Genomics **59,** 24–31 (1999)
Article ID geno.1999.5854, available online at http://www.idealibrary.com on IDEAL®

Interpolated Markov Models for Eukaryotic Gene Finding

Steven L. Salzberg,*,†,[1] Mihaela Pertea,† Arthur L. Delcher,‡,§
Malcolm J. Gardner,* and Hervé Tettelin*

**The Institute for Genomic Research, 9712 Medical Center Drive, Rockville, Maryland 20850; †Department of Computer Science, Johns Hopkins University, Baltimore, Maryland 21218; ‡Department of Computer Science, Loyola College in Maryland, Baltimore, Maryland 21210; and §Celera Genomics, 45 W. Gude Dr., Rockville, Maryland 20850*

Received January 19, 1999; accepted April 13, 1999

Computational gene finding research has emphasized the development of gene finders for bacterial and human DNA. This has left genome projects for some small eukaryotes without a system that addresses their needs. This paper reports on a new system, GLIMMERM, that was developed to find genes in the malaria parasite *Plasmodium falciparum*. Because the gene density in *P. falciparum* is relatively high, the system design was based on a successful bacterial gene finder, GLIMMER. The system was augmented with specially trained modules to find splice sites and was trained on all available data from the *P. falciparum* genome. Although a precise evaluation of its accuracy is impossible at this time, laboratory tests (using RT-PCR) on a small selection of predicted genes confirmed all of those predictions. With the rapid progress in sequencing the genome of *P. falciparum*, the availability of this new gene finder will greatly facilitate the annotation process.

1. INTRODUCTION

The gene finding research community has focused considerable effort on human and bacterial genome sequence analysis. This is not surprising given the attention paid to both areas. The Human Genome Project has produced many millions of nucleotides of sequence, and the importance of rapidly identifying the genes in this sequence cannot be overstated. This task is made difficult by the fact that only 1 to 3% of human genomic sequence is estimated to code for proteins. On the bacterial side, 20 complete bacterial and archaeal genomes have already been published, with dozens more expected in the next 2 years. Gene finders for these prokaryotes have an advantage in that approximately 90% of the DNA of these genomes is coding; thus the task reduces in many cases to choosing between competing reading frames. On the other hand, the demand for accuracy is correspondingly much higher in the prokaryotic world.

[1] To whom correspondence should be addressed. Telephone: (301) 315-2537. Fax: (301) 838-0209. E-mail: salzberg@tigr.org.

In between these two genomic worlds lies a vast array of eukaryotic organisms whose genomes range in size from that of a large prokaryote (on the order of tens of millions of nucleotides) to those that are larger than human (billions of nucleotides). Their gene density tends to be much lower than that of bacteria, but many organisms have a much higher gene density than humans. For example, the genome of the eukaryote *Saccharomyces cerevisiae* has approximately one gene every 5 kb. This corresponds to a gene density of 20%. Recently, chromosome 2 of the malaria parasite *Plasmodium falciparum* was completed (Gardner *et al.*, 1998), and this organism too has a gene density of 20%. The remaining 13 chromosomes from malaria should be completed over the course of the next few years. The much larger (120 million nucleotides) genome of *Arabidopsis thaliana*, which also is expected to have a gene density of approximately 20%, should be completed in the same time frame, and many projects are under way to sequence other small eukaryotes.

Because of their relatively high gene density with respect to human DNA, using a gene finder developed for human sequence (or other organisms with low gene density, including most vertebrates and larger plant genomes) may not be the optimal approach for *P. falciparum* and other small eukaryotes. Prokaryotic gene finders are not well suited to this task because of their inability to handle introns. It is possible to retrain human gene finders using different data (for example, GENSCAN (Burge and Karlin, 1997) has been trained with *Arabidopsis* data), but one still runs the risk that because these systems have been optimized to find genes in DNA that is only 3% coding, they may miss many genes in genomes such as *P. falciparum*.

This paper describes a gene finder developed specifically for small eukaryotes with a gene density of around 20%. This system, GLIMMERM, was built and trained using data from *P. falciparum*, the malaria parasite. It was then used as the principal gene finder for chromosome 2 of *P. falciparum*, which contains 210 genes (209 protein coding genes plus one tRNA) (Gardner *et al.*, 1998). Most of these genes were found by

0888-7543/99 $30.00

GlimmerM, and as described below, some predictions were confirmed by additional laboratory experiments.

The basis of GlimmerM is a dynamic programming algorithm that considers all combinations of possible exons for inclusion in a gene model and chooses the best of these combinations. Dynamic programming (DP) has been the basis of many successful eukaryotic gene finders. Hidden Markov model (HMM) systems use a DP algorithm called Viterbi that is a special case of the algorithm here; these HMM methods include VEIL (Henderson *et al.*, 1997); GENSCAN (Burge and Karlin, 1997), which uses semi-Markov HMMs; and Genie (Kulp *et al.*, 1996), which uses generalized HMMs. Very recently, Wirth (1998) described a gene finder for *P. falciparum* based on generalized HMMs, but it is not yet available for comparison. The Morgan system (Salzberg *et al.*, 1996, 1998a) uses a DP algorithm in combination with a decision tree program, and GeneParser (Snyder and Stormo, 1995) uses DP combined with a neural network program. These latter two DP formulations are most similar to the formulation used for GlimmerM.

