Computational genefinding

A major challenge in the analysis of genomic DNA sequence is to find the functional sites that encode elements responsible for gene structure, regulation and transcription. A variety of computational tools can help to isolate the ‘signal’ from the ‘noise’.

Computational methodology for finding genes and other functional sites in genomic DNA has evolved significantly over the past 20 years (for reviews, see Refs 1–3). The genomic elements that researchers seek include 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 sites4. Local sites such as these are called signals, and methods for detecting them can be called signal sensors. In contrast, extended and variable-length regions, such as exons and introns, are called contents and are recognized by methods that can be called content sensors5.

Conclusion

One should neither have excessive faith in the results of a BLAST run nor blithely disregard them. The BLAST programs are well-tested and reliable indicators of sequence similarity, and their underlying principles are straightforward. Complexities added by the fast algorithms typically need not be carefully considered, because the program and its parameters have been optimized for hundreds of thousands of daily runs. If one is careful about posing the database search experiment and interprets the results with care, sequence comparison methods can be trusted to provide an incomparable wealth of biological information rapidly and easily.

References

1 Altshul, S.F. et al. (1997) Nucleic Acids Res. 25, 3389–3402
4 States, D.J. et al. (1991) Methods 3, 66–70
which each position in the pattern allows a match to any residue, but different costs are associated with matching each residue in each position. 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. Above a given score threshold, the candidate site is predicted to be "true." Each sensor has 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 (reviewed in Ref. 4).

Content sensors

The most important and most studied content sensor is that which predicts coding regions (Ref. 7). In prokaryotes, it is still common to locate genes by simply looking for long open reading frames; this is certainly not adequate for higher eukaryotes. To discriminate coding from noncoding regions in eukaryotes, exon content sensors use statistical models of the nucleotide frequencies and dependencies present in coding structure. The most commonly used statistical models are known as Markov models, which have become popular for genefinding in the computer program GeneMark (2). Neural nets are used to combine several coding measures together with signal sensors for the flanking splice sites in Graal’s exon detector (2). Other content sensors include those for CpG islands, regions that often occur near the beginning of genes, where the dinucleotide CG is more frequent than 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.

Integrated genefinding methods

Signal and content sensors alone cannot solve the genefinding problem: the statistical signals that they are trying to recognize are too weak, and there are dependencies between signals and contents that they cannot capture, such as the possible correlation between splice site strength and exon size (2). During the past five years, several systems that combine signal and content sensors have been developed in an attempt 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, in which the process of breaking down a sequence of DNA into genes, each of which is a series of exons and introns, is likened to the process of parsing a sentence by breaking it down into its constituent grammatical parts. Searls was first to describe gene structure in linguistic terms using a formal grammar (3) and his GenLang gennfiding program, which is based on this idea, was one of the earliest integrated genefinders. As with most integrated genefinders date, GenLang uses dynamic programming to combine candidate exons and other scored regions and sites into a complete gene prediction with a maximal total score. A lucid tutorial on this topic can be found in Ref. 3, with a more detailed explanation in Ref. 13.

The key to success in dynamic programming methods is to develop the right score function to optimize: A fruitful approach in this area 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), and "linguistic" information on what functional features are likely to follow other features (see Fig. 1). This model includes a latent (or "hidden") variable associated with each nucleotide that represents the functional role or position of that nucleotide; for example, 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 (HMMs). Genefinding HMMs can be viewed as stochastic versions of the gene structure grammars used by Searls (3).

Early genefinding HMMs included EcoParse (for E. coli) (4), although recently also used in the annotation of the M. tuberculosis genome (5) and Xpound (for the human genome) (6). More recent programs include GeneMark-HMM (for bacterial genomes) (7), Veil (8) and HMMgene (for the human genome) (9). A slightly more general class of probabilistic models, called generalized HMMs (GHMMs) or (hidden) semi-Markov models, has its roots in GenParse (4) and is more hilly developed in Genie (22) and, subsequently, GenScan (23).

The genefinders above predict gene structure based only on general features of genes, rather than using explicit comparisons to 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 searching a protein database (24).
The homology approach has been taken to its extreme limit in a gene-finding program developed by Gelfand et al. 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 algorithm, is used to derive a putative gene structure by aligning the DNA to the homolog. The major disadvantage of 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.

Computational gene finding as a tool
It is important to distinguish two different goals in gene-finding research. The first 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 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 single computational gene-finding 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 an absence of early in-frame stop codons might be involved in biological start-site selection. One would think that these restrictions would cripple computational gene-finding methods; however, Guigo\(^2\) has shown that using simple weight matrices to find the best combination of splicing site signals, translation start and stop signals, together with the standard syntactic constraints on gene structure (frame consistency, no in-frame stop codons, minimum intron size), gives results on his benchmark dataset that are comparable to those obtained by most of the gene-finders he and Burset tested in 1995 (Ref. 1). These results are not competitive with the older gene-finders that use protein homology, nor with the newer HMM-based methods that use exon coding potential but not homology; but they do indicate a surprising potential for purist gene-finding models. More detailed models of the splicing process, the selection of translation start sites and the process of polyadenylation could significantly improve such purist models and 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 gene-finders 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 together with protein homology and powerful statistical models.

There are other key issues that will affect future research in both of the above gene-finding paradigms. One is the issue of alternative splicing. No currently available gene-finders handle alternative splicing in an effective manner. Intimately linked with this issue is that of gene regulation. The abundant regulatory signals that flank genes and appear in introns (and sometimes in exons), 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 elucidated. This implies that future genefinders will also need to take explicitly into account experimental data relating to differential expression, as well as the other types of data discussed here. It is anticipated that this task will occupy gene-finding researchers for some years to come.

Acknowledgements
I gratefully acknowledge the support of DOE grant DE-FG03-95ER62111 and thank K. Guigo, D. Kulp and M. Reese for helpful suggestions.

References

Multiple-alignment & sequence searches

Comparisons of multiple sequences can reveal gene functions that are not clear from simple sequence homologies. The important parameters in multiple alignment and multiple-sequence-based searches, using an example from Caenorhabditis elegans are described.

It used to be that most new sequences were novel, with no informative similarity to anything in the sequence database. As a result of genome sequencing projects, the situation is now slightly improved. New sequences are often found to be similar to several uncharacterized sequences, defining whole families of novel genes with no informative BLAST or FASTA similarities. However, given a sequence family, powerful alternative similarity search methods can be applied. Software packages are available that can take a multiple sequence alignment and build a profile of it. Profiles incorporate position-specific scoring information that is derived from the frequency with which a given residue is seen in an aligned column. Because sequence families preferentially conserve certain critical residues and motifs, this information can sometimes allow more sensitive database searches to be carried out.

Most new profile software is based on statistical models called hidden Markov models (HMMs). Here, a practical demonstration is given of a multiple-alignment-based

Sean R. Eddy
Dept of Genetics, Washington University School of Medicine, 4546 Scott Ave, St. Louis, MO 63110, USA
eddy@genetics.wustl.edu
http://www.genetics.wustl.edu/eddy/