Finding genes by computer: the state of the art

JAMES W. FICKETT

Discovering new genes, and their functions, can be aided not only by special purpose gene (and coding region) finding software, but also by searches in key databases, and by programs for finding particular sites relevant to gene expression, such as promoters and splice sites. No one software package includes all the necessary tools. I describe here the main kinds of tools; their working principles, strengths and limitations; and how combined evidence from multiple tools can aid in optimum gene identification.

Efficiency is the most frequently quoted reason for availing oneself of computational tools for elucidating the structure of, and assigning a tentative function to, genes: if the DNA sequence is available, almost any kind of computational analysis is cheaper and faster than almost any kind of experimental analysis. Experimental methods for locating genes have grown greatly in power (reviewed in Ref. 1). Yet, in many areas, experimental and computational methods still provide complementary information (see Ref. 2 for a concise statement on the limitations of experimental methods for isolating, expressing and determining the functions of genes). Computational gene identification has recently played prominent roles in, for example, identifying candidate 'disease genes'[3], compiling inventories of possible genes in large-scale genomic sequencing projects[4], and helping to assign tentative functions as the first ever organism-wide gene inventories progress[5,6,35].

One hopes for the day when a sequence, together with relevant results from experiment (e.g. mapped transcript locations), can be submitted for all relevant analysis through a single interface. But today, to make use of the best computational techniques, it is necessary to submit one's sequence to the analysis of several different software packages. My purpose here is to help make this process as efficient as possible by providing a concise guide to current gene identification methods. (For related and more detailed reviews see Refs 7–10.)

I will first describe a conceptual framework to help make sense of the plethora of tools. Next I review the main types of computer tools and, for each, its underlying logic, strengths and limitations. Some comments on practical use are also included, but full details and examples are given elsewhere[8]. Specific techniques, but only a few particular software tools, are mentioned in the text. Some of the most commonly used network-available tools are given in Table 1.

Framework

When an overall gene-finding protocol is carried out either by one integrated program or by a person using several specialized programs, the basic information flow is as follows.

(1) Evidence (both positive and negative) is gathered from several sources:

- Sequence similarity to other features (e.g. repeats) not likely to overlap certain parts of protein-coding genes (e.g. Alu repeats found by BLASTX; Table 1)
- Sequence similarity to other genes (e.g. found by BLASTX, which translates the sequence in all six reading frames, and compares the result against an amino acid sequence database; see BLAST in Table 1)
- Statistical regularity evincing apparent 'codon bias' over a region (this is the foundation stone for all the gene identification programs listed in Table 1, including the widely used GRAIL program)
- Matches to template patterns for functional sites on the DNA (such analysis can be based on very simple patterns, e.g. the well-known consensus sequences for the TATA box and splice junctions, or much more complex reasoning, as in the PromoterScan and NetGene algorithms listed in Table 1)

(2) All the information so gathered is integrated to make as coherent a picture as possible of the overall situation. The rules applied at the integration stage are basically a formalization of common sense. For example, an exon boundary found by a codon bias analysis can be adjusted slightly to take advantage of a better splice site; and codon bias is to be taken more seriously if there is also similarity to a known protein sequence. Using such integration, the programs marked 'integrated gene identification' in Table 1 attempt to predict an overall gene structure with specific exons and introns, rather than just a general plot of coding potential.

For any particular enquiry, only a few of the many gene identification-related programs are relevant. In setting up a protocol, some of the main points to be considered are: (1) for eukaryotic sequences, screening for repeats should precede all other analyses; (2) most programs are organism specific; (3) many programs are specific for either genomic or cDNA data; and (4) the length of the sequence is a major factor. For example, single reads from shotgun sequencing usually cannot be analyzed by the more sophisticated programs expecting to find whole genes in the sequence.

Masking repetitive DNA

It is best to locate and remove interspersed repeats from eukaryotic sequences as the first step in any gene identification analysis[7]. Although such repeats might well overlap regions transcribed by RNA polymerase II, they rarely overlap promoters or the coding portions of exons, so that their location can provide important negative information on the location of gene features. Also, repeats can often confuse other analyses, especially database searches. Several collections of repeats from particular organisms are available, as well as sophisticated programs to match these repeat libraries to particular occurrences in a query sequence ('Repeat analysis' Table 1).