2. METHODS AND ALGORITHMS

The phrase "gene model" will be used to denote a particular combination of exons and introns that the system is considering as a possible gene. The decision about what gene model is best is a combination of the strength of the splice sites and the score of the exons produced by an interpolated Markov model (IMM). The methods for producing the IMM and splice site scores are described next, followed by the description of the dynamic programming algorithm that uses these scores.

2.1. Interpolated Markov Models

Markov chains are a family of methods for computing the probability of an event based on a fixed number of previous events. (More formally, a Markov chain is a sequence of random variables X_i, where the probability distribution for each X_i depends only on $X_{i-1}, \ldots, X_{i-k}$ for some constant k.) In the context of DNA sequence analysis, Markov chains predict a base by examining a fixed number of bases just prior to that base in the sequence. The most common type of Markov chain is a fixed-order chain, in which the number of previous bases to examine is a constant. For example, a fifth-order Markov chain will predict a base by looking at the five previous bases. Markov chains, and fifth-order chains in particular, have proven to be effective at gene prediction in bacterial genomes (Borodovsky and McIninch, 1993; Borodovsky *et al.*, 1995).

IMMs are a generalization of fixed-order Markov chains. The main distinction is that rather than deciding in advance how many bases to consider for each prediction, these models will use varying numbers of bases for each prediction. In some contexts they will use 5 bases, while in others they might use 6 or more bases, and in yet other cases they may use 4 or fewer bases. This allows IMMs to be sensitive to how common a particular oligomer is in a given genome. In a given genome, many 5-mers might occur rarely and should not be used for prediction; here the IMM will fall back on a shorter Markov chain. On the other hand, certain 8-mers may occur very frequently, and for those the IMM can use this longer context and make a better prediction. In addition, the IMM can combine the evidence from the eighth-order Markov chain and the fifth-order chain in such cases. Thus it has all the information available to a fifth-order chain plus additional information. It is also worth noting that both IMMs and fifth-order Markov chains should outperform methods based on codon usage statistics. (Cf. Saul and Battistutta (1988), a codon usage method specific to *P. falciparum*. Note that at the time of that work, much less *Plasmodium* data were available, and higher-order statistics might have been inaccurate as a result.)

IMMs form the basis of the Glimmer system for finding genes in bacteria and archaea (Salzberg *et al.*, 1998b). Glimmer correctly identifies approximately 98% of the genes in bacteria without any human intervention and with a very limited number of false-positives. It has been used as the gene finder for *Borrelia burgdorferi* (Fraser *et al.*, 1997), *Treponema pallidum* (Fraser *et al.*, 1998), *Chlamydia trachomatis* (Stephens *et al.*, 1998), *Thermotoga maritima* (Nelson *et al.*, submitted for publication), and others. Based on the success of Glimmer in bacterial sequence annotation, we thought that IMMs should make a good foundation for eukaryotic gene finding. This is particularly true of small eukaryotes like *P. falciparum* in which the gene density is intermediate between that of prokaryotes and higher eukaryotes.

Details of how to construct an IMM for sequence data can be found in the original Glimmer publication (Salzberg *et al.*, 1998b); GlimmerM uses the same IMM algorithm as that described there. In brief, GlimmerM builds IMMs from a set of DNA sequences chosen for training. For coding regions, it builds three separate IMMs, one for each codon position. (This is known as a 3-periodic Markov model (Borodovsky and McIninch, 1993).) These IMMs include zeroth- through eighth-order Markov chains, as well as weights computed for every oligomer of 8 bases or less that appears in the training data. These weights and Markov models are interpolated to produce a score for each base in any potential coding sequence. The logs of these scores are summed to score each coding region.

2.2. Splice Site Identification

The approach used by GlimmerM to determine the splice sites is similar to that used in the Morgan human gene finding system (Salzberg *et al.*, 1998a). A second-order Markov chain model is used to score a 16-base region around donor sites and a 29-base region around acceptor sites. For both donor and acceptor sites in *P. falciparum*, a wide range of different regions were tested, and these sizes performed best. Two second-order Markov models were built for each type of site. First, a "true" Markov model was created from existing data on known 5′ and 3′ consensus sites. These data were collected by exhaustively combing the literature for every documented exon–intron boundary. A "false" Markov model was built from a large number of randomly chosen false splice sites, i.e., sequences that contained the consensus GT or AG dinucleotide but that were not true splice sites. The score of a site $s_i, s_{i+1}, \ldots, s_j$ was computed by each Markov model according to the formula

$$S(i, j) = \sum_{k=i}^{j} M_{s,k},$$

where

$$M_{s,k} = \ln(f((s_{k-2}, s_{k-1}, s_k), k)/f((s_{k-2}, s_{k-1}), k-1)),$$

and $f(s, k)$ is the frequency of substring s ending at location k. Note that for the leftmost position in the splice site region, M is taken to be the probability given by the zeroth-order Markov model, and for the second position, M is given by the first-order model. The score for a given splice site is computed by taking the difference of the scores obtained from the true site Markov model and the false site model.

