

How does eukaryotic gene prediction work?

Michael R Brent

Computational prediction of gene structure is crucial for interpreting genomic sequences. But how do the algorithms involved work and how accurate are they?

Gene-prediction programs are used primarily to annotate large, contiguous sequences generated by whole-genome sequencing. Most programs used for this purpose aim to predict the complete exon-intron structures of the protein-encoding portions of transcripts (open reading frames or ORFs). Some programs also predict 5' untranslated regions, and a few predict only the boundaries of isolated exons. The resulting predictions are often distributed through web portals such as the University of California Santa Cruz genome browser (<http://genome.ucsc.edu/>), where users can examine and compare predictions in regions of interest. If whole-genome prediction sets are not available for a sequence in question, users can submit genomic sequences to online gene-prediction servers such as the Twinscan/N-SCAN server (<http://mblab.wustl.edu/nsca/>)¹. Gene predictions can also be used as a springboard to obtaining more direct evidence of gene structures through high-throughput reverse transcription (RT)-PCR and sequencing using primers designed on the basis of gene predictions.

What are the major approaches to gene prediction?

Gene-prediction programs can be broadly divided into those whose only inputs are the sequences of one or more genomes (*de novo* predictors) and those that also consider cDNA sequences and/or their predicted translations (expression-based predictors). Originally, *de novo* predictors used only the genome sequence to be annotated (the target sequence), but the past five years have witnessed substantial gains

in accuracy using one or more additional genomes (see below). Expression-based systems work by aligning cDNA or protein sequences to one or more genome locations based on sequence similarity. Some expression-based systems are designed to output only alignments of expressed sequences to the loci from which they were transcribed (native or *cis* alignment); other systems include alignments of expressed sequences transcribed from other loci or even other species (*trans* alignment).

Expression-based systems that use only native alignments tend to produce exon-intron structures that are quite accurate. Their primary limitation is that there are many genomes for which little or no cDNA sequence is available. Even fairly deep sequencing of randomly selected cDNA clones fails to elucidate the structures of the 20–40% of genes in a typical eukaryotic genome that are expressed only at low levels or under rare conditions. Including *trans* alignments in an annotation increases its sensitivity, but the accuracy of *trans* alignments depends on the degree of similarity between the expressed sequence and the locus to which it is aligned. For loci that cannot be annotated by a high identity cDNA or protein alignment, *de novo* systems—which are the primary focus of this tutorial—can provide more accurate predictions.

How well does *de novo* gene prediction work?

The accuracy of *de novo* gene prediction has improved steadily since the introduction of GENSCAN² ten years ago. GENSCAN correctly predicts an ORF at ~10% of human gene loci that contain a known ORF (gene sensitivity). The next major improvement came in 2001, with the advent of dual-genome *de novo* predictors such as TWINSKAN³. Dual-genome predictors use alignments between the target genome and a related genome (the informant),

and can now predict a perfect open reading frame for more than one-third of known protein-encoding human genes⁴. In more compact genomes, exact ORF accuracy can reach 60–70%. In general, accuracy increases as the number and sizes of introns in a genome decrease. Some systems can now use multiple informants, but results so far indicate rapidly diminishing returns as the number of informants increases⁴.

The greatest limitation of GENSCAN was that it predicted too many genes (~45,000 in human) and exons (~315,000 in human), many of which were false positives. For comparison, today's best estimates place the number of human protein-coding genes at 20,000–21,000 (Michele Clamp, personal communication). The best dual-genome predictors have nearly eliminated this problem, but fragments of processed pseudogenes can still show up as false-positive exons. Such fragments can now be eliminated by integrating dual-genome *de novo* predictors with automated pseudogene detectors⁶. In the past, the number of predicted genes was also inflated by the fact that some programs could not process entire chromosomes at once. Splitting chromosomes before processing would split genes that crossed boundaries. The best dual-genome predictors have nearly eliminated this problem, but fragments of processed pseudogenes can still show up as false-positive exons. Such fragments can now be eliminated by integrating dual-genome *de novo* predictors with automated pseudogene detectors⁵. In the past, the number of predicted genes was also inflated by the inability of some programs to process an entire chromosome at once. Disassembling chromosomes before processing would split genes that crossed boundaries. This problem was recently solved by a new, memory-efficient algorithm that eliminates the need for separate analysis of chromosome fragments⁶. With pseudogene detection and whole-chromosome

Michael Brent is at the Center for Genome Sciences, Campus Box 1045, Washington University, One Brookings Drive, Saint Louis, Missouri 63130, USA.
e-mail: brent@cse.wustl.edu

inputs, the number of human genes predicted by the N-SCAN system fell to 20,138—remarkably close to the current estimates cited above, although, of course, predicting the right number of genes does not mean that all the predicted genes are correct!

A detailed analysis of the accuracy of various gene-prediction programs spanning the entire range of methods can be found in reference 7.

How do *de novo* gene predictors work?

