grail [29]: http://compbio.ornl.gov/
MZEF [31]: http://www.cshl.org/genefinder
Procrustes [9]: `http://www-hto.usc.edu/software/procrustes/`

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

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.

References

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

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

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

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

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

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

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

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

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

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

[11] 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.

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

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

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

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

[16] 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.

[17] D. Kulp, D. Haussler, M.G. 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.

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

[19] 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.

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

[21] 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.

[22] David B. Searls. The linguistics of DNA. *American Scientist*, 80:579–591, November–December 1992.

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

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

[25] V. Solovyev, Salamov 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.

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

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

[28] 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.

[29] 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.

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

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

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