After building the models, we scored all the true splice sites and a large selection of randomly chosen false sites. We then set minimum cut-off scores to identify correctly most (or all) true sites and measured how many false-positives we would expect with various thresholds. The splice sites for training the Markov models were taken from the 119 genes (described under Results and Discussion) used to train the IMMs, all of which had laboratory evidence to support them. These genes contained only 81 introns in total, which did not gener-

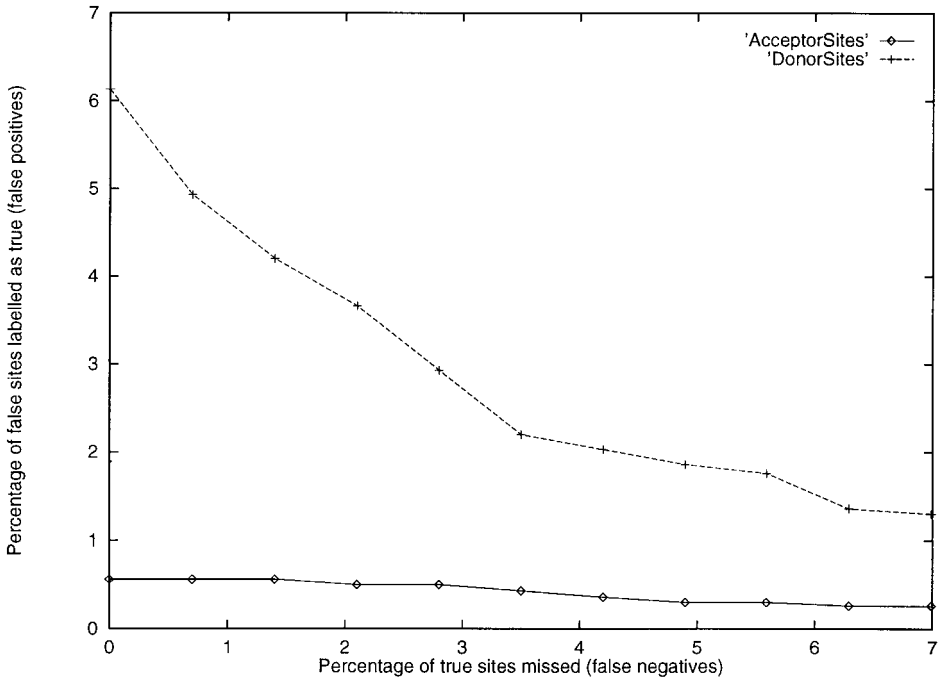

FIG. 1. Trade-off between false-positive rates and false-negative rates for the Markov chain method that recognizes exon–intron splice sites. Data represent the accuracy on sites annotated in chromosome 2 of *P. falciparum*.

ate enough data to produce a very reliable second-order Markov model. Therefore, after an initial training pass using the 81 introns, we used GLIMMERM itself to predict additional introns in chromosome 2, selected the best of these, and added them to the training set. Of course this is a "circular" training protocol, but this represents our attempt to squeeze the best performance we could from limited data. As the sequencing of the remaining chromosomes continues, and as ESTs yield further hard evidence on introns, the available pool of reliable data for training the splice site models should grow dramatically. Alignments with protein sequences from other organisms will provide additional evidence about intron locations. The Markov chain models will consequently improve in accuracy. We intend to continue retraining these models as the genome sequencing progresses.

Figure 1 shows the trade-off in thresholds for the splice site recognition function in *P. falciparum* and shows the trade-off between sensitivity and selectivity for the Markov chain method on the 143 donor and acceptor sites in chromosome 2. Acceptor sites are much easier to recognize: with a false-negative rate of 0% (corresponding to a sensitivity of 100%, meaning that all true sites will be recognized), the false-positive rate—the percentage of AG dinucleotides that will incorrectly be called acceptor sites—is just 0.56%. For donor sites, a 0% false-negative rate corresponds to a rather high 6.1% false-positive rate. Setting the system so that it misses 4 of the 143 (2.8%) donor sites in chromosome 2 would reduce this false-positive rate to 2.9%. The Markov thresholds used here are set so that no true splice sites will be missed.

2.3. Dynamic Programming

GLIMMERM's use of dynamic programming allows it to prune out a large number of possible exon–intron combinations and focus its analysis only on relatively high-scoring combinations (called "parses"). The input to the algorithm is any genomic DNA sequence in FASTA format; small sequences as well as entire chromosomes can be input. The output is a partitioning of the DNA into coding regions interleaved with noncoding regions, on both the main and the complementary strands of the sequence.

As in many other gene finders (Salzberg, 1998), there are a number of assumptions used by GLIMMERM when predicting genes in the DNA sequence. The main assumptions are (1) the coding region of every gene begins with a start codon ATG, (2) a gene has no in-frame stop codons except the very last codon, and (3) each exon is in a consistent reading frame with the previous exon. These constraints significantly enhance the efficiency of computing the optimal gene models, by restricting the search space of the DP algorithm. On the other hand, genuine frameshifts cannot be detected by the system.