De novo gene predictors are based on some variant of hidden Markov models (HMMs)⁸. To get an intuition for HMMs, imagine that you found a scrap of paper containing the typed letters “hst” and suppose that these letters are a fragment of a message written in English. It is not likely that the sender intended to type “hst,” as relatively few English words contain an “h”

followed by “s,” rare examples notwithstanding (Fig. 1a). An alternative hypothesis that solves this problem is that the sender intended to type “hot.” However, typing “o” for “s” is an unlikely error, as “o” is nowhere near “s” on the keyboard. A more likely hypothesis is that the sender intended to type “hat”: “h” is frequently followed by “a” in correctly spelled English words, and “a” is frequently mistyped as “s”. This situation can be modeled by an HMM. In HMM terminology, the intended letters are called hidden states, or simply states, and the letters on the paper are called observations. For each state, the HMM specifies the probability that it will be followed by each other state (transition probabilities) and the probability that it will result in each possible observed letter (emission probabilities). For any particular HMM and any particular sequence of observations, the Viterbi algorithm can be used to efficiently find the most likely sequence of hidden states.

One can imagine a simple application of HMMs to *de novo* gene prediction in which the observations are nucleotides of the target sequence and the hidden states are the functions they serve in RNA processing and translation, such as the first and second base of an intron (splice donor), a base in the middle region of an intron or the first, second and third base of a codon. Most gene predictors use a more elaborate type of HMM called a generalized hidden Markov model (GHMM). The observation corresponding to each state of a GHMM may be a DNA sequence of any length, whereas in ordinary HMMs, the observation is always a single nucleotide. The states correspond to functions such as coding exon, splice donor region of an intron, middle region of an intron and so on (Fig. 1b). As the input DNA sequence does not include the segment boundaries shown in Figure 1b, the Viterbi algorithm for GHMMs must consider the most likely segmentation of the DNA sequence as well as the most likely state sequence for each segmentation.

For each state, there is a model that defines the probability of each possible observation string. For example, consider the first six nucleotides of an intron (splice donor region). In plants and animals, ~99% of introns begin with GT. The third nucleotide is almost always A or G, whereas the fourth is A ~70% of the time. By compiling these statistics for the first six nucleotides of the intron and assuming that the base found in one position does not depend on the bases found in the other positions, one can generate a simple probability model known as a weight matrix⁹. The widely used sequence logos provide a graphical representation of a weight matrix (Fig. 1c). The weight matrix can be used to calculate the