Database searches

Searching for a known homolog is, perhaps, the oldest and most widely understood means of identifying new genes[11,12]. Such searches depend only on evolutionary relatedness, and so are widely applicable.

PII: S0168-9525(96)10038-X

TABLE 1. Internet tools for gene discovery in DNA sequence data[a]

Category	Service	Organism(s)	Address
Repeat analysis	Pythia; give a list of repeats in sequence	Human	pythia@anl.gov
	Repbase; repeat collections	Human and several other collections	ftp://ncbi.nlm.nih.gov; repository/rebase/REF
	BLASTX; tools to mask repeat occurrences	Any	ftp://ncbi.nlm.nih.gov; pub/jmc
Database search	BLAST; search sequence databases	Any	blast@ncbi.nlm.nih.gov
	FASTA; search sequence databases	Any	fasta@ebi.ac.uk
	BLOCKS; search for functional motifs	Any	blocks@howard.fhcrc.org
	ProfileScan	Any	http://ulrec3.unil.ch/software/PFSCAN_form.html
	MotifFinder	Any	motif@genome.ad.jp
Gene identification	FGENEH; integrated gene identification	Human	service@theory.bchs.uh.edu
	GeneID; integrated gene identification	Vertebrate	geneid@bir.cedb.uwf.edu
	GeneMark; coding region identification	Many individual species	genemark@ford.gatech.edu
	GeneParser; integrated gene identification	Human	http://beagle.colorado.edu/~eesnyder/GeneParser.html
	GenLang; integrated gene identification	Dicots, *Drosophila*, vertebrates	genlang@cbil.humgen.upenn.edu
	GRAIL; integrated gene identification	Human	grail@ornl.gov (also graphical interface)
	EcoParse; integrated gene identification	*Escherichia coli*	ecoparse@cse.ucsc.edu
'Signal' recognition	PromoterScan	Eukaryotes	Contact Dan Prestridge at danp@biosci.cbs.umn.edu for FTP
	NetGene	Human	netgene@virus.fki.dth.dk

[a] A few example tools embodying techniques discussed in the text. For each, the category of service is described, the name and a brief description of the service is given, the organism or organisms that can be analyzed are listed, and an address for email, FTP or the World Wide Web (WWW), is listed. Generally the WWW sites are self-explanatory, and documentation for the email services can be obtained by sending a message with the word 'help' to the address given. The category 'gene identification' includes coding region identification as well. Most of the integrated gene identification services can also show the individual gene features (such as coding regions and splice sites) that were predicted in the course of deducing overall gene structure.

A few of the integrated gene-finding services are beginning to include database searches as part of the analysis. However, in most cases the database search step still needs to be done separately by the user. Translating the sequence in all six possible reading frames and using the result as a query against databases of amino acid sequences (using, for example, the well known BLASTX program) and functional motifs (Fig. 1) is usually the single most informative option.

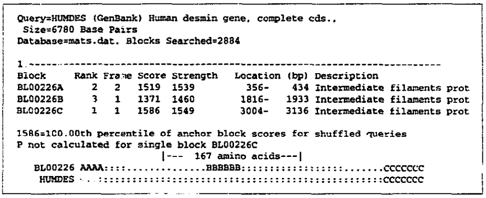
A major advantage of finding a homologous product is, of course, that some of the biology of the protein might be already elucidated. The main limitation of database searching is that, currently, only about half of the new proteins being discovered have a homolog already in the databases and this fraction seems to be increasing rather slowly. Green *et al.*[14] found that: (1) most ancient conserved regions (ACRs, which are roughly defined as regions of protein sequences showing highly significant homologies across phyla) of the protein universe are already known and can be found in current databases; (2) approximately 20–50% of newly found genes contain an ancient conserved region that is represented in the databases; and (3) rarely expressed genes are less likely to contain an ancient conserved region than moderately or highly expressed ones.

A direct search of nucleotide sequence databases will also be valuable, for example, to find conserved regulatory regions (little is known about how often these are detectable) and cDNA fragments[15,36] (which can now help detect a majority of genes, although usually giving little information on gene structure or function).