The dynamic programming algorithm fills in a structure *Parse,* in which each element *Parse* [*t, n, S*] denotes the optimal parse of the subsequence that begins at location *n* and ends at the stop codon at location *S*. The variable *t* specifies the type of signal at *n*, which can be donor, acceptor, start (codon), or stop (codon). More specifically, *Parse* is an ordered list of labeled positions indicating the end-points of a set of exons. For example,

$$Parse[\text{start}, 100, 540] = \langle \text{start}, 100\rangle, \langle \text{donor}, 240\rangle, \langle \text{acceptor}, 380\rangle, \langle \text{stop}, 540\rangle$$

indicates a pair of exons at positions [100 . . . 239] and [380 . . . 539]. A complete gene model is represented as a list *Parse* [start, *n, S*]. Other elements are partial parses, beginning at a location of type *t* ($t \neq$ start) and ending at a stop codon *S*.

The DP algorithm processes the input sequence left to right, looking for stop codons. At each stop codon *S*, it searches back in the 5′ direction and finds all possible genes ending at that stop. It chooses the highest scoring gene to store in *Parse.* More concisely,

$$Parse[t, n, S] = \langle t, n\rangle, Parse[t_{\text{next}}, i, S],$$

where *i* is the location that achieves the maximum score

$$\max_{n<i<S}\{Score(\langle t, n\rangle, Parse[t_{\text{next}}, i, S])\},$$

and t_{next} is the type logically following the type *t* in a parse. For example, if *t* = acceptor, then t_{next} can be either donor or stop. *Score* (*Parse* [*t, n, S*]) is the score given by the IMMs to the coding region obtained by concatenating all the exons in the parse delimited by

Parse [*t*, *n*, *S*]. For example, if *n* is an acceptor site, the algorithm considers all sites *i* that can follow *n* and chooses the best one. These would include donor sites, if *n* is the beginning of an internal exon, and stop codons, if *n* is the final coding exon. Because the algorithm works backward from each stop codon *S*, the entry *Parse* [t_{next}, *i*, *S*] is computed prior to *Parse* [*t*, *n*, *S*]. The only positions that are considered as possible donor and acceptor sites are those that score above the threshold determined by the Markov chains described previously.

The algorithm incorporates special cases for each of the four types *t* to prune the search space further. These are as follows:

1. If the interval (*n* . . . *i*) is the coding portion of an exon, its IMM score must exceed a fixed, preset threshold.
2. If two internal exons (*n* . . . i_1) and (*n* . . . i_2) both result in identical IMM scores, choose the one that maximizes the length of the coding part of the parse. Note that this rule makes GLIMMERM prefer longer gene models.
3. If (*n* . . . *i*) is an intron, then its AT content must be at least 70%. This constraint is based on the observation that all *P. falciparum* introns in the training set had an AT content of above 70%, with only 1% of introns having an AT content under 75%. In contrast, *P. falciparum* exons have an AT content of 70–75%.
4. The length of an intron must be between 50 and 1500 bp; 73 and 1066 bp were the extreme lengths for the introns in the training set.
5. The total length of the coding portions of a gene model represented in *Parse* [start, *n*, *S*] must be greater than 200 bp.
6. If *n* is a stop codon, the algorithm searches backward for all gene models ending at *n*. Many stop codons can be quickly eliminated because they follow too closely another stop codon in the same reading frame. Thus there is no way to create a gene model ending at these stops—any genes ending at the stop would be too short. The high AT content of *P. falciparum* and the resulting high frequency of stop codons make this step particularly effective.

An attempt was made to use IMMs to score introns as well as exons, but this did not improve the results. Therefore, when *t* is a donor site and t_{next} is an acceptor, we have

$$Score(\langle \text{donor}, n\rangle, Parse[\text{acceptor}, i, S]) = Score(Parse[\text{acceptor}, i, S]).$$

The algorithm is run separately on both the direct and the complementary strands of the input. GLIMMERM then makes one more pass over the list of putative genes to reject overlapping genes. If genes overlap by less than a fixed amount (30 bp by default), then the overlap is ignored, and both genes are reported in the output. Most overlapping genes are competing gene models that share a stop codon and have different exon locations. Genes that overlap by more than 30 bp are rescored using the IMM, and the gene with the best score is retained. If the scores of two or more overlapping models differ from the maximum score by less than a small preset amount, then GLIMMERM considers the scores equivalent and outputs all the models as possible genes. In these instances, it marks the longest gene as the preferred model.

2.4. Code Availability

The complete GLIMMERM system is available from the authors; it has already been shared with other malaria genome sequencing centers. The code includes routines for retraining the system on data from other organisms. A version of the system trained on *A. thaliana* genes is currently under development. Total processing time to find all genes in malaria chromosome 2 (approximately one million nucleotides) is about 50 min on a Pentium 450 processor running Linux.

2.5. Annotating a Genome

In its current form, GLIMMERM produces multiple gene models for some genes. When no database matches and no other computational evidence were found to support a GLIMMERM prediction, the chromosome 2 annotation reflects the highest scoring model. Although many of these are likely to be correct, it is undoubtedly the case that some are not. Further investigation is required to confirm these predictions (but see below for laboratory evidence confirming a small subset).