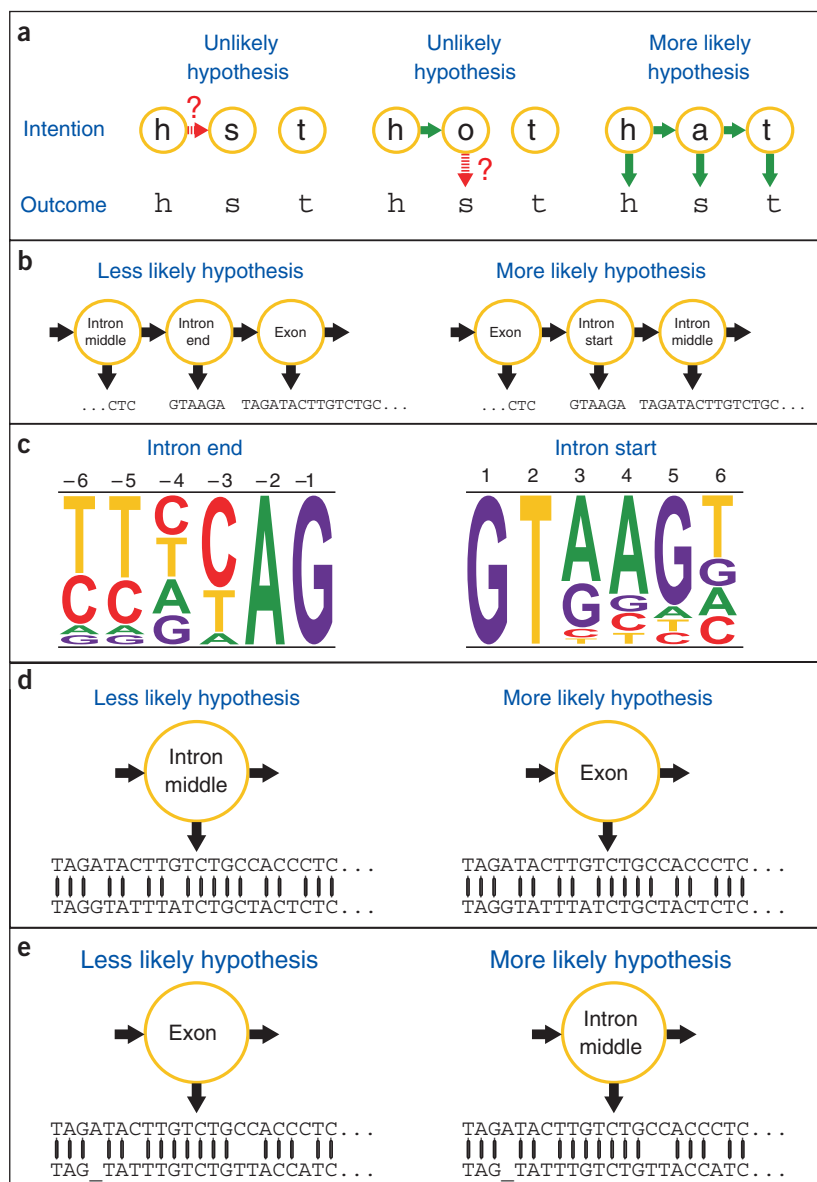


Figure 1 Hidden Markov models form the basis for most gene-prediction algorithms. (a) Use of a hidden Markov model (HMM) to interpret a message containing typographical errors: transition probabilities model letter sequences in correctly spelled English words, whereas emission probabilities model the probability of each possible typographical error. (b) *De novo* gene predictors use generalized hidden Markov models (GHMMs), in which states correspond to variable-length segments of the DNA sequence sharing some common function in transcription, RNA processing or translation. (c) Sequence logos representing weight matrices for the last six bases of an intron (left) and the first six bases of an intron (right). (d,e) For dual-genome predictors, the observations are segments of an alignment between two genomes. The pattern of mismatches and gaps in d suggests a protein-encoding region, whereas the pattern of mismatches and gaps in e suggests a noncoding region.

probability that the start of a randomly selected intron will consist of any given 6-mer. Some states, such as those for the middle regions of introns and exons, use a probability model that allows variable-length observation strings. For any DNA sequence *S*, the probability that the middle region of an intron will consist of *S* is calculated by multiplying the probabilities of each of the bases of *S*, given the five previous bases. For example, the probability of the last A in TAGATA would be estimated by finding all instances of the previous five nucleotides, TAGAT, in the introns of the training set, and computing the fraction of those that are followed by A. These fractions are stored in a large table that the program consults for each position in the putative intron middle *S*.

Dual- and multi-genome *de novo* programs use similar GHMMs, but the observations are segments of an alignment of two or more genomes. For example, **Figure 1d** shows an alignment with no gaps and with mismatches that are separated by multiples of three. This pattern supports the hypothesis that the sub-

stitutions all occur in the third codon position because such substitutions are frequently silent. Thus, this alignment is more likely to correspond to a coding exon than to the middle of an intron. **Figure 1e** shows the same target sequence in an alignment with a frame-shifting gap and two adjacent mismatches, which undermines the hypothesis that it encodes a protein. The most accurate systems also use position-specific substitution models for splice sites and other signals.

Recently, a new variant on the GHMM formalism, the semi-Markov conditional random field (SMCRF) has become a focus of interest for building *de novo* gene-prediction systems^{10,11}. This formalism is more flexible, allowing a wider range of biological features to be incorporated into the model with fewer technical concerns. Although their accuracy on mammalian genomes has not yet exceeded that of the best multi-genome GHMM, SMCRFs show great promise for extending this remarkable ten-year run of steadily increasing gene-prediction accuracy.

COMPETING INTERESTS STATEMENT

The author declares no competing financial interests.

ACKNOWLEDGMENTS

I wish to thank Jeltje van Baren and Tamara Doering for helpful comments and Laura Kyro for artwork. M.R.B. is supported in part by grants from the National Institutes of Health (HG002278, HG003700), the National Science Foundation (0501758) and Monsanto.

1. Hu, P. & Brent, M.R. in *Current Protocols in Bioinformatics* (ed. A.D. Baxevanis) (Wiley, New York; 2003).
2. Burge, C.B. & Karlin, S. *J. Mol. Biol.* **268**, 78–94 (1997).
3. Korf, I., Flicek, P., Duan, D. & Brent, M.R. *Bioinformatics* **17** (Suppl 1), S140–S148 (2001).
4. Gross, S.S. & Brent, M.R. *J. Comput. Biol.* **13**, 379–393 (2006).
5. van Baren, M.J. & Brent, M.R. *Genome Res.* **16**, 678–685 (2006).
6. Keibler, E., Arumugam, M. & Brent, M. *Bioinformatics* **23**, 545–554 (2007).
7. Guigo, R. *et al. Genome Biol.* **7** (Suppl 1), S21–S31 (2006).
8. Eddy, S.R. *Nat. Biotechnol.* **22**, 1315–1316 (2004).
9. D'Haeseleer, P. *Nat. Biotechnol.* **24**, 423–425 (2006).
10. Bernal, A., Crammer, K., Hatzigeorgiou, A. & Pereira, F. *PLoS. Comput. Biol.* **3**, e54 (2007).
11. DeCaprio, D. *et al. Genome Res.* in press (2007).