Codon bias detection

Most computational gene identification relies heavily on recognizing the somewhat diffuse regularities in protein coding regions due to bias in codon usage. Simply tabulating codon frequencies is one example of a coding measure, that is, a rule for calculating a number, or table of numbers, meant to summarize such regularities. Many coding measures have been suggested[16]. Probably the most informative are dicodon counts (i.e. frequency counts for the occurrence of successive codon pairs), some direct measure of periodicity (in this context, periodicity means the tendency of multiple occurrences of the same nucleotide to be found at distances of 3, 6, 9... bp), a measure of homogeneity versus complexity (such as counting long homopolymer runs), and open reading frame occurrence.

Many coding region detection programs are primarily the result of combining the numbers from one or more coding measures (using, for example, probability theory, discriminant analysis techniques from multivariate statistics, or neural net methods from the field of artificial intelligence) to form a single number called a discriminant. Such a combination forms, for example, the primary basis for the well-known GRAIL program[17]. Typically, then, the discriminant is calculated for successive subsequences of fixed length, and the result plotted (Fig. 2).



```
Query=HUMDES (GenBank) Human desmin gene, complete cds.,
 Size=6780 Base Pairs
Database=mats.dat. Blocks Searched=2884

1.-----------------------------------------------------------------------
Block       Rank Frame Score Strength   Location (bp) Description
BL00226A       2   2   1519  1539          356-    434 Intermediate filaments prot
BL00226B       3   1   1371  1460         1816-   1933 Intermediate filaments prot
BL00226C       1   1   1586  1549         3004-   3136 Intermediate filaments prot

1586=100.00th percentile of anchor block scores for shuffled queries
P not calculated for single block BL00226C
                               |---  167 amino acids---|
   BL00226 AAAA:::...............BBBBBB::::::::::::::::::::::.......CCCCCCC
    HUMDES ...::::::::::::::::::::::::::::::::::::::::::::::::::::::CCCCCCC
```

FIGURE 1. Sample output from BLOCKS (Ref. 13), an Internet-accessible motif-searching service. The desmin gene, in FASTA format, was sent to the email address blocks@howard.fhcrc.org. The service accepts either amino acid or nucleotide sequences. The latter case is automatically detected and the sequence is translated in all six frames before being compared with a database of protein sequence motifs. In this case, the top three ranking hits are motif blocks characteristic of the intermediate filament proteins. The third hit is shown. The introns in the genomic sequence prevent the software from recognizing that these three motif blocks are correctly spaced in the actual protein – the diagram at the bottom of the output shows the correct spacing of the three blocks, in the top line, and the occurrence of the third block, without the other two, in the input sequence. However, the three separate matches make it clear that there is an important sequence similarity. The strength of the similarity (higher than any chance similarity found in queries based on shuffled versions of the input sequence, as shown in the 4th and 5th lines from the bottom) make it very likely that genuine homology exists.

Something in the order of 100 bp is required to gain significant information from a coding measure discriminant. More concretely, the following benchmark was carried out[16]. (1) GenBank was divided into successive 108 bp windows; (2) only those fully coding or fully noncoding were saved; (3) half the windows were used to set the parameters in a linear discriminant combination of four measures as described above; and (4) the other half of the windows were used to measure the accuracy of prediction of the resulting discriminant. A correct prediction rate of 88% was found. Thus, coding measures give a rather low resolution picture of coding-region boundaries. However, a major advantage of coding measures is that they can reasonably be applied to fragmentary sequences, for example, single reads of a few hundred base pairs from shotgun sequencing projects. Many coding measures are quite organism specific, and one must look closely to see in what subset of the taxonomic universe a particular service was developed and tested.

Detecting functional sites in the DNA

The measurement of codon bias probably has almost nothing in common with the way a cell recognizes and expresses genes. It will be more enlightening (and probably give better accuracy) when we are able to recognize those locations, such as transcription-factor-binding sites and exon–intron junctions, where the gene expression machinery interacts with the nucleic acid.

One way to summarize the essential information content of these locations (typically called 'signals' by those developing gene identification algorithms) is to give the consensus sequence, consisting of the most common base at each position of an alignment of specific binding sites. Consensus sequences are very useful as a mnemonic device, but are typically not very reliable for discriminating true sites from pseudosites, perhaps because they contain no information on how often other bases can occur. Many algorithms using more sophisticated techniques can give better discrimination. One technique with a basis in physical chemistry is that of the position weight matrix (PWM). A score is assigned to each possible nucleotide at each possible position of the signal. For any particular sequence, considered as a possible occurrence of the signal, the appropriate scores are summed to give a score to a potential site. Under some circumstances this score can approximate the energy of binding for a control (ribonucleo-) protein (reviewed in Refs 19, 20).