The GLIMMERM algorithm was used as one of a suite of tools. Accurate gene identification depends on using every tool available, and the description here should not be taken as implying that GLIMMERM alone can find all genes in *P. falciparum* or any other genome. However, it was a central component in a larger strategy. Other important computational tools used by the malaria chromosome 2 team were as follows: (1) searches of a nonredundant protein sequence database using gapped BLAST and PSI-BLAST (Altschul *et al.*, 1990, 1997); (2) gapped alignments of DNA to protein and EST sequence databases using DDS and DPS (Huang *et al.*, 1997); (3) prediction of putative signal peptides using SignalP (Nielsen *et al.*, 1997); (4) prediction of transmembrane domains with PHThtm (Rost *et al.*, 1995); (5) prediction of nonglobular structures with SEG (Wootton and Federhen, 1996); and (6) a graphical tool to allow annotators to view all the evidence together. In addition, the project used additional aligment tools developed at The Institute for Genomic Research to detect frameshift errors: these tools allow an annotator to detect when a sequence alignment extends beyond the start and stop codons indicated by other tools. In some cases this indicates errors in sequencing, which can be corrected; in other cases it indicates either a genuine frameshift that occurs during translation or a mutation that has changed the length of the translated protein. Any comprehensive annotation effort needs these computational tools and more to produce reasonably accurate gene annotations.

3. RESULTS AND DISCUSSION

GLIMMERM was used as the primary gene finder for chromosome 2 of *P. falciparum*. Chromosome 2 has 209 protein-coding genes spread over approximately one million bases, for a gene density of one gene per 4.5 kb (1/4.5 kb). This contrasts with a density of 1/kb in bacteria, 1/2 kb in yeast, 1/7 kb in *C. elegans*, and 1/50 kb (estimated) in human. Of the 209 protein-coding genes, 43% had at least one intron, and those genes with introns usually had just one or two introns (Gardner *et al.*, 1998). Below we attempt to quantify GLIMMERM's accuracy on these genes.

3.1. Training

To train the IMM, we needed to collect as much coding sequence as possible from *P. falciparum* itself. We exhaustively surveyed the literature to collect every complete sequence that was backed by laboratory evidence. Our survey collected 119 complete coding sequences from 108 GenBank entries representing all 14 chromosomes, of which just 6 genes came from chromosome 2. (This database is available by e-mail upon request from the authors.) Note that by length, chromosome 2 comprises approximately 3% of the genome, so it is unsurprising that just 6/119 genes were from chromosome 2. GenBank contains more than 108 entries from *P. falciparum*, but other entries do not have clear evidence supporting their splice sites. This training set provided the initial data for the splice site models as well.

An important point to emphasize here is that *P.*

TABLE 1

Performance of GlimmerM on Genes Whose Structure Is Completely Known from Independent Laboratory Evidence

Name	Len	Intr	Comment	Common name
PFB0100c	654	1	Perfect match	Knob-associated His-rich prt
PFB0295w	471	0	Perfect match	Adenylosuccinate lyase (OO)
PFB0300c	272	0	Perfect match	Merozoite surface antigen MSP-2
PFB0305c	272	1	Perfect match	Merozoite surface antigen MSP-5 (EGF domain)
PFB0310c	272	1	Perfect match, highest score from 5 models	Merozoite surface antigen MSP-4 (EGF domain)
PFB0340c	997	3	Perfect match, second highest score from 4 models	SERA antigen/papain-like Protease with active Ser
PFB0405w	3135	0	Perfect match, higher score from 2 models	Transmission blocking Target antigen PfS230

Note. All seven genes had perfect matches to the system's predictions, meaning that the start codon, stop codon, and every splice site were correctly predicted. The column headings give the gene name, its length in amino acids, number of introns (Intr), a comment on GlimmerM's prediction, and the common name of the protein.

falciparum has an unusually high 82% AT content. As a consequence of this high AT content, stop codons are very frequent (e.g., TAA will occur especially often) in noncoding DNA. This makes it much more likely that long open reading frames (ORFs) represent coding sequence. This fact was used to generate additional training data for GlimmerM: ORFs greater than 500 bp in the chromosome 2 sequence were assumed to be coding regions and were used in the IMM training. These were added to the list generated by the literature search.

3.2. Accuracy on Known Genes

The 209 genes included in the chromosome 2 annotation were found with GlimmerM's help. To evaluate the accuracy of the system, it is helpful to consider only those genes from this set for which independent evidence can be found to confirm their existence.

The best way to measure the program's accuracy is to consider its accuracy on those proteins whose exon–intron structure is known precisely from laboratory studies. There are seven genes from chromosome 2 of *P. falciparum* that currently fit into this category; i.e., the sequence from start to stop has been completely characterized. Of these seven, six were included in the training set, and one (PFB0100c) was not.

GlimmerM's performance on this small set of genes is shown in Table 1. For the two-exon gene PFB0100c, the only independently confirmed gene that was not included in the training set, the system predicted only one model: the correct one. For all seven of the genes, GlimmerM's output contained a model that matched perfectly. For four of the genes, the correct model was the only one output by the system. For PFB0310c and PFB0405c, GlimmerM produced five and two competing models, respectively, but in each case the highest scoring one was correct. Only for PFB0340c, a four-exon gene, was GlimmerM's correct model not the highest scoring one. The system gave a slightly higher score to a model that used a different donor site for the first exon. GlimmerM's alternate prediction would have a 23-aa insertion in this 997-aa protein.

3.3. Laboratory Tests

An ideal way of measuring the accuracy of GlimmerM precisely would be to test each of its predictions in the laboratory to see whether they are expressed as predicted. Although a complete test of all predictions would be difficult and time-consuming, one careful set of experiments was conducted as part of the chromosome 2 study.

Because many of the proteins predicted by GlimmerM had unusual nonglobular domains, the chromosome 2 project team ran a reverse transcriptase (RT-PCR) experiment for 13 of these genes (Gardner *et al.*, 1998) to determine whether or not they were real. These genes are shown in Table 2. The RT-PCR focused its attention on nonglobular domains, not entire proteins, so it could not confirm every detail of the GlimmerM predictions. In particular, it did not test the exon–intron boundaries for the two genes in this set

TABLE 2

The Set of Genes with Nonglobular Domains for Which RT-PCR Experiments Were Conducted to Confirm Expression

Name	Length	Intr	Common name
PFB0130w	538	0	Prenyl transferase
PFB0145c	1979	0	Hypothetical protein
PFB0180w	560	1	prt with 5′-3′ exonuclease domain
PFB0265c	1516	0	RAD2 endonuclease
PFB0380c	2010	0	Phosphatase (acid phosphatase family)
PFB0435c	1138	7	Predicted amine transporter
PFB0500c	235	0	RAB GTPase
PFB0520w	1233	0	Novel protein kinase
PFB0525w	610	0	Asparaginyl-tRNA synthetase
PFB0685c	885	0	ATP-dependent acyl-CoA synthetase
PFB0720c	899	0	Ori. recognition complex subunit 5 (ATPase)
PFB0755w	1398	0	Hypothetical protein
PFB0880w	426	0	FAD-dependent oxidoreductase

Note. Length is shown in amino acids, and Intr gives the number of introns. In the two genes containing introns, the nonglobular domains are contained within exons.

that contain introns, because the nonglobular domains in those genes do not cross those boundaries. This experiment confirmed that all 13 of the nonglobular domains are expressed; i.e., the predictions for those regions were correct. To our knowledge, this is the first time ever that computational gene predictions provided the impetus for experiments that in turn confirmed the predictions.

Eleven of these 13 genes have sequence homology to known proteins from other organisms. It is worth noting that the nonglobular domains of the *P. falciparum* proteins did *not* occur in the homologs. For example, PFB0180c contains a 176-amino-acid nonglobular insert that is absent from four homologous bacterial exonuclease domains (shown in Fig. 2 of Gardner *et al.*, (1998)). GlimmerM's prediction for this gene was confirmed by amplifying and then sequencing a region that contained the nonglobular domain. This example points out that the presence of a homologous protein sequence does not always produce an accurate gene prediction.

3.4. Comparison on Genes with Homologs

Of the 209 genes in chromosome 2, 119 have homologous proteins in the public sequence databases. (The training set also contained 119 genes, but the identity of these two numbers is merely coincidence.) The existence of homologs, which come from a wide range of other organisms, provides strong independent evidence that these genes are real. We therefore used these genes to make further measurements of GlimmerM's accuracy.

Of the 119 genes, 7 were already mentioned: these are the genes from chromosome 2 whose exon–intron structure was known from previously published laboratory studies. Six of those were included in the training set, which leaves 113 genes in chromosome 2 that were *not* included in the training set and for which we have good hints of their exon–intron structure. Because these are homologs, parts of some genes may not align well, making the predicted exon–intron structure less certain.

GlimmerM finds 98 of these 113 genes (87%) exactly; i.e., the positions of the start codon, the boundaries of each exon and intron, and the stop codon correspond to what is indicated by the alignments to homologous genes. Of these, 22 have competing gene models that score higher, meaning that a human annotator had to examine the output and decide, based on the alignment, to use a model other than the highest-scoring one.

Of the 15 genes that GlimmerM did not find exactly, 14 were found but had slightly modified coding regions. Seven intronless genes were predicted with incorrect start codons. Three 2-exon genes were broken into two genes each. Four 3-exon genes were predicted with an incorrect first exon but correct second and third exons.

Only one of the genes with homologs, ribosomal protein S30, was missed completely; ribosomal proteins often have a strikingly different composition from other genes and are known to be difficult for content-based gene finders to locate. These will not be missed as long as genomic data are searched against databases of known ribosomal proteins.