There have been a few studies (e.g. Ref. 21) showing that a PWM works well to evaluate individual binding sites of a particular kind. Unfortunately, however, using PWMs in isolation for recognizing complex elements of general eukaryotic gene expression, for example, splice sites and promoter sequences, has had relatively limited success. Major reasons probably include context-specific expression mechanisms and cooperativity among multiple binding molecules. It is rare in eukaryotes, for example, for large numbers of genes to have precisely the same complement of proteins involved in the initiation of transcription[22,23].

In most cases no specialized software is needed to apply current knowledge in recognizing signals. For example the 'Kozak rules'[24] can easily be applied by hand to make an educated guess at the translation initiation codon of a known transcript. However, in a few cases a more sophisticated algorithm has been written. For example, the PromoterScan algorithm[25] not only applies a PWM for the TATAA box[26], but also takes into account occurrences of the consensus-sequence binding sites for a large number of general (e.g. Sp1) and tissue-specific (e.g. MYOD) transcription factors; and NetGene (Ref. 27) uses a neural net to combine information on the splice site *per se* with an estimate of coding potential on either side. Although there is still significant room for improvement in the accuracy of such tools (e.g. PromoterScan reportedly finds 70% of known primate promoters, with a false-positive rate of 1 in 5600 bp), they incorporate more information than most of the integrated algorithms mentioned below, and are worth applying separately.

Integrated gene parsing

The first generation of computational aids for gene identification treated mainly the recognition of isolated aspects of genes, for example, splice sites alone, or the regularities of coding regions without reference to signals. But if, for example, a splice site interrupts a coding region, it will help in detection to look for coding region on one side and noncoding on the other. It has been shown that taking into account the overall consistency of putative features significantly increases prediction accuracy. For example, 60% of exons under 50bp missed by the original GRAIL email program can be detected when a simple logical analysis of splicing and frame is added[28].

Integrated gene-finding programs begin by searching for signals and performing a coding region analysis (and sometimes doing homology searches as well), and then, by optimizing some scoring function, attempt to define exons and give one or more tentative gene structures that seem most consistent with all the data at hand (Fig. 3). Increased accuracy and user convenience are the primary forces behind the development of these programs.

Several such integrated algorithms are now freely available (Table 1) and, at least in some circumstances, can give a good idea of gene structure. The main limitations (in this first generation of a new technology) are these: (1) integrated algorithms are currently available for only a few organisms; (2) these algorithms currently assume that there is in the input sequence exactly one entire gene (when the input includes multiple genes or partial genes, the predicted exons can still make sense, but the overall predicted gene structure probably will not); (3) for reasons that are not altogether clear, accuracy can be considerably lower than originally thought, particularly on genes recently discovered[30]; (4) most integrated algorithms are apparently quite sensitive to sequencing errors[30]; and (5) such facets of gene syntax as alternative splicing, overlapping genes and promoter structure remain beyond the reach of current algorithms.

As none of the integrated gene-identification programs is perfect, all embody somewhat different algorithms and all are rapidly evolving, I very strongly suggest analyzing each sequence with several programs ('Gene identification' Table 1) and carefully comparing the results. If the tools are to be used often, it can be worthwhile to analyze a number of test sequences, where the answer is already known, to get a feeling for algorithm capabilities.