In summary, chromosome 2 contains 113 genes that were not included in the set of 119 genes used to train GlimmerM's IMM. Portions of some of these genes, those with ORFs greater than 500 bp, were extracted automatically and added to the IMM; this portion of the training is fully automatic and requires no human intervention. The splice site training also included some data from chromosome 2, as explained above. A similar procedure can be performed on future chromosomes to extract additional splicing data: first use a sequence alignment program to find homologous genes, extract splice sites from those, and add those splice sites to the Markov chain models. This will allow users of the system to improve the system's performance before making a final run on their chromosomes. Assuming this or a similar protocol is followed, the estimates given here should extrapolate reliably to those chromosomes. Of the 113 genes with homologs, GlimmerM is able to annotate automatically 76 (66%) if its top-scoring prediction is assumed correct. If a human annotator is available to confirm or reject predictions, then this number grows to 87% (98/113). In most cases the differences between competing models are small, involving one splice site or the start codon. Information from alignments or from other programs—for example, identification of signal peptides—allowed the human annotators to override GlimmerM's first choice in selected cases.

3.5. Comparison to Chromosome 2 Annotation

Of the 209 genes currently annotated for chromosome 2, GlimmerM finds 178 exactly. Of these, 40 have competing gene models that score higher; human annotators chose a different model for the final annotation. Of the remaining 31 genes, GlimmerM finds the stop codons correctly for 14. Different starts appear in the final annotation for several reasons, for example, the existence of a match to a protein sequence that starts at a different start codon. (Note that it is possible that GlimmerM is still correct in these cases.) The system finds the correct start but the wrong stop codon for 4 genes; this occurs in multiexon genes in which a splice site was missed and one of the exons was incorrectly extended until it hit a stop codon. The 11 remaining partial hits are cases for which GlimmerM predicts some but not all exons correctly; for example, several multiexon genes are each broken into two separate genes.

Only 2 of the 209 genes are missed completely. One is ribosomal protein S30, which was mentioned above. The second is a predicted integral membrane protein of 192 aa predicted by a preliminary version

of GLIMMERM (before retraining the splice site models). A separate program was used to predict the function of this protein; it did not align to any known sequences.

The improved splice site Markov models resulted in GLIMMERM's generating 41 fewer gene models than before. In addition to the one missed gene just described, it generated 5 new gene models. Of these, one appears to encode a genuine protein, and we are currently investigating this to see if it should be added to the published annotation.

A significant caveat to include with these results is that GLIMMERM often produces multiple competing models that the human annotator must resolve. Most genes with three or more exons result in multiple models. The system indicates which model scores the highest, but as indicated above, 40 of the "correct" gene models had alternative parses that scored higher. These alternative parses share some exons but use different splice sites for others. A human annotator looking at additional evidence, such as alignments to homologous proteins or predictions of signal peptides, was able to overrule the system's top choice in these cases. It is likely that in other cases where no evidence besides GLIMMERM's prediction is available, some of the published annotation may still be in error (all such proteins are annotated as hypotheticals). After each set of multiple gene models was collapsed into one model, the gene list still contains 266 genes. (All of the models can be downloaded on the Web at www.tigr.org/~salzberg/GlimmerMchr2output.html.) These means that, since only 209 genes appeared in the final annotation, the annotators eliminated another 57 gene models entirely from the output. These decisions were somewhat subjective: frequently the putative genes were short or they consisted mostly of low-complexity sequence, and this was not enough to convince the human annotators that the genes were real. In many cases the annotators are probably correct, but it is simply impossible at this point to say with confidence that all of the deleted genes are false-positives. Only further evidence will allow us to decide, but this makes clear the importance of continuing to update and improve genome annotation over time.

ACKNOWLEDGMENTS

S.L.S. is supported by the National Human Genome Research Institute at NIH under Grant K01-HG00022-1. S.L.S., A.L.D., and M.P. are supported in part by the National Science Foundation under Grant IRI-9530462. M.J.G. and H.T. were supported by a supplement to NIAID Grant R01 AI40125-01, which was made possible with funds from NIH's Office for Research on Minority Health and Department of the Army Cooperative Agreement DAMD17-98-2-8005.

REFERENCES

Altschul, S., Gish, W., Miller, W., Myers, E., and Lipman, D. (1990). Basic local alignment search tool. *J. Mol. Biol.* **215,** 403–410.

Altschul, S., Madden, T., Schaffer, A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. (1997). Gapped blast and psi-blast: A new generation of protein database search programs. *Nucleic Acids Res.* **25**(17), 3389–3402.

Borodovsky, M., and McIninch, J. (1993). Genemark: Parallel gene recognition for both DNA strands. *Comput. Chem.* **17**(2), 123–133.

Borodovsky, M., McIninch, J., Koonin, E., Rudd, K., Medigue, C., and Danchin, A. (1995). Detection of new genes in the bacterial genome using Markov models for three gene classes. *Nucleic Acids Res.* **23,** 3554–3562.

Burge, C., and Karlin, S. (1997). Prediction of complete gene structures in human genomic DNA. *J. Mol. Biol.* **268,** 78–94.

Fraser, C., Casjens, S., Huang, W., Sutton, G., Clayton, R., Lathigra, R., White, O., Ketchum, K., Dodson, R., Hickey, E., Gwinn, M., Dougherty, B., Tomb, J.-F., Fleischmann, R., Richardson, D., Peterson, J., Kerlavage, A., Quackenbush, Salzberg, S., Hanson, M., van R., Vugt, Palmer, N., Adams, M., Gocayne, J., Weidman, J., Utterback, T., Watthey, L., McDonald, L., Artiach, P., Bowman, C., Garland, S., Fujii, C., Cotton, M., Horst, K., Roberts, K., Hatch, B., Smith, H., and Venter, J. (1997). Genomic sequence of a lyme disease spirochaete, *Borrelia burgdorferi*. *Nature* **390**(6660), 580–586.

Fraser, C., Norris, S., Weinstock, G., White, O., Sutton, G., Clayton, R., Dodson, R., Gwinn, M., Hickey, E., Ketchum, K., Sodergren, E., Hardham, J., McLeod, M., Salzberg, S., Khalak, H., Weidman, J., Howell, J., Chidambaram, M., Utterback, T., Watthey, L., McDonald, L., Artiach, P., Bowman, C., Garland, S., Fujii, C., Cotton, M., Horst, K., Roberts, K., Hatch, B., Smith, H., and Venter, J. (1998). Complete genomic sequence of *Treponema pallidum*, the syphilis spirochete. *Science* **281,** 375–388.

Gardner, M., Tettelin, H., Carucci, D., Cummings, L., Aravind, L., Koonin, E., Shallom, S., Mason, T., Yu, K., Fujii, C., Pederson, J., Shen, K., Jing, J., Aston, C., Lai, Z., Schwartz, D., Pertea, M., Salzberg, S., Zhou, L., Sutton, G., Clayton, R., White, O., Smith, H., Fraser, C., Adams, M., Venter, J., and Hoffman, S. (1998). Chromosome 2 sequence of the human malaria parasite *Plasmodium falciparum*. *Science* **282,** 1126–1132.

Henderson, J., Salzberg, S., and Fasman, K. (1997). Finding genes in human DNA with a hidden Markov model. *J. Computat. Biol.* **4**(2), 127–141.

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

Kulp, D., Haussler, D., Reese, M. G., and Eeckman, F. H. (1996). A generalized hidden Markov model for the recognition of human genes in DNA. *In* "ISMB-96: Proceedings of the Fourth International Conference on Intelligent Systems for Molecular Biology," pp. 134–141, AAAI Press. Menlo Park, CA.

Nelson, K., Clayton, R., Gill, S., Gwinn, M., Dodson, R., Haft, D., Hickey, E., Peterson, J., Nelson, W., Ketchum, K., McDonald, L., Utterback, T., Malek, J., Linher, K., Garrett, M., Stewart, A., Cotton, M., Pratt, M., Phillips, C., Richardson, D., Heidelberg, J., Sutton, G., Fleischmann, R., White, O., Salzberg, S., Smith, H., Venter, J., and Fraser, C. Genome sequence of *Thermotoga maritima:* Evidence for lateral gene transfer between archaea and bacteria. Submitted for publication.

Nielsen, H., Engelbrecht, J., Brunak, S., and von Heijne, G. (1997). Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites. *Protein Eng.* **10**(1), 1–6.

Rost, B., Casadio, R., Fariselli, P., and Sander, C. (1995). Transmembrane helices predicted at 95% accuracy. *Protein Sci.* **4**(3), 521–533.

Salzberg, S. (1998). Decision trees and Markov chains for gene finding. *In* "Computational Methods in Molecular Biology" (S. Salzberg, D. Searls, and S. Kasif, Eds.), pp. 187–203, Elsevier, Amsterdam.

Salzberg, S., Chen, X., Henderson, J., and Fasman, K. (1996). Finding genes in DNA using decision trees and dynamic programming. *In* "ISMB-96: Proceedings of the Fourth International Conference on Intelligent Systems for Molecular Biology" pp. 201–210, AAAI Press, Menlo Park, CA.

Salzberg, S., Delcher, A., Fasman, K., and Henderson, J. (1998a). A decision tree system for finding genes in DNA. *J. Computat. Biol.* **5**(4), 667–680.

Salzberg, S., Delcher, A., Kasif, S., and White, O. (1998b). Microbial gene identification using interpolated Markov models. *Nucleic Acids Res.* **26**(2), 544–548.

Saul, A., and Battistutta, D. (1988). Codon usage in *Plasmodium falciparum*. *Mol. Biochem. Parasitol.* **27,** 35–42.

Snyder, E. E., and Stormo, G. D. (1995). Identification of coding regions in genomic DNA. *J. Mol. Biol.* **248,** 1–18.

Stephens, R., Kalman, S., Lammel, C., Fan, J., Marathe, R., Aravind, L., Mitchell, W., Olinger, L., Tatusov, R., Zhao, Q., Koonin, E., and Davis, R. (1998). Genome sequence of an obligate intracellular pathogen of humans: *Chlamydia trachomatis. Science* **282**(5389), 754–759.

Wirth, A. (1998). "A *Plasmodium falciparum* genefinder," Honours thesis, Department of Mathematics and Statistics, University of Melbourne.

Wootton, J., and Federhen, S. (1996). Analysis of compositionally biased regions in sequence databases. *Methods Enzymol.* **266,** 554–71.