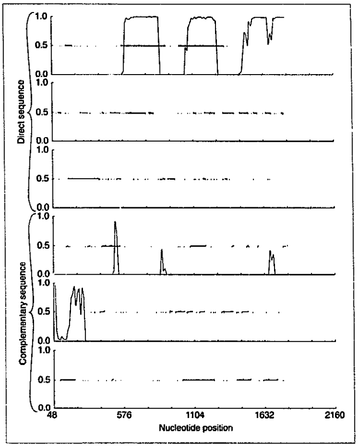
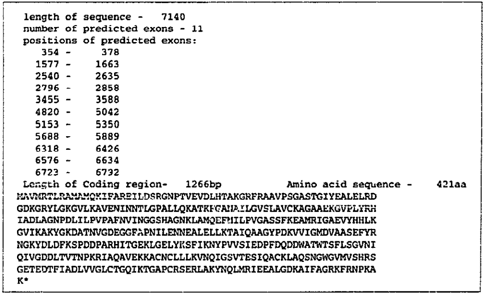


FIGURE 2. Partial sample output from GeneMark (Ref. 18), an email service for coding region identification. GeneMark has seven probabilistic models of DNA, based on counts of hexamers in noncoding regions and in each of the six possible reading frames of coding regions. The program calculates the probability that windows of DNA are noncoding, or should be read in one of the six reading frames.

Future prospects

There are hopeful signs for major improvement in several directions. It has been the case that the best techniques were often not easily accessible to the average user. The situation is getting better, with a number of Internet services easily available (Table 1; Ref. 8), and a World Wide Web (WWW) page that is continually providing more of these services through a single interface[31]. It is still the case, however, that a user wanting access to a suite of state-of-the-art algorithms must either be willing to send data over the Internet (a difficulty if privacy is essential) or hire a programmer to import and install various programs and, in the case of large-scale sequencing, to make a means to automatically submit the sequence to all the programs and distill all the results in a way that makes sense to the end user. A very valuable development would be a framework for tool integration allowing every member of the community to continue independent development, and also allowing someone with relatively little training in programming to integrate any set of such programs into a protocol appropriate for a particular laboratory. Such a framework might be based on email and the WWW.



```
length of sequence -   7140
number of predicted exons - 11
positions of predicted exons:
   354 -    378
  1577 -   1663
  2540 -   2635
  2796 -   2858
  3455 -   3588
  4820 -   5042
  5153 -   5350
  5688 -   5889
  6318 -   6426
  6576 -   6634
  6723 -   6732
Length of Coding region-   1266bp            Amino acid sequence -    421aa
MAVMRTLRAMAMQKIFAREILDSRGNPTVEVDLHTAKGRFRAAVPSGASTGIYEALELRD
GDKGRYLGKGVLKAVENINNTLGPALLQKATKFGANAILGVSLAVCKAGAAEKGVPLYRH
IADLAGNPDLILPVPAFNVINGGSHAGNKLAMQEFMILPVGASSFKEAMRIGAEVYHHLK
GVIKAKYGKDATNVGDEGGFAPNILENNEALELLKTAIQAAGYPDKVVIGMDVAASEFYR
NGKYDLDFKSPDDPARHITGEKLGELYKSFIKNYPVVSIEDPFDQDDWATWTSFLSGVNI
QIVGDDLTVTNPKRIAQAVEKKACNCLLLKVNQIGSVTESIQACKLAQSNGWGVMVSHRS
GETEDTFIADLVVGLCTGQIKTGAPCRSERLAKYNQLMRIEEALGDKAIFAGRKFRNPKA
K*
```

FIGURE 3. Sample output from FGENEH (Ref. 29), an email service for integrated gene identification. The exon structure of the putative gene and the amino acid sequence of the putative product are shown.

Algorithms are getting better at accommodating the needs of real-world data. For example, recent benchmark results[30], and some recent developments (e.g. Ref. 32), take into account the effects of sequencing errors. It is still the case that most integrated gene prediction algorithms make the unrealistic assumption that there is exactly one entire gene in the input sequence, but this will, no doubt, change.

It is remarkable that current algorithms work as well as they do, given that they make use of only a rather small fraction of available biological knowledge. The 'understanding' of genes implicit in any of the current generation of programs could be written down in two or three pages, but, of course, the underlying biology of even fairly general cases is far more complex than this. For example, I know of no program that incorporates an understanding of TATA-less promoters[33] and, because recognizing splice sites is a key challenge in eukaryotic gene identification, some understanding of alternative splicing[34] is bound to be important. If deeper collaborations between computational and experimental biologists can become even a little more frequent, the field will almost certainly be significantly advanced.

Acknowledgements

This work was supported by Public Health Service grant #HG00981-01A1 from the National Center for Human Genome Research. I thank J-M. Claverie, R. Guigs, A. Hatzigeorgiou and B. Roe for valuable comments on the manuscript.

References

1 Church, D.M. *et al.* (1994) *Nat. Genet.* 6, 98–105
2 Caskey, C.T., Eisenberg, R.S., Lander, E.S. and Straus, J. (1995) *Genome Dig.* 2, 6–9
3 Brody, L.C. *et al.* (1995) *Genomics* 25, 238–247
4 Sulston, J. *et al.* (1992) *Nature* 356, 37–41
5 Koonin, E.V., Bork, P. and Sander, C. (1994) *EMBO J.* 13, 493–503
6 Casari, G. *et al.* (1995) *Nature* 376, 647–648
7 Claverie, J-M. (1996) *Meth. Enzymol.* 266, 212–227
8 Fickett, J.W. and Guigs, R. (1996) in *Internet for the Molecular Biologist* (Swindell, S.R., Miller, R.R. and Myers, G., eds), pp. 73–100, Horizon Scientific Press
9 Gelfand, M.S. (1995) *J. Comp. Biol.* 2, 87–115
10 Snyder, E.E. and Stormo, G.D. in *DNA and Protein Sequence Analysis: A Practical Approach* (Bishop, M.J. and Rawlings, C.J., eds), pp. 209–224, IRL Press, (in press)
11 Doolittle, R.F. (1986) *Of URFs and ORFs*, University Science Books
12 Gish, W. and States, D.J. (1993) *Nat. Genet.* 3, 266–272
13 Henikoff, S. and Henikoff, J.G. (1994) *Genomics* 19, 97–107
14 Green, P. *et al.* (1993) *Science* 259, 1711–1716
15 Boguski, M.S., Tolstoshev, C.M. and Bassett, D.E. (1994) *Science* 264, 1993–1994
16 Fickett, J.W. and Tung, C-S. (1992) *Nucleic Acids Res.* 20, 6441–6450
17 Xu, Y. *et al.* (1994) in *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology* (Altman, R. *et al.*, eds), pp. 376–383, AAAI Press
18 Borodovsky, M. and McIninch, J. (1993) *Computers Chem.* 17, 123–134
19 Stormo, G.D. (1990) *Meth. Enzymol.* 183, 211–220
20 von Hippel, P.H. (1994) *Science* 263, 769–770
21 Barrick, D. *et al.* (1994) *Nucleic Acids Res.* 22, 1287–1295
22 Tjian, R. and Maniatis, T. (1994) *Cell* 77, 5–8
23 Koleske, A.J. and Young, R.A. (1995) *Trends Biochem. Sci.* 20, 113–116
24 Kozak, M. (1991) *J. Cell Biol.* 115, 887–903
25 Prestridge, D.S. (1995) *J. Mol. Biol.* 249, 923–932
26 Bucher, P. (1990) *J. Mol. Biol.* 212, 563–578
27 Brunak, S., Engelbrecht, J. and Knudsen, S. (1991) *J. Mol. Biol.* 220, 49–65
28 Einstein, J.R., Mural, R.J., Guan, X. and Uberbacher, E.C. (1992) Oak Ridge National Laboratory report TM-12174
29 Solovyev, V.V., Salamov, A.A. and Lawrence, C.B. (1994) in *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology* (Altman, R. *et al.*, eds), pp. 354–362, AAAI Press
30 Burset, M. and Guigs, R. *Genomics* (in press)
31 http://gc.bcm.tmc.edu:8088/search-launcher/launcher.html
32 Thomas, A. and Skolnick, M.H. (1994) *IMA J. Math. Appl. Med. Biol.* 11, 149–160
33 Zawel, L. and Reinberg, D. (1995) *Annu. Rev. Biochem.* 64, 533–561
34 McKeown, M. (1992) *Annu. Rev. Cell. Biol.* 8, 133–155

References added in proof

35 http://www.sander.embl-heidelberg.de/genequiz/haemophilus.html/
36 http://www.ncbi.nlm.nih.gov/dbEST

J.W. Fickett (ficketjw@molbio.sbpbrd.com) is in the Bioinformatics Group, Mail Code UW 2230, SmithKline Beecham Pharmaceuticals, 709 Swedeland Road, King of Prussia, PA 19406, USA. This work was carried out in the Theoretical Biology and Biophysics Group at Los Alamos National Laboratory.